* [Welcome to CodeIgniter](http://docs.google.com/general/welcome.html)
* [Installation Instructions](http://docs.google.com/installation/index.html)
  + [Downloading CodeIgniter](http://docs.google.com/installation/downloads.html)
  + [Installation Instructions](http://docs.google.com/installation/index.html)
  + [Upgrading From a Previous Version](http://docs.google.com/installation/upgrading.html)
  + [Troubleshooting](http://docs.google.com/installation/troubleshooting.html)
* [CodeIgniter Overview](http://docs.google.com/overview/index.html)
  + [Getting Started](http://docs.google.com/overview/getting_started.html)
  + [CodeIgniter at a Glance](http://docs.google.com/overview/at_a_glance.html)
  + [Supported Features](http://docs.google.com/overview/features.html)
  + [Application Flow Chart](http://docs.google.com/overview/appflow.html)
  + [Model-View-Controller](http://docs.google.com/overview/mvc.html)
  + [Architectural Goals](http://docs.google.com/overview/goals.html)
* [Tutorial](http://docs.google.com/tutorial/index.html)
  + [Static pages](http://docs.google.com/tutorial/static_pages.html)
  + [News section](http://docs.google.com/tutorial/news_section.html)
  + [Create news items](http://docs.google.com/tutorial/create_news_items.html)
  + [Conclusion](http://docs.google.com/tutorial/conclusion.html)
* [Contributing to CodeIgniter](http://docs.google.com/contributing/index.html)
  + [Writing CodeIgniter Documentation](http://docs.google.com/documentation/index.html)
  + [Developer’s Certificate of Origin 1.1](http://docs.google.com/DCO.html)
* [General Topics](http://docs.google.com/general/index.html)
  + [CodeIgniter URLs](http://docs.google.com/general/urls.html)
  + [Controllers](http://docs.google.com/general/controllers.html)
  + [Reserved Names](http://docs.google.com/general/reserved_names.html)
  + [Views](http://docs.google.com/general/views.html)
  + [Models](http://docs.google.com/general/models.html)
  + [Helpers](http://docs.google.com/general/helpers.html)
  + [Using CodeIgniter Libraries](http://docs.google.com/general/libraries.html)
  + [Creating Libraries](http://docs.google.com/general/creating_libraries.html)
  + [Using CodeIgniter Drivers](http://docs.google.com/general/drivers.html)
  + [Creating Drivers](http://docs.google.com/general/creating_drivers.html)
  + [Creating Core System Classes](http://docs.google.com/general/core_classes.html)
  + [Creating Ancillary Classes](http://docs.google.com/general/ancillary_classes.html)
  + [Hooks - Extending the Framework Core](http://docs.google.com/general/hooks.html)
  + [Auto-loading Resources](http://docs.google.com/general/autoloader.html)
  + [Common Functions](http://docs.google.com/general/common_functions.html)
  + [Compatibility Functions](http://docs.google.com/general/compatibility_functions.html)
  + [URI Routing](http://docs.google.com/general/routing.html)
  + [Error Handling](http://docs.google.com/general/errors.html)
  + [Caching](http://docs.google.com/general/caching.html)
  + [Profiling Your Application](http://docs.google.com/general/profiling.html)
  + [Running via the CLI](http://docs.google.com/general/cli.html)
  + [Managing your Applications](http://docs.google.com/general/managing_apps.html)
  + [Handling Multiple Environments](http://docs.google.com/general/environments.html)
  + [Alternate PHP Syntax for View Files](http://docs.google.com/general/alternative_php.html)
  + [Security](http://docs.google.com/general/security.html)
  + [PHP Style Guide](http://docs.google.com/general/styleguide.html)
* [Libraries](http://docs.google.com/index.html)
  + [Benchmarking Class](http://docs.google.com/benchmark.html)
  + [Caching Driver](http://docs.google.com/caching.html)
  + [Calendaring Class](http://docs.google.com/calendar.html)
  + [Shopping Cart Class](http://docs.google.com/cart.html)
  + [Config Class](http://docs.google.com/config.html)
  + [Email Class](http://docs.google.com/email.html)
  + [Encrypt Class](http://docs.google.com/encrypt.html)
  + [Encryption Library](http://docs.google.com/encryption.html)
  + [File Uploading Class](http://docs.google.com/file_uploading.html)
  + [Form Validation](http://docs.google.com/form_validation.html)
  + [FTP Class](http://docs.google.com/ftp.html)
  + [Image Manipulation Class](http://docs.google.com/image_lib.html)
  + [Input Class](http://docs.google.com/input.html)
  + [Javascript Class](http://docs.google.com/javascript.html)
  + [Language Class](http://docs.google.com/language.html)
  + [Loader Class](http://docs.google.com/loader.html)
  + [Migrations Class](http://docs.google.com/migration.html)
  + [Output Class](http://docs.google.com/output.html)
  + [Pagination Class](http://docs.google.com/pagination.html)
  + [Template Parser Class](http://docs.google.com/parser.html)
  + [Security Class](http://docs.google.com/security.html)
  + [Session Library](http://docs.google.com/sessions.html)
  + HTML Table Class
  + [Trackback Class](http://docs.google.com/trackback.html)
  + [Typography Class](http://docs.google.com/typography.html)
  + [Unit Testing Class](http://docs.google.com/unit_testing.html)
  + [URI Class](http://docs.google.com/uri.html)
  + [User Agent Class](http://docs.google.com/user_agent.html)
  + [XML-RPC and XML-RPC Server Classes](http://docs.google.com/xmlrpc.html)
  + [Zip Encoding Class](http://docs.google.com/zip.html)
* [Database Reference](http://docs.google.com/database/index.html)
  + [Quick Start: Usage Examples](http://docs.google.com/database/examples.html)
  + [Database Configuration](http://docs.google.com/database/configuration.html)
  + [Connecting to a Database](http://docs.google.com/database/connecting.html)
  + [Running Queries](http://docs.google.com/database/queries.html)
  + [Generating Query Results](http://docs.google.com/database/results.html)
  + [Query Helper Functions](http://docs.google.com/database/helpers.html)
  + [Query Builder Class](http://docs.google.com/database/query_builder.html)
  + [Transactions](http://docs.google.com/database/transactions.html)
  + [Getting MetaData](http://docs.google.com/database/metadata.html)
  + [Custom Function Calls](http://docs.google.com/database/call_function.html)
  + [Query Caching](http://docs.google.com/database/caching.html)
  + [Database Manipulation with Database Forge](http://docs.google.com/database/forge.html)
  + [Database Utilities Class](http://docs.google.com/database/utilities.html)
  + [Database Driver Reference](http://docs.google.com/database/db_driver_reference.html)
* [Helpers](http://docs.google.com/helpers/index.html)
  + [Array Helper](http://docs.google.com/helpers/array_helper.html)
  + [CAPTCHA Helper](http://docs.google.com/helpers/captcha_helper.html)
  + [Cookie Helper](http://docs.google.com/helpers/cookie_helper.html)
  + [Date Helper](http://docs.google.com/helpers/date_helper.html)
  + [Directory Helper](http://docs.google.com/helpers/directory_helper.html)
  + [Download Helper](http://docs.google.com/helpers/download_helper.html)
  + [Email Helper](http://docs.google.com/helpers/email_helper.html)
  + [File Helper](http://docs.google.com/helpers/file_helper.html)
  + [Form Helper](http://docs.google.com/helpers/form_helper.html)
  + [HTML Helper](http://docs.google.com/helpers/html_helper.html)
  + [Inflector Helper](http://docs.google.com/helpers/inflector_helper.html)
  + [Language Helper](http://docs.google.com/helpers/language_helper.html)
  + [Number Helper](http://docs.google.com/helpers/number_helper.html)
  + [Path Helper](http://docs.google.com/helpers/path_helper.html)
  + [Security Helper](http://docs.google.com/helpers/security_helper.html)
  + [Smiley Helper](http://docs.google.com/helpers/smiley_helper.html)
  + [String Helper](http://docs.google.com/helpers/string_helper.html)
  + [Text Helper](http://docs.google.com/helpers/text_helper.html)
  + [Typography Helper](http://docs.google.com/helpers/typography_helper.html)
  + [URL Helper](http://docs.google.com/helpers/url_helper.html)
  + [XML Helper](http://docs.google.com/helpers/xml_helper.html)

Toggle Table of Contents

[CodeIgniter](http://docs.google.com/index.html)

* [Welcome to CodeIgniter](http://docs.google.com/general/welcome.html)
* [Installation Instructions](http://docs.google.com/installation/index.html)
  + [Downloading CodeIgniter](http://docs.google.com/installation/downloads.html)
  + [Installation Instructions](http://docs.google.com/installation/index.html)
  + [Upgrading From a Previous Version](http://docs.google.com/installation/upgrading.html)
  + [Troubleshooting](http://docs.google.com/installation/troubleshooting.html)
* [CodeIgniter Overview](http://docs.google.com/overview/index.html)
  + [Getting Started](http://docs.google.com/overview/getting_started.html)
  + [CodeIgniter at a Glance](http://docs.google.com/overview/at_a_glance.html)
  + [Supported Features](http://docs.google.com/overview/features.html)
  + [Application Flow Chart](http://docs.google.com/overview/appflow.html)
  + [Model-View-Controller](http://docs.google.com/overview/mvc.html)
  + [Architectural Goals](http://docs.google.com/overview/goals.html)
* [Tutorial](http://docs.google.com/tutorial/index.html)
  + [Static pages](http://docs.google.com/tutorial/static_pages.html)
  + [News section](http://docs.google.com/tutorial/news_section.html)
  + [Create news items](http://docs.google.com/tutorial/create_news_items.html)
  + [Conclusion](http://docs.google.com/tutorial/conclusion.html)
* [Contributing to CodeIgniter](http://docs.google.com/contributing/index.html)
  + [Writing CodeIgniter Documentation](http://docs.google.com/documentation/index.html)
  + [Developer’s Certificate of Origin 1.1](http://docs.google.com/DCO.html)
* [General Topics](http://docs.google.com/general/index.html)
  + [CodeIgniter URLs](http://docs.google.com/general/urls.html)
  + [Controllers](http://docs.google.com/general/controllers.html)
  + [Reserved Names](http://docs.google.com/general/reserved_names.html)
  + [Views](http://docs.google.com/general/views.html)
  + [Models](http://docs.google.com/general/models.html)
  + [Helpers](http://docs.google.com/general/helpers.html)
  + [Using CodeIgniter Libraries](http://docs.google.com/general/libraries.html)
  + [Creating Libraries](http://docs.google.com/general/creating_libraries.html)
  + [Using CodeIgniter Drivers](http://docs.google.com/general/drivers.html)
  + [Creating Drivers](http://docs.google.com/general/creating_drivers.html)
  + [Creating Core System Classes](http://docs.google.com/general/core_classes.html)
  + [Creating Ancillary Classes](http://docs.google.com/general/ancillary_classes.html)
  + [Hooks - Extending the Framework Core](http://docs.google.com/general/hooks.html)
  + [Auto-loading Resources](http://docs.google.com/general/autoloader.html)
  + [Common Functions](http://docs.google.com/general/common_functions.html)
  + [Compatibility Functions](http://docs.google.com/general/compatibility_functions.html)
  + [URI Routing](http://docs.google.com/general/routing.html)
  + [Error Handling](http://docs.google.com/general/errors.html)
  + [Caching](http://docs.google.com/general/caching.html)
  + [Profiling Your Application](http://docs.google.com/general/profiling.html)
  + [Running via the CLI](http://docs.google.com/general/cli.html)
  + [Managing your Applications](http://docs.google.com/general/managing_apps.html)
  + [Handling Multiple Environments](http://docs.google.com/general/environments.html)
  + [Alternate PHP Syntax for View Files](http://docs.google.com/general/alternative_php.html)
  + [Security](http://docs.google.com/general/security.html)
  + [PHP Style Guide](http://docs.google.com/general/styleguide.html)
* [Libraries](http://docs.google.com/index.html)
  + [Benchmarking Class](http://docs.google.com/benchmark.html)
  + [Caching Driver](http://docs.google.com/caching.html)
  + [Calendaring Class](http://docs.google.com/calendar.html)
  + [Shopping Cart Class](http://docs.google.com/cart.html)
  + [Config Class](http://docs.google.com/config.html)
  + [Email Class](http://docs.google.com/email.html)
  + [Encrypt Class](http://docs.google.com/encrypt.html)
  + [Encryption Library](http://docs.google.com/encryption.html)
  + [File Uploading Class](http://docs.google.com/file_uploading.html)
  + [Form Validation](http://docs.google.com/form_validation.html)
  + [FTP Class](http://docs.google.com/ftp.html)
  + [Image Manipulation Class](http://docs.google.com/image_lib.html)
  + [Input Class](http://docs.google.com/input.html)
  + [Javascript Class](http://docs.google.com/javascript.html)
  + [Language Class](http://docs.google.com/language.html)
  + [Loader Class](http://docs.google.com/loader.html)
  + [Migrations Class](http://docs.google.com/migration.html)
  + [Output Class](http://docs.google.com/output.html)
  + [Pagination Class](http://docs.google.com/pagination.html)
  + [Template Parser Class](http://docs.google.com/parser.html)
  + [Security Class](http://docs.google.com/security.html)
  + [Session Library](http://docs.google.com/sessions.html)
  + HTML Table Class
  + [Trackback Class](http://docs.google.com/trackback.html)
  + [Typography Class](http://docs.google.com/typography.html)
  + [Unit Testing Class](http://docs.google.com/unit_testing.html)
  + [URI Class](http://docs.google.com/uri.html)
  + [User Agent Class](http://docs.google.com/user_agent.html)
  + [XML-RPC and XML-RPC Server Classes](http://docs.google.com/xmlrpc.html)
  + [Zip Encoding Class](http://docs.google.com/zip.html)
* [Database Reference](http://docs.google.com/database/index.html)
  + [Quick Start: Usage Examples](http://docs.google.com/database/examples.html)
  + [Database Configuration](http://docs.google.com/database/configuration.html)
  + [Connecting to a Database](http://docs.google.com/database/connecting.html)
  + [Running Queries](http://docs.google.com/database/queries.html)
  + [Generating Query Results](http://docs.google.com/database/results.html)
  + [Query Helper Functions](http://docs.google.com/database/helpers.html)
  + [Query Builder Class](http://docs.google.com/database/query_builder.html)
  + [Transactions](http://docs.google.com/database/transactions.html)
  + [Getting MetaData](http://docs.google.com/database/metadata.html)
  + [Custom Function Calls](http://docs.google.com/database/call_function.html)
  + [Query Caching](http://docs.google.com/database/caching.html)
  + [Database Manipulation with Database Forge](http://docs.google.com/database/forge.html)
  + [Database Utilities Class](http://docs.google.com/database/utilities.html)
  + [Database Driver Reference](http://docs.google.com/database/db_driver_reference.html)
* [Helpers](http://docs.google.com/helpers/index.html)
  + [Array Helper](http://docs.google.com/helpers/array_helper.html)
  + [CAPTCHA Helper](http://docs.google.com/helpers/captcha_helper.html)
  + [Cookie Helper](http://docs.google.com/helpers/cookie_helper.html)
  + [Date Helper](http://docs.google.com/helpers/date_helper.html)
  + [Directory Helper](http://docs.google.com/helpers/directory_helper.html)
  + [Download Helper](http://docs.google.com/helpers/download_helper.html)
  + [Email Helper](http://docs.google.com/helpers/email_helper.html)
  + [File Helper](http://docs.google.com/helpers/file_helper.html)
  + [Form Helper](http://docs.google.com/helpers/form_helper.html)
  + [HTML Helper](http://docs.google.com/helpers/html_helper.html)
  + [Inflector Helper](http://docs.google.com/helpers/inflector_helper.html)
  + [Language Helper](http://docs.google.com/helpers/language_helper.html)
  + [Number Helper](http://docs.google.com/helpers/number_helper.html)
  + [Path Helper](http://docs.google.com/helpers/path_helper.html)
  + [Security Helper](http://docs.google.com/helpers/security_helper.html)
  + [Smiley Helper](http://docs.google.com/helpers/smiley_helper.html)
  + [String Helper](http://docs.google.com/helpers/string_helper.html)
  + [Text Helper](http://docs.google.com/helpers/text_helper.html)
  + [Typography Helper](http://docs.google.com/helpers/typography_helper.html)
  + [URL Helper](http://docs.google.com/helpers/url_helper.html)
  + [XML Helper](http://docs.google.com/helpers/xml_helper.html)

[CodeIgniter](http://docs.google.com/index.html)

* [Docs](http://docs.google.com/index.html) »
* [Libraries](http://docs.google.com/index.html) »
* HTML Table Class

classic layout

HTML Table Class[**¶**](#gjdgxs)

The Table Class provides functions that enable you to auto-generate HTML tables from arrays or database result sets.

* [Using the Table Class](#30j0zll)
  + [Initializing the Class](#1fob9te)
  + [Examples](#3znysh7)
  + [Changing the Look of Your Table](#2et92p0)
* [Class Reference](#tyjcwt)

[**Using the Table Class**](#3dy6vkm)[**¶**](#30j0zll)

[**Initializing the Class**](#1t3h5sf)[**¶**](#1fob9te)

Like most other classes in CodeIgniter, the Table class is initialized in your controller using the $this->load->library() method:

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

Once loaded, the Table library object will be available using:

$this->table

[**Examples**](#4d34og8)[**¶**](#3znysh7)

Here is an example showing how you can create a table from a multi-dimensional array. Note that the first array index will become the table heading (or you can set your own headings using the set\_heading() method described in the function reference below).

$this->load->library('table');  
  
$data = array(  
 array('Name', 'Color', 'Size'),  
 array('Fred', 'Blue', 'Small'),  
 array('Mary', 'Red', 'Large'),  
 array('John', 'Green', 'Medium')  
);  
  
echo $this->table->generate($data);

Here is an example of a table created from a database query result. The table class will automatically generate the headings based on the table names (or you can set your own headings using the set\_heading() method described in the class reference below).

$this->load->library('table');  
  
$query = $this->db->query('SELECT \* FROM my\_table');  
  
echo $this->table->generate($query);

Here is an example showing how you might create a table using discrete parameters:

$this->load->library('table');  
  
$this->table->set\_heading('Name', 'Color', 'Size');  
  
$this->table->add\_row('Fred', 'Blue', 'Small');  
$this->table->add\_row('Mary', 'Red', 'Large');  
$this->table->add\_row('John', 'Green', 'Medium');  
  
echo $this->table->generate();

Here is the same example, except instead of individual parameters, arrays are used:

$this->load->library('table');  
  
$this->table->set\_heading(array('Name', 'Color', 'Size'));  
  
$this->table->add\_row(array('Fred', 'Blue', 'Small'));  
$this->table->add\_row(array('Mary', 'Red', 'Large'));  
$this->table->add\_row(array('John', 'Green', 'Medium'));  
  
echo $this->table->generate();

[**Changing the Look of Your Table**](#2s8eyo1)[**¶**](#2et92p0)

The Table Class permits you to set a table template with which you can specify the design of your layout. Here is the template prototype:

$template = array(  
 'table\_open' => '<table border="0" cellpadding="4" cellspacing="0">',  
  
 'thead\_open' => '<thead>',  
 'thead\_close' => '</thead>',  
  
 'heading\_row\_start' => '<tr>',  
 'heading\_row\_end' => '</tr>',  
 'heading\_cell\_start' => '<th>',  
 'heading\_cell\_end' => '</th>',  
  
 'tbody\_open' => '<tbody>',  
 'tbody\_close' => '</tbody>',  
  
 'row\_start' => '<tr>',  
 'row\_end' => '</tr>',  
 'cell\_start' => '<td>',  
 'cell\_end' => '</td>',  
  
 'row\_alt\_start' => '<tr>',  
 'row\_alt\_end' => '</tr>',  
 'cell\_alt\_start' => '<td>',  
 'cell\_alt\_end' => '</td>',  
  
 'table\_close' => '</table>'  
);  
  
$this->table->set\_template($template);

Note

You’ll notice there are two sets of “row” blocks in the template. These permit you to create alternating row colors or design elements that alternate with each iteration of the row data.

You are NOT required to submit a complete template. If you only need to change parts of the layout you can simply submit those elements. In this example, only the table opening tag is being changed:

$template = array(  
 'table\_open' => '<table border="1" cellpadding="2" cellspacing="1" class="mytable">'  
);  
  
$this->table->set\_template($template);

You can also set defaults for these in a config file.

[**Class Reference**](#17dp8vu)[**¶**](#tyjcwt)

*class* CI\_Table[¶](#3rdcrjn) $function = NULL

Allows you to specify a native PHP function or a valid function array object to be applied to all cell data.

$this->load->library('table');  
  
$this->table->set\_heading('Name', 'Color', 'Size');  
$this->table->add\_row('Fred', '<strong>Blue</strong>', 'Small');  
  
$this->table->function = 'htmlspecialchars';  
echo $this->table->generate();

In the above example, all cell data would be ran through PHP’s htmlspecialchars() function, resulting in:

<td>Fred</td><td>&lt;strong&gt;Blue&lt;/strong&gt;</td><td>Small</td>

generate([*$table\_data = NULL*])[¶](#26in1rg)

| Parameters: | * **$table\_data** (*mixed*) – Data to populate the table rows with |
| --- | --- |
| Returns: | HTML table |
| Return type: | string |

Returns a string containing the generated table. Accepts an optional parameter which can be an array or a database result object.

set\_caption(*$caption*)[¶](#lnxbz9)

| Parameters: | * **$caption** (*string*) – Table caption |
| --- | --- |
| Returns: | CI\_Table instance (method chaining) |
| Return type: | CI\_Table |

Permits you to add a caption to the table.

$this->table->set\_caption('Colors');

set\_heading([*$args = array()*[, *...*]])[¶](#35nkun2)

| Parameters: | * **$args** (*mixed*) – An array or multiple strings containing the table column titles |
| --- | --- |
| Returns: | CI\_Table instance (method chaining) |
| Return type: | CI\_Table |

Permits you to set the table heading. You can submit an array or discrete params:

$this->table->set\_heading('Name', 'Color', 'Size');  
  
$this->table->set\_heading(array('Name', 'Color', 'Size'));

add\_row([*$args = array()*[, *...*]])[¶](#1ksv4uv)

| Parameters: | * **$args** (*mixed*) – An array or multiple strings containing the row values |
| --- | --- |
| Returns: | CI\_Table instance (method chaining) |
| Return type: | CI\_Table |

Permits you to add a row to your table. You can submit an array or discrete params:

$this->table->add\_row('Blue', 'Red', 'Green');  
  
$this->table->add\_row(array('Blue', 'Red', 'Green'));

If you would like to set an individual cell’s tag attributes, you can use an associative array for that cell. The associative key **data** defines the cell’s data. Any other key => val pairs are added as key=’val’ attributes to the tag:

$cell = array('data' => 'Blue', 'class' => 'highlight', 'colspan' => 2);  
$this->table->add\_row($cell, 'Red', 'Green');  
  
// generates  
// <td class='highlight' colspan='2'>Blue</td><td>Red</td><td>Green</td>

make\_columns([*$array = array()*[, *$col\_limit = 0*]])[¶](#44sinio)

| Parameters: | * **$array** (*array*) – An array containing multiple rows’ data * **$col\_limit** (*int*) – Count of columns in the table |
| --- | --- |
| Returns: | An array of HTML table columns |
| Return type: | array |

This method takes a one-dimensional array as input and creates a multi-dimensional array with a depth equal to the number of columns desired. This allows a single array with many elements to be displayed in a table that has a fixed column count. Consider this example:

$list = array('one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine', 'ten', 'eleven', 'twelve');  
  
$new\_list = $this->table->make\_columns($list, 3);  
  
$this->table->generate($new\_list);  
  
// Generates a table with this prototype  
  
<table border="0" cellpadding="4" cellspacing="0">  
<tr>  
<td>one</td><td>two</td><td>three</td>  
</tr><tr>  
<td>four</td><td>five</td><td>six</td>  
</tr><tr>  
<td>seven</td><td>eight</td><td>nine</td>  
</tr><tr>  
<td>ten</td><td>eleven</td><td>twelve</td></tr>  
</table>

set\_template(*$template*)[¶](#2jxsxqh)

| Parameters: | * **$template** (*array*) – An associative array containing template values |
| --- | --- |
| Returns: | TRUE on success, FALSE on failure |
| Return type: | bool |

Permits you to set your template. You can submit a full or partial template.

$template = array(  
 'table\_open' => '<table border="1" cellpadding="2" cellspacing="1" class="mytable">'  
);  
  
$this->table->set\_template($template);

set\_empty(*$value*)[¶](#z337ya)

| Parameters: | * **$value** (*mixed*) – Value to put in empty cells |
| --- | --- |
| Returns: | CI\_Table instance (method chaining) |
| Return type: | CI\_Table |

Lets you set a default value for use in any table cells that are empty. You might, for example, set a non-breaking space:

$this->table->set\_empty(" ");

clear()[¶](#3j2qqm3)

| Returns: | CI\_Table instance (method chaining) |
| --- | --- |
| Return type: | CI\_Table |

Lets you clear the table heading and row data. If you need to show multiple tables with different data you should to call this method after each table has been generated to clear the previous table information. Example:

$this->load->library('table');  
  
$this->table->set\_heading('Name', 'Color', 'Size');  
$this->table->add\_row('Fred', 'Blue', 'Small');  
$this->table->add\_row('Mary', 'Red', 'Large');  
$this->table->add\_row('John', 'Green', 'Medium');  
  
echo $this->table->generate();  
  
$this->table->clear();  
  
$this->table->set\_heading('Name', 'Day', 'Delivery');  
$this->table->add\_row('Fred', 'Wednesday', 'Express');  
$this->table->add\_row('Mary', 'Monday', 'Air');  
$this->table->add\_row('John', 'Saturday', 'Overnight');  
  
echo $this->table->generate();

[Next](http://docs.google.com/trackback.html)   [Previous](http://docs.google.com/sessions.html)

© Copyright 2014 - 2016, British Columbia Institute of Technology. Last updated on Oct 28, 2016.

Built with [Sphinx](http://sphinx-doc.org/) using a [theme](https://github.com/snide/sphinx_rtd_theme) provided by [Read the Docs](https://readthedocs.org).