

# ***CodeIgniter Rocks***

*Versión 3.1.9*

CodeIgniter was created by EllisLab and is now a project of the British Columbia Institute of Technology

© Copyright 2014 - 2018, British Columbia Institute of Technology. Last updated on Jun 12, 2018.

## *Lista de métodos / funciones*

Barcelona, 22 de Junio del 2018

Carlos Valencia Ruiz

[www.codeigniter.xyz](http://www.codeigniter.xyz)

Crear clases auxiliares.....	16
get_instance().....	16
Funciones Comunes.....	17
config_item().....	17
function_usable().....	17
get_mimes().....	18
html_escape().....	18
is_cli().....	18
is_https().....	19
is_php().....	19
is_really_writable().....	19
remove_invisible_characters().....	20
set_status_header().....	20
Funciones de compatibilidad.....	21
password_get_info().....	21
password_hash().....	21
password_needs_rehash().....	22
password_verify().....	22
hash_equals().....	23
hash_pbkdf2().....	23
mb_strlen().....	23
mb_strpos().....	24
mb_substr().....	24
array_column().....	25
hex2bin().....	25
Manejo de errores.....	25
log_message().....	25
show_404().....	26
show_error().....	27
class CI_Benchmark.....	28
class CI_Benchmark.....	28
elapsed_time().....	28
mark().....	28
memory_usage().....	28
class CI_Cache.....	29
class CI_Cache.....	29
cache_info().....	29
clean().....	29
decrement().....	30
delete().....	30
get().....	30

<b>get_metadata()</b> .....	31
<b>increment()</b> .....	31
<b>is_supported()</b> .....	32
<b>save()</b> .....	32
<b>class CI_Calendar</b> .....	33
<b>class CI_Calendar</b> .....	33
<b>adjust_date()</b> .....	33
<b>default_template()</b> .....	34
<b>generate()</b> .....	34
<b>get_day_names()</b> .....	34
<b>get_month_name()</b> .....	34
<b>get_total_days()</b> .....	35
<b>initialize()</b> .....	35
<b>parse_template()</b> .....	36
<b>class CI_Config</b> .....	36
<b>class CI_Config</b> .....	36
<b>\$config</b> .....	36
<b>\$is_loaded</b> .....	36
<b>base_url()</b> .....	36
<b>item()</b> .....	36
<b>load()</b> .....	37
<b>set_item()</b> .....	37
<b>site_url()</b> .....	37
<b>slash_item()</b> .....	38
<b>system_url()</b> .....	38
<b>class CI_Email</b> .....	38
<b>class CI_Email</b> .....	38
<b>attach()</b> .....	38
<b>attachment_cid()</b> .....	39
<b>bcc()</b> .....	40
<b>cc()</b> .....	40
<b>clear()</b> .....	41
<b>from()</b> .....	41
<b>message()</b> .....	42
<b>print_debugger()</b> .....	42
<b>reply_to()</b> .....	43
<b>send()</b> .....	43
<b>set_alt_message()</b> .....	44
<b>set_header()</b> .....	44
<b>subject()</b> .....	45
<b>to()</b> .....	45
<b>class CI_Encryption</b> .....	46
<b>class CI_Encryption</b> .....	46
<b>create_key()</b> .....	46
<b>decrypt()</b> .....	46

encrypt()	46
hkdf()	47
initialize()	48
class CI_Upload	48
class CI_Upload	48
data()	48
display_errors()	50
do_upload()	50
initialize()	51
class CI_Form_validation	51
class CI_Form_validation	51
error()	51
error_array()	52
error_string()	52
has_rule()	52
reset_validation()	52
run()	53
set_data()	53
set_error_delimiters()	53
set_message()	54
set_rules()	54
class CI_FTP	54
class CI_FTP	54
chdir()	55
chmod()	55
close()	55
connect()	56
delete_dir()	57
delete_file()	57
download()	57
list_files()	58
mirror()	58
mkdir()	59
move()	59
rename()	60
upload()	60
class CI_Image_lib	61
class CI_Image_lib	61
clear()	61
crop()	61
display_errors()	62
initialize()	63
resize()	63
rotate()	64
watermark()	65

class CI_Input.....	65
class CI_Input.....	65
\$raw_input_stream.....	65
cookie().....	65
get().....	66
get_post().....	66
get_request_header().....	67
input_stream().....	67
ip_address().....	68
is_ajax_request().....	68
is_cli_request().....	68
method().....	68
post().....	69
post_get().....	70
request_headers().....	70
server().....	71
set_cookie().....	71
user_agent().....	72
valid_ip().....	73
class CI_Lang.....	73
class CI_Lang.....	73
line().....	73
load().....	74
class CI_Loader.....	74
class CI_Loader.....	74
add_package_path().....	74
clear_vars().....	75
config().....	75
database().....	75
dbforge().....	76
dbutil().....	76
driver().....	77
file().....	78
get_package_paths().....	78
get_var().....	79
get_vars().....	79
helper().....	79
is_loaded().....	79
language().....	80
library().....	80
model().....	82
remove_package_path().....	83
vars().....	83
view().....	84
class CI_Migration.....	84

class CI_Migration.....	84
current.....	84
error_string().....	85
find_migrations().....	85
latest().....	85
version().....	85
 class CI_Output.....	 86
class CI_Output.....	86
\$parse_exec_vars.....	86
append_output().....	86
cache().....	86
_display().....	87
enable_profiler.....	87
get_content_type.....	88
get_header().....	88
get_output().....	89
set_content_type.....	89
set_header.....	90
set_output().....	90
set_profiler_sections().....	91
set_status_header().....	91
 class CI_Pagination.....	 92
class CI_Pagination.....	92
create_links().....	92
initialize().....	92
 class CI_Parser.....	 92
class CI_Parser.....	92
parse().....	92
parse_string().....	93
set_delimiters().....	93
 class CI_Security.....	 94
class CI_Security.....	94
entity_decode.....	94
get_csrf_hash().....	94
get_csrf_token_name().....	94
get_random_bytes().....	95
sanitize_filename().....	95
xss_clean().....	96
 class CI_Session.....	 96
class CI_Session.....	96
all_userdata().....	96
flashdata().....	96
_get().....	97
get_flash_keys().....	97

get_temp_keys()	97
&get_userdata()	97
has_userdata()	97
keep_flashdata()	98
mark_as_flash()	98
mark_as_temp()	98
sess_destroy()	98
sess_regenerate()	99
_set()	99
set_flashdata()	100
set_tempdata()	100
set_userdata()	100
tempdata()	100
unmark_flash()	100
unmark_temp()	100
unset_userdata()	101
userdata()	101
class CI_Table	101
class CI_Table	101
\$function	101
add_row()	102
clear()	102
generate()	103
make_columns()	103
set_caption()	104
set_empty()	104
set_heading()	105
set_template()	105
class CI_Trackback	106
class CI_Trackback	106
\$data	106
\$convert_ascii	106
convert_ascii()	106
convert_xml()	106
data()	107
display_errors()	107
extract_urls()	107
get_id()	108
limit_characters()	108
process()	108
receive()	109
send()	109
send_error()	109
send_success()	109
set_error()	110
validate_url()	110

class CI_Typography.....	110
class CI_Typography.....	110
\$protect_braced_quotes.....	110
auto_typography().....	111
format_characters().....	111
nl2br_except_pre().....	112
class CI_Unit_test.....	112
class CI_Unit_test.....	112
active().....	113
report().....	113
result().....	113
run().....	113
set_template().....	114
set_test_items().....	114
use_strict().....	115
URI Class.....	115
class CI_URI.....	115
assoc_to_uri().....	115
rsegment().....	115
rsegment_array().....	116
ruri_string().....	116
ruri_to_assoc().....	116
segment().....	117
segment_array().....	117
slash_rsegment().....	118
slash_segment().....	118
total_rsegments().....	119
total_segments().....	119
uri_string().....	119
uri_to_assoc().....	120
class CI_User_agent.....	121
class CI_User_agent.....	121
accept_charset().....	121
accept_lang().....	121
agent_string().....	122
browser().....	122
charsets().....	122
languages().....	122
mobile().....	123
is_browser().....	123
is_mobile().....	123
is_referral().....	124
is_robot().....	124
parse().....	125
platform().....	125



referrer()	125
robot()	125
version()	126
class CI_Xmlrpc	126
class CI_Xmlrpc	126
display_error()	126
display_response()	126
initialize()	126
method()	127
request()	127
send_error_message()	128
send_request()	128
server()	128
timeout()	129
class CI_Zip	129
class CI_Zip	129
\$compression_level	129
add_data()	130
add_dir()	131
archive()	131
clear_data()	131
download()	132
get_zip()	132
read_dir()	133
read_file()	134
class CI_DB_result	135
class CI_DB_result	135
custom_result_object()	135
custom_row_object()	135
data_seek()	135
field_data()	136
first_row()	136
free_result()	136
last_row()	136
list_fields()	137
next_row()	137
num_fields()	137
num_rows()	137
previous_row()	138
result()	138
result_array()	138
result_object()	138
row()	139
row_array()	139
row_object()	139

set_row()	140
unbuffered_row()	140
class CI_DB_query_builder	140
class CI_DB_query_builder	140
count_all_results()	140
dbprefix()	141
delete()	141
distinct()	141
empty_table()	142
flush_cache()	142
from()	142
get()	143
get_compiled_delete()	143
get_compiled_insert()	143
get_compiled_select()	144
get_compiled_update()	144
get_where()	144
group_by()	145
group_end()	145
group_start()	145
having()	146
insert()	146
insert_batch()	146
join()	147
like()	147
limit()	148
not_group_start()	148
not_like()	148
offset()	149
order_by()	149
or_group_start()	149
or_having()	150
or_like()	150
or_not_group_start()	150
or_not_like()	151
or_where()	151
or_where_in()	151
or_where_not_in()	152
replace()	152
reset_query()	152
select()	153
select_avg()	153
select_max()	153
select_min()	154
select_sum()	154
set()	154
set_dbprefix()	155

set_insert_batch()	155
set_update_batch()	155
start_cache()	156
stop_cache()	156
truncate()	156
update()	156
update_batch()	157
where()	157
where_in()	158
where_not_in()	158
class CI_DB_forge	159
class CI_DB_forge	159
add_column()	159
add_field()	159
add_key()	159
create_database()	160
create_table()	160
drop_column()	161
drop_database()	161
drop_table()	161
modify_column()	162
rename_table()	162
class CI_DB_utility	162
class CI_DB_utility	162
backup()	163
csv_from_result()	163
database_exists()	163
list_databases()	164
optimize_database()	164
optimize_table()	164
repair_table()	164
xml_from_result()	165
class CI_DB_driver	165
class CI_DB_driver	165
affected_rows()	165
cache_delete()	165
cache_delete_all()	166
cache_off()	166
cache_on()	166
cache_set_path()	166
call_function()	166
close()	167
compile_binds()	167
count_all()	167
db_connect()	168

db_pconnect()	168
db_select()	168
db_set_charset()	169
display_error()	169
elapsed_time()	169
escape()	170
escape_identifiers()	170
escape_like_str()	170
escape_str()	171
field_data()	171
field_exists()	171
initialize()	172
insert_string()	172
is_write_type()	172
last_query()	172
list_fields()	173
list_tables()	173
platform()	173
primary()	173
protect_identifiers()	174
query()	175
reconnect()	175
simple_query()	175
table_exists()	176
total_queries()	176
trans_complete()	176
trans_off()	176
trans_start()	176
trans_status()	177
trans_strict()	177
update_string()	177
version()	178
Array Helper	178
element()	178
elements()	179
random_element()	180
CAPTCHA Helper	180
create_captcha()	180
Cookie Helper	181
delete_cookie()	181
get_cookie()	182
set_cookie()	182
Date Helper	183
date_range()	183
days_in_month()	184

gmt_to_local()	184
human_to_unix()	185
local_to_gmt()	185
mdate()	186
mysql_to_unix()	186
nice_date()	187
now()	187
standard_date()	187
timespan()	187
timezones()	188
timezone_menu()	189
unix_to_human()	189
Directory Helper	191
directory_map()	191
Download Helper	193
force_download()	193
File Helper	193
delete_files()	193
get_dir_file_info()	194
get_file_info()	194
get_filenames()	195
get_mime_by_extension()	195
octal_permissions()	196
read_file()	196
symbolic_permissions()	196
write_file()	197
Form Helper	198
form_button()	198
form_checkbox()	199
form_close()	200
form_dropdown()	200
form_error()	202
form_fieldset()	202
form_fieldset_close()	203
form_hidden()	204
form_input()	205
form_label()	207
form_multiselect()	207
form_open()	208
form_open_multipart()	209
form_password()	210
form_prep()	210
form_radio()	210
form_reset()	211
form_submit()	211

form_textarea()	212
form_upload()	212
set_checkbox()	212
set_radio()	213
set_select()	214
set_value()	214
validation_errors()	215
HTML Helper	216
br()	216
doctype()	216
heading()	217
img()	218
link_tag()	219
meta()	220
nbs()	222
ol()	222
ul()	222
Inflector Helper	225
camelize()	225
humanize()	225
is_countable()	226
plural()	226
singular()	226
underscore()	227
Language Helper	227
lang()	227
Number Helper	228
byte_format()	228
Path Helper	229
set_realpath()	229
Security Helper	230
do_hash()	230
encode_php_tags()	230
sanitize_filename()	230
strip_image_tags()	231
xss_clean()	231
String Helper	232
alternator()	232
increment_string()	232
quotes_to_entities()	233
random_string()	233
reduce_double_slashes()	234

reduce_multiples()	234
repeater()	235
strip_quotes()	235
strip_slashes()	235
trim_slashes()	236
Text Helper	236
ascii_to_entities()	236
character_limiter()	237
convert_accented_characters()	237
ellipsize()	238
highlight_code()	239
highlight_phrase()	239
word_censor()	240
word_limiter()	240
word_wrap()	241
Typography Helper	242
auto_typography()	242
entity_decode()	242
nl2br_except_pre()	243
URL Helper	243
anchor()	243
anchor_popup()	244
auto_link()	245
base_url()	246
current_url()	246
index_page()	247
mailto()	247
prep_url()	247
redirect()	248
safe_mailto()	249
site_url()	249
uri_string()	250
url_title()	251
XML Helper	252
xml_convert()	252

## Crear clases auxiliares

---

`get_instance()`

**object** `get_instance()`

Referencia a la instancia de su controlador

### Tipo de devolución

---

## CI\_Controller

Cualquier clase que instancia en los métodos de su controlador puede acceder a los recursos nativos de CodeIgniter usando la función **get\_instance()**, esta función devuelve el objeto CodeIgniter principal

Normalmente, para llamar a cualquiera de los métodos disponibles, CodeIgniter requiere que use **\$this**:

```
$this->load->helper('url');  
$this->load->library('session');  
$this->config->item('base_url');  
// etc.
```

**\$this**, sin embargo, solo funciona dentro de sus controladores, sus modelos o sus vistas, si desea utilizar las clases de CodeIgniter desde sus propias clases personalizadas, puede hacerlo de la siguiente manera:

Primero, asigne el objeto CodeIgniter a una variable:

```
$CI = &get_instance();
```

Una vez que haya asignado el objeto a una variable, usará esa variable en lugar de **\$this**:

```
$CI = &get_instance();  
  
$CI->load->helper('url');  
$CI->load->library('session');  
$CI->config->item('base_url');  
// etc.
```

Si va a usar **get\_instance()** dentro de otra clase, entonces sería mejor si lo asigna a una propiedad, de esta forma, no necesitará llamar a **get\_instance()** en cada método

### Ejemplo

---

```
class Example {  
  
    protected $CI;  
  
    // Usar el constructor, cuando no se puede llamar directamente en la definición de la propiedad
```



```

public function __construct()
{
    // Asignar el súper objeto de CodeIgniter
    $this->CI =& get_instance();
}
public function foo()
{
    $this->CI->load->helper('url');
    redirect();
}

public function bar()
{
    $this->CI->config->item('base_url');
}
}

```

En el ejemplo anterior, ambos métodos **foo()** y **bar()** funcionarán después de crear una instancia de la clase Example, sin la necesidad de llamar a **get\_instance()** en cada uno de ellos

## Funciones Comunes

---

### config\_item()

**mixed config\_item(string \$key)**

#### Descripción

---

La biblioteca de configuración es la forma preferida de acceder a la información de configuración, sin embargo, **config\_item()** se puede usar para recuperar claves individuales, ver la documentación de **Config Library** para más información

#### Parámetros

---

##### key

Clave del elemento de configuración

#### Valores devueltos

---

Valor de la clave de configuración o NULL si no se encuentra

### function\_usable()

**bool function\_usable(string \$function\_name)**

#### Descripción

---

Devuelve TRUE si existe una función y es utilizable, FALSE de lo contrario

Esta función ejecuta una comprobación **function\_exists()** y si la extensión **Suhosin** <<http://www.hardened-php.net/suhosin/>> está cargada, comprueba si no desactiva la función que se está comprobando

Es útil si desea verificar la disponibilidad de funciones como **eval()** y **exec()**, que son peligrosas y pueden estar deshabilitadas en servidores con políticas de seguridad altamente restrictivas

#### Parámetros

---

##### **function\_name**

Nombre de la función

#### Valores devueltos

---

TRUE si la función se puede usar, FALSE si no

#### **Nota**

Esta función se introdujo porque Suhosin terminó la ejecución del script, pero resultó ser un error, una solución ha estado disponible durante algún tiempo (versión 0.9.34), pero desafortunadamente aún no se ha lanzado

#### **get\_mimes()**

##### **array get\_mimes()**

#### Descripción

---

Esta función devuelve una referencia a la array **MIME** desde **application/config/mimes.php**

#### Valores devueltos

---

Array asociativo de tipos de archivos

#### **html\_escape()**

##### **mixed html\_escape(mixed \$var)**

#### Descripción

---

Esta función actúa como un alias para la función **htmlspecialchars()** nativa de PHP, con la ventaja de poder aceptar un array de strings, es útil para evitar **Cross Site Scripting** (XSS)

#### Parámetros

---

##### **var**

Variable para escapar, string o array

#### Valores devueltos

---

String HTML escapada

#### **is\_cli()**

##### **bool is\_cli()**

#### Descripción

---

Devuelve TRUE si la aplicación se ejecuta a través de la línea de comandos y FALSE si no

## Nota

Esta función comprueba si el valor `PHP_SAPI` es 'cli' o si se define la constante `STDIN`

## is\_https()

### bool is\_https()

#### Descripción

---

Devuelve TRUE si se utiliza una conexión segura **HTTPS** y FALSE en cualquier otro caso, incluidas las solicitudes que no son HTTP

#### Devuelve

---

TRUE si actualmente usa HTTP-con-SSL y FALSE si no lo usa

## is\_php()

### bool is\_php(string \$version)

#### Descripción

---

Determina si la versión de PHP utilizada es mayor que el número de versión proporcionado

#### Parámetros

---

##### version

Número de versión

#### Valores devueltos

---

Devuelve TRUE si la versión de PHP instalada es igual o mayor que el número de versión proporcionado, FALSE si la versión de PHP instalada es menor que el número proporcionado

#### Ejemplo

---

```
if (is_php('5.3'))
{
    $str = quoted_printable_encode($str);
}
```

## is\_really\_writable()

### bool is\_really\_writable(string \$file)

#### Descripción

---

Esta función determina si un archivo es realmente modificable al intentar escribirlo primero, por lo general, solo se recomienda en plataformas donde esta información puede no ser confiable

#### Parámetros

---

##### file

Ruta del archivo

#### Valores devueltos

---

TRUE si la ruta es escribible, FALSE si no, **is\_writable ()** devuelve TRUE en los servidores de Windows cuando realmente no puede escribir en el archivo ya que el sistema operativo informa a PHP como FALSE solamente si el atributo de solo lectura está marcado

#### Ejemplo

---

```
if (is_really_writable('file.txt'))
{
    echo "Puedo escribir lo que quiera";
}
else
{
    echo "No se puede escribir en el archivo";
}
```

#### Nota

Vea también el error PHP # 54709 para más información

#### remove\_invisible\_characters()

**string remove\_invisible\_characters(string \$str [ , bool \$url\_encoded = TRUE ])**

#### Descripción

---

Esta función evita insertar caracteres NULL entre caracteres ASCII, como **Java\0script**

#### Parámetros

---

##### str

String de entrada

##### url\_encoded

TRUE elimina también los caracteres codificados en URL

#### Valores devueltos

---

String desinfectada

#### Ejemplo

---

```
remove_invisible_characters('Java\0script'); // Retorna 'Javascript'
```

#### set\_status\_header()

**void set\_status\_header(int \$code [, string \$text = " ])**

#### Descripción

---

Le permite configurar manualmente un encabezado de estado del servidor

#### Parámetros

---

##### code

Código de estado de respuesta HTTP

##### text

Un mensaje personalizado para configurar con el código de estado

valores devueltos

---

void

Ejemplo

---

```
set_status_header(401); // Establece el encabezado como: no autorizado
```

Vea: <http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html> para una lista completa de encabezados

## Funciones de compatibilidad

---

**password\_get\_info()**

**array password\_get\_info(string \$hash)**

Descripción

---

Información sobre la contraseña hash, puede obtener más información consultando el manual de PHP para **password\_get\_info()**

Parámetros

---

**hash**

Hash de la contraseña

Valores devueltos

---

Información sobre la contraseña hash

**password\_hash()**

**string password\_hash(string \$password, int \$algo [, array \$options = array() ])**

Descripción

---

Consulte el manual de PHP para **password\_hash()**

Parámetros

---

**password**

Contraseña en texto plano

**algo**

Algoritmo hashing

**options**

Opciones de hashing

Valores devueltos

---

Contraseña hash o FALSE en caso de error

**Nota**

A menos que proporcione su propia, y válida, salt, esta función tiene una dependencia adicional de una fuente CSPRNG (Generador de números pseudo-aleatorios) disponible

Cada una de las siguientes opciones satisfaría que:

`mcrypt_create_iv()` con `MCRYPT_DEV_RANDOM` - `openssl_random_pseudo_bytes()` - `/dev/random` - `/dev/urandom`

`password_needs_rehash()`

**bool password\_needs\_rehash(string \$hash, int \$algo [, array \$options ])**

#### Descripción

---

Consulte el manual de PHP para **password\_needs\_rehash()**

#### Parámetros

---

##### hash

Hash de la contraseña

##### algo

Algoritmo Hashing

##### options

Opciones de hashing

#### Valores devueltos

---

TRUE si el hash se debe volver a generar para coincidir con el algoritmo y las opciones dados, FALSE en caso contrario

`password_verify()`

**bool password\_verify(string \$password, string \$hash)**

#### Descripción

---

Comprueba que la contraseña coincida con un hash, consulte el manual de PHP para **password\_verify()**

#### Parámetros

---

##### password

Contraseña de texto plano

##### hash

Hash de la contraseña

#### Valores devueltos

---

TRUE si la contraseña coincide con el hash, FALSE si no

`hash_equals()`

**string hash\_equals(string \$known\_string, string \$user\_string)**

## Descripción

---

Consulte el manual de PHP para **hash\_equals()**

## Parámetros

---

### **known\_string**

Cadena conocida

### **user\_string**

Cadena proporcionada por el usuario

## Valores devueltos

---

TRUE si las strings coinciden, de lo contrario FALSE

## hash\_pbkdf2()

```
string hash_pbkdf2(string $algo, string $password, string $salt, int $iterations [, int $length = 0  
[, bool $raw_output = FALSE]])
```

## Descripción

---

Consulte el manual de PHP para **hash\_pbkdf2()**

## Parámetros

---

### **algo**

Algoritmo Hashing

### **password**

Contraseña

### **salt**

Salt hash

### **iterations**

Número de iteraciones a realizar durante la derivación

### **length**

Longitud del string de salida

### **raw\_output**

Si se deben devolver datos binarios en bruto

## Valores devueltos

---

Clave derivado de contraseña o FALSE en caso de error

## mb\_strlen()

```
string mb_strlen(string $str [, string $encoding = NULL])
```

en: Funciones compatibilidad - ámbito global

## Descripción

---

Consulte el manual de PHP para **mb\_strlen()**

## Parámetros

---

### **str**

String de entrada

## encoding

Conjunto de caracteres

### Valores devueltos

---

Número de caracteres en la string de entrada o FALSE si falla

## mb\_strpos()

```
mixed mb_strpos(string $haystack, string $needle [, int $offset = 0 [, string $encoding = NULL] ])
```

### Descripción

---

Consulte el manual de PHP para **mb\_strpos()**

### Parámetros

---

#### haystack

Cadena para buscar en

#### needle

Parte de la string para buscar

#### offset

Compensación de búsqueda

#### encoding

Conjunto de caracteres

### Valores devueltos

---

Posición del carácter numérico donde se encontró **\$needle** o FALSE si no se encontró

## mb\_substr()

```
string mb_substr(string $str, int $start [, int $length = NULL [, string $encoding = NULL]])
```

### Descripción

---

Consulte el manual de PHP para **mb\_substr()**

### Parámetros

---

#### str

String de entrada

#### start

Posición del primer personaje

#### length

Número máximo de caracteres

#### encoding

Conjunto de caracteres

### Valores devueltos

---

Porción de **\$str** especificada por **\$start** y **\$length** o FALSE cuando falla



**array\_column()**

**array** **array\_column**(**array** \$array, **mixed** \$column\_key [, **mixed** \$index\_key = **NULL**])

---

#### Descripción

Consulte el manual de PHP para **array\_column()**

---

#### Parámetros

##### **array**

Un array multidimensional o un array de objetos desde el que extraer una columna de valores

##### **column\_key**

La columna de valores a devolver

##### **index\_key**

La columna a usar como los índices/claves para el array devuelto

---

#### Valores devueltos

Devuelve un array de valores que representa una sola columna del array de entrada

**hex2bin()**

**string** **hex2bin**(**array** \$data)

---

#### Descripción

Consulte el manual de PHP para **hex2bin()**

---

#### Parámetros

##### **data**

La representación hexadecimal de datos

---

#### Valores devueltos

Devuelve la representación binaria de los datos dados o FALSE en caso de error

---

## Manejo de errores

---

**log\_message()**

**void** **log\_message**(**string** \$level, **string** \$message)

---

#### Descripción

Esta función le permite escribir mensajes en sus archivos de registro, debe proporcionar uno de los tres "niveles" en el primer parámetro, indicando qué tipo de mensaje es (**error**, **debug** o **info**), con el mensaje mismo en el segundo parámetro

---

#### Parámetros

##### **level**

Nivel de registro: 'error', 'debug' o 'info'

## message

Mensaje para iniciar sesión

### valores devueltos

---

void

### Ejemplo

---

```
if ($alguna_variable == '')
{
    log_message('error', 'alguna_variable no contenía valor.');
```

```
}
else
{
    log_message('debug', 'alguna_variable está bien establecida');
```

```
}

log_message('info', 'El propósito de alguna_variable es proveer algún valor.');
```

Hay tres tipos de mensajes:

1. **Error Messages:** Estos son errores reales, como errores de PHP o de usuario
2. **Debug Messages:** Estos son mensajes que ayudan a la depuración, por ejemplo, si se ha inicializado una clase, puede registrarla como información de depuración
3. **Informational Messages:** Estos son los mensajes de menor prioridad, simplemente dan información con respecto a algún proceso

### Nota

Para que el archivo de registro se escriba realmente, el directorio **logs/** debe ser de escritura, además, debe establecer la clave **\$config['log\_threshold']** en **application/config/config.php**, puede, por ejemplo, solo querer que se registren los mensajes de error, y no los otros dos tipos, si lo configura en cero, el registro se desactivará

### show\_404()

```
void show_404(string $page = "", bool $log_error = TRUE)
```

### Descripción

---

Esta función mostrará el mensaje de error 404 que se le proporcionó utilizando la plantilla de error adecuada para su ejecución:

```
application/views/errors/html/error_404.php
//o
application/views/errors/cli/error_404.php
```

### Parámetros

---

## page

String URI, la función espera que la string que se le pasa sea la ruta del archivo a la página que no se encuentra, el código de estado de salida se establecerá en **EXIT\_UNKNOWN\_FILE**, tenga en cuenta que CodeIgniter muestra automáticamente el mensaje 404 si no se encuentran los controladores

## log\_error

Si debe registrarse el error, CodeIgniter registra automáticamente cualquier llamada a **show\_404()**, establecer el segundo parámetro opcional en FALSE omitirá el registro

### valores devueltos

---

void

## show\_error()

**void show\_error(mixed \$message, int \$status\_code, string \$heading = 'An Error Was Encountered')**

### Descripción

---

Esta función mostrará el mensaje de error proporcionado utilizando la plantilla de error apropiada para su ejecución:

```
application/views/errors/html/error_general.php
//o
application/views/errors/cli/error_general.php
```

### Parámetros

---

#### message

Mensaje de error

#### status\_code

Código de estado de respuesta HTTP, el parámetro opcional **\$status\_code** determina qué código de estado HTTP se debe enviar con el error, si **\$status\_code** es menor que 100, el código de estado HTTP se establecerá en 500, y el código de estado de salida se establecerá en **status\_code + EXIT\_\_AUTO\_MIN**, si ese valor es mayor que **EXIT\_\_AUTO\_MAX**, o si **\$status\_code** es 100 o superior, el código de estado de salida se establecerá en **EXIT\_ERROR**, puede consultar en **application/config/constants.php** para más detalles

#### heading

Encabezado de la página de error

### valores devueltos

---

void

## class CI\_Benchmark

---

### class CI\_Benchmark

**class CI\_Benchmark**

`elapsed_time()`

`string elapsed_time([ string $point1 = " ", string $point2 = " ", int $decimals = 4])`

---

#### Descripción

Calcula y devuelve la diferencia de tiempo entre dos puntos benchmark, si el primer parámetro está vacío, esta función devuelve la pseudovariable `{elapsed_time}`, esto permite que se muestre el tiempo total de ejecución en una plantilla. La clase **Output Class** intercambiará el valor real de esta variable

---

#### Parámetros

##### **point1**

Un punto marcado particular

##### **point2**

Un punto marcado particular

##### **decimals**

Número de decimales para precisión

---

#### Valores devueltos

Tiempo transcurrido

`mark()`

`void mark(string $name)`

---

#### Descripción

Establece un marcador **benchmark**

---

#### Parámetros

##### **name**

El nombre que desea asignar a su marcador

---

#### Valores devueltos

void

`memory_usage()`

`string memory_usage()`

---

#### Descripción

Devuelve el marcador `{memory_usage}`, esto permite colocarlo en cualquier lugar de una plantilla sin que la memoria se calcule hasta el final, **Output Class** intercambiará el valor real de esta variable

---

#### Valores devueltos

String con el valor de `{memory_usage}`

## class CI\_Cache

---

class CI\_Cache

**class CI\_Cache**

cache\_info()

**mixed cache\_info()**

### Descripción

---

Este método devolverá información de todo el caché

### Valores devueltos

---

Información de toda la Base de datos de caché

### Ejemplo

---

```
var_dump($this->cache->cache_info());
```

### Nota

La información devuelta y la estructura de los datos dependen del adaptador que se está utilizando

clean()

**bool clean()**

### Descripción

---

Este método '**limpia**' todo el caché, si la eliminación de los archivos de caché falla, el método devolverá FALSE

### Valores devueltos

---

TRUE en caso de éxito, FALSE en caso de error

### Ejemplo

---

```
$this->cache->clean();
```

decrement()

**mixed decrement(string \$id [, int \$offset = 1])**

### Descripción

---

Realiza la disminución atómica de un valor almacenado

### Parámetros

---

**id**

ID de caché

## offset

Intervalo/valor para reducir

### Valores devueltos

---

Nuevo valor si tiene éxito, FALSE si falla

### Ejemplo

---

```
// 'iterator' tiene un valor de 6
$this->cache->decrement('iterator'); // 'iterator' ahora es 5
$this->cache->decrement('iterator', 2); // 'iterator' ahora es 3
```

## delete()

### bool delete(string \$id)

#### Descripción

---

Este método elimina un elemento específico del caché

#### Parámetros

---

##### id

Nombre del elemento en caché

#### Valores devueltos

---

TRUE si es eliminado, FALSE si falla

### Ejemplo

---

```
$this->cache->delete('cache_item_id');
```

## get()

### mixed get(string \$id)

#### Descripción

---

Este método intenta recuperar un elemento de la memoria caché

#### Parámetros

---

##### id

Nombre del elemento en caché

#### Valores devueltos

---

Valor del elemento o FALSE si no se encuentra

### Ejemplo

---

```
$foo = $this->cache->get('my_cache_item');
```

`get_metadata()`

**mixed** `get_metadata(string $id)`

---

#### Descripción

Este método devolverá información detallada sobre un elemento específico en la memoria caché

---

#### Parámetros

##### **id**

Nombre del elemento de caché

---

#### Valores devueltos

Metadatos del elemento en caché

---

#### Ejemplo

```
var_dump($this->cache->get_metadata('my_cached_item'));
```

#### **Nota**

La información devuelta y la estructura de los datos dependen del adaptador que se está utilizando

`increment()`

**mixed** `increment(string $id [, int $offset = 1])`

---

#### Descripción

Realiza el incremento atómico de un valor almacenado

---

#### Parámetros

##### **id**

ID de caché

##### **offset**

Intervalo/valor para agregar

---

#### Valores devueltos

Nuevo valor si tiene éxito, FALSE si falla

---

#### Ejemplo

```
// 'iterator' tiene un valor de 2
$this->cache->increment('iterator'); // 'iterator' ahora es 3
$this->cache->increment('iterator', 3); // 'iterator' ahora es 6
```

`is_supported()`

**bool** `is_supported(string $driver)`

---

#### Descripción

Averiguar si el controlador de caché **\$driver** es soportado, este método se llama automáticamente al acceder a los

controladores a través de **`$this->cache->get()`**, sin embargo, si se utiliza los controladores individuales, asegúrese de llamar a este método para asegurarse de que el controlador sea compatible en el entorno de alojamiento

---

#### Parámetros

##### driver

El nombre del controlador de almacenamiento en caché

---

#### Valores devueltos

TRUE si es soportado, FALSE si no

---

#### Ejemplo

```
if ($this->cache->apc->is_supported())
{
    if ($datos = $this->cache->apc->get('my_cache'))
    {
        // hacer cosas.
    }
}
```

#### save()

**string save(string \$id, mixed \$data [, int \$ttl = 60 [, bool \$raw = FALSE]])**

---

#### Descripción

Este método guardará un elemento en el caché, si el almacenamiento falla, el método devolverá FALSE

---

#### Parámetros

##### id

Nombre del elemento de caché

##### data

Los datos para guardar

##### ttl

Time To Live, en segundos (por defecto 60)

##### raw

Si se debe almacenar el valor bruto

---

#### Valores devueltos

TRUE si tiene éxito, FALSE si falla

---

#### Ejemplo

```
$this->cache->save('cache_item_id', 'datos_al_cache');
```

#### Nota

El parámetro **\$raw** solo es utilizado por **APC** y **Memcache**, para permitir el uso de **increment()** y **decrement()**



# class CI\_Calendar

---

class CI\_Calendar

class CI\_Calendar

adjust\_date()

array adjust\_date(int \$month, int \$year)

---

## Descripción

Este método asegura que tiene un mes/año válido, por ejemplo, si envía 13 como mes, el año aumentará y el mes se convertirá en enero

---

## Parámetros

### month

Mes

### year

Año

---

## Valores devueltos

Un array asociativa que contiene mes y año

```
print_r($this->calendar->adjust_date(13, 2017));
```

Resultado:

```
Array
(
    [month] => '01'
    [year]  => '2018'
)
```

`default_template()`

## `array default_template()`

### Descripción

---

Establece la plantilla predeterminada, este método se usa cuando no ha creado su propia plantilla

### Valores devueltos

---

Un array con los valores de plantilla

`generate()`

## `string generate([ int $year = " ", int $month = " ", array $data = array() ])`

### Descripción

---

Genera el calendario

### Parámetros

---

#### **year**

Año

#### **month**

Mes

#### **data**

Datos que se mostrarán en las celdas del calendario

### Valores devueltos

---

Calendario con formato HTML

`get_day_names()`

## `array get_day_names(string $day_type = "")`

### Descripción

---

Devuelve un array con los nombres de días (Sunday, Monday, etc.) según el tipo proporcionado, opciones: largo, corto, abreviado, si no se proporciona **\$day\_type**, o si se proporciona un tipo no válido, devolverá el estilo "**abreviado**"

### Parámetros

---

#### **day\_type**

'long', 'short' o 'abr'

### Valores devueltos

---

Array con los nombres de días

`get_month_name()`

## `string get_month_name(int $month)`

### Descripción

---

Genera un texto con el nombre del mes correspondiente al número de mes proporcionado

## Parámetros

---

### **month**

Mes

## Valores devueltos

---

Nombre del mes

## **get\_total\_days()**

**int get\_total\_days(int \$month, int \$year)**

## Descripción

---

El número de días en un mes dado

### **Nota**

Este método es un alias para la función Date Helper **days\_in\_month()**

## Parámetros

---

### **month**

Mes

### **year**

Año

## Valores devueltos

---

El número de días en el mes especificado

## Ejemplo

---

```
echo $this->calendar->get_total_days(2, 2016); // Presenta: 29
```

## **initialize()**

**object initialize([ array \$config = array()])**

## Descripción

---

Inicia las preferencias del calendario

## Parámetros

---

### **config**

Array asociativo con las preferencias de visualización

## Valores devueltos

---

Instancia CI\_Calendar (method chaining)

`parse_template()`

**object** `parse_template()`

---

#### Descripción

Recolecta los datos dentro de la plantilla **{pseudovariables}** utilizada para mostrar el calendario

---

#### Valores devueltos

Instancia `CI_Calendar` (method chaining)

---

## class `CI_Config`

---

`class CI_Config`

**class** `CI_Config`

`$config`

**\$config**

Array de todos los valores de configuración cargados

`$is_loaded`

**\$is\_loaded**

Array de todos los archivos de configuración cargados

`base_url()`

**string** `base_url()`

---

#### Descripción

Este método recupera la URL de su sitio, más una ruta opcional, como una hoja de estilo o una imagen, a este método se accede normalmente a través de las funciones correspondientes en el **URL Helper**

---

#### Valores devueltos

URL base

`item()`

**mixed** `item(string $item [, string $index = ''])`

---

#### Descripción

Obtiene un elemento de archivo de configuración

---

#### Parámetros

**item**

Nombre del elemento de configuración

**index**

Nombre del índice

## Valores devueltos

---

Valor de elemento de configuración o NULL si no se encuentra

### load()

```
bool load([ string $file = "" [, bool $use_sections = FALSE [, bool $fail_gracefully = FALSE]])
```

#### Descripción

---

Carga un archivo de configuración

#### Parámetros

---

##### file

Nombre del archivo de configuración

##### use\_sections

Si los valores de configuración deben cargarse en su propia sección (índice de la array de configuración principal)

##### fail\_gracefully

Si se devuelve FALSE o se muestra un mensaje de error

#### Valores devueltos

---

TRUE en el éxito, FALSE si falla

### set\_item()

```
void set_item(string $item, string $value)
```

#### Descripción

---

Establece un elemento de archivo de configuración para el valor especificado

#### Parámetros

---

##### item

Nombre del elemento de configuración

##### value

Valor del elemento de configuración

#### valores devueltos

---

void

### site\_url()

```
string site_url()
```

#### Descripción

---

Este método recupera la URL de su sitio, junto con el valor de "**index**" que ha especificado en el archivo de configuración, a este método se accede normalmente a través de las funciones correspondientes en el **URL Helper**

#### Valores devueltos

---

URL del sitio

slash\_item()

**mixed slash\_item(string \$item)**

---

#### Descripción

Este método es idéntico al método **item()** excepto que agrega una barra inclinada al final del elemento

---

#### Parámetros

##### **item**

Nombre del elemento

---

#### Valores devueltos

Valor del elemento de configuración con una barra inclinada o NULL si no se encuentra

system\_url()

**string system\_url()**

---

#### Descripción

Este método recupera la URL de su sistema/directorio CodeIgniter

#### **Nota**

Este método está **DESACONSEJADO** porque fomenta el uso de prácticas de codificación inseguras, su sistema/directorio no debe ser de acceso público

---

#### Valores devueltos

URL que apunta a su sistema/directorio de CI

---

## class CI\_Email

class CI\_Email

**class CI\_Email**

attach()

**object attach(string \$filename [, string \$disposition = " [" , string \$newname = NULL [, string \$mime = "]]])**

---

#### Descripción

Le permite enviar un archivo adjunto, coloque la ruta/nombre del archivo en el primer parámetro, para archivos adjuntos múltiples, use el método varias veces

---

#### Parámetros

##### **filename**

Nombre de archivo

##### **disposition**

'disposición' del archivo adjunto, la mayoría de los clientes de correo electrónico toman su propia decisión independientemente de la especificación MIME utilizada

vea: <https://www.iana.org/assignments/cont-disp/cont-disp.xhtml>

### **newname**

Nombre de archivo personalizado para usar en el correo electrónico

### **mime**

Tipo MIME a usar (útil para datos almacenados en búfer)

### **Valores devueltos**

---

Instancia de CI\_Email (method chaining)

### **Ejemplo**

---

```
$this->email->attach('/path/to/photo1.jpg');  
$this->email->attach('/path/to/photo2.jpg');  
$this->email->attach('/path/to/photo3.jpg');
```

Para usar la disposición predeterminada, archivo adjunto, deje en blanco el segundo parámetro, de lo contrario, use una disposición personalizada:

```
$this->email->attach('imagen.jpg', 'inline');
```

También puede usar una URL:

```
$this->email->attach('http://ejemplo.com/nombre_de_archivo.pdf');
```

Si desea usar un nombre de archivo personalizado, puede usar el tercer parámetro:

```
$this->email->attach('nombre_de_archivo.pdf', 'attachment', 'report.pdf');
```

Si necesita utilizar un string de búfer en lugar de un archivo físico real, puede usar el primer parámetro como búfer, el tercer parámetro como nombre de archivo y el cuarto parámetro como tipo MIME:

```
$this->email->attach($buffer, 'attachment', 'report.pdf', 'application/pdf');
```

## **attachment\_cid()**

**string attachment\_cid(string \$filename)**

### **Descripción**

---

Establece y devuelve **Content-ID** de un archivo adjunto, que le permite insertar un archivo adjunto en línea, imagen, en HTML, el primer parámetro debe ser el nombre del archivo ya adjunto

### **Parámetros**

---

#### **filename**

Archivo adjunto existente

### **Valores devueltos**

---

Adjunto Content-ID o FALSE si no se encuentra

## Ejemplo

```
$nom_archivo = '/img/foto1.jpg';
$this->email->attach($nom_archivo);
foreach ($lista as $direccion)
{
    $this->email->to($direccion);
    $cid = $this->email->attachment_cid($nom_archivo);
    $this->email->message('');
    $this->email->send();
}
```

### Nota

Content-ID debe ser único en cada correo electrónico, debe volver a generarlo para que sea único

## bcc()

**object bcc(mixed \$bcc [, int \$limit = ''])**

### Descripción

Establece las direcciones de correo electrónico de **BCC**, al igual que el método **to()** puede ser un solo correo, una lista delimitada por comas o un array

### Parámetros

#### bcc

Cadena delimitada por comas o un array de direcciones de correo electrónico

#### limit

Número máximo de correos para enviar por lote, si se establece **\$limit** se habilitará el modo **"batch"**, modo por lotes, que enviará los correos en lotes sin que cada lote exceda el valor especificado en **\$limit**

### Valores devueltos

Instancia de CI\_Email (method chaining)

## cc()

**object cc(mixed \$cc)**

### Descripción

Establece las direcciones de correo electrónico **CC**, al igual que en **"to"**, puede ser un solo correo electrónico, una lista delimitada por comas o un array

### Parámetros

#### cc

Cadena delimitada por comas o un array de direcciones de correo electrónico

### Valores devueltos

Instancia de CI\_Email (method chaining)

## clear()

**object clear([ bool \$clear\_attachments = FALSE])**



## Descripción

---

Inicializa todas las variables de correo electrónico a un estado vacío, este método está destinado a utilizarse si ejecuta el método de envío de correo electrónico en bucle, lo que permite que los datos se restablezcan entre ciclos

## Parámetros

---

### **clear\_attachments**

Si borrar o no los archivos adjuntos

## Valores devueltos

---

Instancia de CI\_Email (method chaining)

## Ejemplo

---

```
foreach ($lista as $nombre => $direccion)
{
    $this->email->clear();

    $this->email->to($direccion);
    $this->email->from('su@ejemplo.com ');
    $this->email->subject('Aquí está su información'. $nombre);
    $this->email->message('Hola'. $nombre. 'Aquí está la información que solicitó.');
```

Si establece el parámetro en TRUE, también se borrarán los archivos adjuntos:

```
$this->email->clear(TRUE);
```

## from()

**object from(string \$from [, string \$name = '' [, string \$return\_path = NULL]])**

## Descripción

---

Establece la dirección de correo electrónico, el nombre de la persona que lo envía y la dirección de devolución

## Parámetros

---

### **from**

Dirección de correo electrónico

### **name**

Nombre para mostrar

### **return\_path**

Dirección de correo electrónico en caso de devolución, si el correo no se ha entregado

## Valores devueltos

---

Instancia de CI\_Email (method chaining)

## Ejemplo

---

Establece la dirección de correo electrónico y el nombre de la persona que lo envía:

```
$this->email->from('mi@ejemplo.com', 'Mi nombre');
```

También puede establecer un Return-Path para ayudar a redirigir el correo no entregado:

```
$this->email->from('mi@ejemplo.com', 'Mi nombre', 'email_devuelto@ejemplo.com');
```

### Nota

Return-Path no se puede usar si ha configurado **'smtp'** como su protocolo

## message()

**object message(string \$body)**

### Descripción

Establece el cuerpo del mensaje del correo electrónico

### Parámetros

#### body

Cuerpo del mensaje del correo

### Valores devueltos

Instancia de CI\_Email (method chaining)

### Ejemplo

```
$this->email->message('Este es mi mensaje');
```

## print\_debugger()

**string print\_debugger([ array \$include = array('headers', 'subject', 'body')])**

### Descripción

Devuelve un string que contiene los mensajes del servidor, los encabezados y el mensaje del correo electrónico, útil para la depuración

### Parámetros

#### include

Opcional, la parte del mensaje a imprimir, las opciones son: **'headers'** (encabezados), **'subject'** (asunto), **'body'** (cuerpo)

### Valores devueltos

Datos de depuración formateados

### Ejemplo

```
// Necesita pasar FALSE en el método send() para que los datos no se borren,  
// en caso contrario print_debugger() no tendría nada que mostrar.  
$this->email->send(FALSE);  
  
// Solo imprimirá los encabezados de correo electrónico, excluyendo el mensaje,  
// asunto y cuerpo  
$this->email->print_debugger(array('headers'));
```

### Nota

De manera predeterminada, se imprimirán todos los datos en bruto

## reply\_to()

**object reply\_to(string \$replyto [, string \$name = ''])**

### Descripción

---

Establece la dirección de respuesta, si no se proporciona la información, se utiliza la información del método **from()**

### Parámetros

---

#### replyto

Dirección de correo para respuesta

#### name

Nombre para mostrar para la dirección de correo de respuesta

### Valores devueltos

---

Instancia de CI\_Email (method chaining)

### Ejemplo

---

```
$this->email->reply_to('mi@ejemplo.com', 'Mi nombre');
```

## send()

**bool send([ bool \$auto\_clear = TRUE])**

### Descripción

---

El método de envío de correo electrónico

### Parámetros

---

#### auto\_clear

Si borrar los datos del mensaje automáticamente

### Valores devueltos

---

TRUE en el éxito, FALSE si falla

### Ejemplo

---

```
if ( ! $this->email->send()  
{  
    // Generar error  
}
```

Este método borrará automáticamente todos los parámetros si la solicitud tuvo éxito, para detener este comportamiento pase FALSE:

```
if ($this->email->send(FALSE))  
{  
    // Los parámetros no se borran  
}
```

### Nota

Para utilizar el método **print\_debugger()** debe evitar borrar los parámetros del correo electrónico

## set\_alt\_message()

**object set\_alt\_message(string \$str)**

### Descripción

Establece el cuerpo alternativo del correo electrónico

### Parámetros

#### str

Cuerpo alternativo del mensaje de correo electrónico, si envía un correo en formato HTML , le permite especificar un mensaje alternativo sin formato HTML que se agrega al string de encabezado para las personas que no aceptan correo en HTML, si no configura su propio mensaje, CodeIgniter extraerá el mensaje de su correo en HTML y quitará las etiquetas

### Valores devueltos

Instancia de CI\_Email (method chaining)

### Ejemplo

```
$this->email->set_alt_message('Este es el mensaje alternativo si no acepta HTML');
```

## set\_header()

**object set\_header(string \$header, string \$value)**

### Descripción

Agrega encabezados adicionales al correo electrónico

### Parámetros

#### header

Nombre del encabezado

#### value

Valor de encabezado

### Valores devueltos

Instancia de CI\_Email (method chaining)

---

#### Ejemplo

```
$this->email->set_header('Encabezado1', 'Valor1');  
$this->email->set_header('Encabezado2', 'Valor2');
```

#### subject()

**object subject(string \$subject)**

---

#### Descripción

Establece el asunto del correo electrónico

---

#### Parámetros

##### **subject**

Asunto del correo

---

#### Valores devueltos

Instancia de CI\_Email (method chaining)

---

#### Ejemplo

```
$this->email->subject('Este es mi asunto');
```

#### to()

**object to(mixed \$to)**

---

#### Descripción

Establece las direcciones de correo electrónico de los destinatarios, puede ser un solo correo, una lista delimitada por comas o un array

---

#### Parámetros

##### **to**

Cadena delimitada por comas o un array de direcciones de correo

---

#### Valores devueltos

Instancia de CI\_Email (method chaining)

---

#### Ejemplo

```
$this->email->to('alguien@ejemplo.com');  
$this->email->to('uno@ejemplo.com, dos@ejemplo.com, tres@ejemplo.com');  
$this->email->to(array ('uno@ejemplo.com', 'dos@ejemplo.com', 'tres@ejemplo.com'));
```

## class CI\_Encryption

---

class CI\_Encryption

class CI\_Encryption

create\_key()

string create\_key(int \$length)

Descripción

---

Crea una clave criptográfica obteniendo datos aleatorios de las fuentes del sistema operativo (es decir, /dev/urandom)

Parámetros

---

**length**

Longitud de salida

Valores devueltos

---

Una clave criptográfica pseudoaleatoria con la longitud especificada o FALSE en caso de error

decrypt()

string decrypt(string \$data [, array \$params = NULL])

Descripción

---

Descifra los datos de entrada y los devuelve en texto sin formato

Parámetros

---

**data**

Datos para descifrar

**params**

Parámetros opcionales

Valores devueltos

---

Datos descifrados o FALSE en caso de error

Ejemplo

---

```
echo $this->encryption->decrypt($ciphertext);
```

Consulte la sección **Uso de parámetros personalizados** para obtener información sobre los parámetros opcionales

encrypt()

string encrypt(string \$data [, array \$params = NULL])

Descripción

---

Encripta los datos de entrada y devuelve el texto cifrado

## Parámetros

---

### data

Datos para encriptar

### params

Parámetros opcionales

## Valores devueltos

---

Datos cifrados o FALSE en caso de error

## Ejemplo

---

```
$ciphertext = $this->encryption->encrypt('Mi mensaje secreto');
```

Consulte la sección **Uso de parámetros personalizados** para obtener información sobre los parámetros opcionales

## hkdf()

```
string hkdf(string $key [, string $digest = 'sha512' [, string $salt = NULL [, int $length = NULL [, string $info = '']]]])
```

## Descripción

---

Deriva una clave de otra, presumiblemente más débil, este método se usa internamente para derivar un cifrado y una clave HMAC de su clave de cifrado configurada, está disponible públicamente debido a su propósito general, se describe en RFC 5869, sin embargo, a diferencia de la descripción en RFC 5869, esta implementación no es compatible con SHA1

## Parámetros

---

### key

Material clave de entrada

### digest

Un algoritmo de resumen de familia SHA-2

### salt

Salt opcional

### length

Longitud de salida opcional

### info

Información opcional de contexto / aplicación específica

## Valores devueltos

---

Una clave pseudoaleatoria o FALSE en caso de error

## Ejemplo

---

```
$hmac_key = $this->encryption->hkdf(  
    $clave,  
    'sha512',  
    NULL,
```

```
NULL,  
'authentication'  
);  
// $hmac_key es una clave pseudoaleatoria con una longitud de 64 bytes
```

## initialize()

### object initialize(array \$params)

#### Descripción

---

Inicializa, configura, la biblioteca para usar un controlador, cifrado, modo o clave diferente

#### Parámetros

---

##### params

Parámetros de configuración

#### Valores devueltos

---

Instancia CI\_Encryption (method chaining)

#### Ejemplo

---

```
$this->encryption->initialize(  
    array('mode' => 'ctr')  
);
```

Consulte la sección Configuración de la biblioteca para obtener información detallada

## class CI\_Upload

---

### class CI\_Upload

#### class CI\_Upload

## data()

### mixed data([ string \$index = NULL])

#### Descripción

---

Este es un método que devuelve un array que contiene todos los datos relacionados con el archivo que se cargó

#### Parámetros

---

##### index

Elemento a devolver en lugar de la array completa

#### Valores devueltos

---

Información sobre el archivo cargado

Aquí está el prototipo del array:



```

Array
(
    [file_name] => mypic.jpg
    [file_type] => image/jpeg
    [file_path] => /path/to/your/upload/
    [full_path] => /path/to/your/upload/jpg.jpg
    [raw_name] => mypic
    [orig_name] => mypic.jpg
    [client_name] => mypic.jpg
    [file_ext] => .jpg
    [file_size] => 22.2
    [is_image] => 1
    [image_width] => 800
    [image_height] => 600
    [image_type] => jpeg
    [image_size_str] => width="800" height="200"
)

```

Para obtener un elemento del array:

```
$this->upload->data('file_name'); // Devuelve: mypic.jpg
```

Aquí hay una tabla que explica los elementos del array anterior:

Elemento	Descripción
<b>file_name</b>	Nombre del archivo que se cargó, incluida su extensión
<b>file_type</b>	Tipo MIME del archivo MIME
<b>file_path</b>	Ruta absoluta del archivo en el servidor
<b>full_path</b>	Ruta absoluta del servidor, incluido el nombre del archivo
<b>raw_name</b>	Nombre del archivo, sin la extensión
<b>orig_name</b>	Nombre original del archivo. Solo es útil si usa la opción de nombre encriptado
<b>client_name</b>	Nombre de archivo proporcionado por el agente de usuario del cliente, posiblemente sanitizado
<b>file_ext</b>	Extensión del nombre de archivo, punto incluido
<b>file_size</b>	Tamaño del archivo en kilobytes
<b>is_image</b>	Si el archivo es o no una imagen. 1 = imagen. 0 = no
<b>image_width</b>	Ancho de la imagen
<b>image_height</b>	Alto de la imagen
<b>image_type</b>	Tipo de imagen, generalmente la extensión del nombre de archivo sin el punto
<b>image_size_str</b>	Un string que contiene el ancho y alto (útil para ponerlo en una etiqueta de imagen)

`display_errors()`

**string display\_errors**([ **string \$open** = '<p>' [, **string \$close** = '</p>'])

---

#### Descripción

Recupera cualquier mensaje de error si el método **do\_upload()** devuelve falso, el método no se repite automáticamente, devuelve los datos para que pueda asignarlos como lo necesite

---

#### Parámetros

##### **open**

Apertura de marcado, delimitador de inicio

##### **close**

Marcado de cierre, delimitador de fin

---

#### Valores devueltos

Mensaje(s) de error formateado

---

#### Formateo de errores

Por defecto, el método anterior ajusta cualquier error dentro de las etiquetas <p>, puede establecer sus propios delimitadores de esta manera:

```
$this->upload->display_errors('<p>', '</p>');
```

`do_upload()`

**bool do\_upload**([ **string \$field** = 'userfile'])

---

#### Descripción

Realiza la carga según las preferencias que ha establecido

---

#### Parámetros

##### **field**

Nombre del campo de formulario

---

#### Valores devueltos

TRUE en el éxito, FALSE si falla

#### **Nota**

De forma predeterminada, la rutina de carga espera que el archivo proceda de un campo de formulario denominado **userfile**, y el formulario debe ser del tipo **"multipart"**

```
<form method="post" action="some_action" enctype="multipart/form-data" />
```

Si desea establecer su propio nombre de campo, simplemente pase su valor al método **do\_upload()**:

```
$nombre_campo = "algun_nombre_de_campo";  
$this->upload->do_upload($nombre_campo);
```

**initialize()**

**object initialize**([ **array \$config** = **array()** [, **bool \$reset** = **TRUE**]])

---

**Descripción**

Inicia las preferencias

---

**Parámetros**

**config**

Preferencias

**reset**

Si restablecer las preferencias (que no se proporcionan en **\$config**) a sus valores predeterminados

---

**Valores devueltos**

Instancia CI\_Upload (method chaining)

---

## class CI\_Form\_validation

---

**class CI\_Form\_validation**

**class CI\_Form\_validation**

**error()**

**string error**(**string \$field** [, **string \$prefix** = '' [, **string \$suffix** = ''])

---

**Descripción**

Devuelve el mensaje de error para un campo específico, opcionalmente agregando un prefijo y/o sufijo, generalmente etiquetas HTML

---

**Parámetros**

**field**

Nombre del campo

**prefix**

Prefijo opcional

**suffix**

Sufijo opcional

---

**Valores devueltos**

Cadena de mensaje de error

**error\_array()**

**array error\_array**()

---

**Descripción**

---

Devuelve los mensajes de error como un array

---

#### Valores devueltos

Array de mensajes de error

#### error\_string()

**string error\_string**([ **string \$prefix** = " [, **string \$suffix** = "])

---

#### Descripción

Devuelve todos los mensajes de error ,como se devuelve desde **error\_array()**, formateados como un string y separados por un carácter de nueva línea

---

#### Parámetros

##### **prefix**

Prefijo del mensaje de error

##### **suffix**

Sufijo del mensaje de error

---

#### Valores devueltos

Mensajes de error como un string

#### has\_rule()

**bool has\_rule**(**string \$field**)

---

#### Descripción

Verifica si hay una regla establecida para el campo especificado

---

#### Parámetros

##### **field**

Nombre del campo

---

#### Valores devueltos

TRUE si el campo tiene reglas establecidas, FALSE si no

#### reset\_validation()

**object reset\_validation**()

---

#### Descripción

Le permite restablecer la validación cuando valida más de un array, este método debe invocarse antes de validar cada nuevo array

---

#### Valores devueltos

Instancia CI\_Form\_validation (method chaining)

**run()**

**bool run**([ **string \$group** = ""])

Descripción

Ejecuta las rutinas de validación, opcionalmente, puede pasar el nombre del grupo de validación a través del método, como se describe en la sección **Guardar conjuntos de reglas de validación en un archivo de configuración**

Parámetros

---

**group**

El nombre del grupo de validación para ejecutar

Valores devueltos

---

TRUE en caso de éxito, FALSE si falló la validación

**set\_data()**

**object set\_data**(**array \$data**)

Descripción

---

Le permite establecer un array para la validación, en lugar de usar el array **\$\_POST** predeterminado

Parámetros

---

**data**

Array de datos de validación

Valores devueltos

---

Instancia CI\_Form\_validation (method chaining)

**set\_error\_delimiters()**

**object set\_error\_delimiters**([ **string \$prefix** = '<p>' [, **string \$suffix** = '</p>']] )

Descripción

---

Establece el prefijo y el sufijo por defecto para los mensajes de error

Parámetros

---

**prefix**

Prefijo del mensaje de error

**suffix**

Sufijo del mensaje de error

Valores devueltos

---

Instancia CI\_Form\_validation (method chaining)

**set\_message()**

**object set\_message**(**string \$lang** [, **string \$val** = ""])

Descripción

---

Le permite establecer mensajes de error personalizados, consulte **Configuración de mensajes de error**

#### Parámetros

---

##### lang

La regla para la cual es el mensaje

##### val

El mensaje

#### Valores devueltos

---

Instancia CI\_Form\_validation (method chaining)

#### set\_rules()

```
object set_rules(string $field [, string $label = '' [, mixed $rules = '' [, array $errors = array()]] ])
```

#### Descripción

---

Le permite establecer reglas de validación, como se describe en las secciones:

- **Establecer reglas de validación**
- **Guardar grupos de reglas de validación en un archivo de configuración**

#### Parámetros

---

##### field

Nombre del campo

##### label

Etiqueta de campo

##### rules

Reglas de validación, como una lista de strings separadas por una tubería "|", o como un array con reglas

##### errors

Una lista de mensajes de error personalizados

#### Valores devueltos

---

Instancia CI\_Form\_validation (method chaining)

## class CI\_FTP

---

#### class CI\_FTP

```
class CI_FTP
```

#### chANGEDIR()

```
bool chANGEDIR(string $path [, bool $suppress_debug = FALSE])
```

#### Descripción

---

Cambia el directorio de trabajo actual a la ruta especificada

## Parámetros

---

### path

Ruta del directorio

### suppress\_debug

Si se desactivan los mensajes de depuración para este comando, es útil en caso de que desee utilizar este método como alternativa a **is\_dir()** para FTP

## Valores devueltos

---

TRUE en el éxito, FALSE si falla

## chmod()

### bool chmod(string \$path, int \$perm)

#### Descripción

---

Le permite configurar permisos de archivos, proporcione la ruta al archivo o directorio en el que desea modificar los permisos

#### Parámetros

---

### path

Ruta para modificar los permisos para

### perm

Permisos (octal)

## Valores devueltos

---

TRUE en el éxito, FALSE si falla

## Ejemplo

---

```
// Chmod "jamón_serrano" a 755
$this->ftp->chmod('/publico_html/foo/jamón_serrano/', 0755);
```

## close()

### bool close()

#### Descripción

---

Cierra la conexión a su servidor, se recomienda que use este método cuando termine de cargar

## Valores devueltos

---

TRUE en caso de éxito, FALSE en caso de error

## connect()

### bool connect([ array \$config = array()])

#### Descripción

---

Se conecta e inicia sesión en el servidor FTP, las preferencias de conexión se establecen pasando un array al método, o puede almacenarlas en un archivo de configuración

## Parámetros

---

### config

Valores de conexión

### Valores devueltos

---

TRUE en el éxito, FALSE si falla

Aquí hay un ejemplo que muestra cómo configurar las preferencias de forma manual:

```
$this->load->library('ftp');

$config['hostname'] = 'ftp.ejemplo.com';
$config['username'] = 'su_nombre_de_usuario';
$config['password'] = 'su_contraseña';
$config['port']     = 21;
$config['passive']  = FALSE;
$config['debug']    = TRUE;

$this->ftp->connect($config);
```

### Preferencias de FTP en un archivo de configuración

Si lo prefiere, puede almacenar sus preferencias de FTP en un archivo de configuración, simplemente cree un nuevo archivo llamado **ftp.php**, agregue el array **\$config** en ese archivo, luego guarde el archivo en **application/config/ftp.php** y se usará automáticamente

### Opciones de conexión disponibles

Opción	Predeterminado	Descripción
<b>hostname</b>	n/a	Nombre de host FTP (generalmente algo como: ftp.ejemplo.com)
<b>username</b>	n/a	Nombre de usuario FTP
<b>password</b>	n/a	Contraseña FTP
<b>port</b>	21	FTP Número de puerto del servidor FTP
<b>debug</b>	FALSE	TRUE/FALSE (boolean): Si habilita la depuración para mostrar los mensajes de error
<b>passive</b>	TRUE	TRUE/FALSE (boolean): Si se debe usar el modo pasivo

### delete\_dir()

**bool delete\_dir(string \$filepath)**

### Descripción

---

Permite eliminar un directorio y todo lo que contiene, proporcione la ruta de origen al directorio con una barra inclinada

### Parámetros

---

**filepath**



Ruta al directorio para eliminar

#### Valores devueltos

---

TRUE en el éxito, FALSE si falla

#### **Importante**

**MUCHO CUIDADO** con este método !

Se eliminará de forma recursiva todo dentro de la ruta de acceso, incluidas las subcarpetas y todos los archivos. Asegúrese de que su ruta sea correcta. Intente usar primero **list\_files()** para verificar que la ruta sea la correcta

#### Ejemplo

---

```
$this->ftp->delete_dir('/publico_html/path/a/directorio/');
```

#### **delete\_file()**

**bool delete\_file(string \$filepath)**

#### Descripción

---

Permite eliminar un archivo

#### Parámetros

---

##### **filepath**

Ruta al archivo para eliminar

#### Valores devueltos

---

TRUE en el éxito, FALSE si falla

#### Ejemplo

---

```
// Borra el archivo gasoil.html del directorio 'combustibles'  
$this->ftp->delete_file('/publico_html/combustibles/gasoil.html');
```

#### **download()**

**bool download(string \$rempath, string \$lopath [, string \$mode = 'auto'])**

#### Descripción

---

Descarga un archivo de su servidor, debe proporcionar la ruta remota y la ruta local, y opcionalmente puede establecer el modo

#### Parámetros

---

##### **rempath**

Ruta del archivo remoto

##### **lopath**

Ruta del archivo local

##### **mode**

Modo FTP, por defecto es 'auto' (las opciones son: 'auto', 'binary', 'ascii'), si utiliza el modo 'auto' basará el modo en la extensión de archivo del archivo de origen

## Valores devueltos

---

TRUE en el éxito, FALSE si falla, devuelve FALSE si la descarga no se ejecuta correctamente, incluso si PHP no tiene permiso para escribir el archivo local

## Ejemplo

---

```
$this->ftp->download('/publico_html/mi_archivo.html', '/local/path/a/mi_archivo.html', 'ascii');
```

## list\_files()

**array list\_files**([ **string \$path** = '.'])

### Descripción

---

Le permite recuperar una lista de archivos en su servidor devuelta como un array, debe proporcionar la ruta al directorio deseado

### Parámetros

---

#### path

Ruta del directorio

### Valores devueltos

---

Un array con la lista de archivos o FALSE en caso de error

## Ejemplo

---

```
$list = $this->ftp->list_files('/publico_html/');  
print_r($list);
```

## mirror()

**bool mirror**(**string \$lospath**, **string \$rempath**)

### Descripción

---

Lee de manera recursiva una carpeta local y todo lo que contiene, incluidas las subcarpetas, y crea una réplica mediante FTP en función de ella, cualquiera que sea la estructura del directorio de la ruta original del archivo, se recreará en el servidor, debe proporcionar una ruta de origen y una ruta de destino

### Parámetros

---

#### lospath

Ruta local

#### rempath

Ruta remota

### Valores devueltos

---

TRUE en el éxito, FALSE si falla

## Ejemplo

---

```
$this->ftp->mirror('/path/a/mi_directorio/', '/publico_html/mi_directorio/');
```

## mkdir()

**bool mkdir(string \$path [, int \$permissions = NULL])**

### Descripción

---

Le permite crear un directorio en su servidor, proporcione la ruta que termina en el nombre de la carpeta que desea crear, con una barra al final, los permisos pueden establecerse pasando un valor octal en el segundo parámetro

### Parámetros

---

#### path

Ruta al directorio para crear

#### permissions

Permisos (octal)

### Valores devueltos

---

TRUE en el éxito, FALSE si falla

## Ejemplo

---

```
// Crea una carpeta llamada 'chavo'  
$this->ftp->mkdir('/publico_html/foo/chavo/', 0755);
```

## move()

**bool move(string \$old\_file, string \$new\_file)**

### Descripción

---

Le permite mover un archivo

### Parámetros

---

#### old\_file

Antiguo nombre de archivo

#### new\_file

Nuevo nombre de archivo

### Valores devueltos

---

TRUE en el éxito, FALSE si falla

## Ejemplo

---

```
// Mueve gasoil.html del directorio "bueno" a "malo"  
$this->ftp->move('/publico_html/bueno/gasoil.html', '/publico_html/malo/gasoil.html');
```

## Nota

Si el nombre del archivo de destino es diferente, se cambiará el nombre del archivo

## rename()

**bool rename(string \$old\_file, string \$new\_file [, bool \$move = FALSE])**

### Descripción

---

Le permite cambiar el nombre de un archivo, proporcione el nombre/ruta del archivo de origen y el nuevo nombre/ruta de archivo

### Parámetros

---

#### old\_file

Antiguo nombre de archivo

#### new\_file

Nuevo nombre de archivo

#### move

Si se está realizando un movimiento

### Valores devueltos

---

TRUE en el éxito, FALSE si falla

### Ejemplo

---

```
// Cambia el nombre de verde.html a azul.html
$this->ftp->rename('/publico_html/foo/verde.html', '/publico_html/foo/azul.html');
```

## upload()

**bool upload(string \$lopath, string \$rempath [, string \$mode = 'auto' [, int \$permissions = NULL]])**

### Descripción

---

Carga un archivo a su servidor, debe proporcionar la ruta local y la ruta remota, y opcionalmente puede establecer el modo y los permisos

### Parámetros

---

#### lopath

Ruta del archivo local

#### rempath

Ruta del archivo remoto

#### mode

Modo FTP, por defecto es 'auto' (las opciones son: 'auto', 'binario', 'ascii'), si se utiliza el modo 'auto', automático, basará el modo en la extensión de archivo del archivo de origen.

#### permissions

Permisos de archivos, si se establece, los permisos deben pasarse como un valor octal

### Valores devueltos

---

TRUE en el éxito, FALSE si falla

## Ejemplo

---

```
$this->ftp->upload('/local/path/a/mi_archivo.html','/publico_html/mi_archivo.html','ascii', 0775);
```

## class CI\_Image\_lib

---

class CI\_Image\_lib

**class CI\_Image\_lib**

clear()

**void clear()**

### Descripción

---

Restablece todos los valores utilizados al procesar una imagen, debe llamarlo si está procesando imágenes en un bucle

### Ejemplo

---

```
$this->image_lib->clear();
```

### valores devueltos

---

void

crop()

**bool crop()**

### Descripción

---

Recorta una imagen, funciona de forma casi idéntica al método de cambio de tamaño, excepto que requiere que establezca preferencias para los ejes X e Y (en píxeles) especificando dónde recortar

Todas las preferencias enumeradas en la tabla de preferencias están disponibles para este método excepto estas:

**rotation\_angle**, **create\_thumb** y **new\_image**

### Valores devueltos

---

TRUE en caso de éxito, FALSE en caso de error

### Ejemplo

---

El método requiere que establezca preferencias para los ejes X e Y (en píxeles) especificando dónde recortar, como aquí:

```
$config['x_axis'] = 100;  
$config['y_axis'] = 40;
```

Aquí hay un ejemplo que muestra cómo puedes recortar una imagen:

```
$config['image_library'] = 'imagemagick';
$config['library_path'] = '/usr/X11R6/bin/';
$config['source_image'] = '/path/a/imagen/mipic.jpg';
$config['x_axis'] = 100;
$config['y_axis'] = 60;

$this->image_lib->initialize($config);

if ( ! $this->image_lib->crop() )
{
    echo $this->image_lib->display_errors();
}
```

### Nota

Sin una interfaz visual, es difícil recortar imágenes, por lo que este método no es muy útil a menos que tenga la intención de construir dicha interfaz, eso es exactamente lo que hicimos utilizando para el módulo de galería de fotos en ExpressionEngine, el CMS que desarrollamos, agregamos una interfaz de usuario JavaScript que permite seleccionar el área de recorte

### display\_errors()

**string display\_errors([ string \$open = '<p> [, string \$close = '</p>']])**

#### Descripción

---

Devuelve todos los errores detectados formateados como un string

#### Parámetros

---

##### open

Etiqueta de apertura del mensaje de error

##### close

Etiqueta de cierre del mensaje de error

#### Valores devueltos

---

Mensajes de error

#### Ejemplo

---

```
echo $this->image_lib->display_errors();
```

### initialize()

**bool initialize([ array \$props = array()])**

#### Descripción

---

Inicializa la clase para procesar una imagen

#### Parámetros

---

##### props

#### Valores devueltos

---

TRUE en caso de éxito, FALSE en el caso de configuraciones no válidas

**resize()**

**bool resize()**

#### Descripción

---

Cambia el tamaño de una imagen, permite cambiar el tamaño de la imagen original, crear una copia (con o sin cambio de tamaño) o crea una imagen en miniatura

A efectos prácticos, no hay diferencia entre crear una copia y crear una miniatura, excepto que tendrá un indicador de miniatura como parte del nombre (es decir, `mipic_thumb.jpg`)

Todas las preferencias enumeradas en la tabla de preferencias están disponibles para este método, excepto estas tres: **rotation\_angle**, **x\_axis** e **y\_axis**

#### Valores devueltos

---

Devuelve: TRUE en caso de éxito, FALSE en caso de error

#### Ejemplos

---

### Crear una miniatura

El método de cambio de tamaño creará un archivo en miniatura (y conserva el original) si establece esta preferencia en TRUE:

```
$config['create_thumb'] = TRUE;
```

Esta única preferencia determina si se crea o no una miniatura

### Creando una Copia

El método de cambio de tamaño creará una copia del archivo de imagen, y conserva el original, si con esta preferencia establece una ruta y/o un nuevo nombre de archivo:

```
$config['new_image'] = '/path/a/nueva_imagen.jpg';
```

#### Notas con respecto a esta preferencia:

- Si solo se especifica el nuevo nombre de imagen, se colocará en el mismo directorio
- Si solo se especifica la ruta, la nueva imagen se colocará en el destino con el mismo nombre que el original
- Si se especifican tanto la ruta como el nombre de la imagen, se ubicará en su propio destino y se le asignará el nuevo nombre

### Cambiar el tamaño de la imagen original

Si no se utiliza ninguna de las dos preferencias enumeradas anteriormente (**create\_thumb** y **new\_image**), el método **resize()** se centrará en la imagen original para su procesamiento

## rotate()

### bool rotate()

#### Descripción

---

Rota una imagen, el método requiere que el ángulo de rotación se establezca a través de su preferencia

Hay 5 opciones de rotación:

1. **90** - gira en sentido antihorario 90 grados
2. **180** - gira en sentido antihorario 180 grados
3. **270** - gira en sentido antihorario 270 grados
4. **hor** - voltea la imagen horizontalmente
5. **vrt** - voltea la imagen verticalmente

#### Valores devueltos

---

TRUE en caso de éxito, FALSE en caso de error

#### Ejemplos

---

El ángulo de rotación se establece a través de su preferencia:

```
$config['rotation_angle'] = '90';
```

Aquí se muestra cómo puede rotar una imagen:

```
$config['image_library'] = 'netpbm';
$config['library_path'] = '/usr/bin/';
$config['source_image'] = '/path/a/imagen/mipic.jpg';
$config['rotation_angle'] = 'hor';

$this->image_lib->initialize($config);

if ( ! $this->image_lib->rotate() )
{
    echo $this->image_lib->display_errors();
}
```

## watermark()

### bool watermark()

#### Descripción

---

Crea una marca de agua sobre una imagen, consulte la sección **Marca de agua** para obtener más información

#### Valores devueltos

---

Devuelve TRUE en caso de éxito, FALSE en caso de error



# class CI\_Input

---

class CI\_Input

## class CI\_Input

\$raw\_input\_stream

## \$raw\_input\_stream

### Descripción

---

Propiedad de solo lectura que devolverá los datos de entrada **php://** tal como están, la propiedad se puede leer varias veces

cookie()

## mixed cookie([ mixed \$index = NULL [, bool \$xss\_clean = NULL]])

### Descripción

---

Retorna el valor de **\$\_COOKIE** , es idéntico a **post()** y **get()**, solo que busca datos de cookies

### Parámetros

---

#### index

Nombre del cookie

#### xss\_clean

Si debe aplicarse el filtrado XSS

### Valores devueltos

---

**\$\_COOKIE** si no se han proporcionado los parámetros, de lo contrario el valor de **COOKIE** si se encuentra o **NULL** si no

### Ejemplos

---

Este método es idéntico a **post()** y **get()**, solo que busca datos de cookies:

```
$this->input->cookie('algun_cookie');  
$this->input->cookie('algun_cookie', TRUE); // con filtro XSS
```

Para devolver un array de múltiples valores de cookies, pase todas las claves requeridas como un array:

```
$this->input->cookie(array('algun_cookie1', 'algun_cookie2'));
```

### Nota

A diferencia de la función **get\_cookie()** de **Cookie Helper**, este método **NO** antepone el valor configurado en **\$config['cookie\_prefix']**

get()

## mixed get([ mixed \$index = NULL [, bool \$xss\_clean = NULL]])

### Descripción

---

Retorna el valor de **\$\_GET**, para devolver un array con todos los elementos **GET** llame sin ningún parámetro, este método es idéntico a **post()**, solo que obtiene datos **GET**

#### Parámetros

---

##### index

Nombre del elemento GET

##### xss\_clean

Si se debe aplicar el filtrado XSS

#### Valores devueltos

---

**\$\_GET** si no se han proporcionado los parámetros, de lo contrario el valor GET si se encuentra o NULL si no

#### Ejemplos

---

```
$this->input->get('algun_dato', TRUE);
```

Para devolver todos los elementos **GET** y pasarlos a través del filtro XSS, establezca el primer parámetro a NULL mientras configura el segundo parámetro como TRUE:

```
$this->input->get(NULL, TRUE); // devuelve todos los elementos de GET filtrados XSS
$this->input->get(NULL, FALSE); // devuelve todos los elementos GET sin filtro XSS
```

Para devolver un array de múltiples elementos **GET**, pase todas las claves requeridas como un array:

```
$this->input->get(array('elemento1', 'elemento2'));
```

Se aplica la misma regla para recuperar los parámetros con el filtrado XSS activado, establezca el segundo parámetro en TRUE:

```
$this->input->get(array('elemento1', 'elemento2'), TRUE);
```

#### get\_post()

**mixed get\_post(string \$index [, bool \$xss\_clean = NULL])**

#### Descripción

---

Retorna el valor de **\$\_GET** y **\$\_POST**, el método funciona más o menos igual que **post()** y **get()**, en este caso combinados, buscará en **\$\_GET** y **\$\_POST** los datos, buscando primero en **\$\_GET** y luego en **\$\_POST**

#### Parámetros

---

##### index

Nombre del elemento GET/POST

##### xss\_clean

Si debe aplicarse el filtrado XSS

#### Valores devueltos

---

Valor GET/POST si se encuentra, NULL si no

#### Ejemplo

---

Este método funciona de la misma manera que **post\_get()** solo que busca datos GET primero:

```
$this->input->get_post('algun_dato', TRUE);
```

### Nota

Este método solía actuar EXACTAMENTE como **post\_get()**, pero su comportamiento ha cambiado en CodeIgniter 3.0

### get\_request\_header()

**string get\_request\_header(string \$index [, bool \$xss\_clean = FALSE])**

#### Descripción

Devuelve un único miembro del array de encabezados de solicitud o NULL si no se encuentra el encabezado buscado

#### Parámetros

##### index

Nombre del encabezado de solicitud HTTP

##### xss\_clean

Si debe aplicarse el filtrado XSS

#### Valores devueltos

Un encabezado de solicitud HTTP o NULL si no se encuentra

#### Ejemplo

```
$this->input->get_request_header('alguna-header', TRUE);
```

### input\_stream()

**mixed input\_stream([ mixed \$index = NULL [, bool \$xss\_clean = NULL]])**

#### Descripción

Este método es idéntico a **get()**, **post()** y **cookie()**, solo que obtiene los datos de la ruta de entrada **php://**

#### Parámetros

##### index

Nombre clave

##### xss\_clean

Si debe aplicarse el filtrado XSS

#### Valores devueltos

Array de flujo de entrada si no se suministran los parámetros, de lo contrario el valor especificado si se encuentra o NULL si no

### ip\_address()

**string ip\_address()**

#### Descripción

Devuelve la dirección IP para el usuario actual, si la dirección IP no es válida, el método devolverá '0.0.0.0'

#### Valores devueltos

Dirección IP del visitante o '0.0.0.0' si no es válido

#### Ejemplo

---

```
echo $this->input->ip_address();
```

#### Importante

Este método tiene en cuenta la configuración **\$config['proxy\_ips']** y devolverá la dirección

HTTP\_X\_FORWARDED\_FOR,

HTTP\_CLIENT\_IP, HTTP\_X\_CLIENT\_IP o HTTP\_X\_CLUSTER\_CLIENT\_IP informada para las direcciones IP permitidas

**is\_ajax\_request()**

**bool is\_ajax\_request()**

#### Descripción

---

Indica si es una solicitud Ajax, comprueba si el encabezado del servidor **HTTP\_X\_REQUESTED\_WITH** se ha establecido y devuelve booleano TRUE si esta o FALSE si no

#### Valores devueltos

---

TRUE si es una solicitud Ajax, FALSE si no

**is\_cli\_request()**

**bool is\_cli\_request()**

#### Nota

Este método es **OBSOLETO** y ahora es solo un alias para la función **is\_cli()**

**method()**

**string method([ bool \$upper = FALSE ])**

#### Descripción

---

Devuelve **\$ \_SERVER['REQUEST\_METHOD']**, con la opción de establecerlo en mayúsculas o minúsculas

#### Parámetros

---

##### upper

Si se debe devolver el nombre del método de solicitud en mayúscula o minúscula

#### Valores devueltos

---

Método de solicitud HTTP

#### Ejemplo

---

```
echo $this->input->method(TRUE); // Resultado: POST
echo $this->input->method(FALSE); // Resultado: post
echo $this->input->method(); // Resultado: post
```

## post()

**mixed** `post([ mixed $index = NULL [, bool $xss_clean = NULL]])`

### Descripción

---

Retorna el valor de **\$\_POST**, para devolver un array con todos los elementos POST llame sin ningún parámetro

### Parámetros

---

#### index

Nombre del elemento POST

#### xss\_clean

Si se debe aplicar el filtrado XSS

### Valores devueltos

---

**\$\_POST** si no se han proporcionado los parámetros, de lo contrario el valor POST si se encuentra o NULL si no

### Ejemplos

---

El primer parámetro contendrá el nombre del elemento POST que está buscando:

```
$this->input->post('algun_dato');
```

El método devuelve NULL si el elemento que está intentando recuperar no existe

El segundo parámetro opcional le permite ejecutar el filtro XSS, se habilita estableciendo el segundo parámetro en TRUE o configurando **\$config['global\_xss\_filtering']** en TRUE:

```
$this->input->post('algun_dato', TRUE);
```

Para devolver todos los elementos POST y pasarlos a través del filtro XSS, establezca el primer parámetro a NULL mientras configura el segundo parámetro como booleano TRUE:

```
$this->input->post(NULL, TRUE); // devuelve todos los elementos de POST filtrados XSS
$this->input->post(NULL, FALSE); // devuelve todos los elementos POST sin filtro XSS
```

Para devolver un array de múltiples elementos POST, pase todas las claves requeridas como un array:

```
$this->input->post(array('elemento1', 'elemento2'));
```

Se aplica la misma regla, para recuperar los parámetros con el filtrado XSS activado, establezca el segundo parámetro en TRUE:

```
$this->input->post(array('elemento1', 'elemento2'), TRUE);
```

## post\_get()

**mixed** `post_get(string $index [, bool $xss_clean = NULL])`

### Descripción

---

Retorna el valor de **\$\_POST** y **\$\_GET**, el método funciona más o menos igual que **post()** y **get()**, en este caso combinados, buscará en **\$\_POST** y **\$\_GET** los datos, buscando primero en **\$\_POST** y luego en **\$\_GET**

## Parámetros

---

### index

Nombre del elemento POST/GET

### xss\_clean

Si debe aplicarse el filtrado XSS

## Valores devueltos

---

Valor POST/GET si se encuentra, NULL si no

## Ejemplo

---

Busca en **\$\_POST** y **\$\_GET** los datos, buscando primero en **\$\_POST** y luego en **\$\_GET**:

```
$this->input->post_get('algun_dato', TRUE);
```

## request\_headers()

**array request\_headers([ bool \$xss\_clean = FALSE])**

### Descripción

---

Devuelve un array de encabezados **HTTP**, útil si está corriendo en un entorno que no es **Apache** donde **apache\_request\_headers()** no está soportado

## Parámetros

---

### xss\_clean

Si debe aplicarse el filtrado XSS

## Valores devueltos

---

Un array de encabezados de solicitud HTTP

## Ejemplo

---

```
$headers = $this->input->request_headers();
```

## server()

**mixed server(mixed \$index [, bool \$xss\_clean = NULL])**

### Descripción

---

Retorna el valor de **\$\_SERVER**, es idéntico a los métodos **post()**, **get()** y **cookie()**, solo recupera los datos del servidor **\$\_SERVER**

## Parámetros

---

### index

Nombre del elemento

### xss\_clean

Si debe aplicarse el filtrado XSS

## Valores devueltos

---

El valor de **\$\_SERVER** si se encuentra o NULL si no

## Ejemplos

---

```
$this->input->server('algun_dato');
```

Para devolver un array de múltiples valores \$ \_SERVER, pase todas las claves requeridas como un array :

```
$this->input->server(array('SERVER_PROTOCOL', 'REQUEST_URI'));
```

## set\_cookie()

```
void set_cookie(mixed $name = '' [, string $value = '' [, int $expire = '' [, string $domain = '' [, string $path = '/' [, string $prefix = '' [, bool $secure = NULL [, bool $httponly = NULL]]]]]])
```

### Descripción

---

Establece una cookie que contiene los valores que especifique

### Parámetros

---

#### name

Nombre de la cookie o una array de parámetros

#### value

Valor de la cookie

#### expire

Tiempo de expiración de la cookie en segundos

#### domain

Dominio de cookies

#### path

Ruta de las cookies

#### prefix

Prefijo de nombre de cookie

#### secure

Si solo se transfiere la cookie a través de HTTPS

#### httponly

Si solo hacer que la cookie sea accesible para las solicitudes HTTP (sin JavaScript)

Hay dos formas de pasar información a este método para que se pueda establecer una cookie: **Método de array** y de **Parámetros discretos**:

### Método de array

Usando este método, un array asociativo se pasa al primer parámetro:

```
$cookie = array(
    'name' => 'Nombre de la Cookie',
    'value' => 'El valor',
    'expire' => '86500',
    'domain' => '.algun-dominio.com',
    'path' => '/',
    'prefix' => 'mi_prefijo_',
    'secure' => TRUE
);
$this->input->set_cookie($cookie);
```

## Notas

Solo se requiere el nombre y el valor, para eliminar una cookie, configúrela con un espacio en blanco en caducidad

La caducidad se establece en segundos, que se agregará a la hora actual, no incluya el tiempo, sino solo el número de segundos a partir del momento en que desea que la cookie sea válida, si la caducidad se establece en cero, la cookie solo durará mientras el navegador esté abierto

Para las cookies de todo el sitio, independientemente de cómo se solicite su sitio, agregue su URL al dominio comenzando con un punto, como este: **.su\_dominio.com**

La ruta generalmente no es necesaria ya que el método establece una ruta raíz

El prefijo solo es necesario si necesita evitar colisiones de nombres con otras cookies idénticas para su servidor

Los parámetros **httponly** y **secure**, cuando se omiten, se ajustan de manera predeterminada a sus configuraciones de **\$config['cookie\_httponly']** y **\$config['cookie\_secure']**

## Parámetros discretos

Si lo prefiere, puede configurar la cookie pasando datos usando parámetros individuales:

```
$this->input->set_cookie($name, $value, $expire, $domain, $path, $prefix, $secure);
```

## user\_agent()

**mixed user\_agent([ bool \$xss\_clean = NULL ])**

### Descripción

---

Devuelve la cadena de agente de usuario (navegador web) que utiliza el usuario actual o NULL si no está disponible, consulte **User Agent Class** para conocer los métodos que extraen información de la cadena del agente de usuario

### Parámetros

---

#### **xss\_clean**

Si debe aplicarse el filtrado XSS

### Valores devueltos

---

Cadena de agente de usuario o NULL si no está establecido



## Ejemplo

---

```
echo $this->input->user_agent();
```

## valid\_ip()

**bool valid\_ip(string \$ip [, string \$which = ''])**

### Descripción

---

Toma una dirección IP como entrada y devuelve TRUE o FALSE (booleano) dependiendo de si es válida o no

### Parámetros

---

#### ip

Dirección IP

#### which

Protocolo IP ('ipv4' o 'ipv6') que especifica un formato de IP, el valor predeterminado comprueba ambos formatos

### Valores devueltos

---

TRUE si la dirección es válida, FALSE si no

## Ejemplo

---

```
if ( ! $this->input->valid_ip($ip) )
{
    echo 'No es válida';
}
else
{
    echo 'IP válida';
}
```

## class CI\_Lang

---

### class CI\_Lang

**class CI\_Lang**

## line()

**string line(string \$line [, bool \$log\_errors = TRUE])**

### Descripción

---

Obtiene una línea de traducción de los archivos de idioma ya cargados, según la clave de la línea

### Parámetros

---

#### line

Nombre de la clave de línea de idioma

#### log\_errors

Si se registra un error si la línea no se encuentra

### Valores devueltos

---

Cadena de línea de idioma o FALSE en caso de error

## load()

```
mixed load(mixed $langfile [, string $idiom = '' [, bool $return = FALSE [, bool $add_suffix = TRUE [, string $alt_path = '']]])
```

### Descripción

---

Carga un archivo de idioma

### Parámetros

---

#### langfile

Archivo de idioma para cargar o array con múltiples archivos

#### idiom

Nombre del idioma (por ejemplo "spanish")

#### return

Si se debe devolver el array cargado de traducciones

#### add\_suffix

Si se agrega el sufijo '\_lang' al nombre del archivo de idioma

#### alt\_path

Una ruta alternativa para buscar el archivo de idioma

### Valores devueltos

---

Un array de líneas de idioma si **\$return** está establecido en TRUE, en caso contrario void

## class CI\_Loader

---

### class CI\_Loader

#### class CI\_Loader

## add\_package\_path()

```
object add_package_path(string $path [, bool $view_cascade = TRUE])
```

### Descripción

---

Agrega una ruta para que la clase **Loader** la anteponga en solicitudes posteriores de recursos, como ejemplo, el paquete de aplicación "**Leonardo**" anterior tiene una biblioteca llamada **Leonardo.php**, en nuestro controlador, haríamos lo siguiente:

```
$this->load->add_package_path(APPPATH.'third_party/leonardo/')->library('leonardo');
```

### Parámetros

---

#### path

Ruta que se agrega

#### view\_cascade

Si se deben usar vistas en cascada

## Valores devueltos

---

Instancia de CI\_Loader (method chaining)

**clear\_vars()**

**object clear\_vars()**

## Descripción

---

Borra las variables de vista en caché

## Valores devueltos

---

Una instancia de CI\_Loader (method chaining)

**config()**

**bool config(string \$file [, bool \$use\_sections = FALSE [, bool \$fail\_gracefully = FALSE]])**

## Descripción

---

Este método es un alias del método de carga del archivo de configuración: **\$this->config->load()**

## Parámetros

---

### file

Nombre del archivo de configuración

### use\_sections

Si los valores de configuración deben cargarse en su propia sección

### fail\_gracefully

Si solo devuelve FALSE en caso de fallo

## Valores devueltos

---

TRUE en el éxito, FALSE si falla

**database()**

**mixed database([ mixed \$params = '' [, bool \$return = FALSE [, bool \$query\_builder = NULL]])**

## Descripción

---

Este método le permite cargar la clase de la Base de datos, los dos parámetros son opcionales, por favor, mire la sección

**Bases de datos** para más información

## Parámetros

---

### params

Nombre del grupo de Base de datos u opciones de configuración

### return

Si se debe devolver el objeto de Base de datos cargado

### query\_builder

Si se carga el Query Builder

## Valores devueltos

---

Instancia de CI\_DB cargada o FALSE en caso de error si **\$return** está establecido a TRUE, de lo contrario la instancia de CI\_Loader (method chaining)

#### dbforge()

**mixed dbforge**([ **object \$db = NULL** [, **bool \$return = FALSE**]])

##### Descripción

---

Carga la clase **Database Forge**

##### Parámetros

---

###### db

Objeto de Base de datos

###### return

Si se devuelve la instancia de **Database Forge**

##### Valores devueltos

---

Instancia de CI\_DB\_forge cargada si **\$return** es TRUE, de lo contrario la instancia de CI\_Loader (method chaining)

#### dbutil()

**mixed dbutil**([ **object \$db = NULL** [, **bool \$return = FALSE**]])

##### Descripción

---

Carga la clase **Database Utilities**

##### Parámetros

---

###### db

Objeto de Base de datos

###### return

Si se devuelve la instancia de **Database Utilities**

##### Valores devueltos

---

Instancia de CI\_DB\_utility cargada si **\$return** es TRUE, de lo contrario la instancia de CI\_Loader (method chaining)

driver()

**object driver(mixed \$library [, array \$params = NULL [, string \$object\_name]])**

---

#### Descripción

Este método se usa para cargar las bibliotecas de los controladores, actúa de manera muy similar al método **library()**

---

#### Parámetros

##### **library**

Nombre de la biblioteca como un string o un array con múltiples bibliotecas

##### **params**

Array opcional de parámetros para pasar al constructor de la biblioteca cargada

##### **object\_name**

Nombre de objeto opcional para asignar a la biblioteca

---

#### Valores devueltos

Instancia de CI\_Loader (method chaining)

---

#### Ejemplo

Si desea utilizar sesiones con CodeIgniter, el primer paso es cargar el session driver dentro de su controlador:

```
$this->load->driver('session');
```

Una vez cargada, la biblioteca estará lista para usar, usando **\$this->session**

Los archivos del controlador deben almacenarse en un subdirectorio dentro del directorio principal **"libraries"**, o dentro de su directorio personal **application/libraries**, el subdirectorio debe coincidir con el nombre de la clase principal. Lea la descripción de **Drivers** para más detalles

Además, se pueden cargar varias bibliotecas de controladores al mismo tiempo pasando un array de controladores al método de carga:

```
$this->load->driver(array('session', 'cache'));
```

#### Opciones de configuración

El segundo parámetro, opcional, le permite pasar opciones de configuración, por lo general, pasará estos como un array:

```
$config = array(
    'sess_driver'          => 'cookie',
    'sess_encrypt_cookie' => true,
    'encryption_key'     => 'clave_secreta'
);

$this->load->driver('session', $config);
```

Las opciones de configuración generalmente también se pueden establecer a través de un archivo de configuración, cada biblioteca lo explica en detalle en su propia sección, por lo tanto, lea la información relativa a cada una

#### Asignación de un controlador a un nombre de objeto diferente

Si el tercer parámetro, opcional, está en blanco, la biblioteca se asignará a un objeto con el mismo nombre que la clase padre, por ejemplo, si la biblioteca se llama **Session**, se asignará a una variable llamada **`$this->session`**

Si prefiere establecer sus propios nombres de clase, puede pasar su valor al tercer parámetro:

```
$this->load->library('session', '', 'mi_session');

// Ahora se accede a la clase de sesión usando:
$this->mi_session
```

## file()

**mixed file(string \$path [, bool \$return = FALSE])**

### Descripción

---

Este es un método genérico de carga de archivos, proporcione la ruta de archivo y el nombre en el primer parámetro y se abrirá y leerá el archivo, por defecto, los datos se envían a su navegador, al igual que un archivo de vista, pero si establece el segundo parámetro como booleano TRUE, en su lugar devolverá los datos como un string

### Parámetros

---

#### path

Ruta del archivo

#### return

Si se debe devolver el archivo cargado

### Valores devueltos

---

Contenidos del archivo si **\$return** está establecido en TRUE, de lo contrario la instancia de CI\_Loader (method chaining)

## get\_package\_paths()

**array get\_package\_paths([ bool \$include\_base = TRUE])**

### Descripción

---

Devuelve todas las rutas de paquetes disponibles actualmente

### Parámetros

---

#### include\_base

Si se debe incluir BASEPATH

### Valores devueltos

---

Array de rutas de paquetes

`get_var()`

**mixed** `get_var(string $key)`

---

#### Descripción

Este método verifica el array asociativo de variables disponibles para sus vistas, esto es útil si por alguna razón se establece una variable en una biblioteca u otro método de controlador usando `$this->load->vars()`

---

#### Parámetros

##### **key**

Clave de nombre de variable

---

#### Valores devueltos

Valor si se encuentra la clave, NULL si no

`get_vars()`

**array** `get_vars()`

---

#### Descripción

Este método recupera todas las variables disponibles para sus vistas

---

#### Valores devueltos

Un array de todas las variables de vista asignadas

`helper()`

**object** `helper(mixed $helpers)`

---

#### Descripción

Este método carga archivos auxiliares **Helper**

---

#### Parámetros

##### **helpers**

Nombre del helper como un string o un array que contiene múltiples helpers, el nombre es sin la extensión `_helper.php`

---

#### Valores devueltos

Instancia de `CI_Loader` (method chaining)

`is_loaded()`

**mixed** `is_loaded(string $class)`

---

#### Descripción

Le permite verificar si una clase ya se ha cargado o no

---

#### **Nota**

La palabra "clase" aquí se refiere a bibliotecas y controladores

## Parámetros

---

### class

Nombre de la clase

### Valores devueltos

---

Nombre de propiedad Singleton si se encuentra, FALSE si no

Si la clase solicitada se ha cargado, el método devuelve su nombre asignado en el superobjeto de CI y FALSE si no es así:

```
$this->load->library('form_validation');  
$this->load->is_loaded('Form_validation'); // devuelve 'form_validation'  
  
$this->load->is_loaded('Librería_que_no_existe'); // devuelve FALSE
```

### **Importante**

Si tiene más de una instancia de una clase (asignada a diferentes propiedades), se devolverá la primera

```
$this->load->library('form_validation', $config, 'fv');  
$this->load->library('form_validation');  
  
$this->load->is_loaded('Form_validation'); // devuelve 'fv'
```

## language()

**object language(mixed \$files [, string \$lang = "])**

### Descripción

---

Este método es un alias del método de carga del lenguaje: **`$this->lang->load()`**

### Parámetros

---

#### files

Nombre del archivo de idioma o un array de múltiples archivos de idioma

#### lang

Nombre del idioma

### Valores devueltos

---

Instancia de CI\_Loader (method chaining)

## library()

**object library(mixed \$library [, array \$params = NULL [, string \$object\_name = NULL]])**

### Descripción

---

Este método se usa para cargar clases principales

### Parámetros

---



## library

Nombre de la biblioteca como un string o un array con múltiples bibliotecas

## params

Array opcional de parámetros para pasar al constructor de la biblioteca cargada

## object\_name

Nombre de objeto opcional para asignar la biblioteca

## Valores devueltos

---

Instancia de CI\_Loader (method chaining)

## Ejemplo

---

### **Nota**

Usamos los términos "clase" y "bibliotecas" indistintamente

Si desea enviar un correo electrónico con CodeIgniter, el primer paso es cargar la clase **email** dentro de su controlador:

```
$this->load->library('email');
```

Una vez cargada, la biblioteca esta lista para usarse, usando:

```
$this->email
```

Los archivos de la biblioteca pueden almacenarse en subdirectorios dentro del directorio principal "**libraries**", o dentro de su directorio **application/libraries**, para cargar un archivo ubicado en un subdirectorio, simplemente incluya la ruta, relativa al directorio "**libraries**"

Por ejemplo, si tiene un archivo ubicado en:

```
libraries/pintores/Goya.php
```

Lo cargará usando:

```
$this->load->library('pintores/goya');
```

Puede anidar el archivo en tantos subdirectorios como desee

Además, se pueden cargar múltiples bibliotecas al mismo tiempo pasando un array de bibliotecas al método de carga:

```
$this->load->library(array('email', 'table'));
```

## Opciones de configuración

El segundo parámetro, opcional, le permite opcionalmente pasar la configuración, por lo general, pasará estos como un array:

```

$config = array (
    'mailtype' => 'html',
    'charset' => 'utf-8',
    'priority' => '1'
);

$this->load->library('email', $config);

```

Las opciones de configuración generalmente también se pueden establecer a través de un archivo de configuración, cada biblioteca lo explica en detalle, por lo tanto, lea la información relativa a cada una

Tenga en cuenta que cuando se proporcionan varias bibliotecas en un array en el primer parámetro, cada una recibirá la misma información de parámetros

### Asignar una biblioteca a un nombre de objeto diferente

Si el tercer parámetro, opcional, está en blanco, la biblioteca generalmente se asignará a un objeto con el mismo nombre que la biblioteca, por ejemplo, si la biblioteca se llama **Calendar**, se asignará a una variable llamada **\$this->calendar**

Si prefiere establecer sus propios nombres de clase, puede pasar su valor en el tercer parámetro:

```

$this->load->library('calendar', NULL, 'mi_calendario');

// Ahora se accede a la clase de calendario usando:
$this->mi_calendario

```

Tenga en cuenta que cuando se suministran varias bibliotecas en un array en el primer parámetro, este parámetro se descarta

### model()

**object model(mixed \$model [, string \$name = " [, string \$db\_conn = FALSE]])**

#### Descripción

---

Este método se usa para cargar sus archivos de modelo

#### Parámetros

---

##### model

Nombre del modelo o un array que contiene varios modelos

##### name

Si desea que a su modelo se le asigne un nombre de objeto diferente, puede especificarlo mediante este parámetro:

```

$this->load->model('nombre_modelo', 'ieso');
$this->ieso->method();

```

##### db\_conn

Opcional, configuración de la Base de datos a cargar

#### Valores devueltos

---

Instancia de CI\_Loader (method chaining)

#### Ejemplo

---

```
$this->load->model('nombre_del_modelo');
```

Si su modelo está ubicado en un subdirectorio, incluya la ruta relativa desde su directorio de modelos, por ejemplo, si tiene un modelo ubicado en **application/models/blog/Consultas.php**, lo cargará usando:

```
$this->load->model('blog/consultas');
```

#### **remove\_package\_path()**

**object remove\_package\_path([ string \$path = '' ])**

#### Descripción

---

Cuando su controlador haya terminado de usar los recursos de un paquete de aplicaciones, y particularmente si tiene otros paquetes de aplicaciones con los que desea trabajar, para que el cargador ya no busque recursos en ese directorio es posible que desee eliminar la ruta del paquete, para eliminar la última ruta agregada, simplemente llame al método sin parámetros, o para eliminar una ruta específica del paquete, especifique la misma ruta previamente dada a **add\_package\_path ()**:

```
$this->load->remove_package_path(APPPATH.'third_party/leonardo/');
```

#### Parámetros

---

##### **path**

Ruta para eliminar

#### Valores devueltos

---

Instancia de CI\_Loader (method chaining)

#### **vars()**

**object vars(mixed \$vars [, mixed \$val = '' ])**

#### Descripción

---

Este método toma un array asociativo como entrada y genera variables usando la función PHP **extract()**, este método produce el mismo resultado que usar el segundo parámetro del método **\$this->load->view ()** anterior, la razón por la que utilizar este método de forma independiente es si desea establecer algunas variables globales en el constructor de su controlador y hacer que estén disponibles en cualquier archivo de vista cargado desde cualquier método, puede tener múltiples llamadas a este método, los datos se almacenan en caché y se fusionan en un array para la conversión a variables

#### Parámetros

---

##### **vars**

Un array de variables o un solo nombre de variable

##### **val**

Valor de variable opcional

#### Valores devueltos

---

## view()

**mixed view(string \$view [, array \$vars = array() [, bool return = FALSE]])**

### Descripción

---

Este método se usa para cargar sus archivos de vista. En la sección **Views** se muestra cómo se usa este método

### Parámetros

---

#### view

Es el nombre del archivo de vista que desea cargar, no necesita especificar la extensión **.php**, a menos que esté usando otra distinta

#### vars

Parámetro opcional que puede tomar como entrada un array asociativo o un objeto, que se ejecuta mediante la función **extract()** de PHP para convertir a variables que se pueden usar en sus archivos de vista

#### return

Si se devuelve la vista cargada, parámetro opcional que le permite cambiar el comportamiento del método para que devuelva los datos como un string en lugar de enviarlos al navegador, puede ser útil si quiere procesar los datos de alguna forma, si lo establece a TRUE, devolverá datos, por defecto es FALSE, que los envía al navegador. Recuerde asignarla a una variable si quiere que los datos sean devueltos:

```
$string = $this->load->view('mi_archivo', '', TRUE);
```

### Valores devueltos

---

El string de contenido si **\$return** está establecido en TRUE, de lo contrario la instancia de CI\_Loader (method chaining)

## class CI\_Migration

---

### class CI\_Migration

**class CI\_Migration**

### current

**mixed current()**

### Descripción

---

Migra hasta la versión actual, lo que está configurado para **\$config['migration\_version']** en **application/config/migration.php**

### Valores devueltos

---

TRUE si no se encuentran migraciones, string de versión actual en caso de éxito, FALSE en caso de error

### error\_string()

**string error\_string()**

### Descripción

---

Devuelve un string con los errores que se detectaron al realizar una migración

### Valores devueltos

---

Mensajes de error

### find\_migrations()

#### array find\_migrations()

### Descripción

---

Se devuelve un array de nombres de archivos de migración que se encuentran en la propiedad **migration\_path**

### Valores devueltos

---

Un array de archivos de migración

### latest()

#### mixed latest()

### Descripción

---

Esto funciona de forma muy similar a **current()**, pero en lugar de buscar en **\$config['migration\_version']** la clase **Migration** utilizará la migración más reciente que se encuentre en el sistema de archivos

### Valores devueltos

---

String de versión actual en caso de éxito, FALSE en caso de error

### version()

#### mixed version(mixed \$target\_version)

### Descripción

---

La versión se puede utilizar para deshacer cambios o avanzar de forma programática a versiones específicas, funciona igual que **current()** pero ignora **\$config['migration\_version']**

### Parámetros

---

#### target\_version

Versión de migración para procesar

### Valores devueltos

---

TRUE si no se encuentran migraciones, string de versión actual en caso de éxito, FALSE en caso de error

### Ejemplo

---

```
$this->migration->version(5);
```

# class CI\_Output

---

class CI\_Output

**class CI\_Output**

\$parse\_exec\_vars

**\$parse\_exec\_vars = TRUE;**

Habilita/deshabilita el análisis de las pseudovariables **{elapsed\_time}** y **{memory\_usage}**

CodeIgniter analizará esos tokens en su salida de forma predeterminada, para desactivar esto, establezca esta propiedad en FALSE en su controlador:

```
$this->output->parse_exec_vars = FALSE;
```

append\_output()

**object append\_output(string \$output)**

**Descripción**

---

Añade datos en el string de salida

**Parámetros**

---

**output**

Datos de salida adicionales para agregar

**Valores devueltos**

---

Instancia de CI\_Output (method chaining)

**Ejemplo**

---

```
$this->output->append_output($datos);
```

cache()

**object cache(int \$time)**

**Descripción**

---

Almacena en caché la página actual durante la cantidad de minutos especificada, para obtener más información, consulte la sección de **almacenamiento en caché**

**Parámetros**

---

**time**

Tiempo de caducidad de caché en minutos

**Valores devueltos**

---

Instancia de CI\_Output (method chaining)

## `_display()`

`void _display([ string $output = ''])`

### Descripción

---

Envía datos de salida finalizados al navegador junto con los encabezados de servidor, también detiene los temporizadores de referencia

### Parámetros

---

#### **output**

Anulación de datos de salida

### **Nota**

Este método se llama automáticamente al final de la ejecución del script, no tendrá que llamarlo manualmente a menos que esté cancelando la ejecución del script usando **exit()** o **die()** en su código

### valores devueltos

---

void

### Ejemplo

---

```
$response = array('status' => 'OK');

$this->output
->set_status_header(200)
->set_content_type('application/json', 'utf-8')
->set_output(json_encode($response, JSON_PRETTY_PRINT | JSON_UNESCAPED_UNICODE |
JSON_UNESCAPED_SLASHES))
->_display();
exit;
```

### **Nota**

Llamar a este método manualmente sin abortar la ejecución del script generará una salida duplicada

## `enable_profiler`

`object enable_profiler([ bool $val = TRUE])`

### Descripción

---

Le permite habilitar/deshabilitar el **Profiler**, que mostrará los puntos de referencia y otros datos en la parte inferior de sus páginas para fines de depuración y optimización

### Parámetros

---

#### **val**

Si habilitar o deshabilitar el Profiler

### Valores devueltos

---

Instancia de CI\_Output (method chaining)

### Ejemplo

---

Para habilitar el generador de perfiles, coloque la siguiente línea en cualquier lugar dentro de sus métodos de Controlador:

```
$this->output->enable_profiler(TRUE);
```

Cuando se habilite, se generará un informe y se insertará en la parte inferior de sus páginas

Para deshabilitar el generador de perfiles, debe usar:

```
$this->output->enable_profiler(FALSE);
```

## get\_content\_type

### string get\_content\_type()

#### Descripción

---

Devuelve el encabezado **HTTP Content-Type** que está actualmente en uso, excluyendo el valor del conjunto de caracteres

#### Valores devueltos

---

String de tipo de contenido

#### Ejemplo

---

```
$mime= $this->output->get_content_type();
```

#### **Nota**

Si no se establece, el valor de retorno predeterminado es **'text/html'**

## get\_header()

### mixed get\_header(string \$header)

#### Descripción

---

Devuelve el valor del encabezado HTTP solicitado o NULL si el encabezado solicitado no está establecido

#### Parámetros

---

##### **header**

Nombre del encabezado HTTP

#### Valores devueltos

---

Encabezado de respuesta HTTP o NULL si no se encuentra

#### Ejemplo

---

```
$this->output->set_content_type('text/plain', 'UTF-8');  
echo $this->output->get_header('content-type');  
// Salidas: text/plain; charset=utf-8
```

#### **Nota**



- El nombre del encabezado se compara de una manera insensible a mayúsculas y minúsculas
- Los encabezados sin formato enviados a través de la función del encabezado nativo de PHP() también se detectan

## get\_output()

### string get\_output()

#### Descripción

Le permite recuperar manualmente cualquier salida que se haya enviado para su almacenamiento en la clase output, tenga en cuenta que los datos solo se podrán recuperar de este método si se han enviado previamente a la clase **output** mediante una de las funciones de CodeIgniter como **\$this->load->view()**

#### Valores devueltos

String de salida

#### Ejemplo

```
$string = $this->output->get_output();
```

## set\_content\_type

### object set\_content\_type(string \$mime\_type [, string \$charset = NULL])

#### Descripción

Le permite configurar el tipo de MIME de su página para que pueda servir datos JSON, JPEG, XML, etc. fácilmente

#### Parámetros

##### mime\_type

String con el tipo MIME

##### charset

Conjunto de caracteres

Puede establecer el conjunto de caracteres del documento, usando este parámetro:

```
$this->output->set_content_type('css', 'utf-8');
```

#### Valores devueltos

Instancia de CI\_Output (method chaining)

#### Ejemplo

```
$this->output->set_content_type('application/json')->set_output(json_encode(array('foo' => 'bar')));  
  
// También podría usar ".jpeg" que tendrá el punto eliminado antes de buscar en config/mimes.php  
$this->output->set_content_type('jpeg')->set_output(file_get_contents('files/algo.jpg'));
```

#### Importante

Asegúrese de que cualquier string **no\_mime** que pase a este método exista en **application/config/mimes.php** o no tendrá efecto

## set\_header

**object set\_header(string \$header [, bool \$replace = TRUE])**

### Descripción

Le permite configurar manualmente los encabezados del servidor, que la clase de salida enviará por usted cuando entregue la visualización renderizada final

### Parámetros

#### header

Encabezado de respuesta HTTP

#### replace

Si se reemplaza el antiguo valor del encabezado, si ya está configurado

### Valores devueltos

Instancia de CI\_Output (method chaining)

### Ejemplo

```
$this->output->set_header('HTTP/1.0 200 OK');  
$this->output->set_header('HTTP/1.1 200 OK');  
$this->output->set_header('Last-Modified: '.gmdate('D, d M Y H:i:s', $last_update).' GMT');  
$this->output->set_header('Cache-Control: no-store, no-cache, must-revalidate');  
$this->output->set_header('Cache-Control: post-check=0, pre-check=0');  
$this->output->set_header('Pragma: no-cache');
```

## set\_output()

**object set\_output(string \$output)**

### Descripción

Le permite establecer manualmente el string de salida final

### Parámetros

#### output

Cadena para establecer el resultado de salida

### Valores devueltos

Instancia de CI\_Output (method chaining)

### Ejemplo

```
$this->output->set_output($datos);
```

## Importante

Si configura su salida manualmente, debe ser lo último que haga en el método desde lo que lo llama, por ejemplo, si construye una página en uno de sus métodos de controlador, no configure la salida hasta el final

**set\_profiler\_sections()**

**object set\_profiler\_sections(array \$sections)**

#### Descripción

Le permite habilitar/deshabilitar secciones específicas del **Profiler** cuando está habilitado, consulte la documentación de **Profiler** para obtener más información

#### Parámetros

##### sections

Secciones de Profiler

#### Valores devueltos

Instancia de CI\_Output (method chaining)

**set\_status\_header()**

**object set\_status\_header([ int \$code = 200 [, string \$text = '']])**

#### Descripción

Le permite configurar manualmente un encabezado de estado del servidor

#### Parámetros

##### code

Código de estado HTTP

##### text

Mensaje opcional

#### Valores devueltos

Instancia de CI\_Output (method chaining)

#### Ejemplo

```
//Establece el encabezado como: no autorizado
$this->output->set_status_header(401);
```

Vea aquí para una lista completa de encabezados: <http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html>

#### Nota

Este método es un alias para la función común: **set\_status\_header()**

## class CI\_Pagination

---

class CI\_Pagination

**class CI\_Pagination**

create\_links()

**string create\_links()**

Descripción

---

Devuelve una barra de "**paginación**" que contiene los enlaces generados o una string vacía si solo hay una página

Valores devueltos

---

Paginación con formato HTML

initialize()

**object initialize**([ **array \$params = array()**])

Descripción

---

Inicializa la clase de paginación con sus opciones preferidas

Parámetros

---

**params**

Parámetros de configuración

Valores devueltos

---

Instancia CI\_Pagination (method chaining)

## class CI\_Parser

---

class CI\_Parser

**class CI\_Parser**

parse()

**string parse**(**string \$template**, **array \$data** [, **bool \$return = FALSE**])

Descripción

---

Analiza una plantilla en la ruta y variables proporcionadas

Parámetros

---

**template**

Ruta para del archivo de vista

## data

Variables con los datos

## return

Si solo devolverá la plantilla analizada

## Valores devueltos

---

String de plantilla analizada

## parse\_string()

**string parse\_string(string \$template, array \$data [, bool \$return = FALSE])**

## Descripción

---

Este método funciona exactamente como **parse()**, solo que acepta la plantilla como un string en lugar de cargar un archivo de vista

## Parámetros

---

### template

String con la plantilla

### data

Variables con los datos

### return

Si solo devolverá la plantilla analizada

## Valores devueltos

---

String de plantilla analizada

## set\_delimiters()

**void set\_delimiters([ string \$l = '{', string \$r = '}]')**

## Descripción

---

Establece los delimitadores, apertura y cierre, para una **"etiqueta"** pseudovariante en una plantilla

## Parámetros

---

### l

Delimitador izquierdo

### r

Delimitador derecho

## valores devueltos

---

void

## class CI\_Security

---

class CI\_Security

### class CI\_Security

entity\_decode

**string** entity\_decode(**string** \$str [, **string** \$charset = NULL])

#### Descripción

---

Este método funciona de forma muy parecida a la función **html\_entity\_decode()** nativa de PHP en el modo **ENT\_COMPAT**, solo trata de detectar entidades HTML que no terminan en un punto y coma porque algunos navegadores lo permiten, si el parámetro **\$charset** se deja vacío, entonces se usará el valor configurado en **\$config['charset']**

#### Parámetros

---

##### **str**

String de entrada

##### **charset**

Conjunto de caracteres de la string de entrada

#### Valores devueltos

---

Cadena decodificada

get\_csrf\_hash()

**string** get\_csrf\_hash()

#### Descripción

---

Devuelve el valor hash CSRF, útil en combinación con **get\_csrf\_token\_name()** para crear formularios manualmente o enviar solicitudes AJAX POST válidas

#### Valores devueltos

---

Hash CSRF

get\_csrf\_token\_name()

**string** get\_csrf\_token\_name()

#### Descripción

---

Devuelve el nombre del token CSRF, el valor **\$config['csrf\_token\_name']**

#### Valores devueltos

---

Nombre del token CSRF

`get_random_bytes()`

**string** `get_random_bytes(int $length)`

---

#### Descripción

Usado para generar tokens CSRF y XSS, método conveniente para obtener bytes aleatorios adecuados a través de **mcrypt\_create\_iv( )**, **/dev/urandom** o **openssl\_random\_pseudo\_bytes ( )**, en ese orden, si uno de ellos está disponible

---

#### Parámetros

##### **length**

Longitud de salida

---

#### Valores devueltos

Una secuencia binaria de bytes aleatorios o FALSE en caso de error

---

#### **Nota**

No se garantiza que la salida sea criptográficamente segura

`sanitize_filename()`

**string** `sanitize_filename(string $str [, bool $relative_path = FALSE])`

---

#### Descripción

Intenta desinfectar los nombres de los archivos para evitar intentos de cruce de directorios y otras amenazas de seguridad, es útil para los archivos que se suministran a través de la entrada del usuario

---

#### Parámetros

##### **str**

Nombre de archivo / ruta

##### **relative\_path**

Si desea conservar cualquier directorio en la ruta del archivo

---

#### Valores devueltos

Nombre/ruta del archivo desinfectado

---

#### Ejemplo

```
$filename = $this->security->sanitize_filename($this->input->post('nombre_archivo'));
```

Si es aceptable que la entrada del usuario incluya rutas relativas, por ejemplo

**archivo/en/alguna/carpeta/aprobada.txt**, puede establecer el segundo parámetro opcional **\$relative\_path** en TRUE:

```
$filename = $this->security->sanitize_filename($this->input->post('nombre_archivo'), TRUE);
```

`xss_clean()`

`mixed xss_clean(mixed $str [, bool $is_image = FALSE])`

#### Descripción

---

Intenta eliminar exploits XSS de los datos de entrada y devuelve el string limpio, si el segundo parámetro opcional se establece en verdadero, devolverá TRUE si la imagen es segura de utilizar y FALSE si se detectan datos maliciosos en ella

#### Parámetros

---

##### **str**

Cadena de entrada o un array de strings

##### **is\_image**

Opcional, permite que el método se use para comprobar imágenes

#### Valores devueltos

---

Datos XSS-clean

#### **Importante**

Este método no es adecuado para filtrar valores de atributos HTML. Use `html_escape()` para ello

## class CI\_Session

---

`class CI_Session`

`class CI_Session`

`all_userdata()`

`array all_userdata()`

#### **Nota**

Este método está **ABANDONADO**. Use `userdata()` sin parámetros en su lugar

`flashdata()`

`mixed flashdata([ mixed $key = NULL])`

#### **Nota**

Este es un método heredado que se mantiene solo por compatibilidad con versiones anteriores de aplicaciones anteriores. Debe acceder directamente a `$_SESSION` en su lugar

`__get()`

`mixed __get(string $key)`



## Descripción

---

Un método mágico que le permite usar `$this->session->item` en lugar de `$_SESSION['item']`, si eso es lo que prefiere, también devolverá el ID de sesión llamando a `session_id()` si intenta acceder a `$this->session->session_id`

## Parámetros

---

### key

Clave de elemento de sesión

## Valores devueltos

---

El elemento de datos de la sesión solicitada, o NULL si no existe

## get\_flash\_keys()

**array** `get_flash_keys()`

## Descripción

---

Obtiene una lista de todos los elementos `$_SESSION` que se han marcado como `'flashdata'`

## Valores devueltos

---

Array que contiene las claves de todos los elementos `'flashdata'`

## get\_temp\_keys()

**array** `get_temp_keys()`

## Descripción

---

Obtiene una lista de todos los elementos `$_SESSION` que se han marcado como `'tempdata'`

## Valores devueltos

---

Un array que contiene las claves de todos los elementos `'tempdata'`

## &get\_userdata()

**array** `&get_userdata()`

### **Nota**

Este es un método heredado que se mantiene solo por compatibilidad con versiones anteriores de aplicaciones anteriores

## has\_userdata()

**bool** `has_userdata(string $key)`

### **Nota**

Este es un método heredado que se mantiene solo por compatibilidad con versiones anteriores de aplicaciones anteriores, es solo un alias para `isset($_SESSION[$key])` , úselo en su lugar

## keep\_flashdata()

**bool** `keep_flashdata(mixed $key)`

### Nota

Este es un método heredado que se mantiene solo por compatibilidad con versiones anteriores, es un alias para el método **mark\_as\_flash()**

**mark\_as\_flash()**

**bool mark\_as\_flash(mixed \$key)**

#### Descripción

Marca una clave de elemento **\$\_SESSION** (o varias) como **'flashdata'**

#### Parámetros

##### key

Clave para marcar como flashdata, o un array de múltiples claves

#### Valores devueltos

TRUE en el éxito, FALSE si falla

**mark\_as\_temp()**

**bool mark\_as\_temp(mixed \$key [, int \$ttl = 300])**

#### Descripción

Marca una clave de elemento **\$\_SESSION** (o varias) como **'tempdata'**

#### Parámetros

##### key

Clave para marcar como tempdata, o un array de múltiples claves

##### ttl

Valor de tiempo de vida, en segundos, para los tempdatos

#### Valores devueltos

TRUE en el éxito, FALSE si falla

**sess\_destroy()**

**void sess\_destroy()**

### Nota

Este método es solo un alias para la función **session\_destroy()** nativa de PHP

#### Descripción

Destruye la sesión actual

### Nota

Esta debe ser la última función relacionada con la sesión a la que llama, todos los datos de la sesión se perderán después de hacer eso

valores devueltos

---

void

**sess\_regenerate()**

**void sess\_regenerate**([ **bool \$destroy = FALSE**])

### Nota

Este método es solo un alias para la función nativa **session\_regenerate\_id()** de PHP

### Descripción

---

Vuelve a generar la ID de la sesión, destruyendo opcionalmente los datos de la sesión actual

### Parámetros

---

#### **destroy**

Si se destruyen los datos de la sesión

valores devueltos

---

void

**\_\_set()**

**void \_\_set**(**string \$key**, **mixed \$value**)

### Descripción

---

Un método mágico que le permite asignar elementos a **\$\_SESSION** accediendo a ellos como **\$this->session->propiedad de sesión**

### Parámetros

---

#### **key**

Clave de elemento de sesión

#### **value**

Valor para asignar a la clave del elemento de sesión

valores devueltos

---

void

### Ejemplo

---

```
$this->session->foo = 'milú';  
// Resultado:  
// $_SESSION['foo'] = 'milú';
```

**set\_flashdata()**

**void set\_flashdata**(**mixed \$data** [, **mixed \$value = NULL**])

### Nota

Este es un método heredado que se mantiene solo por compatibilidad con versiones anteriores

### set\_tempdata()

```
void set_tempdata(mixed $data [, mixed $value = NULL, int $ttl = 300])
```

### Nota

Este es un método heredado que se mantiene solo por compatibilidad con versiones anteriores de aplicaciones anteriores

### set\_userdata()

```
void set_userdata(mixed $data [, mixed $value = NULL])
```

### Nota

Este es un método heredado que se mantiene solo por compatibilidad con versiones anteriores de aplicaciones anteriores

### tempdata()

```
mixed tempdata([ mixed $key = NULL])
```

### Nota

Este es un método heredado que se mantiene solo por compatibilidad con versiones anteriores de aplicaciones anteriores, debe acceder directamente a **\$\_SESSION** en su lugar

### unmark\_flash()

```
void unmark_flash(mixed $key)
```

#### Descripción

---

Desmarca una clave de elemento **\$\_SESSION** (o varias) como **'flashdata'**

#### Parámetros

---

#### key

Clave que se debe anular como flashdata, o un array de múltiples claves

#### valores devueltos

---

void

### unmark\_temp()

```
void unmark_temp(mixed $key)
```

#### Descripción

---

Desmarca una clave de elemento **\$\_SESSION** (o varias) como **'tempdata'**

#### Parámetros

---

#### key

Clave que se debe eliminar como tempdata, o un array de múltiples claves

valores devueltos

---

void

**unset\_userdata()**

**void unset\_userdata(mixed \$key)**

**Nota**

Este es un método heredado que se mantiene solo por compatibilidad con versiones anteriores de aplicaciones anteriores. Es solo un alias para **unset(\$\_SESSION[\$key])** , úselo en su lugar

**userdata()**

**mixed userdata([ mixed \$key = NULL])**

**Nota**

Este es un método heredado que se mantiene solo por compatibilidad con versiones anteriores de aplicaciones anteriores, debe acceder directamente a **\$\_SESSION** en su lugar

## class CI\_Table

---

**class CI\_Table**

**class CI\_Table**

**\$function**

**\$function = NULL**

Le permite especificar una función PHP nativa o un objeto de array de función válida para aplicar a todos los datos de celda

**Ejemplo**

---

```
$this->load->library('table');

$this->table->set_heading('Pintor', 'Obra', 'Fecha');
$this->table->add_row('Salvador Dalí', '<strong>Relojes blandos</strong>', '1931');

$this->table->function = 'htmlspecialchars';
echo $this->table->generate();
```

En el ejemplo anterior, todos los datos de la celda se ejecutarán a través de la función **htmlspecialchars()** de PHP, lo que da como resultado:

```
<td>Salvador Dalí</td><td>&lt;strong&gt;Relojes blandos&lt;/strong&gt;</td><td>1931</td>
```

**add\_row()**

**object add\_row([ mixed \$args = array() [, ...]])**

## Descripción

---

Le permite agregar una fila a su tabla, puede enviar un array o varias strings

## Parámetros

---

### args

Una array o varias strings que contienen los valores de la fila

## Valores devueltos

---

Instancia de CI\_Table (method chaining)

## Ejemplo

---

```
$this->table->add_row('Gaudí', 'Jujol', 'Domènech i Montaner');

$this->table->add_row(array('Gaudí', 'Jujol', 'Domènech i Montaner'));
```

Si desea establecer los atributos de etiqueta de una celda individual, puede usar un array asociativo para esa celda, los datos de clave asociativos definen los datos de la celda, cualquier otro par **clave =>valor** se agrega como **clave = atributos** a la etiqueta:

```
$celda = array('data' => 'Gaudí', 'class' => 'highlight', 'colspan' => 2);
$this->table->add_row($celda, 'Jujol', 'Domènech i Montaner');

// Produce :
// <td class='highlight' colspan='2'>Gaudí</td><td>Jujol</td><td>Domènech i Montaner</td>
```

## clear()

### object clear()

## Descripción

---

Le permite borrar el encabezado de la tabla y los datos de la fila, si necesita mostrar varias tablas con datos diferentes, debe llamar a este método después de que se haya generado cada tabla para borrar la información de la tabla anterior

## Valores devueltos

---

Instancia CI\_Table (method chaining)

## Ejemplo

---

```
$this->load->library('table');

$this->table->set_heading('Obra', 'Arquitecto', 'Fecha');
$this->table->add_row('Sagrada Familia', 'Gaudí', '1883');
$this->table->add_row('Hospital de San Pablo', 'Domènech i Montaner', '1902');
$this->table->add_row('Trencadís', 'Jujol', '1906');
```

```
echo $this->table->generate();

$this->table->clear();

$this->table->set_heading('Obra', 'Escritor', 'Publicación');
$this->table->add_row('El Buscón', 'Quevedo', '1626');
$this->table->add_row('El Quijote', 'Cervantes', '1605');
$this->table->add_row('El proceso', 'Kafka', '1925');

echo $this->table->generate();
```

## generate()

**string generate**([ **mixed \$table\_data = NULL**])

### Descripción

---

Devuelve un string que contiene la tabla generada, acepta un parámetro opcional que puede ser un array o un objeto de resultado de Base de datos

### Parámetros

---

#### **table\_data**

Datos para llenar las filas de la tabla

### Valores devueltos

---

Tabla HTML

## make\_columns()

**array make\_columns**([ **array \$array = array()** [, **int \$col\_limit = 0**])

### Descripción

---

Este método toma un array unidimensional como entrada y crea una array multidimensional con una profundidad igual al número de columnas deseado, esto permite que una única array con muchos elementos se muestre en una tabla que tiene un conteo de columnas fijo

### Parámetros

---

#### **array**

Un array que contiene datos de múltiples filas

#### **col\_limit**

Número de columnas en la tabla

### Valores devueltos

---

Un array de columnas de tabla HTML

### Ejemplo

---

```
$lista = array('uno', 'dos', 'tres', 'cuatro', 'cinco', 'seis', 'siete', 'ocho', 'nueve', 'diez', 'once', 'doce');
```

```

$nueva_lista = $this->table->make_columns($lista, 3);

$this->table->generate($nueva_lista);

// Genera una tabla con este prototipo
<table border="0" cellpadding="4" cellspacing="0">
  <tr>
    <td>uno</td><td>dos</td><td>tres</td>
  </tr><tr>
    <td>cuatro</td><td>cinco</td><td>seis</td>
  </tr><tr>
    <td>siete</td><td>ocho</td><td>nueve</td>
  </tr><tr>
    <td>diez</td><td>once</td><td>doce</td></tr>
</table>

```

## set\_caption()

**object set\_caption(string \$caption)**

### Descripción

---

Permite agregar un título a la tabla

### Parámetros

---

#### caption

Título de la tabla

### Valores devueltos

---

Instancia de CI\_Table (method chaining)

### Ejemplo

---

```
$this->table->set_caption('Pintores');
```

## set\_empty()

**object set\_empty(mixed \$value)**

### Descripción

---

Le permite establecer un valor predeterminado para usar en cualquier celda de la tabla que esté vacía

### Parámetros

---

#### value

Valor para poner en celdas vacías

### Valores devueltos

---

Instancia de CI\_Table (method chaining)

### Ejemplo

---



Establecer un espacio sin ruptura:

```
$this->table->set_empty("&nbsp;");
```

## set\_heading()

**object set\_heading**([ **mixed \$args = array()** [, ...]])

### Descripción

---

Le permite configurar el encabezado de la tabla, puede enviar un array o varias strings

### Parámetros

---

#### args

Una array o varias strings que contienen los títulos de columna de la tabla

### Valores devueltos

---

Instancia de CI\_Table (method chaining)

### Ejemplo

---

```
$this->table->set_heading('Pintor', 'Obra', 'Fecha');

$this->table->set_heading(array('Pintor', 'Obra', 'Fecha'));
```

## set\_template()

**bool set\_template**(array \$template)

### Descripción

---

Le permite configurar su plantilla, puede enviar una plantilla completa o parcial

### Parámetros

---

#### template

Un array asociativa que contiene valores de plantilla

### Valores devueltos

---

TRUE en el éxito, FALSE si falla

### Ejemplo

---

```
$plantilla = array(
    'table_open' => '<table border="1" cellpadding="2" cellspacing="1" class="mytable">'
);

$this->table->set_template($plantilla);
```

## class CI\_Trackback

---

class CI\_Trackback

**class CI\_Trackback**

\$data

```
$data = array('url' => '', 'title' => '', 'excerpt' => '', 'blog_name' => '', 'charset' => '')
```

Array de datos para el Trackback

\$convert\_ascii

```
$convert_ascii = TRUE
```

Si se deben convertir los caracteres ASCII y MS Word a entidades HTML

convert\_ascii()

```
string convert_ascii(string $str)
```

Descripción

---

Convierte texto ASCII y caracteres especiales de MS Word en entidades HTML

Parámetros

---

**str**

String de entrada

Valores devueltos

---

String convertida

convert\_xml()

```
string convert_xml(string $str)
```

Descripción

---

Convierte caracteres XML reservados a entidades

Parámetros

---

**str**

String de entrada

Valores devueltos

---

String convertida

**data()**

**string data(string \$item)**

---

**Descripción**

Devuelve un solo elemento del array de datos de respuesta

---

**Parámetros**

**item**

Clave de datos

---

**Valores devueltos**

Valor de datos o un string vacío si no se encuentra

**display\_errors()**

**string display\_errors([ string \$open = '<p>' [, string \$close = '</p>']])**

---

**Descripción**

Devuelve mensajes de error formateados en HTML o un string vacío si no hay errores

---

**Parámetros**

**open**

Etiqueta de apertura

**close**

Etiqueta de cierre

---

**Valores devueltos**

Mensajes de error en formato HTML

**extract\_urls()**

**array extract\_urls(string \$urls)**

---

**Descripción**

Este método permite el envío de múltiples trackbacks, toma un string de URL, separadas por coma o espacio, y coloca cada URL en un array

---

**Parámetros**

**urls**

Lista de URL separadas por comas

---

**Valores devueltos**

Array de URL

**get\_id()**

**string get\_id(string \$url)**

### Descripción

---

Busca y devuelve un ID de traceback URL

### Parámetros

---

#### url

URL de referencia

### Valores devueltos

---

ID de URL o FALSE en caso de error

### limit\_characters()

```
string limit_characters(string $str [, int $n = 500 [, string $end_char = '&#8230;']])
```

### Descripción

---

Limita el string en función del recuento de caracteres, preservará palabras completas

### Parámetros

---

#### str

String de entrada

#### n

Número máximo de caracteres

#### end\_char

Carácter para poner al final del string

### Valores devueltos

---

String recortado

### process()

```
bool process(string $url, string $data)
```

### Descripción

---

Abre una conexión de socket y pasa los datos al servidor

### Parámetros

---

#### url

URL objetivo

#### data

Datos Raw POST

### Valores devueltos

---

TRUE en el éxito, FALSE si falla

### receive()

```
bool receive()
```

### Descripción

---

Este método valida los datos de traceback entrantes, si los datos son válidos, se incorporan para que pueda insertarse en una Base de datos con **\$this->data**

### Valores devueltos

---

TRUE en caso de éxito, FALSE en caso de error

### send()

**bool send(array \$tb\_data)**

### Descripción

---

Enviar traceback

### Parámetros

---

#### **tb\_data**

Datos de referencia

### Valores devueltos

---

TRUE en el éxito, FALSE si falla

### send\_error()

**void send\_error([ string \$message = 'Incomplete information'])**

### Descripción

---

Responde a una petición traceback con un mensaje de error

### Parámetros

---

#### **message**

Mensaje de error

### **Nota**

Este método terminará la ejecución del script

### valores devueltos

---

void

### send\_success()

**void send\_success()**

### Descripción

---

Responde a una petición traceback con un mensaje de éxito

### **Nota**

Este método terminará la ejecución del script

valores devueltos

---

void

set\_error()

**void set\_error(string \$msg)**

Descripción

---

Establece un registro como un mensaje de error

Parámetros

---

**msg**

Mensaje de error

valores devueltos

---

void

validate\_url()

**void validate\_url(string &\$url)**

Descripción

---

Simplemente agrega el prefijo **http://** si aún no está presente en la URL

Parámetros

---

**url**

URL de referencia

valores devueltos

---

void

## class CI\_Typography

---

class CI\_Typography

**class CI\_Typography**

\$protect\_braced\_quotes

**\$protect\_braced\_quotes = FALSE**

Al utilizar la Typography library junto con la **Template Parser library**, a menudo puede ser conveniente proteger las comillas simples y dobles entre llaves, para habilitar esto, establezca la propiedad de clase **protect\_braced\_quotes** en TRUE

Ejemplo

---

```
$this->load->library('typography');  
$this->typography->protect_braced_quotes = TRUE;
```

## auto\_typography()

**string auto\_typography(string \$str [, bool \$reduce\_linebreaks = FALSE])**

### Descripción

---

Formatea texto para que sea HTML semántica y tipográficamente correctos, toma como entrada un string y lo devuelve con el siguiente formato:

- Rodea párrafos dentro de <p> </p> ,busca saltos de línea dobles para identificar los párrafos
- Los saltos de línea individuales se convierten en <br />, excepto los que aparecen dentro de las etiquetas <pre>
- Los elementos a nivel de bloque, como las etiquetas <div>, no están incluidos en los párrafos, pero sí su texto contenido, si contiene párrafos
- Las comillas se convierten correctamente en comillas tipográficas enfrentadas, excepto aquellas que aparecen dentro de las etiquetas
- Los apóstrofes se convierten en apóstrofes tipográficos
- Los guiones dobles, sea como -- estos o como estos--otros, se convierten en em—dashes
- Tres puntos correlativos sea precediendo o siguiendo a una palabras, se convierten en (...)
- Los espacios dobles siguiendo sentencias se convierten en &nbsp; que imitan el doble espaciado
- Los espacios dobles que siguen las oraciones se convierten en non-breaking spaces para imitar el doble espacio

### Parámetros

---

#### str

String de entrada

#### reduce\_linebreaks

Opcional, si es TRUE reduce más de dos linebreaks (salto de línea '\n') consecutivos a dos

### Valores devueltos

---

Cadena tipográfica HTML-safe

### Ejemplo

---

```
$string = $this->typography->auto_typography($string);
```

### Nota

El formateo tipográfico puede ser intensivo en el uso del procesador, particularmente si hay mucho contenido a formatear, si elige esta función puede querer considerar cachear sus páginas

## format\_characters()

**string format\_characters(string \$str)**

### Descripción

---

Este método es similar a **auto\_typography()**, excepto que solo hace conversión de caracteres:

- Las comillas se convierten correctamente en comillas tipográficas enfrentadas, excepto aquellas que aparecen dentro de las etiquetas
- Las cotizaciones se convierten en entidades de cotización rizada que se enfrentan correctamente, excepto aquellas que aparecen dentro de las etiquetas
- Los apóstrofes se convierten en apóstrofes tipográficos
- Los guiones dobles, sea como -- estos o como estos--otros, se convierten en em—dashes
- Tres puntos correlativos sea precediendo o siguiendo a una palabras, se convierten en (...)

- Los espacios dobles siguiendo sentencias se convierten en &nbsp; que imitan el doble espaciado

#### Parámetros

---

#### **str**

String de entrada

#### Valores devueltos

---

String formateada

#### Ejemplo

---

```
$string = $this->typography->format_characters($string);
```

#### **nl2br\_except\_pre()**

**string nl2br\_except\_pre(string \$str)**

#### Descripción

---

Convierte las nuevas líneas en etiquetas **<br />** a menos que aparezcan dentro de las etiquetas **<pre>**, esta función es idéntica a la función **nl2br()** nativa de PHP, excepto que ignora las etiquetas **<pre>**

#### Parámetros

---

#### **str**

String de entrada

#### Valores devueltos

---

String formateada

#### Ejemplo

---

```
$string = $this->typography->nl2br_except_pre($string);
```

## class CI\_Unit\_test

---

#### **class CI\_Unit\_test**

**class CI\_Unit\_test**

#### **active()**

**void active([ bool \$state = TRUE])**

#### Descripción

---

Habilita/deshabilita la prueba unitaria

#### Parámetros

---

#### **state**

Si se deben habilitar las pruebas



valores devueltos

---

void

**report()**

**string** report([ array \$result = array()])

**Descripción**

---

Genera un informe sobre las pruebas ya completadas

**Parámetros**

---

**result**

Array que contiene los resultados de las pruebas

**Valores devueltos**

---

Informe de prueba

**result()**

**array** result([ array \$results = array()])

**Descripción**

---

Devuelve datos de resultados de pruebas sin procesar

**Parámetros**

---

**results**

Lista de resultados de las pruebas

**Valores devueltos**

---

Array de datos de resultados sin procesar

**run()**

**string** run(mixed \$test [, mixed \$expected = TRUE [, string \$test\_name = 'undefined' [, string \$notes = '' ]]])

**Descripción**

---

Ejecuta pruebas unitarias

**Parámetros**

---

**test**

Datos de prueba

**expected**

Resultado esperado

**test\_name**

Nombre de la prueba

**notes**

Cualquier nota que se adjunte a la prueba

## Valores devueltos

---

Informe de prueba

**set\_template()**

**void set\_template(string \$template)**

### Descripción

---

Establece la plantilla para mostrar los resultados de las pruebas

### Parámetros

---

#### **template**

Plantilla de resultados de prueba

### valores devueltos

---

void

**set\_test\_items()**

**void set\_test\_items(array \$items)**

### Descripción

---

Establece una lista de elementos que deberían ser visibles en las pruebas. Las opciones válidas son:

- test\_name
- test\_datatype
- res\_datatype
- result
- file
- line
- notes

### Parámetros

---

#### **items**

Array con la lista de elementos de prueba visibles

### valores devueltos

---

void

**use\_strict()**

**void use\_strict([ bool \$state = TRUE])**

### Descripción

---

Habilita/deshabilita la comparación de tipos estrictos en las pruebas

### Parámetros

---

#### **state**

Indicador de estado estricto

### valores devueltos

---

void

## URI Class

---

class CI\_URI

**class CI\_URI**

assoc\_to\_uri()

**string assoc\_to\_uri(array \$array)**

### Descripción

---

Toma un array asociativo como entrada y genera un string URI a partir de el, las claves de array se incluirán en el string

### Parámetros

---

#### array

Array de entrada de pares clave/valor

### Valores devueltos

---

Cadena de URI

### Ejemplo

---

```
$array = array('producto' => 'zapato', 'material' => 'gamuza', 'color' => 'azul');
$str = $this->uri->assoc_to_uri($array);

// Crea: producto/zapato/material/gamuza/color/azul
```

rsegment()

**mixed rsegment(int \$n [, mixed \$no\_result = NULL])**

### Descripción

---

Este método es idéntico a **segment()**, excepto que le permite recuperar un segmento específico de su URI reencaminado en caso de que esté utilizando la característica de enrutamiento URI Routing de CodeIgniter

### Parámetros

---

#### n

Número de índice del segmento

#### no\_result

Qué devolver si no se encuentra el segmento buscado

### Valores devueltos

---

Valor del segmento indicado o el valor **\$no\_result** si no se encuentra

rsegment\_array()

**array rsegment\_array()**

### Descripción

---

Este método es idéntico a **segment\_array()**, excepto que devuelve el array de segmentos en su URI redirigido en caso de que esté utilizando la característica de enrutamiento URI Routing de CodeIgniter

### Valores devueltos

---

Array de segmentos de URI enrutados

### ruri\_string()

**string ruri\_string()**

### Descripción

---

Este método es idéntico a **uri\_string()**, excepto que devuelve el URI redirigido en caso de que esté utilizando la característica de enrutamiento URI Routing de CodeIgniter

### Valores devueltos

---

String URI enrutada

### ruri\_to\_assoc()

**array ruri\_to\_assoc([ int \$n = 3 [, array \$default = array()])**

### Descripción

---

Este método es idéntico a **uri\_to\_assoc()**, excepto que crea un array asociativo utilizando el URI redirigido en caso de que esté utilizando la característica de enrutamiento URI Routing de CodeIgniter

### Parámetros

---

#### **n**

Número de índice del segmento

#### **default**

Valores predeterminados

### Valores devueltos

---

Array asociativo de segmentos URI

### segment()

**mixed segment(int \$n [, mixed \$no\_result = NULL])**

### Descripción

---

Le permite recuperar un segmento específico, donde **n** es el número de segmento que desea recuperar, los segmentos están numerados de izquierda a derecha

Por ejemplo, si su URL completa es esta:

```
http://ejemplo.com/index.php/noticias/nacional/gobierno/suben_las_pensiones
```

Los números de segmento serán estos:

1. noticias
2. nacional
3. gobierno
4. suben\_las\_pensiones

## Parámetros

---

### n

Número de índice del segmento

### no\_result

Opcional, valor a devolver si no se encuentra el segmento buscado, se establece por defecto a NULL y permite establecer el valor de retorno del método cuando falta el segmento de URI solicitado

Por ejemplo, esto le indicaría al método que devuelva el número cero en caso de fallo:

```
$product_id = $this->uri->segment(3, 0);
```

Ayuda a evitar tener que escribir código como este:

```
if ($this->uri->segment(3) === FALSE)
{
    $product_id = 0;
}
else
{
    $product_id = $this->uri->segment(3);
}
```

## Valores devueltos

---

Valor de segmento o el valor de **\$no\_result** si no se encuentra

## segment\_array()

### array segment\_array()

#### Descripción

---

Devuelve un array que contiene los segmentos de URI

#### Valores devueltos

---

Array de segmentos URI

#### Ejemplo

---

```
$segs = $this->uri->segment_array();

foreach ($segs as $segment)
{
    echo $segment;
    echo '<br />';
}
```

## slash\_rsegment()

**string slash\_rsegment(int \$n [, string \$where = 'trailing'])**

### Descripción

---

Este método es idéntico a **slash\_segment()**, excepto que le permite agregar barras un segmento específico de su URI redirigido en caso de que esté utilizando la característica URI Routing de CodeIgniter

### Parámetros

---

#### **n**

Número de índice del segmento

#### **where**

Dónde agregar la barra ('trailing' o 'leading')

### Valores devueltos

---

Valor del segmento con una barra inclinada al comienzo/final, o una barra inclinada si no se encuentra

## slash\_segment()

**string slash\_segment(int \$n [, string \$where = 'trailing'])**

### Descripción

---

Este método es casi idéntico a **segment()**, excepto que agrega las barras al comienzo y/o final según el valor del segundo parámetro. Si no se usa este parámetro, se agrega una barra al final

### Parámetros

---

#### **n**

Número de índice del segmento

#### **where**

Dónde agregar la barra ('trailing' o 'leading')

### Valores devueltos

---

Valor del segmento con una barra inclinada al comienzo/final, o una barra inclinada si no se encuentra

### Ejemplo

---

```
$this->uri->slash_segment(3);  
$this->uri->slash_segment(3, 'leading');  
$this->uri->slash_segment(3, 'both');
```

Valores devueltos :

1. segmento/
2. /segmento
3. /segmento/

## total\_rsegments()

**int total\_rsegments()**

## Descripción

---

Este método es idéntico a **total\_segments()**, excepto que devuelve el número total de segmentos en su URI redirigido en caso de que esté utilizando la característica de enrutamiento URI Routing de CodeIgniter

## Valores devueltos

---

Recuento de segmentos de URI enrutados

## **total\_segments()**

### **int total\_segments()**

## Descripción

---

Devuelve la cantidad total de segmentos

## Valores devueltos

---

Recuento de segmentos de URI

## **uri\_string()**

### **string uri\_string()**

## Descripción

---

Devuelve un string con el URI completo

## Valores devueltos

---

String URI

## Ejemplo

---

Si esta es la URL completa:

```
http://ejemplo.com/index.php/noticias/local/345
```

El método devolverá esto:

```
noticias/local/345
```

## **uri\_to\_assoc()**

### **array uri\_to\_assoc([ int \$n = 3 [, array \$default = array()])**

## Descripción

---

Este método le permite convertir segmentos URI en un array asociativo de pares clave/valor

Considere este URI:

```
index.php/usuario/buscar/nombre/carmen/pais/spain/genero/mujer
```

Con este método, puedes convertir el URI en una array asociativa con este prototipo:

```
[array]
(
  'nombre' => 'carmen'
  'país'   => 'spain'
  'genero' => 'mujer'
)
```

El primer parámetro le permite establecer un desplazamiento, que por defecto es **3** ya que su URI normalmente contendrá un par de controlador/método en los segmentos primero y segundo:

```
$array = $this->uri->uri_to_assoc(3);
echo $array['nombre'];
```

El segundo parámetro le permite establecer nombres de clave predeterminados, de modo que el array devuelto contendrá siempre los índices esperados, incluso si faltan en el URI:

```
$default = array('nombre', 'genero', 'país', 'tipo', 'ordenar');
$array = $this->uri->uri_to_assoc(3, $default);
```

Si el URI no contiene un valor valor predeterminado, un índice de array se establecerá a ese nombre, con valor NULL. Por último, si no se encuentra un valor correspondiente para una clave dada, si hay un número impar de segmentos de URI, el valor se establecerá en NULL

#### Parámetros

---

##### **n**

Número de índice del segmento

##### **default**

Valores predeterminados

#### Valores devueltos

---

Serie de segmentos URI asociativos



## class CI\_User\_agent

---

class CI\_User\_agent

**class CI\_User\_agent**

accept\_charset()

**bool accept\_charset([ string \$charset = 'utf-8'])**

### Descripción

---

Permite determinar si el agente de usuario acepta un juego de caracteres en particular

### Parámetros

---

#### charset

Conjunto de caracteres

### Valores devueltos

---

TRUE si el conjunto de caracteres es aceptado, FALSE si no

### Ejemplo

---

```
if ($this->agent->accept_charset('utf-8'))
{
    echo 'Su navegador soporta UTF-8!';
}
```

### Nota

Este método no suele ser muy confiable ya que algunos navegadores no proporcionan información de conjunto de caracteres, e incluso entre los que sí lo hacen, no siempre es preciso

accept\_lang()

**bool accept\_lang([ string \$lang = 'en'])**

### Descripción

---

Determina si el agente de usuario acepta un idioma en particular

### Parámetros

---

#### lang

Clave de idioma

### Valores devueltos

---

TRUE si se acepta el idioma proporcionado, FALSE si no

### Ejemplo

---

```
if ($this->agent->accept_lang('en'))
{
    echo 'Acepta el inglés!';
}
```

## Nota

Este método no suele ser muy confiable, ya que algunos navegadores no proporcionan información de idioma, e incluso entre los que sí lo hacen, no siempre es preciso

## agent\_string()

### string agent\_string()

#### Descripción

---

Obtener el string de agente de usuario completo

Normalmente, será algo como esto:

```
Mozilla/5.0 (Macintosh; U; Intel Mac OS X; en-US; rv:1.8.0.4) Gecko/20060613 Camino/1.0.2
```

#### Valores devueltos

---

String completa de agente de usuario o un string vacío

## browser()

### string browser()

#### Descripción

---

Obtener el nombre del navegador web que visita su sitio

#### Valores devueltos

---

Navegador detectado o un string vacío

## Charsets()

### array Charsets()

#### Descripción

---

Obtener una lista del conjuntos de caracteres aceptados por el agente de usuario

#### Valores devueltos

---

Un array con la lista de conjuntos de caracteres aceptados

## languages()

### array languages()

#### Descripción

---

Obtiene los idiomas admitidos por el agente de usuario

#### Valores devueltos

---

Una array con la lista de idiomas aceptados

mobile()

## string mobile()

### Descripción

---

Averiguar el nombre del dispositivo móvil que visita su sitio

### Valores devueltos

---

Marca del dispositivo móvil detectado o una string vacía

is\_browser()

## bool is\_browser([ string \$key = NULL])

### Descripción

---

Devuelve TRUE/FALSE si el agente de usuario es un navegador web conocido

### Parámetros

---

#### key

Opcional, Nombre del navegador

### Valores devueltos

---

TRUE si el agente de usuario es un navegador (especificado), FALSE si no

### Ejemplo

---

```
if ($this->agent->is_browser('Safari'))
{
    echo 'Está usando Safari.';
}
elseif ($this->agent->is_browser())
{
    echo 'Está usando un navegador.';
}
```

### Nota

La string **"Safari"** en este ejemplo es una clave de array en la lista de definiciones de navegador. Puede encontrar esta lista en **application/config/user\_agents.php** si desea agregar nuevos navegadores o cambiar las opciones

is\_mobile()

## bool is\_mobile([ string \$key = NULL])

### Descripción

---

Devuelve TRUE/FALSE si el agente de usuario es un dispositivo móvil conocido

### Parámetros

---

#### key

Opcional, nombre del dispositivo móvil

### Valores devueltos

---

TRUE si el agente de usuario es un dispositivo móvil (especificado), FALSE si no

#### Ejemplo

```
if ($this->agent->is_mobile('iphone'))
{
    $this->load->view('iphone/home');
}
elseif ($this->agent->is_mobile())
{
    $this->load->view('mobile/home');
}
else
{
    $this->load->view('web/home');
}
```

#### is\_referral()

### bool is\_referral()

#### Descripción

Averiguar si el agente de usuario fue derivado desde otro sitio

#### Valores devueltos

TRUE si el agente de usuario es una referencia, FALSE si no

#### is\_robot()

### bool is\_robot([ string \$key = NULL])

#### Descripción

Devuelve TRUE/FALSE si el agente de usuario es un robot conocido

#### Parámetros

##### key

Opcional, nombre del robot

#### Valores devueltos

TRUE si el agente de usuario es un robot (especificado), FALSE si no

#### Nota

La biblioteca de agente de usuario solo contiene las definiciones de robot más comunes, no es una lista completa de robots, hay cientos de ellos, por lo que buscarlos no sería muy eficiente. Si encuentra que algunos bots que comúnmente visitan su sitio no se encuentran en la lista, puede agregarlos a su **archivo**

**application/config/user\_agents.php**

#### parse()

### void parse(string \$string )

## Descripción

---

Analiza un string personalizado de agente de usuario, es diferente de la información facilitada para el visitante actual

## Parámetros

---

### **string**

Un string personalizada de agente de usuario

## valores devueltos

---

void

## platform()

### **string platform()**

## Descripción

---

Averiguar la plataforma que visualiza su sitio, Linux, Windows, OS X, etc.

## Valores devueltos

---

Sistema operativo detectado o una string vacía

## referrer()

### **string referrer()**

## Descripción

---

El referente, si el agente de usuario fue referido desde otro sitio

Normalmente lo probará de la siguiente manera:

```
if ($this->agent->is_referral())
{
    echo $this->agent->referrer();
}
```

## Valores devueltos

---

Referrer detectado o una string vacía

## robot()

### **string robot()**

## Descripción

---

Obtener el nombre del robot que visita su sitio

## Valores devueltos

---

Nombre del robot detectado o una string vacía

## version()

### **string version()**

## Descripción

---

Obtener el número de versión del navegador web que visita su sitio

## Valores devueltos

---

Versión del navegador detectado o una string vacía

# class CI\_Xmlrpc

---

class CI\_Xmlrpc

**class CI\_Xmlrpc**

display\_error()

**string display\_error()**

## Descripción

---

Devuelve un mensaje de error como un string si su solicitud falló por alguna razón

## Valores devueltos

---

String de mensaje de error

## Ejemplo

---

```
echo $this->xmlrpc->display_error();
```

display\_response()

**mixed display\_response()**

## Descripción

---

Devuelve la respuesta del servidor remoto una vez que se recibe la solicitud, la respuesta normalmente será un array asociativo

## Valores devueltos

---

Respuesta

## Ejemplo

---

```
$this->xmlrpc->display_response();
```

initialize()

**void initialize([ array \$config = array()])**

## Descripción

---

Inicializa la biblioteca XML-RPC

## Parámetros

---

**config**

Array asociativo que contiene su configuración

## valores devueltos

---

void

## method()

**void method(string \$function)**

### Descripción

---

Establece el método que se solicitará desde el servidor XML-RPC

### Parámetros

---

#### **function**

Nombre del método

## valores devueltos

---

void

### Ejemplo

---

```
$this->xmlrpc->method('method');
```

Donde **method** es el nombre del método

## request()

**void request(array \$incoming)**

### Descripción

---

Toma un array de datos y construye la solicitud a enviarse al servidor XML-RPC

### Parámetros

---

#### **incoming**

Datos solicitados

## valores devueltos

---

void

### Ejemplo

---

```
$pregunta = array(array('My Photoblog', 'string'), 'http://www.suite.com/photoblog/');  
$this->xmlrpc->request($pregunta);
```

## send\_error\_message()

**object send\_error\_message(int \$number, string \$message)**

### Descripción

---

Permite enviar un mensaje de error desde su servidor al cliente

## Parámetros

---

### **number**

Número de error

### **mensaje**

Mensaje de error

## Valores devueltos

---

Instancia XML\_RPC\_Response

## Ejemplo

---

```
return $this->xmlrpc->send_error_message(123, 'Datos no disponibles');
```

## send\_request()

### **bool send\_request()**

## Descripción

---

Método de envío de solicitud, devuelve TRUE/FALSE basado en el éxito o error, lo que permite su uso condicional

## Valores devueltos

---

TRUE en caso de éxito, FALSE en caso de error

## server()

### **void server(string \$url [, int \$port = 80 [, string \$proxy = FALSE [, int \$proxy\_port = 8080]])**

## Descripción

---

Establece la URL y el número de puerto del servidor al que se enviará una solicitud

## Parámetros

---

### **url**

URL del servidor XML-RPC

### **port**

Puerto del servidor

### **proxy**

Proxy opcional

### **proxy\_port**

Puerto de escucha proxy

## valores devueltos

---

void

## Ejemplo

---

```
$this->xmlrpc->server('http://www.algunasite.com/pings.php', 80);
```



La autenticación HTTP básica también es compatible, simplemente agréguela a la URL del servidor:

```
$this->xmlrpc->server('http://user:pass@localhost/', 80);
```

**timeout()**

**void timeout(int \$seconds = 5)**

**Descripción**

---

Establece un período de tiempo de espera, en segundos, después del cual se cancelará la solicitud

**Parámetros**

---

**seconds**

Tiempo de espera en segundos

**valores devueltos**

---

void

**Ejemplo**

---

```
$this->xmlrpc->timeout(6);
```

Este período de tiempo se usará tanto para una conexión inicial al servidor remoto, como para obtener una respuesta del mismo, asegúrese de configurar el tiempo de espera antes de llamar a **send\_request()**

## class CI\_Zip

---

**class CI\_Zip**

**class CI\_Zip**

**\$compression\_level**

**\$compression\_level = 2**

El nivel de compresión a usar.

Puede tener un rango de 0 a 9, siendo 9 la compresión más alta y 0 sin compresión:

```
$this->zip->compression_level = 0;
```

**add\_data()**

**void add\_data(mixed \$filepath [, array \$data = NULL])**

**Descripción**

---

Agrega datos al archivo Zip, puede funcionar tanto en modo archivos individuales como múltiples

**Parámetros**

---

**filepath**

---

Una única ruta de archivo o un array de archivo => datos

## data

Contenido del archivo (ignorado si \$filepath es un array)

### valores devueltos

---

void

### Ejemplo

---

Al agregar un solo archivo, el primer parámetro debe contener el nombre que le gustaría darle al archivo y el segundo debe contener el contenido del archivo:

```
$archivo = 'mis_datos1.txt';
$datos = 'Un string con los datos!';
$this->zip->add_data($archivo, $datos);

$archivo = 'mis_datos2.txt';
$datos = 'Un string con otros datos!';
$this->zip->add_data($archivo, $datos);
```

Al agregar múltiples archivos, el primer parámetro debe contener pares **archivo => contenido** y el segundo parámetro se ignora:

```
$datos = array(
    'mis_datos1.txt' => 'Un string con los datos!',
    'mis_datos2.txt' => 'Un string con otros datos!'
);
```

Si desea que sus datos comprimidos se organicen en subdirectorios, simplemente incluya la ruta como parte del nombre de archivo:

```
$archivo = 'cervantes/el_quijote.txt';
$datos = 'En un lugar de la mancha...';

$this->zip->add_data($archivo, $datos);
```

El ejemplo anterior coloca **el\_quijote.txt** dentro de un directorio llamado cervantes

## add\_dir()

**void add\_dir(mixed \$directory)**

### Descripción

---

Crear un directorio, por lo general, este método es innecesario ya que puede colocar sus datos en directorios cuando usa **\$this->zip->add\_data()**, pero si desea crear un directorio vacío, puede hacerlo

### Parámetros

---

#### directory

Cadena de nombre de directorio o un array con múltiples directorios

## valores devueltos

---

void

## Ejemplo

---

```
$this->zip->add_dir('un_directorio'); // Crea un directorio llamado "un_directorio"
```

## archive()

### **bool** archive(**string** \$filepath)

#### Descripción

---

Escribe el archivo Zip en un directorio en su servidor, envíe una ruta de servidor válida que termine con el nombre del archivo, asegúrese de que el directorio sea escribible (755 generalmente está bien)

#### Parámetros

---

##### **filepath**

Ruta al archivo zip de destino

#### Valores devueltos

---

TRUE en el éxito, FALSE si falla

## Ejemplo

---

```
// Crea un archivo llamado mi_archivo.zip
$this->zip->archive('/path/a/directorio/mi_archivo.zip');
```

## clear\_data()

### **void** clear\_data()

#### Descripción

---

La clase Zip guarda en caché los datos zip para que no sea necesario volver a compilar el archivo Zip para cada método que utilice anteriormente, sin embargo, si necesita crear múltiples archivos Zip, cada uno con datos diferentes, puede borrar el caché entre llamadas

## valores devueltos

---

void

## Ejemplo

---

```
$nombre = 'obelix.txt';
$datos = 'Están locos estos romanos';

$this->zip->add_data($nombre, $datos);
$zip_file = $this->zip->get_zip();

$this->zip->clear_data();

$nombre = 'foto.jpg';

// Leer el contenido del archivo
$this->zip->read_file("/path/a/foto.jpg");
$this->zip->download('mifoto.zip');
```

## download()

**void download(string \$filename = 'backup.zip')**

### Descripción

---

Hace que el archivo Zip se descargue del servidor, debe pasar el nombre al que desea que se llame el archivo comprimido

### Parámetros

---

#### filename

Nombre de archivo

### valores devueltos

---

void

### Ejemplo

---

```
// El archivo se llamará "incontinencia_suma.zip"
$this->zip->download('incontinencia_suma.zip');
```

### Nota

No mostrar ningún dato en el controlador que llama este método, ya que envía varios encabezados de servidor que hacen que la descarga suceda y el archivo se trata como binario

## get\_zip()

**string get\_zip()**

### Descripción

---

Devuelve los datos del archivo Zip, en general, no necesitará este método a menos que desee hacer algo único con los datos

### Valores devueltos

---

Contenido del archivo Zip

### Ejemplo

---

```
$nom = 'la_vida_del_buscón.txt';
$datos = '... nunca mejora su estado quien muda solamente de lugar, y no de vida y costumbres.';

$this->zip->add_data($nom, $datos);

$zip_file = $this->zip->get_zip();
```

## read\_dir()

**bool read\_dir(string \$path [, bool \$preserve\_filepath = TRUE [, string \$root\_path = NULL]])**

### Descripción

---

Le permite comprimir un directorio, y sus contenidos, que ya existe en algún lugar de su servidor, proporcione una ruta al directorio y la clase zip leerá recursivamente y lo volverá a crear como un archivo Zip, todos los archivos contenidos dentro de la ruta de acceso suministrada serán codificados, al igual que cualquier subdirectorio que contenga

### Parámetros

---

#### path

Ruta al directorio

#### preserve\_filepath

Si se debe mantener la ruta original

#### root\_path

Parte de la ruta para excluir del directorio de archivos

### Valores devueltos

---

TRUE en el éxito, FALSE si falla

### Ejemplo

---

```
$path = '/path/a/su/directorio/';
$this->zip->read_dir($path);

// Descarga el archivo a su escritorio. lo llama "mi_copia.zip"
$this->zip->download('mi_copia.zip');
```

Por defecto, el archivo Zip colocará todos los directorios listados en el primer parámetro dentro del zip, si desea ignorar el árbol que precede al directorio de destino, puede pasar FALSE en el segundo parámetro:

```
$path = '/path/a/su/directorio/';

$this->zip->read_dir($path, FALSE);
```

Esto creará un ZIP con un directorio llamado **"directorio"** , con todos sus subdirectorios almacenados, pero no incluirá **/path/a/su/** como parte de la ruta

## read\_file()

**bool read\_file(string \$path [, mixed \$archive\_filepath = FALSE])**

### Descripción

---

Le permite comprimir un archivo que ya existe en algún lugar de su servidor, proporcione una ruta de archivo y la clase zip lo leerá y lo agregará al archivo

#### Parámetros

---

##### **path**

Ruta al archivo

##### **archive\_filepath**

Nuevo nombre de archivo/ruta (string) o (booleano) si se debe mantener la ruta de archivo original

#### Valores devueltos

---

TRUE en el éxito, FALSE si falla

#### Ejemplo

---

```
$path = '/path/a/foto.jpg';

$this->zip->read_file($path);

//Descarga el archivo a su escritorio. lo llama "mi_copia.zip"
$this->zip->download('mi_copia.zip');
```

Si desea que el archivo Zip mantenga la estructura de directorio, pase TRUE (booleano) en el segundo parámetro:

```
$path = '/path/a/foto.jpg';

$this->zip->read_file($path, TRUE);

//Descarga el archivo a su escritorio. lo llama "mi_copia.zip"
$this->zip->download('mi_copia.zip');
```

En el ejemplo anterior, **foto.jpg** se colocará en el directorio **/path/a/**

También puede especificar un nuevo nombre (ruta incluida) para el archivo agregado sobre la marcha:

```
$path = '/path/a/foto.jpg';
$nuevo_path = '/nuevo/path/alguna_foyo.jpg';

$this->zip->read_file($path, $nuevo_path);

// Descargar el archivo ZIP que contiene /nuevo/path/alguna_foyo.jpg
$this->zip->download('mi_archivo.zip');
```

## class CI\_DB\_result

---

class CI\_DB\_result

**class CI\_DB\_result**

custom\_result\_object()

**array custom\_result\_object(string \$class\_name)**

### Descripción

---

Devuelve los resultados de la consulta como un array de filas, donde cada fila es una instancia de la clase especificada

### Parámetros

---

**class\_name**

Nombre de clase para las filas resultantes

### Valores devueltos

---

Array que contiene las filas recuperadas

custom\_row\_object()

**object custom\_row\_object(int \$n, string \$type)**

### Descripción

---

Devuelve la fila de resultados solicitados como una instancia de la clase solicitada

### Parámetros

---

**n**

Índice de la fila de resultados a devolver

**type**

Nombre de la clase para la fila resultante

### Valores devueltos

---

La fila solicitada o NULL si no existe

data\_seek()

**bool data\_seek([ int \$n = 0])**

### Descripción

---

Mueve el puntero de la fila de resultados internos al desplazamiento deseado

### Parámetros

---

**n**

Índice de la fila de resultados que se devolverá a continuación

#### Valores devueltos

---

TRUE en el éxito, FALSE si falla

#### field\_data()

### array field\_data()

#### Descripción

---

Genera un array de objetos **stdClass** que contienen metadatos de campo

#### Valores devueltos

---

Array que contiene metadatos de campo

#### first\_row()

### mixed first\_row([ string \$type = 'object'])

#### Descripción

---

Devuelve la primera fila del conjunto de resultados

#### Parámetros

---

#### type

Tipo del resultado solicitado array, objeto o nombre de clase

#### Valores devueltos

---

Primera fila del conjunto de resultados, o NULL si no existe

#### free\_result()

### void free\_result()

#### Descripción

---

Libera un conjunto de resultados

Uso: ver **Métodos de ayuda de resultados**

#### valores devueltos

---

void

#### last\_row()

### mixed last\_row([ string \$type = 'object'])

#### Descripción

---

Devuelve la última fila del conjunto de resultados

#### Parámetros

---

#### type

Tipo del resultado solicitado array, objeto o nombre de clase



## Valores devueltos

---

Última fila del conjunto de resultados, o NULL si no existe

`list_fields()`

**array** `list_fields()`

## Descripción

---

Devuelve una array que contiene los nombres de campo en el conjunto de resultados

## Valores devueltos

---

Array de nombres de columna

`next_row()`

**mixed** `next_row([ string $type = 'object'])`

## Descripción

---

Devuelve la siguiente fila del conjunto de resultados

## Parámetros

---

### type

Tipo del resultado solicitado array, objeto o nombre de clase

## Valores devueltos

---

Fila siguiente del conjunto de resultados, o NULL si no existe

`num_fields()`

**int** `num_fields()`

## Descripción

---

Devuelve el número de campos en el conjunto de resultados

Uso: ver **Métodos de ayuda de resultados**

## Valores devueltos

---

Número de campos en el conjunto de resultados

`num_rows()`

**int** `num_rows()`

## Descripción

---

Devuelve el número de filas en el conjunto de resultados

Uso: ver **Métodos de ayuda de resultados**

## Valores devueltos

---

Número de filas en el conjunto de resultados

`previous_row()`

`mixed previous_row([ string $type = 'object'])`

---

#### Descripción

Devuelve la fila anterior del conjunto de resultados

---

#### Parámetros

#### type

Tipo del resultado solicitado array, objeto o nombre de clase

---

#### Valores devueltos

Fila anterior del conjunto de resultados, o NULL si no existe

`result()`

`array result([ string $type = 'object'])`

---

#### Descripción

Un contenedor para los métodos `result_array()`, `result_object()` y `custom_result_object()`

Uso: ver **Arrays de resultados**

---

#### Parámetros

#### type

Tipo de resultados solicitados array, objeto o nombre de clase

---

#### Valores devueltos

Array que contiene las filas recuperadas

`result_array()`

`array result_array()`

---

#### Descripción

Devuelve los resultados de la consulta como un array de filas, donde cada fila es en sí misma un array asociativo

Uso: ver **Arrays de resultados**

---

#### Valores devueltos

Array que contiene las filas recuperadas

`result_object()`

`array result_object()`

---

#### Descripción

Devuelve los resultados de la consulta como un array de filas, donde cada fila es un objeto de tipo **stdClass**

Uso: ver **Arrays de resultados**

---

#### Valores devueltos

Array que contiene las filas recuperadas

**row()**

**mixed row**([ **int \$n = 0** [, **string \$type = 'object'**]])

---

#### Descripción

Un contenedor para los métodos **row\_array()**, **row\_object()** y **custom\_row\_object()**

Uso: ver **Filas de resultados**

---

#### Parámetros

**n**

Índice de la fila de resultados de la consulta que se devolverá

**type**

Tipo del resultado solicitado array, objeto o nombre de clase

---

#### Valores devueltos

La fila solicitada o NULL si no existe

**row\_array()**

**array row\_array**([ **int \$n = 0**])

---

#### Descripción

Devuelve la fila de resultado solicitada como un array asociativo

Uso: ver **Filas de resultados**

---

#### Parámetros

**n**

Índice de la fila de resultados de la consulta que se devolverá

---

#### Valores devueltos

La fila solicitada o NULL si no existe

**row\_object()**

**object row\_object**([ **int \$n = 0**])

---

#### Descripción

Devuelve la fila de resultado solicitada como un objeto de tipo **stdClass**

Uso: ver **Filas de resultados**

---

#### Parámetros

**n**

Índice de la fila de resultados de la consulta que se devolverá

---

#### Valores devueltos

La fila solicitada o NULL si no existe

`set_row()`

`void set_row(mixed $key [, mixed $value = NULL])`

---

#### Descripción

Asigna un valor a una columna en particular

---

#### Parámetros

##### **key**

Nombre de la columna o array de pares clave/valor

##### **value**

Valor para asignar a la columna, **\$key** es un solo nombre de campo

---

#### valores devueltos

void

`unbuffered_row()`

`mixed unbuffered_row([ string $type = 'object'])`

---

#### Descripción

Obtiene la siguiente fila de resultados y la devuelve en la forma solicitada

Uso: ver **Filas de resultados**

---

#### Parámetros

##### **type**

Tipo del resultado solicitado array, objeto o nombre de clase

---

#### Valores devueltos

Fila siguiente del conjunto de resultados o NULL si no existe

---

## class CI\_DB\_query\_builder

---

`class CI_DB_query_builder`

`class CI_DB_query_builder`

`count_all_results()`

`int count_all_results([ string $table = " ", bool $reset = TRUE])`

---

#### Descripción

Genera un string de consulta específica de la plataforma que cuenta todos los registros devueltos por una consulta del generador de consultas

---

#### Parámetros

##### **table**

Nombre de la tabla

## reset

Si restablecer valores para **SELECT**

### Valores devueltos

---

Número de filas en el resultado de la consulta

## dbprefix()

```
string dbprefix([ string $table = '' ])
```

### Descripción

---

Antepone un prefijo de Base de datos, si existe uno en la configuración

### Parámetros

---

#### table

El nombre de la tabla, a la que se antepone el prefijo

### Valores devueltos

---

El nombre de la tabla prefijada

## delete()

```
mixed delete([ mixed $table = '' [, string $where = '' [, int $limit = NULL [, bool $reset_data = TRUE]]]])
```

### Descripción

---

Compila y ejecuta una consulta **DELETE**

### Parámetros

---

#### table

La tabla(s) para borrar; string o array

#### where

La cláusula **WHERE**

#### limit

La cláusula **LIMIT**

#### reset\_data

TRUE para restablecer la cláusula "**write**" de la consulta

### Valores devueltos

---

Instancia `CI_DB_query_builder` (method chaining) o FALSE en caso de error

## distinct()

```
object distinct([ bool $val = TRUE ])
```

### Descripción

---

Agrega una cláusula **DISTINCT** a la parte **SELECT** de la consulta

### Parámetros

---

## val

Valor deseado de la bandera "**DISTINCT**"

### Valores devueltos

---

Instancia CI\_DB\_query\_builder (method chaining)

## empty\_table()

**bool** empty\_table([ string \$table = '' ])

### Descripción

---

Elimina todos los registros de una tabla a través de una declaración **DELETE**

### Parámetros

---

#### table

Nombre de la tabla

### Valores devueltos

---

TRUE en el éxito, FALSE en la falla

## flush\_cache()

**object** flush\_cache()

### Descripción

---

Vacía el caché del generador de consultas

### Valores devueltos

---

Instancia de CI\_DB\_query\_builder (method chaining)

## from()

**object** from(mixed \$from)

### Descripción

---

Especifica la cláusula **FROM** de una consulta

### Parámetros

---

#### from

Nombre(s) de la tabla; string o array

### Valores devueltos

---

Instancia CI\_DB\_query\_builder (method chaining)

## get()

**object** get([ string \$table = '' [, int \$limit = NULL [, int \$offset = NULL]])

## Descripción

---

Compila y ejecuta una instrucción **SELECT** basada en los métodos ya llamados de **Query Builder**

## Parámetros

---

### table

La tabla para consultar

### limit

La cláusula **LIMIT**

### offset

La cláusula **OFFSET**

## Valores devueltos

---

Instancia CI\_DB\_result (method chaining)

## get\_compiled\_delete()

```
string get_compiled_delete([ string $table = '' [, bool $reset = TRUE]])
```

## Descripción

---

Compila una declaración **DELETE** y la devuelve como un string

## Parámetros

---

### table

Nombre de la tabla

### reset

Si se restablecen los valores de QB actuales o no

## Valores devueltos

---

La instrucción SQL compilada como una string

## get\_compiled\_insert()

```
string get_compiled_insert([ string $table = '' [, bool $reset = TRUE]])
```

## Descripción

---

Compila una instrucción **INSERT** y la devuelve como un string

## Parámetros

---

### table

Nombre de la tabla

### reset

Si se restablecen los valores de QB actuales o no

## Valores devueltos

---

La instrucción SQL compilada como una string

`get_compiled_select()`

**string** `get_compiled_select`([ **string** `$table` = " [, **bool** `$reset` = **TRUE**]])

---

#### Descripción

Compila una instrucción **SELECT** y la devuelve como un string

---

#### Parámetros

##### **table**

Nombre de la tabla

##### **reset**

Si se restablecen los valores de QB actuales o no

---

#### Valores devueltos

La instrucción SQL compilada como una string

`get_compiled_update()`

**string** `get_compiled_update`([ **string** `$table` = " [, **bool** `$reset` = **TRUE**]])

---

#### Descripción

Compila una instrucción **UPDATE** y la devuelve como un string

---

#### Parámetros

##### **table**

Nombre de la tabla

##### **reset**

Si se restablecen los valores de QB actuales o no

---

#### Valores devueltos

La instrucción SQL compilada como una string

`get_where()`

**object** `get_where`([ **mixed** `$table` = " [, **string** `$where` = **NULL** [, **int** `$limit` = **NULL** [, **int** `$offset` = **NULL**]])

---

#### Descripción

Igual que `get()`, pero también permite que **WHERE** se agregue directamente

---

#### Parámetros

##### **table**

La (s) tabla (s) de donde obtener datos; string o array

##### **where**

La cláusula **WHERE**

##### **limit**

La cláusula **LIMIT**



## offset

La cláusula **OFFSET**

### Valores devueltos

---

Instancia CI\_DB\_result (method chaining)

## group\_by()

**object group\_by**(mixed \$by [, bool \$escape = NULL])

### Descripción

---

Agrega una cláusula **GROUP BY** a una consulta

### Parámetros

---

#### by

Campo (s) para agrupar; string o array

#### escape

Si se deben escapar valores e identificadores

### Valores devueltos

---

Instancia CI\_DB\_query\_builder (method chaining)

## group\_end()

**object group\_end**()

### Descripción

---

Termina una expresión de grupo

### Valores devueltos

---

Instancia DB\_query\_builder

## group\_start()

**object group\_start**()

### Descripción

---

Inicia una expresión de grupo, usando **AND** para las condiciones dentro de él

### Valores devueltos

---

Instancia de CI\_DB\_query\_builder (method chaining)

## having()

**object having**(mixed \$key [, string \$value = NULL [, string \$escape = NULL]])

### Descripción

---

Agrega una cláusula **HAVING** a una consulta, separando múltiples llamadas con **AND**

---

#### Parámetros

---

##### key

Identificador (string) o array asociativo de pares campo / valor

##### value

Valor buscado si **\$key** es un identificador

##### escape

Si se deben escapar valores e identificadores

---

#### Valores devueltos

---

Instancia CI\_DB\_query\_builder (method chaining)

#### insert()

```
bool insert([ string $table = '' [, array $set = NULL [, bool $escape = NULL]])
```

---

#### Descripción

---

Compila y ejecuta una instrucción **INSERT**

---

#### Parámetros

---

##### table

Nombre de la tabla

##### set

Una array asociativa de pares campo / valor

##### escape

Si se deben escapar valores e identificadores

---

#### Valores devueltos

---

TRUE en el éxito, FALSE en la falla

#### insert\_batch()

```
mixed insert_batch(string $table [, array $set = NULL [, bool $escape = NULL [, int $batch_size = 100]])
```

---

#### Descripción

---

Compila y ejecuta las instrucciones **INSERT** por lotes

---

#### Parámetros

---

##### table

Nombre de la tabla

##### set

Datos para insertar

##### escape

Si se deben escapar valores e identificadores

##### batch\_size

Recuento de filas para insertar a la vez

## Valores devueltos

---

Número de filas insertadas o FALSE en caso de error

### **Nota**

Cuando se proporcionan **\$batch\_size**, se ejecutarán varias consultas **INSERT**, cada una de las cuales intenta insertar filas hasta **\$batch\_size**

## join()

**object join(string \$table, string \$cond [, string \$type = " [, bool \$escape = NULL]])**

### Descripción

---

Agrega una cláusula **JOIN** a una consulta

### Parámetros

---

#### **table**

Nombre de tabla para unirse

#### **cond**

La condición **JOIN ON**

#### **type**

El tipo JOIN

#### **escape**

Si se deben escapar valores e identificadores

### Valores devueltos

---

Instancia CI\_DB\_query\_builder (method chaining)

## like()

**object like(string \$field [, string \$match = " [, string \$side = 'both' [, bool \$escape = NULL]])**

### Descripción

---

Agrega una cláusula **LIKE** a una consulta, separando múltiples llamadas con **AND**

### Parámetros

---

#### **field**

Nombre del campo

#### **match**

Parte del texto para hacer coincidir

#### **side**

En qué lado de la expresión colocar el comodín '%'

#### **escape**

Si se deben escapar valores e identificadores

### Valores devueltos

---

Instancia CI\_DB\_query\_builder (method chaining)

**limit()**

**object limit(int \$value [, int \$offset = 0])**

**Descripción**

---

Agrega cláusulas **LIMIT** y **OFFSET** a una consulta

**Parámetros**

---

**value**

Número de filas para limitar los resultados

**offset**

Número de filas para saltar

**Valores devueltos**

---

Instancia CI\_DB\_query\_builder (method chaining)

**not\_group\_start()**

**object not\_group\_start()**

**Descripción**

---

Inicia una expresión grupal, usando **AND NOT** para las condiciones dentro de ella

**Valores devueltos**

---

Instancia de CI\_DB\_query\_builder (method chaining)

**not\_like()**

**object not\_like(string \$field [, string \$match = '' [, string \$side = 'both' [, bool \$escape = NULL ]]])**

**Descripción**

---

Agrega una cláusula **NOT LIKE** a una consulta, separando múltiples llamadas con **AND**

**Parámetros**

---

**field**

Nombre del campo

**match**

Parte del texto para hacer coincidir

**side**

En qué lado de la expresión colocar el comodín '%'

**escape**

Si se deben escapar valores e identificadores

**Valores devueltos**

---

Instancia CI\_DB\_query\_builder (method chaining)

**offset()**

**object offset(int \$offset)**

**Descripción**

---

Agrega una cláusula **OFFSET** a una consulta

**Parámetros**

---

**offset**

Número de filas para saltar

**Valores devueltos**

---

Instancia CI\_DB\_query\_builder (method chaining)

**order\_by()**

**object order\_by(string \$orderby [, string \$direction = " [, bool \$escape = NULL]])**

**Descripción**

---

Agrega una cláusula **ORDER BY** a una consulta

**Parámetros**

---

**orderby**

Campo para ordenar

**direction**

La orden solicitada **ASC**, **DESC** o **RANDOM**

**escape**

Si se deben escapar valores e identificadores

**Valores devueltos**

---

Instancia CI\_DB\_query\_builder (method chaining)

**or\_group\_start()**

**object or\_group\_start()**

**Descripción**

---

Inicia una expresión de grupo, utilizando **OR** para las condiciones dentro de él

**Valores devueltos**

---

Instancia de CI\_DB\_query\_builder (method chaining)

**or\_having()**

**object or\_having(mixed \$key [, string \$value = NULL [, bool \$escape = NULL]])**

**Descripción**

---

Agrega una cláusula **HAVING** a una consulta, separando múltiples llamadas con **OR**

## Parámetros

---

### key

Identificador (string) o array asociativo de pares campo / valor

### value

Valor buscado si **\$key** es un identificador

### escape

Si se deben escapar valores e identificadores

## Valores devueltos

---

Instancia CI\_DB\_query\_builder (method chaining)

### or\_like()

```
object or_like(string $field [, string $match = '' [, string $side = 'both' [, bool $escape = NULL] ]])
```

## Descripción

---

Agrega una cláusula **LIKE** a una consulta, separando clases múltiples con **OR**

## Parámetros

---

### field

Nombre del campo

### match

Parte del texto para hacer coincidir

### side

En qué lado de la expresión colocar el comodín '%'

### escape

Si se deben escapar valores e identificadores

## Valores devueltos

---

Instancia CI\_DB\_query\_builder (method chaining)

### or\_not\_group\_start()

```
object or_not_group_start()
```

## Descripción

---

Inicia una expresión de grupo, usando **NOT** para las condiciones dentro de él

## Valores devueltos

---

Instancia de CI\_DB\_query\_builder (method chaining)

### or\_not\_like()

```
object or_not_like(string $field [, string $match = '' [, string $side = 'both' [, bool $escape = NULL]])
```

## Descripción

---

Agrega una cláusula **NOT LIKE** a una consulta, separando múltiples llamadas con **OR**

---

#### Parámetros

##### field

Nombre del campo

##### match

Parte del texto para hacer coincidir

##### side

En qué lado de la expresión colocar el comodín '%'

##### escape

Si se deben escapar valores e identificadores

---

#### Valores devueltos

Instancia CI\_DB\_query\_builder (method chaining)

#### or\_where()

```
object or_where(mixed $key [, mixed $value = NULL [, bool $escape = NULL]])
```

---

#### Descripción

Genera la parte **WHERE** de la consulta, separa llamadas múltiples con **OR**

---

#### Parámetros

##### key

Nombre del campo a comparar, o array asociativo

##### value

Si es una sola clave, en comparación con este valor

##### escape

Si se deben escapar valores e identificadores

---

#### Valores devueltos

Instancia DB\_query\_builder

#### or\_where\_in()

```
object or_where_in([ string $key = NULL [, array $values = NULL [, bool $escape = NULL]])
```

---

#### Descripción

Genera **WHERE campo IN('item','item')** enlazada con **OR** si corresponde

---

#### Parámetros

##### key

El campo para buscar

##### values

Los valores buscados

##### escape

Si se deben escapar valores e identificadores

---

#### Valores devueltos

## or\_where\_not\_in()

**object or\_where\_not\_in**([string \$key = NULL [, array \$values = NULL [, bool \$escape = NULL ]]])

### Descripción

---

Genera un **WHERE campo NOT IN('item', 'item')** en la consulta unida con **OR** si corresponde

### Parámetros

---

#### key

El campo para buscar

#### values

Los valores buscados

#### escape

Si se deben escapar valores e identificadores

### Valores devueltos

---

Instancia DB\_query\_builder

## replace()

**bool replace**([string \$table = '' [, array \$set = NULL]])

### Descripción

---

Compila y ejecuta una instrucción **REPLACE**

### Parámetros

---

#### table

Nombre de la tabla

#### set

Un array asociativa de pares campo / valor

### Valores devueltos

---

TRUE en el éxito, FALSE en la falla

## reset\_query()

**object reset\_query**()

### Descripción

---

Restablece el estado actual del generador de consultas, útil cuando se quiere construir una consulta que se puede cancelar bajo ciertas condiciones

### Valores devueltos

---

Instancia de CI\_DB\_query\_builder (method chaining)

## select()

**object select**([string \$select = '\*' [, bool \$escape = NULL]])



## Descripción

---

Agrega una cláusula **SELECT** a una consulta

## Parámetros

---

### select

La parte **SELECT** de una consulta

### escape

Si se deben escapar valores e identificadores

## Valores devueltos

---

Instancia CI\_DB\_query\_builder (method chaining)

## select\_avg()

```
object select_avg([ string $select = " ", string $alias = "" ])
```

## Descripción

---

Agrega una cláusula **SELECT AVG(campo)** a una consulta

## Parámetros

---

### select

Campo para calcular el promedio de

### alias

Alias para el nombre del valor resultante

## Valores devueltos

---

Instancia CI\_DB\_query\_builder (method chaining)

## select\_max()

```
object select_max([ string $select = " ", string $alias = "" ])
```

## Descripción

---

Agrega una cláusula **SELECT MAX (campo)** a una consulta

## Parámetros

---

### select

Campo para calcular el máximo de

### alias

Alias para el nombre del valor resultante

## Valores devueltos

---

Instancia CI\_DB\_query\_builder (method chaining)

## select\_min()

```
object select_min([ string $select = " ", string $alias = "" ])
```

## Descripción

---

Agrega una cláusula **SELECT MIN (campo)** a una consulta

## Parámetros

---

### select

Campo para calcular el mínimo de

### alias

Alias para el nombre del valor resultante

## Valores devueltos

---

Instancia CI\_DB\_query\_builder (method chaining)

## select\_sum()

```
object select_sum([ string $select = " [, string $alias = "" ]])
```

## Descripción

---

Agrega una cláusula **SELECT SUM(campo)** a una consulta

## Parámetros

---

### select

Campo para calcular la suma

### alias

Alias para el nombre del valor resultante

## Valores devueltos

---

Instancia CI\_DB\_query\_builder (method chaining)

## set()

```
object set(mixed $key [, string $value = " [, bool $escape = NULL ]])
```

## Descripción

---

Agrega pares de campo / valor que se pasarán más tarde a **insert()**, **update()** o **replace()**

## Parámetros

---

### key

Nombre de campo, o un array de pares de campo / valor

### value

Valor de campo, si **\$key** es un solo campo

### escape

Si se deben escapar valores e identificadores

## Valores devueltos

---

Instancia CI\_DB\_query\_builder (method chaining)

## set\_dbprefix()

```
string set_dbprefix([ string $prefix = "" ])
```

## Descripción

---

Establece el prefijo de la Base de datos, sin tener que volver a conectar

## Parámetros

---

### prefix

El nuevo prefijo a usar

## Valores devueltos

---

El prefijo DB en uso

## set\_insert\_batch()

**object set\_insert\_batch(mixed \$key [, string \$value = " [, bool \$escape = NULL]])**

## Descripción

---

Agrega pares de campo / valor para ser insertados en una tabla más tarde a través de **insert\_batch()**

## Parámetros

---

### key

Nombre del campo o una array de pares de campo / valor

### value

Valor de campo, si **\$key** es un solo campo

### escape

Si se deben escapar valores e identificadores

## Valores devueltos

---

Instancia CI\_DB\_query\_builder (method chaining)

## set\_update\_batch()

**object set\_update\_batch(mixed \$key [, string \$value = ' ', bool \$escape = NULL]])**

## Descripción

---

Agrega pares de campo / valor que se actualizarán en una tabla más tarde mediante **update\_batch()**

## Parámetros

---

### key

Nombre del campo o una array de pares de campo / valor

### value

Valor de campo, si **\$key** es un solo campo

### escape

Si se deben escapar valores e identificadores

## Valores devueltos

---

Instancia CI\_DB\_query\_builder (method chaining)

## start\_cache()

**object start\_cache()**

## Descripción

---

Inicia el caché del generador de consultas

## Valores devueltos

---

Instancia de `CI_DB_query_builder` (method chaining)

## stop\_cache()

**object stop\_cache()**

## Descripción

---

Detiene el caché del generador de consultas

## Valores devueltos

---

Instancia de `CI_DB_query_builder` (method chaining)

## truncate()

**bool truncate([ string \$table = '' ])**

## Descripción

---

Ejecuta una declaración **TRUNCATE** en una tabla

## Parámetros

---

### table

Nombre de la tabla

## Valores devueltos

---

TRUE en el éxito, FALSE en la falla

## update()

**bool update([ string \$table = '' [, array \$set = NULL [, string \$where = NULL [, int \$limit = NULL ]]])**

## Descripción

---

Compila y ejecuta una instrucción **UPDATE**

## Parámetros

---

### table

Nombre de la tabla

### set

Un array asociativo de pares campo / valor

### where

La cláusula **WHERE**

### limit

La cláusula **LIMIT**

## Valores devueltos

---

TRUE en el éxito, FALSE en la falla

## update\_batch()

```
mixed update_batch(string $table [, array $set = NULL [, string $value = NULL [, int $batch_size = 100]])
```

### Descripción

---

Compila y ejecuta instrucciones **UPDATE** por lotes

### Parámetros

---

#### table

Nombre de la tabla

#### set

Nombre de campo, o una array asociativa de pares de campo / valor

#### value

Valor del campo, si **\$set** es un solo campo

#### batch\_size

Recuento de condiciones para agrupar en una sola consulta

### Valores devueltos

---

Número de filas actualizadas o FALSE en caso de error

### Nota

Cuando se proporcionan pares de campo / valor de más de **\$batch\_size**, se ejecutarán múltiples consultas, cada una manejando pares de campo / valor de **\$batch\_size**

## where()

```
object where(mixed $key [, mixed $value = NULL [, bool $escape = NULL]])
```

### Descripción

---

Genera la parte **WHERE** de la consulta, separa llamadas múltiples con **AND**

### Parámetros

---

#### key

Nombre del campo a comparar, o array asociativo

#### value

Si es una sola clave, en comparación con este valor

#### escape

Si se deben escapar valores e identificadores

### Valores devueltos

---

Instancia DB\_query\_builder

## where\_in()

**object** `where_in`([ **string** `$key` = **NULL** [, **array** `$values` = **NULL** [, **bool** `$escape` = **NULL**]])

### Descripción

---

Genera un **WHERE campo IN('item', 'item')** en la consulta SQL unida con **AND** si corresponde

### Parámetros

---

#### **key**

Nombre del campo a examinar

#### **values**

Array de valores objetivo

#### **escape**

Si se deben escapar valores e identificadores

### Valores devueltos

---

Instancia `DB_query_builder`

## where\_not\_in()

**object** `where_not_in`([ **string** `$key` = **NULL** [, **array** `$values` = **NULL** [, **bool** `$escape` = **NULL**]])

### Descripción

---

Genera **WHERE campo NOT IN('item', 'item')**, junto con **AND** si corresponde

### Parámetros

---

#### **key**

El campo para buscar

#### **values**

Los valores buscados

#### **escape**

Si se deben escapar valores e identificadores

### Valores devueltos

---

Instancia `DB_query_builder`

## class CI\_DB\_forge

---

### class `CI_DB_forge`

**class** `CI_DB_forge`

## add\_column()

**bool** `add_column`(**string** `$table` [, **array** `$field` = `array()` [, **string** `$_after` = **NULL**]])

## Descripción

---

Agrega una columna a una tabla, se usa para modificar una tabla existente, acepta el mismo conjunto de campos que el anterior, y se puede usar para un número ilimitado de campos adicionales

Uso: ver **Agregar una columna a una tabla**

## Parámetros

---

### table

Nombre de tabla para agregar la columna

### field

Definición de columna

### \_after

Columna para la cláusula AFTER (obsoleta)

## Valores devueltos

---

TRUE en el éxito, FALSE si falla

## add\_field()

**object add\_field(array \$field)**

## Descripción

---

Agrega un campo al conjunto que se usará para crear una tabla

Uso: ver **Agregar campos**

## Parámetros

---

### field

Definición de campo para agregar

## Valores devueltos

---

Instancia de CI\_DB\_forge (method chaining)

## add\_key()

**object add\_key(array \$key [, bool \$primary = FALSE])**

## Descripción

---

Agrega un índice al conjunto que se usará para crear una tabla

Uso: Ver **Agregar índice**

## Parámetros

---

### key

Nombre de un campo clave

### primary

Establézcalo en TRUE si debe ser una clave principal o una normal, los índices no primarios de varias columnas se deben enviar como un array

## Valores devueltos

---

Instancia de CI\_DB\_forge (method chaining)

`create_database()`

**bool** `create_database(string $db_name)`

---

#### Descripción

Crea una nueva Base de datos

Uso: vea **Crear y eliminar Bases de datos**

---

#### Parámetros

##### **db\_name**

Nombre de la Base de datos para crear

---

#### Valores devueltos

TRUE en el éxito, FALSE si falla

`create_table()`

**bool** `create_table(string $table [, string $if_not_exists = FALSE [, array $attributes = array()]] )`

---

#### Descripción

Crea una nueva tabla

Uso: ver **Crear una tabla**

---

#### Parámetros

##### **table**

Nombre de la tabla para crear

##### **if\_not\_exists**

Establézcalo en TRUE para agregar una cláusula **IF NOT EXISTS**

##### **attributes**

Un array asociativo de atributos de tabla

---

#### Valores devueltos

TRUE en el éxito, FALSE si falla

`drop_column()`

**bool** `drop_column(string $table, array $column_name)`

---

#### Descripción

Borra una columna de una tabla

Uso: vea **Borrar una columna de la tabla**

---

#### Parámetros

##### **table**

Nombre de la tabla



## column\_name

El nombre de la columna a borrar

### Valores devueltos

---

TRUE en el éxito, FALSE si falla

## drop\_database()

**bool** drop\_database(string \$db\_name)

### Descripción

---

Borra una Base de datos

Uso: vea **Crear y eliminar Bases de datos**

### Parámetros

---

#### db\_name

Nombre de la Base de datos a borrar

### Valores devueltos

---

TRUE en el éxito, FALSE si falla

## drop\_table()

**bool** drop\_table(string \$table\_name [, string \$if\_exists = FALSE])

### Descripción

---

Borra una tabla

Uso: vea **Borrar una tabla**

### Parámetros

---

#### table

Nombre de la tabla a borrar

#### if\_exists

Establézcalo en TRUE para agregar una cláusula **IF EXISTS**

### Valores devueltos

---

TRUE en el éxito, FALSE si falla

## modify\_column()

**bool** modify\_column(string \$table, array \$field)

### Descripción

---

Modifica una columna de tabla

Uso: Ver **Modificar una columna de una tabla**

## Parámetros

---

### **table**

Nombre de la tabla

### **field**

Definición de columna

## Valores devueltos

---

TRUE en el éxito, FALSE si falla

### **rename\_table()**

**bool** **rename\_table**(**string** \$table\_name, **string** \$new\_table\_name)

## Descripción

---

Cambia el nombre de una tabla

Uso: ver **Cambiar el nombre de una tabla**

## Parámetros

---

### **table**

Nombre actual de la tabla

### **new\_table\_name**

Nuevo nombre de la tabla

## Valores devueltos

---

TRUE en el éxito, FALSE si falla

## **class CI\_DB\_utility**

---

### **class CI\_DB\_utility**

**class** **CI\_DB\_utility**

### **backup()**

**string** **backup**([ **array** \$params = **array**()])

## Descripción

---

Realiza una copia de seguridad de la Base de datos, según las preferencias del usuario

## Parámetros

---

### **params**

Un array asociativo de opciones

## Valores devueltos

---

String raw/(g) zipped de la consulta SQL

## csv\_from\_result()

```
string csv_from_result(object $query [, string $delim = ',' [, string $newline = "\n" [, string $enclosure = '"']]])
```

### Descripción

---

Transforma un objeto de resultado de Base de datos en un documento CSV

### Parámetros

---

#### query

Un objeto de resultado de Base de datos

#### delim

Separador de campo

#### newline

Carácter de nueva línea

#### enclosure

Delimitador

### Valores devueltos

---

Un string con el archivo CSV generado

## database\_exists()

```
bool database_exists(string $database_name)
```

### Descripción

---

Verifica la existencia de una Base de datos

### Parámetros

---

#### database\_name

Nombre de la Base de datos

### Valores devueltos

---

TRUE si la Base de datos existe, FALSE si no

## list\_databases()

```
array list_databases()
```

### Descripción

---

Recupera una lista de todos los nombres de las Bases de datos

### Valores devueltos

---

Array de nombres de las Bases de datos encontradas

## optimize\_database()

```
array optimize_database()
```

#### Descripción

---

Optimiza la Base de datos

#### Valores devueltos

---

Array de mensajes de optimización o FALSE en caso de error

#### `optimize_table()`

**array optimize\_table(string \$table\_name)**

#### Descripción

---

Optimiza una tabla de la Base de datos

#### Parámetros

---

##### **table\_name**

Nombre de la tabla para optimizar

#### Valores devueltos

---

Array de mensajes de optimización o FALSE en caso de error

#### `repair_table()`

**array repair\_table(string \$table\_name)**

#### Descripción

---

Repara una tabla de Base de datos

#### Parámetros

---

##### **table\_name**

Nombre de la tabla para reparar

#### Valores devueltos

---

Matriz de mensajes de reparación o FALSE en caso de error

#### `xml_from_result()`

**string xml\_from\_result(object \$query [, array \$params = array()])**

#### Descripción

---

Transforma un objeto de resultado de Base de datos en un documento XML

#### Parámetros

---

##### **query**

Un objeto de resultado de Base de datos

##### **params**

Un array asociativo de preferencias

#### Valores devueltos

---

Un string con el documento XML generado

## class CI\_DB\_driver

---

class CI\_DB\_driver

**class CI\_DB\_driver**

affected\_rows()

**int affected\_rows()**

### Descripción

---

Devuelve el número de filas modificadas por la última consulta ejecutada, útil para verificar cuántas filas se crearon, actualizaron o eliminaron

### Valores devueltos

---

Número de filas afectadas

cache\_delete()

**bool cache\_delete([ string \$segment\_one = " [, string \$segment\_two = "])**

### Descripción

---

Elimine los archivos de caché asociados con un URI particular

### Parámetros

---

#### **segment\_one**

Primer segmento de URI

#### **segment\_two**

Segundo segmento de URI

### Valores devueltos

---

TRUE en el éxito, FALSE si falla

cache\_delete\_all()

**bool cache\_delete\_all()**

### Descripción

---

Eliminar todos los archivos de caché

### Valores devueltos

---

TRUE en caso de éxito, FALSE en caso de error

cache\_off()

**bool cache\_off()**

### Descripción

---

Deshabilita el almacenamiento en memoria caché de resultados

#### Valores devueltos

---

TRUE si el almacenamiento en caché está activado, FALSE si no

#### cache\_on()

**bool** cache\_on()

#### Descripción

---

Habilita el almacenamiento en memoria caché de resultados

#### Valores devueltos

---

TRUE si el almacenamiento en caché está activado, FALSE si no

#### cache\_set\_path()

**void** cache\_set\_path([ **string** \$path = '' ])

#### Descripción

---

Establece la ruta del directorio que se utilizará para el almacenamiento en caché

#### Parámetros

---

##### **path**

Ruta al directorio de caché

#### valores devueltos

---

void

#### call\_function()

**string** call\_function(**string** \$function)

#### Descripción

---

Ejecuta una función PHP nativa, utilizando un contenedor independiente de la plataforma

#### Parámetros

---

##### **function**

Nombre de la función

#### Valores devueltos

---

El resultado de la función

#### close()

**void** close()

#### Descripción

---

Cierra la conexión a la Base de datos

#### valores devueltos

---

void

**compile\_binds()**

**string compile\_binds(string \$sql, array \$binds)**

---

**Descripción**

Compila una consulta SQL con los valores de enlace pasados

---

**Parámetros**

**sql**

La instrucción SQL

**binds**

Un array de datos vinculantes

---

**Valores devueltos**

La declaración de SQL actualizada

**count\_all()**

**int count\_all([ string \$table = '' ])**

---

**Descripción**

Devuelve el número total de filas en una tabla, o 0 si no se proporcionó ninguna tabla

---

**Parámetros**

**table**

Nombre de la tabla

---

**Valores devueltos**

Recuento de filas para la tabla especificada

**db\_connect()**

**mixed db\_connect(bool \$persistent = TRUE)**

---

**Descripción**

Establece una conexión con la Base de datos

---

**Parámetros**

**persistent**

Si se debe establecer una conexión persistente o una conexión regular

---

**Valores devueltos**

Base de datos recurso / objeto o FALSE en caso de error

**Nota**

El valor devuelto depende del controlador subyacente en uso, por ejemplo, se devolverá una instancia de mysqli con el controlador 'mysqli'

**db\_pconnect()**

**mixed db\_pconnect()**

**Descripción**

---

Establece una conexión persistente con la Base de datos

**Valores devueltos**

---

Devuelve: recurso / objeto de conexión de Base de datos o FALSE en caso de error

**Nota**

Este método es solo un alias para **db\_connect(TRUE)**

**db\_select()**

**bool db\_select([ string \$database = '' ])**

**Descripción**

---

Selecciona / cambia la Base de datos actual

**Parámetros**

---

**database**

Nombre de la Base de datos

**Valores devueltos**

---

TRUE en el éxito, FALSE si falla

**db\_set\_charset()**

**bool db\_set\_charset(string \$charset)**

**Descripción**

---

Establecer conjunto de caracteres del cliente

**Parámetros**

---

**charset**

Nombre del conjunto de caracteres

**Valores devueltos**

---

TRUE en el éxito, FALSE si falla

**display\_error()**

**string display\_error([ string \$error = '' [, string \$swap = '' [, bool \$native = FALSE]])**



## Descripción

---

Muestra un mensaje de error y detiene la ejecución del script

## Parámetros

---

### **error**

El mensaje de error

### **swap**

Cualquier valor de "intercambio"

### **native**

Si se localiza el mensaje

## Valores devueltos

---

Muestra la pantalla de error de DB, el mensaje se muestra utilizando la plantilla

**application/views/errors/error\_db.php**

## elapsed\_time()

**string elapsed\_time([ int \$decimals = 6 ])**

## Descripción

---

Calcula el tiempo transcurrido de la consulta agregada

## Parámetros

---

### **decimales**

La cantidad de decimales

## Valores devueltos

---

El tiempo transcurrido de la consulta agregada, en microsegundos

## escape()

**mixed escape(mixed \$str)**

## Descripción

---

Escapa los datos de entrada en función del tipo, incluidos booleanos y NULL

## Parámetros

---

### **str**

El valor para escapar, o un array múltiple

## Valores devueltos

---

Los valores escapados

## escape\_identifiers()

**mixed escape\_identifiers(mixed \$item)**

## Descripción

---

Escape de identificadores de SQL, como columna, tabla y nombres

#### Parámetros

---

##### item

El elemento o conjunto de elementos para escapar

#### Valores devueltos

---

El (los) elemento(s) de entrada escapados

#### escape\_like\_str()

**mixed** escape\_like\_str(**mixed** \$str)

#### Descripción

---

Escape LIKE strings

#### Parámetros

---

##### str

Un valor de string o array múltiple

#### Valores devueltos

---

Lo(s) string(s) escapado(s)

Similar a **escape\_str()**, pero también escapará a los caracteres comodín % y \_, para que no causen falsos positivos en condiciones **LIKE**

#### **Importante**

El método **escape\_like\_str()** usa '!' (Signo de exclamación) para escapar los caracteres especiales para las condiciones **LIKE**, debido a que este método escapa strings parciales que envolvería entre comillas usted mismo, no puede agregar automáticamente la condición **ESCAPE '!'** por usted, por lo que tendrá que hacerlo manualmente

#### escape\_str()

**mixed** escape\_str(**mixed** \$str [, **bool** \$like = **FALSE**])

#### Descripción

---

Escapa de los valores de string

#### Parámetros

---

##### str

Un valor de string o array múltiple

##### like

Si la string se usará o no en una condición LIKE

#### Valores devueltos

---

Lo(s) string(s) escapado(s)

#### **Advertencia**

Los strings devueltos **NO incluyen comillas** a su alrededor

`field_data()`

**array field\_data(string \$table)**

**Descripción**

---

Obtiene una lista que contiene datos de campo sobre una tabla

**Parámetros**

---

**table**

El nombre de la tabla

**Valores devueltos**

---

Matriz de elementos de datos de campo o FALSE en caso de error

`field_exists()`

**bool field\_exists(string \$field\_name, string \$table\_name)**

**Descripción**

---

Determina si existe un campo en particular

**Parámetros**

---

**table\_name**

El nombre de la tabla

**field\_name**

El nombre del campo

**Valores devueltos**

---

TRUE si ese campo existe en esa tabla, FALSE si no

`initialize()`

**bool initialize()**

**Descripción**

---

Inicializa la configuración de la Base de datos, establece una conexión con la Base de datos

**Valores devueltos**

---

TRUE en caso de éxito, FALSE en caso de error

`insert_string()`

**string insert\_string(string \$table, array \$data)**

**Descripción**

---

Genera un string de instrucción **INSERT**

**Parámetros**

---

## table

La tabla objetivo

## data

Un array asociativo de pares clave / valor

## Valores devueltos

---

La instrucción SQL INSERT, como un string

## is\_write\_type()

**bool** `is_write_type(string $sql)`

## Descripción

---

Determina si una consulta es de tipo "**escritura**" (como **INSERT, UPDATE, DELETE**) o "**lectura**" (es decir, **SELECT**)

## Parámetros

---

### sql

La instrucción SQL

## Valores devueltos

---

TRUE si la declaración SQL es de "**tipo de escritura**", FALSE si no

## last\_query()

**string** `last_query()`

## Descripción

---

Devuelve la última consulta que se ejecutó

## Valores devueltos

---

La última consulta ejecutada

## list\_fields()

**array** `list_fields(string $table)`

## Descripción

---

Obtiene una lista de los nombres de campo en una tabla

## Parámetros

---

### table

El nombre de la tabla

## Valores devueltos

---

Array de nombres de campo o FALSE en caso de error

## list\_tables()

**array** `list_tables([ bool $constrain_by_prefix = FALSE ])`

## Descripción

---

Obtiene una lista de las tablas en la Base de datos actual

#### Parámetros

---

#### **constrain\_by\_prefix**

TRUE para unir los nombres de las tablas con el **dbprefix** configurado

#### Valores devueltos

---

Array de nombres de tabla o FALSE en caso de error

#### **platform()**

#### **string platform()**

#### Descripción

---

El nombre de la plataforma en uso (mysql, mssql, etc ...)

#### Valores devueltos

---

Nombre de la plataforma

#### **primary()**

#### **string primary(string \$table)**

#### Descripción

---

Recupera la clave principal de una tabla

#### Parámetros

---

#### **table**

Nombre de la tabla

#### Valores devueltos

---

El nombre de la clave principal, FALSE si no hay

#### **Nota**

Si la plataforma de la Base de datos no es compatible con la detección de clave primaria, se puede suponer que el nombre de la primera columna es la clave principal

#### **protect\_identifiers()**

#### **string protect\_identifiers(string \$item [, bool \$prefix\_single = FALSE [, bool protect\_identifiers = NULL [, bool \$field\_exists = TRUE]]])**

#### Descripción

---

Toma un nombre de columna o tabla, opcionalmente con un alias, y aplica el **dbprefix** configurado

#### Parámetros

---

#### **item**

El ítem para trabajar con

**prefix\_single**

Si se aplica el **dbprefix** incluso si el elemento de entrada es un identificador único

**protect\_identifiers**

Si se deben citar identificadores

**field\_exists**

Si el elemento suministrado contiene un nombre de campo o no

**Valores devueltos**

---

El item modificado

Es necesaria cierta lógica para tratar los nombres de las columnas que incluyen la ruta

Considere una consulta como esta:

```
SELECT * FROM hostname.database.table.column AS c FROM hostname.database.table
```

O una consulta con aliasing:

```
SELECT m.member_id, m.member_name FROM members AS m
```

Dado que el nombre de la columna puede incluir hasta cuatro segmentos, host, DB, table, column, o también tiene un prefijo de alias, necesitamos hacer un poco de trabajo para resolverlo e insertar el prefijo de la tabla, si existe, en el posición correcta, y escape solo los identificadores correctos

Este método es utilizado ampliamente por la clase **Query Builder**

## query()

**mixed query(string \$sql [, array \$binds = FALSE [, bool \$return\_object = NULL]])**

### Descripción

---

Ejecuta una consulta SQL

### Parámetros

---

#### sql

La instrucción SQL para ejecutar

#### binds

Un array de datos vinculantes

#### return\_object

Si se devuelve un objeto resultado o no

### Valores devueltos

---

TRUE luego de la ejecución exitosa de una consulta de "**tipo escritura**", FALSE al fallar, un objeto CI\_DB\_result (method chaining) para consultas de "**tipo lectura**", acepta un string SQL como entrada y devuelve un objeto resultante al ejecutar con éxito una consulta de tipo "**lectura**"

## reconnect()

**bool reconnect()**

### Descripción

---

Mantiene / restablece la conexión a la Base de datos si no se han enviado consultas durante un período de tiempo que exceda el tiempo de espera inactivo del servidor

### Valores devueltos

---

TRUE en caso de éxito, FALSE en caso de error

## simple\_query()

**mixed simple\_query(string \$sql)**

### Descripción

---

Una versión simplificada del método **query()**, apropiado para usar cuando no necesita obtener un objeto de resultado o simplemente enviar una consulta a la Base de datos y no preocuparse por el resultado

### Parámetros

---

#### sql

La instrucción SQL para ejecutar

### Valores devueltos

---

Cualquiera que sea la función de "**consulta**" del controlador subyacente

## table\_exists()

**bool table\_exists(string \$table\_name)**

### Descripción

---

Determina si existe una tabla en particular

### Parámetros

---

#### **table\_name**

El nombre de la tabla

### Valores devueltos

---

TRUE si esa tabla existe, FALSE si no

### total\_queries()

#### **int total\_queries()**

### Descripción

---

Devuelve el número total de consultas que se han ejecutado hasta el momento

### Valores devueltos

---

El número total de consultas ejecutadas

### trans\_complete()

#### **bool trans\_complete()**

### Descripción

---

Completa la transacción

### Valores devueltos

---

TRUE en caso de éxito, FALSE en caso de error

### trans\_off()

#### **void trans\_off()**

### Descripción

---

Desactiva las transacciones en tiempo de ejecución

### valores devueltos

---

void

### trans\_start()

#### **bool trans\_start([ bool \$test\_mode = FALSE ])**

### Descripción

---

Comienza una transacción

### Parámetros

---

#### **test\_mode**



Indicador de modo de prueba

---

#### Valores devueltos

---

TRUE en el éxito, FALSE si falla

**trans\_status()**

**bool trans\_status()**

---

#### Descripción

---

Le permite recuperar el indicador de estado de la transacción para determinar si ha fallado

---

#### Valores devueltos

---

TRUE si la transacción fue exitosa, FALSE si falló

**trans\_strict()**

**void trans\_strict([ bool \$mode = TRUE ])**

---

#### Descripción

---

Activar / desactivar el modo "**estricto**" de transacción, cuando el modo estricto está habilitado, si está ejecutando varios grupos de transacciones y un grupo falla, todos los grupos subsiguientes se revertirán, si el modo estricto está deshabilitado, cada grupo se trata de forma autónoma, lo que significa que un fallo de un grupo no afectará a ninguna otra

---

#### Parámetros

---

##### **mode**

Indicador de modo estricto

---

#### valores devueltos

---

void

**update\_string()**

**string update\_string(string \$table, array \$data, mixed \$where)**

---

#### Descripción

---

Genera un string de instrucción **UPDATE**

---

#### Parámetros

---

##### **table**

La tabla objetivo

##### **data**

Una array asociativo de pares clave / valor

##### **where**

Las condiciones de la declaración **WHERE**

---

#### Valores devueltos

---

La instrucción SQL **UPDATE**, como un string

**version()**

**string version()**

---

**Descripción**

Número de versión de la Base de datos

---

**Valores devueltos**

La versión de la Base de datos que se está utilizando

---

## Array Helper

---

**element()**

**mixed element(string \$item, array \$array [, bool \$default = NULL])**

---

**Descripción**

Le permite buscar un elemento de un array, la función prueba si el índice del array está establecido y si tiene el valor

---

**Parámetros**

**item**

Elemento a recuperar del array

**array**

Array de entrada

**default**

Qué devolver si el array no es válido

---

**Valores devueltos**

Si existe el valor, se devuelve, si no existe devuelve NULL o lo que se haya especificado como predeterminado en el tercer parámetro

---

**Ejemplo**

```
$array = array(
    'color' => 'rojo',
    'forma' => 'octogono',
    'peso' => ''
);

echo element('color', $array); // presenta: "rojo"
echo element('peso', $array, 'nolose'); // presenta: "nolose"
```

**elements()**

**mixed elements(string \$items, array \$array [, bool \$default = NULL])**

---

**Descripción**

---

Le permite obtener una cantidad de elementos de un array, la función prueba si cada uno de los índices del array está establecido

## Parámetros

---

### item

Elemento a recuperar del array

### array

Array de entrada

### default

Qué devolver si el array no es válido

## Valores devueltos

---

Si existe el valor, se devuelve, si no existe devuelve NULL o lo que se haya especificado como predeterminado en el tercer parámetro

## Ejemplo

---

```
$array = array(
    'forma'    => 'octogono',
    'color'    => 'rojo',
    'radio'    => '10',
    'diámetro' => '20'
);
$mis_figuras = elements(array('forma','color','altura'), $array);
```

Devolverá el siguiente array:

```
Array(
    'forma' => 'octogono',
    'color' => 'rojo',
    'altura' => NULL
);
```

Puede establecer el tercer parámetro a cualquier valor predeterminado que desee :

```
$mis_figuras = elements(array ('forma','color','altura'), $array, 'nolose');
```

Devolverá:

```
Array(
    'forma' => 'octogono',
    'color' => 'rojo',
    'altura' => 'nolose'
);
```

Es útil al enviar el array **\$\_POST** a uno de sus Modelos, evita que los usuarios envíen datos **POST** adicionales para ser ingresados en sus tablas :

```
$this->load->model('post_model');

$this->post_model->update(
    elements(array('id', 'título', 'contenido'), $_POST)
);
```

Esto asegura que solo los campos **id**, **título** y **contenido** se envíen para actualizarse

**random\_element()**

**mixed random\_element(array \$array)**

---

#### Descripción

Toma un array como entrada y devuelve un elemento aleatorio de el

---

#### Parámetros

##### **array**

Array de entrada

---

#### Valores devueltos

Un elemento aleatorio del array

---

#### Ejemplo

```
$citas = array (
    "Una vez terminado el juego, el rey y el peón vuelven a la misma caja",
    "Cuando los ricos se hacen la guerra, son los pobres los que mueren",
    "El mundo de los olores es el mundo de los recuerdos",
    "Dejar hacer pudiendo evitar, es aprobar",
    "Actuamos en este mundo como si tuviéramos uno de repuesto",
    "La vejez es la perdida de la curiosidad"
);

echo random_element($citas);
```

---

## CAPTCHA Helper

---

**create\_captcha()**

**array create\_captcha([ array \$data = " [, string \$img\_path = " [, string \$img\_url = " [, string \$font\_path = " ]])**

---

#### Descripción

Toma una serie de información para generar un **CAPTCHA** como entrada y crea la imagen según sus especificaciones, devolviendo un array de datos asociativos sobre la imagen

---

#### Parámetros

##### **data**

Array de datos para el **CAPTCHA**

### **img\_path**

Ruta para crear la imagen (**DESACONSEJADO**, proporciónelo en el array **\$data**)

### **img\_url**

URL a la carpeta de imágenes **CAPTCHA** (**DESACONSEJADO**, proporciónelo en el array **\$data**)

### **font\_path**

Ruta del servidor a la fuente (**DESACONSEJADO**, proporciónelo en el array **\$data**)

#### Valores devueltos

---

array('word' => \$word, 'time' => \$now, 'image' => \$img)

```
array(  
    'image' => IMAGE_TAG  
    'time' => TIMESTAMP (in microtime)  
    'word' => CAPTCHA_WORD  
);
```

**image:** es la de la etiqueta:

```

```

**time** : es la micro timestamp usada como el nombre de la imagen sin la extensión de archivo, será un número como este: 1139612155.3422

**word:** Es la palabra que aparece en la imagen del captcha, que si no se proporciona a la función, será un string aleatorio

#### **Nota**

El uso de los parámetros **\$img\_path**, **\$img\_url** y **\$font\_path** está **DESACONSEJADO**, proporciónelos en el conjunto **\$data** en su lugar

## Cookie Helper

---

### **delete\_cookie()**

```
void delete_cookie(string $name [, string $domain = " [, string $path = '/' [, string $prefix =  
" ] ])
```

#### Descripción

---

Permite eliminar una cookie, a menos que haya establecido una ruta personalizada u otros valores, solo se necesita el nombre de la cookie

#### Parámetros

---

##### **name**

Nombre de la cookie

##### **domain**

Dominio de cookies (generalmente: .sudominio.com)

### path

Ruta de las cookies

### prefix

Prefijo de nombre de cookie

### valores devueltos

---

void

### Ejemplo

---

```
delete_cookie('nombre');
```

Esta función es idéntica a **set\_cookie()**, excepto que no tiene el valor y los parámetros de caducidad, puede enviar un array de valores en el primer parámetro o puede establecer parámetros discretos:

```
delete_cookie($nombre, $dominio, $ruta, $prefijo);
```

### get\_cookie()

**mixed get\_cookie(string \$index [, bool \$xss\_clean = NULL])**

#### Descripción

---

Esta función le proporciona una sintaxis más amigable para obtener cookies del navegador, consulte la **Input Library** para una descripción detallada de su uso, ya que esta función actúa de similar a **CI\_Input::cookie()**, excepto que antepone **config['cookie\_prefix']** si se encuentra configurado en **application/config/config.php**

#### Parámetros

---

##### index

Nombre de la cookie

##### xss\_clean

Si se debe aplicar el filtrado XSS al valor devuelto

#### Valores devueltos

---

El valor de la cookie o NULL si no se encuentra

### set\_cookie()

**void set\_cookie(mixed \$name [, string \$value = '' [, int \$expire = '' [, string \$domain = '' [, string \$path = '/' [, string \$prefix = '' [, bool \$secure = NULL [, bool \$httponly = NULL]]]]]])**

#### Descripción

---

Esta función auxiliar le brinda una sintaxis más amigable para configurar las cookies del navegador, consulte la **Input Library** para obtener una descripción de su uso, ya que esta función es un alias de **CI\_Input::set\_cookie()**

#### Parámetros

---

##### name

Nombre de la cookie o array asociativo de todos los parámetros disponibles para esta función

##### value

Valor de la cookie

**expire**

Número de segundos hasta la caducidad

**domain**

Dominio de cookies (generalmente: .sudominio.com)

**path**

Ruta de las cookies

**prefix**

Prefijo de nombre de cookie

**secure**

Si solo enviar la cookie a través de HTTPS

**httponly**

Si ocultar la cookie de JavaScript

**valores devueltos**

---

void

## Date Helper

---

**date\_range()**

```
array date_range([ int $unix_start = " ", int $mixed = " ", bool $is_unix = TRUE [, string $format = 'Y-m-d'] ])]
```

**Descripción**

---

Devuelve una lista de fechas dentro de un período especificado

**Parámetros**

---

**unix\_start**

Marca de tiempo UNIX de la fecha de inicio del rango

**mixed**

Marca de tiempo UNIX de la fecha de finalización del rango o intervalo en días

**is\_unix**

Establecido en FALSE si **\$mixed** no es una marca de tiempo

**format**

Formato de fecha de salida, igual que en **date()**

**Valores devueltos**

---

Una variedad de fechas

**Ejemplo**

---

```
$rango = date_range('2012-01-01', '2012-01-15');
echo "Primeros 15 días de 2012:";
foreach ($rango as $fecha)
{
    echo $fecha."\n";
}
```

**days\_in\_month()**

**int days\_in\_month**([ **int \$month** = 0[, **int \$year** = '']])

#### Descripción

---

Devuelve la cantidad de días en un mes / año determinado, tiene en cuenta si es año bisiesto

#### Parámetros

---

##### **month**

Un mes numérico

##### **year**

Un año numérico, si está vacío, se usará el año actual

#### Valores devueltos

---

Recuento de días en el mes especificado

#### Ejemplo

---

```
echo days_in_month(06, 2018);
```

#### **Nota**

Esta función es un alias de la función nativa **cal\_days\_in\_month()**, si está disponible

**gmt\_to\_local()**

**int gmt\_to\_local**([ **int \$time** = '' [, **string \$timezone** = 'UTC' [, **bool \$dst** = FALSE]])

#### Descripción

---

Toma como entrada una marca de tiempo UNIX (referenciada a GMT), y la convierte a una marca de tiempo localizada según la zona horaria y el horario de verano enviados

#### Parámetros

---

##### **time**

Indicación de fecha y hora UNIX

##### **timezone**

Zona horaria

##### **dst**

Si el DST está activo

#### Valores devueltos

---

Marca de tiempo UNIX



## Ejemplo

---

```
$timestamp = 1140153693;
$timezone = 'UM8';
$daylight_saving = TRUE;
echo gm_to_local($timestamp, $timezone, $daylight_saving);
```

## Nota

Para obtener una lista de zonas horarias, consulte la referencia en la parte inferior de esta página

## human\_to\_unix()

**int human\_to\_unix([ int \$datestr = '' ])**

### Descripción

---

Lo opuesto a la función **unix\_to\_time()**, toma un tiempo "**humano**" como entrada y lo devuelve como Unix, esta función es útil si acepta fechas formateadas como "**humanas**" enviadas desde un formulario. Devuelve FALSE (booleano) si la cadena de la fecha pasada no está formateada como se indicó anteriormente

### Parámetros

---

#### **datestr**

Cadena de fecha

### Valores devueltos

---

Marca de tiempo UNIX o FALSE en caso de error

## Ejemplo

---

```
$ahora = time();
$humano = unix_to_human($ahora);
$unix = human_to_unix($humano);
```

## local\_to\_gmt()

**int local\_to\_gmt([ time \$time = '' ])**

### Descripción

---

Toma una marca de tiempo UNIX como entrada y la devuelve como GMT

### Parámetros

---

#### **time**

Indicación de fecha y hora UNIX

### Valores devueltos

---

Marca de tiempo UNIX

## Ejemplo

---

```
$gmt = local_to_gmt(time());
```

## mdate()

**string mdate**([ **string \$datestr** = " [, **int \$time** = "])

### Descripción

---

Esta función es idéntica **date()** de PHP, excepto que le permite usar códigos de fecha de al estilo de MySQL, donde cada letra de código va precedida de un signo de porcentaje, por ejemplo: %Y %m %d etc.

### Parámetros

---

#### **datestr**

Cadena de fecha

#### **time**

Indicación de fecha y hora UNIX

### Valores devueltos

---

Fecha con formato de MySQL

El beneficio de hacer las fechas de esta forma es que no hay que preocuparse de escapar los caracteres que no son códigos de fecha, como tendría que hacerlo normalmente con la función **date()**

### Ejemplo

---

```
$fecha_string = 'Año: %Y Mes: %m Día: %d - %h:%i %a';
$time = time();
echo mdate($fecha_string, $time);
```

Si no se incluye una marca de tiempo en el segundo parámetro, se usará la fecha/hora actual

## mysql\_to\_unix()

**int mysql\_to\_unix**([ **string \$time** = "])

### Descripción

---

Toma una marca de tiempo MySQL como entrada y la devuelve como una marca de tiempo UNIX

### Parámetros

---

#### **time**

Marca de tiempo MySQL

### Valores devueltos

---

Marca de tiempo UNIX

### Ejemplo

---

```
$unix = mysql_to_unix('20061124092345');
```

## nice\_date()

### **Nota**

Esta función está **DESACONSEJADA**, use la clase nativa de PHP **DateTime** en su lugar

**now()**

**int now**([ **string \$timezone = NULL**])

#### Descripción

Devuelve la hora actual como una marca de tiempo UNIX, referenciada a la hora local de su servidor o cualquier zona horaria compatible con PHP, según el valor "**time reference**" del archivo de configuración, si no tiene la intención de establecer su referencia de hora maestra a cualquier otra zona horaria compatible con PHP (que normalmente hará si ejecuta un sitio que permite a cada usuario establecer sus propias configuraciones de zona horaria), no hay beneficio de usar esta función respecto de la función **time()** de PHP

#### Parámetros

##### **timezone**

Zona horaria, si no se proporciona una zona horaria, devolverá **time()** en función de la configuración de referencia de hora

#### Valores devueltos

Marca de tiempo UNIX

```
echo now('Australia/Victoria');
```

**standard\_date()**

#### Nota

Función **DESACONSEJADA**. Use la función nativa `date()` combinada con las constantes de formato de `DateTime`:

```
echo date(DATE_RFC822, time());
```

**timespan()**

**string timespan**([ **int \$seconds = 1** [, **string \$time = "** [, **int \$units = "**]])

#### Descripción

Formatea una marca de tiempo UNIX para que parezca similar a esto:

```
1 año, 10 meses, 2 semanas, 5 días, 10 horas, 16 minutos
```

El propósito más común para esta función es mostrar cuánto tiempo ha transcurrido desde algún punto en el tiempo en el pasado hasta ahora

#### Parámetros

##### **seconds**

Número de segundos, tiene que contener una marca de tiempo UNIX

## time

Marca de tiempo UNIX, tiene que contener una marca de tiempo que sea mayor que la primera, si está vacío, se usará la hora actual

## units

Número de unidades de tiempo para mostrar, parámetro opcional, limita el número de unidades de tiempo para mostrar

### Valores devueltos

---

Diferencia horaria formateada

### Ejemplo

---

```
$fecha_post = '1079621429';  
$ahora = time();  
$unidades = 2;  
echo timespan($fecha_post, $ahora, $unidades);
```

### Nota

El texto generado por esta función se encuentra en el siguiente archivo de idioma:

**language/<su\_idioma>/date\_lang.php**

## timezones()

**int timezones([ string \$tz = '' ])**

### Descripción

---

Toma una referencia de zona horaria (para obtener una lista de zonas horarias válidas, consulte la "**Referencia de zona horaria**" a continuación) y devuelve el número de horas de compensación UTC

### Parámetros

---

#### tz

Una zona horaria numérica

### Valores devueltos

---

Diferencia horaria de UTC

### Ejemplo

---

```
echo timezones('UM5');
```

Esta función es útil cuando se utiliza con **timezone\_menu()**

## timezone\_menu()

```
string timezone_menu([ string $default = 'UTC' [, string $class = '' [, string $name = 'timezones' [, mixed $attributes = '' ]]])
```

### Descripción

---

Genera un menú desplegable de zonas horarias, este menú es útil si ejecuta un sitio en el que los usuarios pueden establecer su valor de zona horaria local, consulte la **referencia de la zona horaria** para ver los valores de este menú

### Parámetros

---

#### default

Zona horaria , le permite establecer el estado "**selected**" del menú. Por ejemplo, para configurar la hora del Pacífico como predeterminada, hará esto:

```
echo timezone_menu('UM8');
```

#### class

Nombre de la clase CSS para el menú

#### name

Nombre del menú

#### attributes

Establecer uno o más atributos HTML en la etiqueta de selección generada

### Valores devueltos

---

Menú desplegable HTML con husos horarios

## unix\_to\_human()

```
string unix_to_human([ int $time = '' [, bool $seconds = FALSE [, string $fmt = 'us' ]]])
```

### Descripción

---

Toma una marca de tiempo UNIX como entrada y la devuelve en un formato humanamente legible con este prototipo:

```
YYYY-MM-DD HH:MM:SS AM/PM
```

Puede ser útil si necesita mostrar una fecha en un campo de formulario para enviar

### Parámetros

---

#### time

Indicación de fecha y hora UNIX

#### seconds

Si se muestran los segundos

#### fmt

Formato (us o euro)

### Valores devueltos

---

Fecha formateada

### Ejemplo

---

La hora se puede formatear con o sin segundos, y se puede configurar en formato europeo o estadounidense, si solo se envía la marca de tiempo, devolverá el tiempo sin segundos formateado para los EE. UU:

```
$now = time();
echo unix_to_human($now); // hora de EE. UU. sin segundos
echo unix_to_human($now, TRUE, 'us'); // Tiempo de EE. UU. con segundos
echo unix_to_human($now, TRUE, 'eu'); // Tiempo Europeo con segundos
```

## Referencia de zona horaria

La siguiente tabla indica cada zona horaria y su ubicación

Tenga en cuenta que algunas de las listas de ubicaciones se han resumido para mayor claridad y formato

TimeZone	Ubicación
UM12	(UTC - 12:00) Baker/Howland Island
UM11	(UTC - 11:00) Samoa Time Zone, Niue
UM10	(UTC - 10:00) Hawaii-Aleutian Standard Time, Cook Islands
UM95	(UTC - 09:30) Marquesas Islands
UM9	(UTC - 09:00) Alaska Standard Time, Gambier Islands
UM8	(UTC - 08:00) Pacific Standard Time, Clipperton Island
UM7	(UTC - 07:00) Mountain Standard Time
UM6	(UTC - 06:00) Central Standard Time
UM5	(UTC - 05:00) Eastern Standard Time, Western Caribbean
UM45	(UTC - 04:30) Venezuelan Standard Time
UM4	(UTC - 04:00) Atlantic Standard Time, Eastern Caribbean
UM35	(UTC - 03:30) Newfoundland Standard Time
UM3	(UTC - 03:00) Argentina, Brazil, French Guiana, Uruguay
UM2	(UTC - 02:00) South Georgia/South Sandwich Islands
UM1	(UTC - 1:00) Azores, Cape Verde Islands
UTC	(UTC) Greenwich Mean Time, Western European Time
UP1	(UTC +1:00) Central European Time, West Africa Time
UP2	(UTC +2:00) Central Africa Time, Eastern European Time
UP3	(UTC +3:00) Moscow Time, East Africa Time
UP35	(UTC +3:30) Iran Standard Time
UP4	(UTC +4:00) Azerbaijan Standard Time, Samara Time
UP45	(UTC +4:30) Afghanistan
UP5	(UTC +5:00) Pakistan Standard Time, Yekaterinburg Time
UP55	(UTC +5:30) Indian Standard Time, Sri Lanka Time
UP575	(UTC +5:45) Nepal Time

UP6	(UTC +6:00) Bangladesh Standard Time, Bhutan Time, Omsk Time
UP65	(UTC +6:30) Cocos Islands, Myanmar
UP7	(UTC +7:00) Krasnoyarsk Time, Cambodia, Laos, Thailand, Vietnam
UP8	(UTC +8:00) Australian Western Standard Time, Beijing Time
UP875	(UTC +8:45) Australian Central Western Standard Time
UP9	(UTC +9:00) Japan Standard Time, Korea Standard Time, Yakutsk
UP95	(UTC +9:30) Australian Central Standard Time
UP10	(UTC +10:00) Australian Eastern Standard Time, Vladivostok Time
UP105	(UTC +10:30) Lord Howe Island
UP11	(UTC +11:00) Srednekolymysk Time, Solomon Islands, Vanuatu
UP115	(UTC +11:30) Norfolk Island
UP12	(UTC +12:00) Fiji, Gilbert Islands, Kamchatka, New Zealand
UP1275	(UTC +12:45) Chatham Islands Standard Time
UP13	(UTC +13:00) Phoenix Islands Time, Tonga
UP14	(UTC +14:00) Line Islands

## Directory Helper

---

`directory_map()`

**array** `directory_map`(string `$source_dir` [, int `$directory_depth` = 0 [, bool `$hidden` = FALSE]])

### Descripción

---

Lee la ruta del directorio especificado y crea un array con él y todo su contenido

### Parámetros

---

#### **source\_dir**

Ruta al directorio fuente

#### **directory\_depth**

Profundidad de directorios para recorrer (0 = totalmente recursivo, 1 = directorio actual, etc.)

#### **hidden**

Si se incluyen directorios ocultos

### Valores devueltos

---

Un array de archivos

### Ejemplo

---

```
$map = directory_map('./mydirectory/');
```

### Nota

Las rutas son casi siempre relativas a su archivo **index.php** principal

Los subdirectorios contenidos dentro del directorio también se incluyen, si desea controlar la profundidad de recursión, puede hacerlo utilizando el segundo parámetro, una profundidad de 1 solo recorre el directorio de nivel superior :

```
$map = directory_map('./mydirectory/', 1);
```

Por defecto, los archivos ocultos no se incluyen en el array devuelto, para anular este comportamiento, puede establecer el tercer parámetro como TRUE:

```
$map = directory_map('./mydirectory/', FALSE, TRUE);
```

Cada nombre de directorio será un índice de array, mientras que sus archivos contenidos serán indexados numéricamente

Aquí hay un ejemplo de un array típico:

```
Array (
  [libraries] => Array
  (
    [0] => benchmark.html
    [1] => config.html
    ["database/"] => Array
    (
      [0] => query_builder.html
      [1] => binds.html
      [2] => configuration.html
      [3] => connecting.html
      [4] => examples.html
      [5] => fields.html
      [6] => index.html
      [7] => queries.html
    )
  [2] => email.html
  [3] => file_uploading.html
  [4] => image_lib.html
  [5] => input.html
  [6] => language.html
  [7] => loader.html
  [8] => pagination.html
  [9] => uri.html
  )
)
```



## Download Helper

---

`force_download()`

**`void force_download([ string $filename = '' [, mixed $data = '' [, bool $set_mime = FALSE]])`**

### Descripción

---

Genera encabezados de servidor que fuerzan la descarga de datos a su escritorio, útil con descargas de archivos

### Parámetros

---

#### **filename**

Nombre para el archivo descargado

#### **data**

Contenidos del archivo, el archivo de datos, si tiene valor `NULL` y **\$filename** es una ruta de archivo legible existente, su contenido se leerá en su lugar

#### **set\_mime**

Si se trata de enviar el tipo MIME real, si tiene valor `TRUE`, se enviará el tipo de archivo MIME real, basado en la extensión del nombre de archivo, de modo que si su navegador tiene un controlador para ese tipo, puede usarlo

### valores devueltos

---

`void`

### Ejemplo

---

```
$datos = 'Aquí hay algo de texto!';
$nombre = 'mitext.txt';
force_download($nombre, $datos);
```

Si desea descargar un archivo existente de su servidor, debe hacer lo siguiente:

```
// El contenido de foto.jpg se leerá automáticamente
force_download('/path/a/foto.jpg', NULL);
```

## File Helper

---

`delete_files()`

**`bool delete_files(string $path [, bool $del_dir = FALSE [, bool $htdocs = FALSE]])`**

### Descripción

---

Elimina TODOS los archivos contenidos en la ruta suministrada

### Parámetros

---

#### **path**

Ruta del directorio

## del\_dir

Si también eliminar directorios, si se establece en TRUE, también se eliminarán los directorios contenidos en la ruta raíz suministrada

## htdocs

Si se salta la eliminación de **.htaccess** y los archivos de página de índice

### Valores devueltos

---

TRUE en caso de éxito, FALSE en caso de error

### Ejemplo

---

```
delete_files('./path/al/directorio/');  
  
delete_files('./path/al/directorio/', TRUE);
```

### Nota

Los archivos deben ser de escritura o propiedad del sistema para ser eliminados

## get\_dir\_file\_info()

**array get\_dir\_file\_info(string \$source\_dir, bool \$top\_level\_only)**

### Descripción

---

Lee el directorio especificado y crea un array que contiene el nombre de archivo, tamaño, fecha y permisos, los subdirectorios contenidos en la ruta especificada solo se leen si se fuerzan al enviar el segundo parámetro a FALSE, ya que puede tratarse de una operación lenta

### Parámetros

---

#### source\_dir

Ruta del directorio

#### top\_level\_only

Si se debe mirar solo el directorio especificado, excluyendo los subdirectorios

### Valores devueltos

---

Un array que contiene información sobre los contenidos del directorio suministrado

### Ejemplo

---

```
$models_info = get_dir_file_info(APPPATH.'models/');
```

## get\_file\_info()

**array get\_file\_info(string \$file [, array \$returned\_values = array('name', 'server\_path', 'size', 'date')])**

### Descripción

---

Obtener información de un archivo, nombre, ruta, tamaño, fecha de modificación y permisos

## Parámetros

---

### file

Ruta del archivo

### returned\_values

Qué información devolver, las opciones válidas son : name, size, date, readable, writeable, executable y fileperms

## Valores devueltos

---

Un array con la información requerida o FALSE si falla

### get\_filenames()

**array get\_filenames(string \$source\_dir [, bool \$include\_path = FALSE])**

## Descripción

---

Toma una ruta de servidor como entrada y devuelve un array que contiene los nombres de todos los archivos que contiene, la ruta del archivo se puede agregar opcionalmente a los nombres de archivo estableciendo el segundo parámetro en TRUE

## Parámetros

---

### source\_dir

Ruta del directorio

### include\_path

Si se debe incluir la ruta como parte de los nombres de archivo

## Valores devueltos

---

Un array de nombres de archivos

## Ejemplo

---

```
$controllers = get_filenames(APPPATH.'controllers/');
```

### get\_mime\_by\_extension()

**string get\_mime\_by\_extension(string \$filename)**

## Descripción

---

Traduce la extensión de nombre de archivo en un tipo MIME basado en **config/mimes.php**

## Parámetros:

---

### filename

Nombre de archivo

## Valores devueltos

---

Cadena de tipo MIME o FALSE en si no puede determinar el tipo o leer el archivo de configuración MIME

## Ejemplo

---

```
$nomarch = 'algun_archivo.png';  
echo $nomarch.' tiene un tipo mime de '.get_mime_by_extension($nomarch);
```

### Nota

Esta no es una forma precisa de determinar los tipos MIME de archivos, y está aquí estrictamente por conveniencia

**No debe usarse con fines de seguridad**

## octal\_permissions()

**string octal\_permissions(int \$perms)**

### Descripción

---

Toma permisos numéricos, como los devueltos por **fileperms()**, y devuelve una notación octal de tres caracteres de permisos de archivos

### Parámetros

---

#### perms

Permisos

### Valores devueltos

---

String de permisos octal

### Ejemplo

---

```
echo octal_permissions(fileperms('./index.php')); // 644
```

## read\_file()

### Nota

Esta función está **DESACONSEJADA**. Use la función nativa de PHP : **file\_get\_contents()** en su lugar

## symbolic\_permissions()

**string symbolic\_permissions(int \$perms)**

### Descripción

---

Toma permisos numéricos, como los devueltos por **fileperms()**, y devuelve la notación simbólica estándar de los permisos de archivos

### Parámetros

---

#### perms

Permisos

### Valores devueltos

---

String de permisos simbólicos

### Ejemplo

---

```
echo symbolic_permissions(fileperms('./index.php')); // -rw-r--r--
```

## write\_file()

**bool write\_file(string \$path, string \$data [, string \$mode = 'wb'])**

### Descripción

---

Escribe datos en el archivo especificado en la ruta, si el archivo no existe, la función lo crea

### Parámetros

---

#### path

Ruta del archivo

#### data

Datos para escribir en el archivo

#### mode

Modo **fopen()**

### Valores devueltos

---

TRUE si la escritura fue exitosa, FALSE en caso de error

### Ejemplo

---

```
$datos = 'Algunos datos de archivo';  
if ( ! write_file('./path/a/archivo.php', $datos))  
{  
    echo 'No se puede escribir el archivo';  
}  
else  
{  
    echo 'Archivo escrito!';  
}
```

Opcionalmente, puede establecer el modo de escritura a través del tercer parámetro:

```
write_file('./path/to/file.php', $datos, 'r+');
```

El modo predeterminado es **'wb'**, por favor, consulte la guía de usuario de PHP para conocer las opciones de modo

### Nota

La ruta es relativa al archivo **index.php** de su sitio principal, **NO a su controlador** ni a sus archivos, CodeIgniter utiliza un controlador frontal para que las rutas sean siempre relativas al índice principal del sitio

### Nota

Esta función adquiere un bloqueo exclusivo en el archivo mientras escribe en él

# Form Helper

---

## form\_button()

```
string form_button([ string $data = " [, string $content = " [, mixed $extra = "" ]])
```

---

### Descripción

Le permite generar un elemento de botón estándar, puede pasar el nombre y el contenido del botón en el primer y segundo parámetro

---

### Parámetros

#### data

Nombre del botón

#### contenido

Etiqueta de botón

#### extra

Atributos adicionales que se agregarán a la etiqueta, ya sea como un array o un string literal

---

### Valores devueltos

Una etiqueta de botón HTML

---

### Ejemplo

```
echo form_button('name','content');  
// Would produce: <button name="name" type="button">Content</button>
```

O puede pasar un array asociativo que contiene todos los datos que desea que contenga su formulario:

```
$datos = array(  
    'name'    => 'button',  
    'id'      => 'button',  
    'value'   => 'true',  
    'type'    => 'reset',  
    'content' => 'Reset'  
);  
  
echo form_button($datos);  
// Would produce: <button name="button" id="button" value="true" type="reset">Reset</button>
```

Si desea que su formulario contenga algunos datos adicionales, como JavaScript, puede pasarlo como un string en el tercer parámetro:

```
$js = 'onClick="alguna_function()";  
echo form_button('mi_button', 'Pulsame', $js);
```

## form\_checkbox()

**string form\_checkbox**([ **array** \$data = " [, **string** \$value = " [, **bool** \$checked = FALSE [, **mixed** \$extra = "]]]])

### Descripción

---

Le permite generar un campo de casilla de verificación

### Parámetros

---

#### data

Datos de atributos de campo

#### value

Valor del campo

#### checked

Si marcar la casilla como marcada

#### extra

Atributos adicionales que se agregarán a la etiqueta, ya sea como un array o un string literal

### Valores devueltos

---

Una etiqueta de entrada HTML checkbox

### Ejemplo

---

```
echo form_checkbox('noticias', 'accept', TRUE);  
// Produce: <input type="checkbox" name="noticias" value="accept" checked="checked" />
```

El tercer parámetro contiene un booleano TRUE / FALSE para determinar si la casilla debe ser marcada o no.

De forma similar a las otras funciones de formulario en este asistente, también puede pasar un array de atributos a la función:

```
$datos = array(  
    'name'    => 'noticias',  
    'id'      => 'id_noticias',  
    'value'   => 'accept',  
    'checked' => TRUE,  
    'style'   => 'margin:10px'  
);  
  
echo form_checkbox($datos);  
// Produce: <input type="checkbox" name="noticias" id="id_noticias" value="accept" checked="checked"  
style="margin:10px" />
```

Además, como con otras funciones, si desea que la etiqueta contenga datos adicionales como JavaScript, puede pasarla como un string en el cuarto parámetro:

```
$js = 'onClick="alguna_function()";  
echo form_checkbox('noticias', 'accept', TRUE, $js);
```

O puede pasarlo como una array:

```
$js = array('onClick' => 'alguna_funcion()');  
echo form_checkbox('noticias', 'accept', TRUE, $js);
```

## form\_close()

**string form\_close**([ **string \$extra** = '' ])

### Descripción

---

Produce una etiqueta de cierre **</ form>**, la única ventaja de usar esta función es que le permite pasar datos que se agregarán debajo de la etiqueta

### Parámetros

---

#### **extra**

Cualquier cosa para agregar después de la etiqueta de cierre

### Valores devueltos

---

Una etiqueta de cierre de formulario HTML

### Ejemplo

---

```
$string = '</div></div>';  
echo form_close($string);  
  
// Produce: </form> </div></div>
```

## form\_dropdown()

**string form\_dropdown**([ **string \$name** = '' [, **array \$options** = **array()** [, **array \$selected** = **array()** [, **mixed \$extra** = '' ]]]])

### Descripción

---

Le permite crear un campo desplegable estándar, el primer parámetro contendrá el nombre del campo, el segundo contendrá un array asociativo de opciones y el tercero el valor que desea seleccionar, también puede pasar un array de elementos múltiples a través del tercer parámetro, y CodeIgniter creará una selección múltiple para usted

### Parámetros

---

#### **name**

Nombre del campo

#### **options**

Un array asociativo de opciones para ser listadas

#### **selected**

Lista de campos para marcar con el atributo seleccionado

#### **extra**

Atributos adicionales que se agregarán a la etiqueta, ya sea como un array o un string literal

### Valores devueltos

---

Una lista desplegable HTML de campo de selección

### Ejemplo

---



```

$opciones = array(
    'pequeña' => 'Camisa pequeña',
    'mediana' => 'Camisa mediana',
    'grande' => 'Camisa grande',
    'extra' => 'Camisa extragrande',
);

$camisas_en_venta = array('pequeña', 'grande');
echo form_dropdown('camisas', $opciones, 'grande');

/*
Produciría:
<select name="camisas">
    <option value="pequeña">Camisa pequeña</option>
    <option value="mediana">Camisa mediana</option>
    <option value="grande" selected="selected">Camisa grande</option>
    <option value="extra">Camisa extragrande</option>
</select>
*/

echo form_dropdown('camisas', $opciones, $camisas_en_venta);

/*
Produciría:
<select name="camisas" multiple="multiple">
    <option value="pequeña" selected="selected">Camisa pequeña</option>
    <option value="mediana">Camisa mediana</option>
    <option value="grande" selected="selected">Camisa grande</option>
    <option value="extra">Camisa extragrande</option>
</select>
*/

```

Si desea que el **<select>** de apertura contenga datos adicionales, como un atributo id o JavaScript, puede pasarlo como un string en el cuarto parámetro:

```

$js = 'id="camisas" onChange="alguna_funcion();"';
echo form_dropdown('camisas', $opciones, 'grande', $js);

```

O puede pasarlo como un array:

```

$js = array(
    'id' => 'camisas',
    'onChange' => 'alguna_funcion()';
);
echo form_dropdown('camisas', $opciones, 'grande', $js);

```

Si el array pasado como **\$opciones** es un array multidimensional, **form\_dropdown()** producirá un **<optgroup>** con la clave de array como etiqueta

`form_error()`

**string form\_error**([ **string \$field** = '' [, **string \$prefix** = '' [, **string \$suffix** = '']] )

---

#### Descripción

Devuelve un mensaje de error de la **Form Validation Library**, asociado con el nombre de campo especificado, opcionalmente puede especificar las etiquetas de apertura y cierre para colocar el mensaje de error

---

#### Parámetros

##### **field**

Nombre del campo

##### **prefix**

Error al abrir la etiqueta

##### **suffix**

Error al cerrar la etiqueta

---

#### Valores devueltos

Mensaje (s) de error de validación de formulario con formato HTML

---

#### Ejemplo

```
// Suponiendo que el valor del campo 'username' fuera incorrecto:
echo form_error('mi_campo', '<div class="error">', '</div>');

// Produce: <div class="error">Mensaje de error asociado con el campo "username".</div>
```

`form_fieldset()`

**string form\_fieldset**([ **string \$legend\_text** = '' [, **array \$attributes** = **array()**]] )

---

#### Descripción

Le permite generar campos **fieldset / legend**

---

#### Parámetros

##### **legend\_text**

Texto para poner en la etiqueta **<legend>**

##### **attributes**

Atributos que se establecerán en la etiqueta **<fieldset>**

---

#### Valores devueltos

Una etiqueta de apertura de conjunto de campos HTML

---

#### Ejemplo

```

echo form_fieldset('Dirección');
echo "<p>aquí el contenido del fieldset</p>\n";
echo form_fieldset_close();

/*
Produce:

<fieldset>
  <legend>Dirección</legend>
  <p>aquí el contenido del formulario</p>
</fieldset>
*/

```

De forma similar a otras funciones, puede enviar un array asociativo en el segundo parámetro si prefiere establecer atributos adicionales:

```

$atributos = array(
  'id'    => 'inf_contacto',
  'class' => 'inf_contacto'
);

echo form_fieldset('Información de contacto', $atributos);
echo "<p>aquí el contenido del fieldset</p>\n";
echo form_fieldset_close();

/*
Produce:
<fieldset id="inf_contacto" class="inf_contacto">
  <legend>Información de contacto</legend>
  <p>aquí el contendo del form</p>
</fieldset>
*/

```

## form\_fieldset\_close()

**string form\_fieldset\_close**([ string \$extra = '' ])

### Descripción

Produce una etiqueta de cierre **</ fieldset>**, la única ventaja de usar esta función es que le permite pasar datos que se agregarán debajo de la etiqueta

### Parámetros

#### extra

Cualquier cosa para agregar después de la etiqueta de cierre

### Valores devueltos

Una etiqueta de cierre de campo de HTML

### Ejemplo

```
$string = '</div></div>';  
echo form_fieldset_close($string);  
// Produce: </fieldset></div></div>
```

## form\_hidden()

**string form\_hidden(string \$name [, string \$value = ""])**

### Descripción

---

Le permite generar campos de entrada ocultos

### Parámetros

---

#### name

Nombre del campo

#### value

Valor del campo

### Valores devueltos

---

Una etiqueta de campo de entrada oculto HTML

### Ejemplo

---

Puede enviar una string de nombre / valor para crear un campo:

```
form_hidden('nom_usuario', 'jujol');  
  
// Produce: <input type="hidden" name="nom_usuario" value="jujol" />
```

... o puede enviar un array asociativo para crear múltiples campos:

```
$datos = array(  
    'nombre' => 'Salvador Dalí',  
    'email'  => 'salvador@example.com',  
    'url'    => 'http://ejemplo.com'  
);  
  
echo form_hidden($datos);  
  
/*  
Produciría:  
<input type="hidden" name="nombre" value="Salvador Dalí" />  
<input type="hidden" name="email" value="salvador@example.com" />  
<input type="hidden" name="url" value="http://example.com" />  
*/
```

También puede pasar un array asociativo al valor de campo:

```

$datos = array(
    'nombre' => 'Antonio Gaudí',
    'email' => 'antonio@example.com',
    'url' => 'http://example.com'
);

echo form_hidden('mi_array', $datos);

/*
Produciría:
<input type="hidden" name="mi_array[nombre]" value="Antonio Gaudí" />
<input type="hidden" name="mi_array[email]" value="antonio@example.com" />
<input type="hidden" name="mi_array[url]" value="http://example.com" />
*/
*/

```

Si desea crear campos de entrada ocultos con atributos adicionales:

```

$datos = array(
    'type' => 'hidden',
    'name' => 'email',
    'id' => 'mail_oculto',
    'value' => 'laura@example.com',
    'class' => 'mail_oculto'
);

echo form_input($datos);

/*
Produce:
<input type="hidden" name="email" value="laura@example.com" id="mail_oculto" class="mail_oculto"
/>
*/

```

## form\_input()

**string form\_input([ array \$data = " [, string \$value = " [, mixed \$extra = "" ] ] ] )**

### Descripción

---

Le permite generar un campo de entrada de texto estándar

### Parámetros

---

#### data

Datos de atributos de campo

#### value

Valor del campo

#### extra

Atributos adicionales que se agregarán a la etiqueta, ya sea como un array o un string literal

#### Valores devueltos

---

Una etiqueta de campo de entrada de texto HTML

#### Ejemplo

---

Como mínimo tiene que pasar el nombre del campo en el primer parámetro y el valor en el segundo:

```
echo form_input('nom_usuario', 'antonio_gaudí');
```

O puede pasar un array asociativo que contiene todos los datos que desea que contenga su formulario:

```
$datos = array(
    'name'      => 'nom_usuario',
    'id'        => 'nom_usuario',
    'value'     => 'antonio_gaudí',
    'maxlength' => '100',
    'size'      => '50',
    'style'     => 'width:50%'
);

echo form_input($datos);

/*
Produce:

<input type="text" name="nom_usuario" value="antonio_gaudí" id="nom_usuario" maxlength="100"
size="50" style="width:50%" />
*/
```

Si desea que su formulario contenga algunos datos adicionales, como JavaScript, puede pasarlo como un string en el tercer parámetro:

```
$js = 'onClick="alguna_function()";
echo form_input('nom_usuario', 'antonio_gaudí', $js);
```

O puede pasarlo como un array:

```
$js = array('onClick' => 'some_function();');
echo form_input('nom_usuario', 'antonio_gaudí', $js);
```

## form\_label()

```
string form_label([ string $label_text = '' [, string $id = '' [, mixed $attributes = array()]]])
```

### Descripción

---

Le permite generar una **<label>**

### Parámetros

---

#### label\_text

Texto para poner en la etiqueta **<label>**

#### id

ID del elemento de formulario para el que estamos haciendo una etiqueta

#### attributes

Atributos de HTML

### Valores devueltos

---

Una etiqueta de etiqueta de campo HTML

### Ejemplo

---

```
echo form_label('Cuál es su nombre?', 'nom_usuario');  
// Produce: <label for="nom_usuario">Cuál es su nombre?</label>
```

De forma similar a otras funciones, puede enviar un array asociativo en el tercer parámetro si prefiere establecer atributos adicionales:

```
$atributos = array(  
    'class' => 'micustomclass',  
    'style' => 'color: #000;'  
);  
  
echo form_label('Cuál es su nombre?', 'nom_usuario', $atributos);  
// Produce: <label for="nom_usuario" class="micustomclass" style="color: #000;">Cuál es su  
//          nombre?</label>
```

## form\_multiselect()

```
string form_multiselect([ string $name = '' [, array $options = array() [, array $selected = array() [, mixed $extra = '']]]])
```

### Descripción

---

Le permite crear un campo multiselect estándar, el primer parámetro contendrá el nombre del campo, el segundo parámetro contendrá un array asociativo de opciones y el tercer parámetro contendrá el valor o valores que desea seleccionar

El uso de parámetros es idéntico al uso de **form\_dropdown()** anterior, excepto, por supuesto, que el nombre del campo necesitará usar la sintaxis del array **POST**, por ejemplo **foo[ ]**

### Parámetros

---

**name**

Nombre del campo

**options**

Un array asociativo de opciones para ser listadas

**selected**

Lista de campos para marcar con el atributo seleccionado

**extra**

Atributos adicionales que se agregarán a la etiqueta, ya sea como un array o un string literal

**Valores devueltos**

---

Una etiqueta de campo de selección múltiple HTML desplegable

**form\_open()**

**string form\_open([ string \$action = " ", array \$attributes = " ", array \$hidden = array() ])**

**Descripción**

---

Crea una etiqueta de apertura de formulario con una URL base creada a partir de sus preferencias de configuración, opcionalmente, le permite agregar atributos de formulario y campos de entrada ocultos, y siempre agregará el atributo accept-charset basado en el valor del conjunto de caracteres en su archivo de configuración

El beneficio principal al utilizar esta etiqueta en lugar de la codificación estricta de su propio HTML es que permite que su sitio sea más portátil si cambian sus URL

**Parámetros**

---

**action**

Cadena URI de acción / destino de formulario

**attributes**

Atributos de HTML

**hidden**

Un array de definiciones de campos ocultos

**Valores devueltos**

---

Una etiqueta de apertura de formulario HTML

**Ejemplo**

---

```
echo form_open('email/envia');
```

El ejemplo anterior crea un formulario que apunta a su URL base más los segmentos "**email/envia**", como este:

```
<form method="post" accept-charset="utf-8" action="http://ejemplo.com/index.php/email/envia">
```

**Agregar atributos**

Los atributos se pueden agregar pasando un array asociativo al segundo parámetro, como este:

```
$atributos = array('class' => 'email', 'id' => 'mi_form');
echo form_open('email/envia', $atributos);
```



Alternativamente, puede especificar el segundo parámetro como un string:

```
echo form_open('email/envia', 'class="email" id="mi_form");
```

Los ejemplos anteriores crean un formulario similar a este:

```
<form method="post" accept-charset="utf-8" action="http://ejemplo.com/index.php/email/envio" class="email" id="mi_form">
```

## Agregar campos de entrada ocultos

Los campos ocultos se pueden agregar pasando un array asociativo al tercer parámetro, así:

```
$oculto = array('usuario_nombre' => 'Carmen', 'miembro_id' => '234');
echo form_open('email/enviar', '', $oculto);
```

Puede omitir el segundo parámetro pasándole cualquier valor falso

El ejemplo anterior crearía un form similar a este:

```
<form method="post" accept-charset="utf-8" action="http://ejemplo.com/index.php/email/enviar">
    <input type="hidden" name="usuario_nombre" value="Carmen" />
    <input type="hidden" name="miembro_id" value="234" />
```

## form\_open\_multipart()

**string form\_open\_multipart**([ **string \$action** = " [, **array \$attributes** = **array()** [, **array \$hidden** = **array()** ] ] ])

### Descripción

---

Esta función es idéntica a la anterior **form\_open()**, excepto que agrega un atributo **multipart**, necesario si desea usar el formulario para cargar archivos

### Parámetros

---

#### **action**

Cadena URI de acción / destino de formulario

#### **attributes**

Atributos de HTML

#### **hidden**

Un array de definiciones de campos ocultos

### Valores devueltos

---

Una etiqueta de apertura de formulario multipart HTML

## form\_password()

**string form\_password**([ **array \$data** = " [, **string \$value** = " [, **mixed \$extra** = " ] ] ])

### Descripción

---

Esta función es idéntica a la función `form_input()` anterior, excepto que utiliza el tipo de entrada "**password**"

---

#### Parámetros

##### **data**

Datos de atributos de campo

##### **value**

Valor del campo

##### **extra**

Atributos adicionales que se agregarán a la etiqueta, ya sea como un array o un string literal

---

#### Valores devueltos

Una etiqueta de campo de entrada de contraseña HTML

#### form\_prep()

##### **Nota**

Esta función esta **DESACONSEJADA** y es solo un alias para la función común `html_escape()` , usela en su lugar

#### form\_radio()

```
string form_radio([ array $data = " [, string $value = " [, bool $checked = FALSE [, mixed $extra = "]]])
```

---

#### Descripción

Esta función es idéntica en todos los aspectos a la función `form_checkbox()` anterior excepto que usa el tipo de entrada "**radio**"

---

#### Parámetros

##### **data**

Datos de atributos de campo

##### **value**

Valor del campo

##### **checked**

Si marcar el botón de radio como marcado

##### **extra**

Atributos adicionales que se agregarán a la etiqueta, ya sea como un array o un string literal

---

#### Valores devueltos

Una etiqueta de entrada de radio HTML

#### form\_reset()

```
string form_reset([ string $data = " [, string $value = " [, mixed $extra = "]]])
```

---

#### Descripción

Le permite generar un botón de reinicio estándar, el uso es idéntico a `form_submit()`

## Parámetros

---

### **data**

Nombre del botón

### **value**

Valor del botón

### **extra**

Atributos adicionales que se agregarán a la etiqueta, ya sea como un array o un string literal

## Valores devueltos

---

Una etiqueta de botón de reinicio de entrada HTML

## form\_submit()

```
string form_submit([ string $data = " [, string $value = " [, mixed $extra = "" ]])
```

## Descripción

---

Le permite generar un botón de envío estándar

## Parámetros

---

### **data**

Nombre del botón

### **value**

Valor del botón

### **extra**

Atributos adicionales que se agregarán a la etiqueta, ya sea como un array o un string literal

## Valores devueltos

---

Una etiqueta de envío de entrada HTML

## Ejemplo

---

```
echo form_submit('mi_submit', 'Submit Post!');

// Produce: <input type="submit" name="mi_submit" value="Submit Post!" />
```

De forma similar a otras funciones, puede enviar un array asociativo en el primer parámetro si prefiere establecer sus propios atributos, el tercer parámetro le permite agregar datos adicionales a su formulario, como JavaScript

## form\_textarea()

**string form\_textarea**([ array \$data = " [, string \$value = " [, mixed \$extra = "]]])

### Descripción

---

Esta función es idéntica a la función **form\_input()** anterior, excepto que genera un tipo "textarea"

### Nota

En lugar de los atributos **maxlength** y **size** en el ejemplo anterior, en su lugar se especificarán **rows** y **cols**

### Parámetros

---

#### data

Datos de atributos de campo

#### value

Valor del campo

#### extra

Atributos adicionales que se agregarán a la etiqueta, ya sea como un array o un string literal

### Valores devueltos

---

Una etiqueta HTML textarea

## form\_upload()

**string form\_upload**([ array \$data = " [, string \$value = " [, mixed \$extra = "]]])

### Descripción

---

Esta función es idéntica a la función **form\_input()** anterior, excepto que utiliza el tipo de entrada "file", lo que permite su uso para cargar archivos

### Parámetros

---

#### data

Datos de atributos de campo

#### value

Valor del campo

#### extra

Atributos adicionales que se agregarán a la etiqueta, ya sea como un array o un string literal

### Valores devueltos

---

Una etiqueta de campo de entrada de carga de archivo HTML

## set\_checkbox()

**string set\_checkbox**(string \$field [, string \$value = " [, string \$default = FALSE]])

### Descripción

---

Le permite mostrar una casilla de verificación en el estado en que fue enviada, el primer parámetro debe contener el nombre de la casilla de verificación, el segundo parámetro debe contener su valor y el tercer parámetro, opcional, le permite establecer un elemento como predeterminado, use TRUE / FALSE

## Parámetros

---

### field

Nombre del campo

### value

Valor para verificar

### default

Si el valor también es uno predeterminado

## Valores devueltos

---

Atributo '**checked**' o una string vacía

## Ejemplo

---

```
<input type="checkbox" name="mi_check" value="1" <?php echo set_checkbox('mi_check', '1'); ?> />
<input type="checkbox" name="mi_check" value="2" <?php echo set_checkbox('mi_check', '2'); ?> />
```

## set\_radio()

**string set\_radio(string \$field [, string \$value = '' [, string \$default = FALSE]])**

## Descripción

---

Le permite mostrar botones de opción en el estado en que fueron enviados, esta función es idéntica a la función **set\_checkbox()** anterior

## Parámetros

---

### field

Nombre del campo

### value

Valor para verificar

### default

Si el valor también es uno predeterminado

## Valores devueltos

---

Atributo '**checked**' o una string vacía

## Ejemplo

---

```
<input type="radio" name="mi_radio" value="1" <?php echo set_radio('mi_radio', '1', TRUE); ?> />
<input type="radio" name="mi_radio" value="2" <?php echo set_radio('mi_radio', '2'); ?> />
```

## Nota

Si está utilizando la clase **Form Validation**, siempre debe especificar una regla para su campo, incluso si está vacía, para que funcionen las funciones **set\_\***(), esto se debe a que si se define un objeto de **Form Validation**, el control para el conjunto **set\_\***() se transfiere a un método de la clase en lugar de a la función auxiliar genérica

## set\_select()

**string set\_select(string \$field [, string \$value = '' [, string \$default = FALSE]])**

## Descripción

---

Si usa un menú **<select>**, esta función le permite visualizar el elemento del menú que se seleccionó, el primer parámetro debe contener el nombre del menú de selección, el segundo parámetro debe contener el valor de cada elemento y el tercer parámetro, opcional, le permite establecer un elemento como predeterminado, use TRUE / FALSE

## Parámetros

---

### field

Nombre del campo

### value

Valor para verificar

### default

Si el valor también es uno predeterminado

## Valores devueltos

---

Atributo **'seleccionado'** o una string vacía

## Ejemplo

---

```
<select name="mi_select">
  <option value="uno" <?php echo set_select('mi_select', 'uno', TRUE); ?> >Uno</option>
  <option value="dos" <?php echo set_select('mi_select', 'dos'); ?> >Dos</option>
  <option value="tres" <?php echo set_select('mi_select', 'tres'); ?> >Tres</option>
</select>
```

## set\_value()

**string set\_value(string \$field [, string \$default = " [, bool \$html\_escape = TRUE]])**

## Descripción

---

Le permite establecer el valor de un formulario de entrada o área de texto, debe proporcionar el nombre del campo a través del primer parámetro de la función, el segundo parámetro, opcional, le permite establecer un valor predeterminado para el formulario, el tercer parámetro, opcional, le permite desactivar el escapado de HTML del valor, en caso de que necesite usar esta función en combinación con, por ejemplo, **form\_input()** y evitar el doble escape

## Parámetros

---

### field

Nombre del campo

### default

Valor predeterminado

### html\_escape

Si se debe desactivar el escape de HTML del valor

## Valores devueltos

---

Valor de campo

## Ejemplo

---

```
<input type="text" name="cantidad" value="<?php echo set_value('cantidad', '0'); ?>" size="50" />
```

El formulario anterior mostrará "0" cuando se cargue por primera vez

### Nota

Si ha cargado la **Form Validation Library** y ha establecido una regla de validación para el nombre de campo en uso de este ayudante, reenviará la llamada al propio método **set\_value()** de la **Form Validation Library**, de lo contrario, esta función se basa en **\$\_POST** para el valor del campo

`validation_errors()`

**string validation\_errors**([ **string \$prefix** = " " [, **string \$suffix** = ""]])

#### Descripción

---

De forma similar a la función **form\_error()**, devuelve todos los mensajes de error de validación producidos por la **Form Validation Library**, con etiquetas de apertura y cierre opcionales alrededor de cada uno de los mensajes

#### Parámetros

---

##### prefix

Error al abrir la etiqueta

##### sufijo

Error al cerrar la etiqueta

#### Valores devueltos

---

Mensaje(s) de error de validación de formulario con formato HTML

#### Ejemplo

---

```
echo validation_errors('<span class="error">', '</span>');

/*
Produce, por ejemplo:

<span class="error">El campo "email" no contiene una dirección válida!</span>
<span class="error">El campo "contraseña" no coincide con el campo "repetir_contraseña"!</span>
*/
```

br()

## Nota

Esta función está **DESACONSEJADA**. Utilice la función nativa de PHP: **str\_repeat()** en combinación con **<br />** en su lugar

doctype()

**string doctype([ string \$type = 'xhtml1-strict'])**

## Descripción

Genera declaraciones de tipo de documento o DTD. XHTML 1.0 Strict se usa de manera predeterminada, pero hay muchos tipos de **DocType** disponibles

## Parámetros

### type

DocType name

## Valores devueltos

Etiqueta HTML DocType

## Ejemplo

```
echo doctype(); // <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

echo doctype('html4-trans'); // <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
```

La siguiente es una lista de opciones de '**DOCTYPE**', tipo de documento, son configurables y extraídos de: **application/config/doctypes.php**

Document type	Opción	Resultado
<b>XHTML 1.1</b>	xhtml11	<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<b>XHTML 1.0 Strict</b>	xhtml1-strict	<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<b>XHTML 1.0 Transitional</b>	xhtml1-trans	<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<b>XHTML 1.0 Frameset</b>	xhtml1-frame	<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
<b>XHTML Basic 1.1</b>	xhtml-basic11	<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML Basic 1.1//EN" "http://www.w3.org/TR/xhtml-basic/xhtml-basic11.dtd">
<b>HTML 5</b>	html5	<!DOCTYPE html>



<b>HTML 4 Strict</b>	html4-strict	<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<b>HTML 4 Transitional</b>	html4-trans	<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<b>HTML 4 Frameset</b>	html4-frame	<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN" "http://www.w3.org/TR/html4/frameset.dtd">
<b>MathML 1.01</b>	mathml1	<!DOCTYPE math SYSTEM "http://www.w3.org/Math/DTD/mathml1/mathml.dtd">
<b>MathML 2.0</b>	mathml2	<!DOCTYPE math PUBLIC "-//W3C//DTD MathML 2.0//EN" "http://www.w3.org/Math/DTD/mathml2/mathml2.dtd">
<b>SVG 1.0</b>	svg10	<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.0//EN" "http://www.w3.org/TR/2001/REC-SVG-20010904/DTD/svg10.dtd">
<b>SVG 1.1 Full</b>	svg11	<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN" "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<b>SVG 1.1 Basic</b>	svg11-basic	<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1 Basic//EN" "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11-basic.dtd">
<b>SVG 1.1 Tiny</b>	svg11-tiny	<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1 Tiny//EN" "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11-tiny.dtd">
<b>XHTML+MathML+SVG (XHTML host)</b>	xhtml-math-svg-xh	<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1 plus MathML 2.0 plus SVG 1.1//EN" "http://www.w3.org/2002/04/xhtml-math-svg/xhtml-math- svg.dtd">
<b>XHTML+MathML+SVG (SVG host)</b>	xhtml-math-svg-sh	<!DOCTYPE svg:svg PUBLIC "-//W3C//DTD XHTML 1.1 plus MathML 2.0 plus SVG 1.1//EN" "http://www.w3.org/2002/04/xhtml-math-svg/xhtml- math-svg.dtd">
<b>XHTML+RDFa 1.0</b>	xhtml-rdfa-1	<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML+RDFa 1.0//EN" "http://www.w3.org/MarkUp/DTD/xhtml-rdfa-1.dtd">
<b>XHTML+RDFa 1.1</b>	xhtml-rdfa-2	<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML+RDFa 1.1//EN" "http://www.w3.org/MarkUp/DTD/xhtml-rdfa-2.dtd">

`heading()`

**string heading**([ **string \$data** = " [, **string \$h** = '1' [, **mixed \$attributes** = "]]])

#### Descripción

Le permite crear etiquetas de encabezado HTML, el primer parámetro contendrá los datos, el segundo el tamaño del encabezado

#### Parámetros

##### **data**

Contenido

##### **h**

Nivel de la etiqueta H, H1, H2,...

##### **attributes**

Atributos de HTML

#### Valores devueltos

Etiqueta de encabezado HTML

#### Ejemplo

```
echo heading('Bienvenido!', 3);
// Produce: <h3>Bienvenido!</ H3>
```

Además, para agregar atributos a la etiqueta de encabezado, como clases HTML, ids o estilos en línea, un tercer parámetro acepta un string o un array:

```
echo heading('Bienvenido!', 3, 'class="carmen"');
echo heading('Cómo está?', 4, array('id' => 'pregunta', 'class' => 'julia'));
```

El código anterior produce:

```
<h3 class="carmen">Bienvenido!</h3>
<h4 id="question" class="julia">Cómo está?</h4>
```

## img()

**string** **img**([ **string** **\$src** = " [, **bool** **\$index\_page** = **FALSE** [, **array** **\$attributes** = "]])

### Descripción

---

Le permite crear etiquetas HTML **<img />**, el primer parámetro contiene la fuente de la imagen

### Parámetros

---

#### **src**

Datos fuente de la imagen

#### **index\_page**

Si se trata **\$src** como un string de URI enrutada

#### **attributes**

Atributos de HTML

### Valores devueltos

---

Etiqueta de imagen HTML

### Ejemplo

---

```
echo img('images/picture.jpg');
// Produce: 
```

Hay un segundo parámetro opcional que es un valor TRUE / FALSE que especifica si el **src** debe tener la página especificada por **\$config['index\_page']** agregada a la dirección que crea, presumiblemente, esto sería si estuviera usando un controlador de medios:

```
echo img('images/picture.jpg', TRUE);
// Produce: 
```

Además, un array asociativo se puede pasar a la función **img()** para un control completo sobre todos los atributos y valores, si no se proporciona un atributo **alt**, CodeIgniter generará un string vacío:

```

$imagen_propiedades = array(
    'src' => 'images/picture.jpg',
    'alt' => 'Un pequeño gatito, un dulce micifú',
    'class' => 'post_images',
    'width' => '200',
    'height' => '200',
    'title' => 'Algo nuevo,super, super original...',
    'rel' => 'lightbox'
);

img($imagen_propiedades);

// 

```

## link\_tag()

```

string link_tag([ string $href = " ", string $rel = 'stylesheet' [, string $type = 'text/css'
[, string $title = " ", string $media = " ", bool $index_page = FALSE]]]]])

```

### Descripción

---

Le permite crear etiquetas HTML `<link />`, esto es útil para enlaces de hojas de estilo, así como otros enlaces

### Parámetros

---

#### href

Qué estamos vinculando

#### rel

Tipo de relación, parámetro opcional

#### type

Tipo del documento relacionado, parámetro opcional

#### title

Título del enlace, parámetro opcional

#### media

Tipo de medios, parámetro opcional

#### index\_page

Si se trata **\$src** como una string de URI enrutada, parámetro opcional

### Valores devueltos

---

Etiqueta de enlace HTML

### Ejemplo

---

**index\_page** es un valor booleano que especifica si **href** debe tener la página especificada por **\$config['index\_page']** agregada a la dirección que crea:

```

echo link_tag('css/mystyles.css');
// gives <link href="http://site.com/css/mystyles.css" rel="stylesheet" type="text/css" />

echo link_tag('favicon.ico', 'shortcut icon', 'image/ico');
// <link href="http://site.com/favicon.ico" rel="shortcut icon" type="image/ico" />

echo link_tag('feed', 'alternate', 'application/rss+xml', 'My RSS Feed');
// <link href="http://site.com/feed" rel="alternate" type="application/rss+xml"
// title="My RSS Feed" />

```

Además, un array asociativo se puede pasar a la función **link()** para un control completo sobre todos los atributos y valores:

```

$link = array(
    'href' => 'css/printer.css',
    'rel' => 'stylesheet',
    'type' => 'text/css',
    'media' => 'print'
);

echo link_tag($link);
// <link href="http://site.com/css/printer.css" rel="stylesheet" type="text/css" media="print" />

```

## meta()

**string meta([ string \$name = '' [, string \$content = '' [, string \$type = 'name' [, string \$newline = "n"]]])**

### Descripción

---

Generar meta etiquetas, puede pasar strings a la función, o arrays simples, o multidimensionales

### Parámetros

---

#### name

Meta nombre

#### content

Meta contenido

#### type

Meta type

#### newline

Carácter Newline

### Valores devueltos

---

HTML meta tag

### Ejemplo

---

```

echo meta('description', 'Mi gran sitio');
// Produce: <meta name="description" content="Mi gran sitio" />

echo meta('Content-type', 'text/html; charset=utf-8', 'equiv');
// Tenga en cuenta el tercer parámetro. Puede ser "equiv" or "name"
// Produce: <meta http-equiv="Content-type" content="text/html; charset=utf-8" />

echo meta(array('name' => 'robots', 'content' => 'no-cache'));
// Produce: <meta name="robots" content="no-cache" />

$meta = array(
    array(
        'name' => 'robots',
        'content' => 'no-cache'
    ),
    array(
        'name' => 'description',
        'content' => 'Mi gran sitio'
    ),
    array(
        'name' => 'keywords',
        'content' => 'dalí, gaudí, mortadelo, asterix,kafka, Lovecraft'
    ),
    array(
        'name' => 'robots',
        'content' => 'no-cache'
    ),
    array(
        'name' => 'Content-type',
        'content' => 'text/html; charset=utf-8', 'type' => 'equiv'
    )
);

echo meta($meta);
// Produce:
//
// <meta name="robots" content="no-cache" />
// <meta name="description" content=" Mi gran sitio" />
// <meta name="keywords" content=" dalí, gaudí, mortadelo, asterix,kafka, Lovecraft" />
// <meta name="robots" content="no-cache" />
// <meta http-equiv="Content-type" content="text/html; charset=utf-8" />

```

nbs()

### Nota

Esta función está **DESACONSEJADA**. Utilice la función nativa de PHP: **str\_repeat()** en combinación con **&nbsp;** en su lugar

**ol()**

**string ol(array \$list, array \$attributes = '')**

#### Descripción

---

Idéntico a **ul()**, solo que produce la etiqueta **<ol>** para las listas ordenadas en lugar de **<ul>**

#### Parámetros

---

##### **list**

Entradas de la lista

##### **attributes**

Atributos de HTML

#### Valores devueltos

---

Lista ordenada con formato HTML

**ul()**

**string ul(array \$list [, array \$attributes = ''])**

#### Descripción

---

Le permite generar listas HTML sin ordenar a partir de arrays simples o multidimensionales

#### Parámetros

---

##### **list**

Entradas de la lista

##### **attributes**

Atributos de HTML

#### Valores devueltos

---

Lista desordenada con formato HTML

#### Ejemplo

---

```
$lista = array(
    'verde',
    'rojo',
    'violeta'
);
```

```
$atributos = array(
    'class' => 'objetos',
    'id'    => 'mi_lista'
);

echo ul($lista, $atributos);
```

El código anterior produce:

```
<ul class="objetos" id="mi_lista">
  <li>verde</li>
  <li>rojo</li>
  <li>violeta</li>
</ul>
```

Aquí hay un ejemplo más complejo, usando un array multidimensional:

```
$atributos = array(
    'class' => 'objetos',
    'id'    => 'mi_lista'
);

$lista = array(
    'colores' => array(
        'verde',
        'rojo',
        'violeta'
    ),
    'formas' => array(
        'triángulos',
        'cuadrados',
        'círculos' => array(
            'elipse',
            'óvalo',
            'esfera'
        )
    ),
    'huesos' => array(
        'cráneo',
        'oreja' => array(
            'oído medio' => array(
                'martillo',
                'yunque',
                'estribo'
            ),
            'mano',
            'brazo',
```

```

        'vértebras'
    )
)
);

echo ul($lista, $atributos);

```

El código anterior producirá esto:

```

<ul class="objetos" id="mi_lista">
  <li>colores
    <ul>
      <li>verde</li>
      <li>rojo</li>
      <li>violeta</li>
    </ul>
  </li>
  <li>formas
    <ul>
      <li>triángulos</li>
      <li>cuadrados</li>
      <li>círculos
        <ul>
          <li>elipse</li>
          <li>óvalo</li>
          <li>esfera</li>
        </ul>
      </li>
    </ul>
  </li>
  <li>huesos
    <ul>
      <li>cráneo</li>
      <li>oreja
        <ul>
          <li>oído medio
            <ul>
              <li>martillo</li>
              <li>yunque</li>
              <li>estribo</li>
            </ul>
          </li>
          <li>mano</li>
          <li>brazo</li>
          <li>vértebras</li>
        </ul>
      </li>
    </ul>
  </li>
</ul>

```



```
</li>
</ul>
```

## Inflector Helper

---

### camelize()

**string camelize(string \$str)**

#### Descripción

---

Cambia un string de palabras separadas por espacios o guiones bajos a camel case

#### Parámetros

---

##### **str**

String de entrada

#### Valores devueltos

---

String camelizada

#### Ejemplo

---

```
echo camelize('todo_mi_cariño_para_ti'); // Presenta: ' todoMiCariñoParaTi'
```

### humanize()

**string humanize(string \$str [, string \$separator = '\_'])**

#### Descripción

---

Toma varias palabras separadas por guiones bajos y agrega espacios entre ellas, cada palabra está en mayúscula

#### Parámetros

---

##### **str**

String de entrada

##### **separator**

Separador de entrada

#### Valores devueltos

---

String humanizada

#### Ejemplo

---

```
echo humanize('todo_mi_cariño_para_ti'); // Presenta: 'Todo Mi Cariño Para Ti'
```

Para usar guiones en lugar de guiones bajos:

```
echo humanize('todo-mi-cariño-para-ti', '-'); // Presenta: 'Todo Mi Cariño Para Ti'
```

**is\_countable()**

**bool is\_countable(string \$word)**

---

**Descripción**

Comprueba si la palabra dada tiene una versión plural

---

**Parámetros**

**word**

String de entrada

---

**Valores devueltos**

TRUE si la palabra es contable o FALSE si no

---

**Ejemplo**

```
is_countable('equipment'); // Devuelve: FALSE
```

**plural()**

**string plural(string \$str)**

---

**Descripción**

Cambia una palabra singular a plural

---

**Parámetros**

**str**

String de entrada

---

**Valores devueltos**

Una palabra plural

---

**Ejemplo**

```
echo plural('conejo'); // Presenta: 'conejos'
```

**singular()**

**string singular(string \$str)**

---

**Descripción**

Cambia una palabra plural a singular

---

**Parámetros**

**str**

String de entrada

---

**Valores devueltos**

Una palabra singular

## Ejemplo

---

```
echo singular('conejos'); // Presenta: 'conejo'
```

## underscore()

**string underscore(string \$str)**

### Descripción

---

Toma múltiples palabras separadas por espacios y las subraya

### Parámetros

---

#### str

String de entrada

### Valores devueltos

---

String que contiene guiones bajos en lugar de espacios

## Ejemplo

---

```
echo underscore('todo mi cariño para ti'); // Presenta: 'todo_mi_cariño_para_ti'
```

# Language Helper

---

## lang()

**string lang(string \$line [, string \$for = "" [, array \$attributes = array()])**

### Descripción

---

Esta función devuelve una línea de texto de un archivo de idioma cargado con sintaxis simplificada, puede ser más deseable para los archivos de vista que **CI\_Lang::line()**

### Parámetros

---

#### line

Clave de línea de idioma

#### for

Atributo HTML "**for**" ,**ID** del elemento para el que estamos creando la etiqueta

#### attributes

Cualquier atributo HTML adicional

### Valores devueltos

---

La línea del idioma, en una etiqueta '**label**' de HTML, si el parámetro **\$for** no está vacío

## Ejemplo

---

```
echo lang('language_key');  
// Presenta: la línea de idioma  
  
echo lang('language_key', 'form_item_id', array('class' => 'mi_clase'));  
// Presenta: <label for="form_item_id" class="mi_clase">línea de idioma</label>
```

## Number Helper

---

### byte\_format()

**string** `byte_format(mixed $num [, int $precision = 1])`

#### Descripción

---

Formatea los números como bytes, según el tamaño, y agrega el sufijo apropiado

#### Parámetros

---

##### **num**

Número de bytes

##### **precision**

Precisión del punto flotante

#### Valores devueltos

---

String con los datos formateados

#### Ejemplos

---

```
echo byte_format(456); // Devuelve 456 Bytes  
echo byte_format(4567); // Devuelve 4.5 KB  
echo byte_format(45678); // Devuelve 44.6 KB  
echo byte_format(456789); // Devuelve 447.8 KB  
echo byte_format(3456789); // Devuelve 3.3 MB  
echo byte_format(12345678912345); // Devuelve 1.8 GB  
echo byte_format(123456789123456789); // Devuelve 11,228.3 TB
```

Un segundo parámetro opcional le permite establecer la precisión del resultado:

```
echo byte_format(45678, 2); // Devuelve 44.61 KB
```

#### **Nota**

El texto generado por esta función se encuentra en el siguiente archivo de idioma:

**language/<your\_lang>/number\_lang.php**

# Path Helper

---

## set\_realpath()

**string set\_realpath(string \$path [, bool \$check\_existence = FALSE])**

### Descripción

---

Esta función devolverá una ruta de servidor sin enlaces simbólicos o estructuras de directorio relativas. Un segundo argumento opcional provocará que se active un error si la ruta no se puede resolver

### Parámetros

---

#### path

Ruta

#### check\_existence

Si se debe verificar si la ruta realmente existe

### Valores devueltos

---

Un path absoluto

### Ejemplo

---

```
$file = '/etc/php5/apache2/php.ini';
echo set_realpath($file); // Imprime '/etc/php5/apache2/php.ini'

$non_existent_file = '/path/to/non-exist-file.txt';

// Muestra un error, ya que el path no se puede resolver
echo set_realpath($non_existent_file, TRUE);

// Imprime: '/path/to/non-exist-file.txt'
echo set_realpath($non_existent_file, FALSE);

$directory = '/etc/php5';
echo set_realpath($directory); // Imprime '/etc/php5/'

$non_existent_directory = '/path/to/nowhere';

// Muestra un error, ya que el path no se puede resolver
echo set_realpath($non_existent_directory, TRUE);

// Imprime '/path/to/nowhere'
echo set_realpath($non_existent_directory, FALSE);
```

# Security Helper

---

Security Helper contiene funciones relacionadas con la seguridad

## do\_hash()

### Nota

Esta función está **DESACONSEJADA**. Use la función nativa de PHP: **hash()** en su lugar

## encode\_php\_tags()

**string encode\_php\_tags(string \$str)**

### Descripción

---

Esta es una función de seguridad que convierte etiquetas PHP a entidades

### Nota

**xss\_clean()** lo hace automáticamente, si lo usa

### Parámetros

---

#### str

String de entrada

### Valores devueltos

---

String con formato seguro

### Ejemplo

---

```
$string = encode_php_tags($string);
```

## sanitize\_filename()

**string sanitize\_filename(string \$filename)**

### Descripción

---

Proporciona protección contra el recorrido del directorio

Esta función es un alias para **CI\_Security::sanitize\_filename()**, para obtener más información, consulte la documentación de la **Security Library**

### Parámetros

---

#### filename

Nombre de archivo

## Valores devueltos

---

Nombre de archivo desinfectado

### strip\_image\_tags()

**string strip\_image\_tags(string \$str)**

#### Descripción

---

Esta es una función de seguridad que quitará etiquetas de imagen de un string. Deja la URL de la imagen como texto sin formato

Esta función es un alias para **CI\_Security::strip\_image\_tags()**, para obtener más información, consulte la documentación de la **Security Library**

#### Parámetros

---

##### **str**

String de entrada

#### Valores devueltos

---

La string de entrada sin etiquetas de imagen

#### Ejemplo

---

```
$string = strip_image_tags($string);
```

### xss\_clean()

**string xss\_clean(string \$str [, bool \$is\_image = FALSE])**

#### Descripción

---

Proporciona filtrado de Cross Site Script Hack

Esta función es un alias para **CI\_Input::xss\_clean()**, para obtener más información, consulte la documentación de la **Input Library**

#### Parámetros

---

##### **str**

Datos de entrada

##### **is\_image**

Si se trata de una imagen

#### Valores devueltos

---

Cadena XSS-clean

# String Helper

---

## alternator()

**mixed alternator(mixed \$args)**

### Descripción

---

Permite que dos o más elementos alternen entre sí al ciclar dentro de un bucle

### Parámetros

---

#### args

Cantidad variable de argumentos

### Valores devueltos

---

Cadena(s) alternadas

### Ejemplo

---

```
for ($i = 0; $i < 10; $i++)
{
    echo alternator('string uno', 'string dos');
}
```

Puede agregar tantos parámetros como desee y con cada iteración de su ciclo, se devolverá el siguiente elemento :

```
for ($i = 0; $i < 10; $i++)
{
    echo alternator('uno', 'dos', 'tres', 'cuatro', 'cinco');
}
```

### Nota

Para usar múltiples llamadas separadas a esta función, simplemente llame a la función sin argumentos para reinicializar

## increment\_string()

**string increment\_string(string \$str [, string \$separator = '\_' [, int \$first = 1]])**

### Descripción

---

Incrementa un string añadiéndole un número o aumentando el número, útil para crear "copias" o un archivo o duplicar el contenido de la base de datos que tiene valores únicos o slugs

### Parámetros

---

#### str

String de entrada

#### separator

Separador para agregar al número

#### first

Número inicial



## Valores devueltos

---

Un string incrementado

## Ejemplo

---

```
echo increment_string('archivo', '_'); // "archivo_1"
echo increment_string('archivo', '-', 2); // "archivo-2"
echo increment_string('archivo_4'); // "archivo_5"
```

## quotes\_to\_entities()

**string** `quotes_to_entities(string $str)`

## Descripción

---

Convierte las comillas simples y dobles en una cadena a las entidades HTML correspondientes

## Parámetros

---

### str

String de entrada

## Valores devueltos

---

Cadena con comillas convertidas a entidades HTML

## Ejemplo

---

```
$string = "Joe's \"dinner\"";
$string = quotes_to_entities($string); //resultado: "Joe&#39;s &quot;dinner&quot;"
```

## random\_string()

**string** `random_string([ string $type = 'alnum' [, int $len = 8])`

## Descripción

---

Genera un string aleatorio según el tipo y la longitud que especifique, útil para crear contraseñas o generar hashes aleatorios

## Parámetros

---

### type

Tipo de aleatorización

Las siguientes opciones están disponibles:

- **alpha**: Un string con letras mayúsculas y minúsculas solamente
- **alnum**: Un string alfanumérico con mayúsculas y minúsculas
- **basic**: Un número aleatorio basado en **mt\_rand()**
- **numeric**: Un string numérico
- **nozero**: Un string numérico sin ceros
- **md5**: Un número aleatorio encriptado basado en **md5()** (longitud fija de 32)
- **sha1**: Un número aleatorio encriptado basado en **sha1()** (longitud fija de 40)

### len

Longitud del string de salida

## Valores devueltos

---

Un string aleatorio

## Ejemplo

---

```
echo random_string('alnum', 16);
```

## reduce\_double\_slashes()

**string reduce\_double\_slashes(string \$str)**

### Descripción

---

Convierte barras diagonales dobles en un string de una sola barra, excepto las que se encuentran en los prefijos de protocolo URL (por ejemplo, http: //)

### Parámetros

---

#### str

String de entrada

### Valores devueltos

---

Un string con barras normalizadas

## Ejemplo

---

```
$string = "http://ejemplo.com//index.php";  
  
echo reduce_double_slashes($string);  
// resultado: "http://ejemplo.com/index.php"
```

## reduce\_multiples()

**string reduce\_multiples(string \$str [, string \$character = " " [, bool \$trim = FALSE]])**

### Descripción

---

Reduce múltiples instancias de un carácter particular que ocurren uno detrás de otro

### Parámetros

---

#### str

Texto para buscar

#### character

Carácter para reducir

#### trim

Si se establece en TRUE, eliminará las apariciones del carácter especificado al principio y al final del string

### Valores devueltos

---

String reducida

## Ejemplo

---

```
$string = "Carmen, María,, Raquel, Helena";  
$string = reduce_multiples($string,","); // Produce: "Carmen, María, Raquel, Helena"
```

Eliminar las apariciones del carácter al principio y al final del string :

```
$string = ",Carmen, María,, Raquel, Helena,";  
$string = reduce_multiples($string, " , ", TRUE); //Produce: "Carmen, María, Raquel, Helena"
```

## repeater()

### Nota

Esta función esta **DESACONSEJADA**, use la función nativa de PHP: **str\_repeat()** en su lugar

## strip\_quotes()

**string strip\_quotes(string \$str)**

### Descripción

Elimina comillas simples y dobles de un string

### Parámetros

#### str

String de entrada

### Valores devueltos

Cadena sin las comillas

### Ejemplo

```
$string = "Gallina's \"ponedoras\"";  
$string = strip_quotes($string); // Produce: "Gallinas ponedoras"
```

## strip\_slashes()

**mixed strip\_slashes(mixed \$data)**

### Descripción

Elimina cualquier barra de un array de strings

### Parámetros

#### data

Cadena de entrada o un array de strings

### Valores devueltos

String(s) sin las barras

## Ejemplo

---

```
$str = array(
    'pregunta' => 'Tu nombre es S\'cooby-Doo?',
    'respuesta' => 'No, mi nombre es R\'antanplan.'
);

$str = strip_slashes($str);
```

Lo anterior devolverá la siguiente array:

```
array(
    'pregunta' => "Tu nombre es S'cooby-Doo?",
    'respuesta' => "No, mi nombre es R'antanplan."
);
```

### Nota

Por razones históricas, esta función también aceptará y manejará las entradas de string, sin embargo, esto lo hace solo como un alias de **stripslashes()**

## trim\_slashes()

### Nota

Esta función está **DESACONSEJADA**, use la función nativa de PHP: **trim()** en su lugar: **trim(\$str, '/')**;

## Text Helper

---

## ascii\_to\_entities()

**string** **ascii\_to\_entities** (**string** \$str)

### Descripción

---

Convierte valores ASCII en entidades de caracteres, incluidos los caracteres ASCII y MS Word altos que pueden causar problemas cuando se utilizan en una página web, de modo que se pueden mostrar de forma coherente independientemente de la configuración del navegador o almacenarse de forma fiable en una base de datos

Existe cierta dependencia de los juegos de caracteres soportados por su servidor, por lo que puede no ser 100% confiable en todos los casos, pero en su mayor parte debe identificar correctamente los caracteres fuera del rango normal (como los caracteres acentuados)

### Parámetros

---

#### str

String de entrada

## Valores devueltos

---

Un string con valores ASCII convertidos en entidades

## Ejemplo

---

```
$string = ascii_to_entities($string);
```

## character\_limiter()

**string character\_limiter(string \$str [, int \$n = 500 [, string \$end\_char = '&#8230;']])**

### Descripción

---

Trunca un string a la cantidad de caracteres especificados, mantiene la integridad de las palabras, por lo que el recuento de caracteres puede ser un poco más o menos de lo que especifique

### Parámetros

---

#### **str**

String de entrada

#### **n**

Número de caracteres

#### **end\_char**

Carácter final (usualmente una elipsis)

## Valores devueltos

---

Cadena de caracteres limitados

## Ejemplo

---

```
$string = "Aquí hay una bonita string de texto que consta de doce palabras.";
$string = character_limiter($string, 20);
// Devuelve: Aquí hay una bonita..
```

El tercer parámetro es un sufijo opcional agregado al string; si no se declara, se usa una elipse

### **Nota**

Si necesita truncar a un número exacto de caracteres, consulte la función **ellipsize()** a continuación

## convert\_accented\_characters()

**string convert\_accented\_characters(string \$str)**

### Descripción

---

Transcribe los caracteres ASCII altos a equivalentes ASCII bajos, útil cuando los caracteres no ingleses deben usarse donde solo se usan con seguridad los caracteres ASCII estándar, por ejemplo, en las URL

### Parámetros

---

#### **str**

String de entrada

## Valores devueltos

---

Un string con caracteres acentuados convertidos

## Ejemplo

---

```
$string = convert_accented_characters($string);
```

## Nota

Esta función utiliza un archivo de configuración complementario **application/config/foreign\_chars.php** para definir el conjunto de la transliteración

## ellipsize()

**string ellipsize(string \$str, int \$max\_length [, mixed \$position = 1 [, string \$ellipsis = '&hellip;'] ]])**

## Descripción

---

Esta función quitará las etiquetas de una string, la dividirá en una longitud máxima definida e insertará puntos suspensivos

## Parámetros

---

### str

String de entrada

### max\_length

Límite de longitud de string

### position

Posición para dividir en (int o float), el tercer parámetro es donde en la string los puntos suspensivos deberían aparecer de 0 a 1, de izquierda a derecha, por ejemplo. un valor de 1 colocará los puntos suspensivos a la derecha de la string, 5 en el medio y 0 a la izquierda

### ellipsis

Qué usar como puntos suspensivos, por defecto inserta **'&hellip;'**

## Valores devueltos

---

String elíptica

## Ejemplo

---

```
$str = 'en_un_lugar_de_la_mancha_de_cuyo_nombre.jpg';  
echo ellipsize($str, 30, .5);  
  
// Produce: en_un_lugar_de_...cuyo_nombre.jpg
```

## highlight\_code()

**string highlight\_code(string \$str)**

## Descripción

---

Colorea una string de código (PHP, HTML, etc.)

#### Parámetros

---

##### **str**

String de entrada

#### Valores devueltos

---

String con código resaltado a través de HTML

#### Ejemplo

---

```
$string = highlight_code($string);
```

La función utiliza la función **highlight\_string()** de PHP, por lo que los colores utilizados son los especificados en su archivo **php.ini**

#### highlight\_phrase()

```
string highlight_phrase(string $str, string $phrase [, string $tag_open = '<mark>' [, string $tag_close = '</mark>']])
```

#### Descripción

---

Resaltará una frase dentro de una string de texto

#### Parámetros

---

##### **str**

String de entrada

##### **phrase**

Frase a resaltar

##### **tag\_open**

Etiqueta de apertura utilizada para resaltar

##### **tag\_close**

Etiqueta de cierre para resaltar

#### Valores devueltos

---

Cadena con una frase resaltada en HTML

#### Ejemplo

---

```
$string = "En un lugar de la Mancha de cuyo nombre";  
echo highlight_phrase($string, "Mancha", '<span style="color:#990000;">', '</span>');
```

El código anterior imprime:

```
En un lugar de la<span style="color:#990000;">Mancha</span>de cuyo nombre
```

#### **Nota**

Esta función solía usar la etiqueta **<strong>** de forma predeterminada. Los navegadores antiguos pueden no ser compatibles con la nueva etiqueta de marca HTML5, por lo que se recomienda que inserte el siguiente código CSS en su hoja de estilos si necesita admitir dichos navegadores:

```
mark {
  background: #ff0;
  color: #000;
};
```

## word\_censor()

**string word\_censor(string \$str, array \$censored [, string \$replacement = ''])**

### Descripción

---

Le permite censurar palabras dentro de un string de texto, el primer parámetro contendrá el string original

### Parámetros

---

#### **str**

String de entrada

#### **censored**

Un array de palabras que usted no permite

#### **replacement**

Opcional, puede contener un valor de reemplazo para las palabras, si no se especifica, se reemplazan con signos de libra: #####

### Valores devueltos

---

String censurada

### Ejemplo

---

```
$párrafo = 'En un lugar de la Mancha de cuyo nombre';
$censurado = array('Monegros', 'Mancha', 'Barcelona', 'Madrid');
$string = word_censor($párrafo, $censurado, 'Galaxia');

echo $string; // Produce: En un lugar de la Galaxia de cuyo nombre
```

## word\_limiter()

**string word\_limiter(string \$str [, int \$limit = 100 [, string \$end\_char = '&#8230;']])**

### Descripción

---

Trunca una string a la cantidad de palabras especificadas

### Parámetros

---

#### **str**

String de entrada

#### **limit**

Límite



## end\_char

Carácter final, por defecto puntos suspensivos ...

### Valores devueltos

---

String truncada

### Ejemplo

---

```
$string = "En un lugar de la Mancha de cuyo nombre";  
$string = word_limiter($string, 4);  
echo $string; // Produce: En un lugar de...
```

## word\_wrap()

**string word\_wrap(string \$str [, int \$charlim = 76])**

### Descripción

---

Divide un texto en la porciones del número de caracteres especificado, ajustado a palabras completas

### Parámetros

---

#### str

String de entrada

#### charlim

Límite de caracteres

### Valores devueltos

---

String dividido ajustado al número indicado y respetando palabras completas

### Ejemplo

---

```
$string = "En un lugar de la Mancha de cuyo nombre";  
echo word_wrap($string, 12);  
  
// Produce:  
// En un lugar  
// de la Mancha  
// de cuyo  
// nombre
```

# Typography Helper

---

## auto\_typography()

**string auto\_typography(string \$str [, bool \$reduce\_linebreaks = FALSE])**

### Descripción

---

Formatea texto para que sea HTML semántica y tipográficamente correcto, esta función es un alias para **CI\_Typography::auto\_typography()**, Puede obtener más información, consultando la **Typography Library**

### Parámetros

---

#### **str**

String de entrada

#### **reduce\_linebreaks**

Si se deben reducir las instancias múltiples de líneas dobles nuevas a dos

### Valores devueltos

---

String tipográfica con formato HTML

### Ejemplo

---

```
$string = auto_typography($string);
```

### **Nota**

El formateo tipográfico puede usar intensamente al procesador, particularmente si tiene un mucho contenido para formatear, si elige usar esta función, debería considerar cachear sus páginas

## entity\_decode()

**string entity\_decode(string \$str, string \$charset = NULL)**

### Descripción

---

Esta función es un alias para **CI\_Security::entity\_decode()**, para obtener más información, consulte la documentación de la **Security Library**

### Parámetros

---

#### **str**

String de entrada

#### **charset**

Conjunto de caracteres

### Valores devueltos

---

String entidades HTML decodificadas

`nl2br_except_pre()`

**string nl2br\_except\_pre(string \$str)**

#### Descripción

---

Convierte caracteres de nueva línea a etiquetas `<br />` a menos que éstas aparezcan dentro de etiquetas `<pre>`, esta función es idéntica a la función nativa `nl2br()` PHP, excepto que ignora las etiquetas `<pre>`

#### Parámetros

---

##### **str**

String de entrada

#### Valores devueltos

---

String con saltos de línea con formato HTML

#### Ejemplo

---

```
$string = nl2br_except_pre($string);
```

## URL Helper

---

`anchor()`

**string anchor(string \$uri = "", string \$title = "", mixed \$attributes = "")**

#### Descripción

---

Crea un enlace de anclaje HTML estándar basado en la URL de su sitio

#### Parámetros

---

##### **uri**

String URI, puede contener cualquier segmento que desee adjuntar a la URL, al igual que con la función `site_url()` anterior, los segmentos pueden ser un string o un array

##### **title**

Título de anclaje, si lo deja en blanco, se usará la URL

##### **attributes**

Atributos de HTML, un string simple o un array asociativo con los atributos que le gustaría agregar al enlace

#### Valores devueltos

---

Hipervínculo HTML (etiqueta de anclaje)

#### **Nota**

Si está creando enlaces que son internos a su aplicación, no incluya la URL base (`http://...`), esto se agregará automáticamente a partir de la información especificada en su archivo de configuración, incluya solo los segmentos de URI que desee adjuntar a la URL

#### Ejemplo

---

```
echo anchor('news/local/123', 'La noticia', 'title="Título de la noticia"');
// Imprime: <a href="http://ejemplo.com/index.php/news/local/123"
// title="Título de la noticia">La noticia</a>

echo anchor('news/local/123', 'La noticia', array('title' => 'La mejor noticia!'));
// Imprime: <a href="http://ejemplo.com/index.php/news/local/123"
// title="La mejor noticia!">La noticia</a>

echo anchor('', 'Clic aquí');
// Imprime: <a href="http://ejemplo.com">Clic aquí</a>
```

## anchor\_popup()

**string anchor\_popup(string \$uri = "", string \$title = "", mixed \$attributes = FALSE)**

### Descripción

---

Casi idéntica a la función **anchor()** excepto que abre la URL en una nueva ventana, puede especificar atributos de ventana de JavaScript en el tercer parámetro para controlar cómo se abre la ventana, si el tercer parámetro no está configurado, simplemente abrirá una nueva ventana con la configuración del navegador

### Parámetros

---

#### uri

String URI

#### title

Título de anclaje

#### attributes

Atributos de HTML

### Valores devueltos

---

Hipervínculo emergente

### Ejemplo

---

```
$atts = array(
    'width'      => 800,
    'height'     => 600,
    'scrollbars' => 'yes',
    'status'     => 'yes',
    'resizable'  => 'yes',
    'screenx'    => 0,
    'screeny'    => 0,
    'window_name' => '_blank'
);

echo anchor_popup('news/local/123', 'Click aquí!', $atts);
```

### Nota

- Los atributos anteriores son los valores predeterminados de la función, por lo que solo necesita establecer los que son diferentes de lo que necesita. Si desea que la función use todos sus valores predeterminados simplemente pase un array vacío en el tercer parámetro:

```
echo anchor_popup('news/local/123', 'Click aquí!', array());
```

- **window\_name** no es realmente un atributo, sino un argumento para el método JavaScript **window.open()** <http://www.w3schools.com/jsref/met\_win\_open.asp>, que acepta un nombre de ventana o un destino de ventana
- Cualquier otro atributo distinto de los enumerados anteriormente se analizará como un atributo HTML para la etiqueta de anclaje

## auto\_link()

**string auto\_link(string \$str, string \$type = 'both', bool \$popup = FALSE)**

### Descripción

---

Convierte automáticamente las URL y las direcciones de correo electrónico contenidas en un string en enlaces

### Parámetros

---

#### str

String de entrada

#### type

Tipo de enlace ('**email**', '**url**' o '**both**'), determina si las URL y los correos electrónicos se convierten o solo uno o el otro, si el parámetro no está especificado el comportamiento predeterminado es '**both**', los enlaces de correos electrónico están codificados como **safe\_mailto()**

#### popup

Si se deben crear enlaces emergentes, determina si los enlaces se muestran en una nueva ventana

### Valores devueltos

---

String de enlace

### Ejemplo

---

```
$string = auto_link($string);
```

Convierte solo URLs:

```
$string = auto_link($string, 'url');
```

Convierte solo direcciones de correo electrónico:

```
$string = auto_link($string, 'email');
```

El tercer parámetro determina si los enlaces se muestran en una nueva ventana :

```
$string = auto_link($string, 'both', TRUE);
```

## base\_url()

**string base\_url(string \$uri = "", string \$protocol = NULL)**

### Descripción

Devuelve la URL base de su sitio, como se especifica en su archivo de configuración, esta función devuelve lo mismo que **site\_url()**, sin que se agregue **index\_page** o **url\_suffix**, también como en **site\_url()**, puede suministrar segmentos como un string o un array

Esta función es un alias para **CI\_Config::base\_url()**, para obtener más información, consulte la **Config Library**

### Parámetros

#### uri

String URI

#### protocol

Protocolo, p. ej. 'Http' o 'https'

### Valores devueltos

URL base

### Ejemplo

```
echo base_url();
```

Con string:

```
echo base_url("blog/post/123");
```

El ejemplo anterior devolvería algo así como: **http://ejemplo.com/blog/post/123**

Esto es útil porque a diferencia de **site\_url()**, puede suministrar un string a un archivo, como una imagen o una hoja de estilo :

```
echo base_url("images/icons/edit.png");
```

Esto daría algo como: **http://ejemplo.com/images/icons/edit.png**

## current\_url()

**string current\_url()**

### Descripción

Devuelve la URL completa, incluidos los segmentos, de la página que se está viendo actualmente

### Nota

Llamar a esta función es lo mismo que hacer esto: **site\_url(uri\_string());**

### Valores devueltos

La URL actual

`index_page()`

**mixed** `index_page()`

---

#### Descripción

Devuelve su sitio **index\_page**, como se especifica en su archivo de configuración

---

#### Valores devueltos

Valor de 'index\_page'

---

#### Ejemplo

```
echo index_page();
```

`mailto()`

**string** `mailto(string $email, string $title = "", mixed $attributes = "")`

---

#### Descripción

Crea un enlace de correo electrónico HTML estándar "**mail to**"

---

#### Parámetros

##### **email**

Dirección de correo electrónico

##### **title**

Título de anclaje

##### **attributes**

Atributos de HTML

---

#### Valores devueltos

Un hipervínculo "**mail to**"

---

#### Ejemplo

```
echo mailto('me@my-site.com', 'Click Here to Contact Me');
```

Al igual que con **anchor()**, puede establecer atributos utilizando el tercer parámetro:

```
$attributes = array('title' => 'Mail me');
echo mailto('me@my-site.com', 'Contact Me', $attributes);
```

`prep_url()`

**string** `prep_url(string $str = "")`

---

#### Descripción

Esta función agregará **http://** en caso de que falte un prefijo de protocolo de una URL

---

#### Parámetros

##### **str**

String URL

## Valores devueltos

---

String de URL con prefijo de protocolo

## Ejemplo

---

Pase la string de URL a la función de esta forma:

```
$url = prep_url('example.com');
```

## redirect()

**void redirect(string \$uri = '', string \$method = 'auto', string \$code = NULL)**

### Descripción

---

Hace una "redirección de encabezado" a la URI especificada, si especifica la URL completa del sitio, se construye ese enlace, pero para enlaces locales, simplemente proveer los segmentos URI al controlador que quiere direccionar para crear el enlace, la función construye la URL basándose en los valores del archivo de configuración

### Parámetros

---

#### uri

String URI

#### method

Método de re direccionamiento ('auto', 'location' o 'refresh'), el valor predeterminado es 'auto', que intentará elegir de forma inteligente el método en función del entorno del servidor, 'location' location es más rápida pero menos confiable en los servidores IIS

#### code

Código de respuesta HTTP (generalmente 302 o 303)

### valores devueltos

---

void

## Ejemplo

---

```
if ($logged_in == FALSE)
{
    redirect('/login/form/');
}

// with 301 redirect
redirect('/article/13', 'location', 301);
```

## Nota



- Para que esta función funcione, debe usarse antes de que se genere algo en el navegador, ya que utiliza los encabezados del servidor
- Para obtener un control más detallado de los encabezados, debe utilizar el método **set\_header()** de la **Output Library**
- Para los usuarios de IIS: si oculta el encabezado HTTP del servidor, el método automático no detectará IIS; en ese caso, se recomienda utilizar explícitamente el método de actualización
- Cuando se usa el método de ubicación, se seleccionará automáticamente un código de estado HTTP de 303 cuando se acceda a la página por medio de POST y se use HTTP/1.1

### **Importante**

Esta función terminará la ejecución del script

### **safe\_mailto()**

```
string safe_mailto(string $email, string $title = "", mixed $attributes = "")
```

#### **Descripción**

---

Idéntico a la función **mailto()**, excepto que escribe una versión ofuscada de la etiqueta mailto utilizando números ordinales escritos con JavaScript para evitar que los bots de correo basura capturen la dirección de correo electrónico

#### **Parámetros**

---

##### **email**

Dirección de correo electrónico

##### **title**

Título de anclaje

##### **attributes**

Atributos de HTML

#### **Valores devueltos**

---

Un hipervínculo de "**mail to**" ofuscado

### **site\_url()**

```
string site_url([ string $uri = "" [, string $protocol = NULL]])
```

#### **Descripción**

---

Devuelve la URL de su sitio, como se especifica en su archivo de configuración, el archivo **index.php** (o el que haya establecido como el **index\_page** en su archivo de configuración) se agregará a la URL, al igual que cualquier segmento URI que pase a la función, más el **url\_suffix** como se establece en su archivo de configuración

Se le recomienda utilizar esta función cuando necesite generar una URL local para que sus páginas se vuelvan más portables en caso de que su URL cambie, los segmentos se pueden pasar opcionalmente a la función como un string o un

array, esta función es un alias para **CI\_Config :: site\_url ()**, para obtener más información, consulte la documentación de **Config Library**

#### Parámetros

---

##### uri

String URI

##### protocol

Protocolo, p. ej. 'Http' o 'https'

#### Valores devueltos

---

Sitio URL

#### Ejemplo

---

Con string:

```
echo site_url('news/local/123');

// Devuelve algo así como: http://ejemplo.com/index.php/news/local/123
```

Con array :

```
$segmentos = array('news', 'local', '123');
echo site_url($segmentos);
```

#### uri\_string()

### string uri\_string()

#### Descripción

---

Devuelve los segmentos de URI de cualquier página que contenga esta función, esta función es un alias para **CI\_Config::uri\_string()**, para obtener más información, consulte la documentación de la **Config Library**

#### Valores devueltos

---

String URI

#### Ejemplo

---

Si la URL fue esta:

```
http://ejemplo.com/blog/opiniones/123
```

La función devolvería:

```
blog/opiniones/123
```

#### url\_title()

### string url\_title(string \$str, string \$separator = '-', bool \$lowercase = FALSE)

#### Descripción

---

Toma un string como entrada y crea un string URL amigable para los humanos, es útil, por ejemplo, si tiene un blog en el que le gustaría usar el título de sus entradas en la URL

#### Parámetros

---

##### **str**

String de entrada

##### **separator**

Separador de palabras

##### **lowercase**

Si se debe transformar el string de salida a minúsculas

#### Valores devueltos

---

String con formato de URL

#### Ejemplo

---

```
$título = "Más vale prevenir que curar";
$url_título = url_title($título);

// Produce: Más-vale-prevenir-que-curar

$título = "Más/vale/prevenir/?%que#curar";
$url_título = url_title($título);

// Produce: Másvaleprevenirquecurar
```

El segundo parámetro determina el delimitador de palabra, por defecto, se usan guiones, las opciones preferidas son: - (guión) o \_ (guión bajo):

```
$título = "Más vale prevenir que curar";
$url_título = url_title($título, 'underscore');

// Produce: Más_vale_prevenir_que_curar
```

#### **Nota**

El uso anterior de 'guion' y 'subrayado' como el segundo parámetro está **DESACONSEJADO**

El tercer parámetro determina si los caracteres minúsculos son forzados o no, por defecto no lo son:

```
$título = "Qué pasa con CSS?";
$url_title = url_title($título, 'underscore', TRUE);

// Produce: qué_pasa_con_css
```

## XML Helper

---

## xml\_convert()

**string xml\_convert(string \$str [, bool \$protect\_all = FALSE])**

### Descripción

---

Toma un string como entrada y convierte los siguientes caracteres XML en entidades

Ampersands: **&**

Menos que y más que caracteres: **<>**

Comillas simples y dobles: **'"**

Guiones: **-**

Esta función ignora los Ampersands si son parte de entidades de caracteres numerados existentes, e.g. e.g. `&#123;`

### Parámetros

---

#### **str**

La string de texto para convertir

#### **protect\_all**

Si se debe proteger todo el contenido que se parece a una entidad potencial en lugar de solo las entidades numeradas, p. `&foo;`

### Valores devueltos

---

String convertida XML

### Ejemplo

---

```
$string = '<p>Aquí hay un párrafo & una entidad (&#123;).</p>';  
$string = xml_convert($string);  
echo $string;
```

Produce:

```
&lt;p&gt;Aquí hay un párrafo &amp; una entidad ({}).&lt;/p&gt;
```

Documento alojado en: [www.codeigniter.xyz](http://www.codeigniter.xyz)