* [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
  + [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](http://docs.google.com/table.html)
  + [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
  + [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](http://docs.google.com/table.html)
  + [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) »
* Calendaring Class

classic layout

Calendaring Class[**¶**](#gjdgxs)

The Calendar class enables you to dynamically create calendars. Your calendars can be formatted through the use of a calendar template, allowing 100% control over every aspect of its design. In addition, you can pass data to your calendar cells.

* [Using the Calendaring Class](#30j0zll)
  + [Initializing the Class](#1fob9te)
  + [Displaying a Calendar](#3znysh7)
  + [Passing Data to your Calendar Cells](#2et92p0)
  + [Setting Display Preferences](#tyjcwt)
  + [Showing Next/Previous Month Links](#3dy6vkm)
  + [Creating a Calendar Template](#1t3h5sf)
* [Class Reference](#4d34og8)

[**Using the Calendaring Class**](#2s8eyo1)[**¶**](#30j0zll)

[**Initializing the Class**](#17dp8vu)[**¶**](#1fob9te)

Like most other classes in CodeIgniter, the Calendar class is initialized in your controller using the $this->load->library function:

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

Once loaded, the Calendar object will be available using:

$this->calendar

[**Displaying a Calendar**](#3rdcrjn)[**¶**](#3znysh7)

Here is a very simple example showing how you can display a calendar:

$this->load->library('calendar');  
echo $this->calendar->generate();

The above code will generate a calendar for the current month/year based on your server time. To show a calendar for a specific month and year you will pass this information to the calendar generating function:

$this->load->library('calendar');  
echo $this->calendar->generate(2006, 6);

The above code will generate a calendar showing the month of June in 2006. The first parameter specifies the year, the second parameter specifies the month.

[**Passing Data to your Calendar Cells**](#26in1rg)[**¶**](#2et92p0)

To add data to your calendar cells involves creating an associative array in which the keys correspond to the days you wish to populate and the array value contains the data. The array is passed to the third parameter of the calendar generating function. Consider this example:

$this->load->library('calendar');  
  
$data = array(  
 3 => 'http://example.com/news/article/2006/06/03/',  
 7 => 'http://example.com/news/article/2006/06/07/',  
 13 => 'http://example.com/news/article/2006/06/13/',  
 26 => 'http://example.com/news/article/2006/06/26/'  
);  
  
echo $this->calendar->generate(2006, 6, $data);

Using the above example, day numbers 3, 7, 13, and 26 will become links pointing to the URLs you’ve provided.

Note

By default it is assumed that your array will contain links. In the section that explains the calendar template below you’ll see how you can customize how data passed to your cells is handled so you can pass different types of information.

[**Setting Display Preferences**](#lnxbz9)[**¶**](#tyjcwt)

There are seven preferences you can set to control various aspects of the calendar. Preferences are set by passing an array of preferences in the second parameter of the loading function. Here is an example:

$prefs = array(  
 'start\_day' => 'saturday',  
 'month\_type' => 'long',  
 'day\_type' => 'short'  
);  
  
$this->load->library('calendar', $prefs);  
  
echo $this->calendar->generate();

The above code would start the calendar on saturday, use the “long” month heading, and the “short” day names. More information regarding preferences below.

| Preference | Default | Options | Description |
| --- | --- | --- | --- |
| **template** | None | None | A string or array containing your calendar template. See the template section below. |
| **local\_time** | time() | None | A Unix timestamp corresponding to the current time. |
| **start\_day** | sunday | Any week day (sunday, monday, tuesday, etc.) | Sets the day of the week the calendar should start on. |
| **month\_type** | long | long, short | Determines what version of the month name to use in the header. long = January, short = Jan. |
| **day\_type** | abr | long, short, abr | Determines what version of the weekday names to use in the column headers. long = Sunday, short = Sun, abr = Su. |
| **show\_next\_prev** | FALSE | TRUE/FALSE (boolean) | Determines whether to display links allowing you to toggle to next/previous months. See information on this feature below. |
| **next\_prev\_url** | controller/method | A URL | Sets the basepath used in the next/previous calendar links. |
| **show\_other\_days** | FALSE | TRUE/FALSE (boolean) | Determines whether to display days of other months that share the first or last week of the calendar month. |

[**Showing Next/Previous Month Links**](#35nkun2)[**¶**](#3dy6vkm)

To allow your calendar to dynamically increment/decrement via the next/previous links requires that you set up your calendar code similar to this example:

$prefs = array(  
 'show\_next\_prev' => TRUE,  
 'next\_prev\_url' => 'http://example.com/index.php/calendar/show/'  
);  
  
$this->load->library('calendar', $prefs);  
  
echo $this->calendar->generate($this->uri->segment(3), $this->uri->segment(4));

You’ll notice a few things about the above example:

* You must set the “show\_next\_prev” to TRUE.
* You must supply the URL to the controller containing your calendar in the “next\_prev\_url” preference. If you don’t, it will be set to the current *controller/method*.
* You must supply the “year” and “month” to the calendar generating function via the URI segments where they appear (Note: The calendar class automatically adds the year/month to the base URL you provide.).

[**Creating a Calendar Template**](#1ksv4uv)[**¶**](#1t3h5sf)

By creating a calendar template you have 100% control over the design of your calendar. Using the string method, each component of your calendar will be placed within a pair of pseudo-variables as shown here:

$prefs['template'] = '  
  
 {table\_open}<table border="0" cellpadding="0" cellspacing="0">{/table\_open}  
  
 {heading\_row\_start}<tr>{/heading\_row\_start}  
  
 {heading\_previous\_cell}<th><a href="{previous\_url}"><<</a></th>{/heading\_previous\_cell}  
 {heading\_title\_cell}<th colspan="{colspan}">{heading}</th>{/heading\_title\_cell}  
 {heading\_next\_cell}<th><a href="{next\_url}">>></a></th>{/heading\_next\_cell}  
  
 {heading\_row\_end}</tr>{/heading\_row\_end}  
  
 {week\_row\_start}<tr>{/week\_row\_start}  
 {week\_day\_cell}<td>{week\_day}</td>{/week\_day\_cell}  
 {week\_row\_end}</tr>{/week\_row\_end}  
  
 {cal\_row\_start}<tr>{/cal\_row\_start}  
 {cal\_cell\_start}<td>{/cal\_cell\_start}  
 {cal\_cell\_start\_today}<td>{/cal\_cell\_start\_today}  
 {cal\_cell\_start\_other}<td class="other-month">{/cal\_cell\_start\_other}  
  
 {cal\_cell\_content}<a href="{content}">{day}</a>{/cal\_cell\_content}  
 {cal\_cell\_content\_today}<div class="highlight"><a href="{content}">{day}</a></div>{/cal\_cell\_content\_today}  
  
 {cal\_cell\_no\_content}{day}{/cal\_cell\_no\_content}  
 {cal\_cell\_no\_content\_today}<div class="highlight">{day}</div>{/cal\_cell\_no\_content\_today}  
  
 {cal\_cell\_blank} {/cal\_cell\_blank}  
  
 {cal\_cell\_other}{day}{/cal\_cel\_other}  
  
 {cal\_cell\_end}</td>{/cal\_cell\_end}  
 {cal\_cell\_end\_today}</td>{/cal\_cell\_end\_today}  
 {cal\_cell\_end\_other}</td>{/cal\_cell\_end\_other}  
 {cal\_row\_end}</tr>{/cal\_row\_end}  
  
 {table\_close}</table>{/table\_close}  
';  
  
$this->load->library('calendar', $prefs);  
  
echo $this->calendar->generate();

Using the array method, you will pass key => value pairs. You can pass as many or as few values as you’d like. Omitted keys will use the default values inherited in the calendar class.

Example:

$prefs['template'] = array(  
 'table\_open' => '<table class="calendar">',  
 'cal\_cell\_start' => '<td class="day">',  
 'cal\_cell\_start\_today' => '<td class="today">'  
);  
  
$this->load->library('calendar', $prefs);  
  
echo $this->calendar->generate();

[**Class Reference**](#44sinio)[**¶**](#4d34og8)

*class* CI\_Calendar[¶](#2jxsxqh) initialize([*$config = array()*])[¶](#z337ya)

| Parameters: | * **$config** (*array*) – Configuration parameters |
| --- | --- |
| Returns: | CI\_Calendar instance (method chaining) |
| Return type: | CI\_Calendar |

Initializes the Calendaring preferences. Accepts an associative array as input, containing display preferences.

generate([*$year = ''*[, *$month = ''*[, *$data = array()*]]])[¶](#3j2qqm3)

| Parameters: | * **$year** (*int*) – Year * **$month** (*int*) – Month * **$data** (*array*) – Data to be shown in the calendar cells |
| --- | --- |
| Returns: | HTML-formatted calendar |
| Return type: | string |

Generate the calendar.

get\_month\_name(*$month*)[¶](#1y810tw)

| Parameters: | * **$month** (*int*) – Month |
| --- | --- |
| Returns: | Month name |
| Return type: | string |

Generates a textual month name based on the numeric month provided.

get\_day\_names(*$day\_type = ''*)[¶](#4i7ojhp)

| Parameters: | * **$day\_type** (*string*) – ‘long’, ‘short’, or ‘abr’ |
| --- | --- |
| Returns: | Array of day names |
| Return type: | array |

Returns an array of day names (Sunday, Monday, etc.) based on the type provided. Options: long, short, abr. If no $day\_type is provided (or if an invalid type is provided) this method will return the “abbreviated” style.

adjust\_date(*$month*, *$year*)[¶](#2xcytpi)

| Parameters: | * **$month** (*int*) – Month * **$year** (*int*) – Year |
| --- | --- |
| Returns: | An associative array containing month and year |
| Return type: | array |

This method makes sure that you have a valid month/year. For example, if you submit 13 as the month, the year will increment and the month will become January:

print\_r($this->calendar->adjust\_date(13, 2014));

outputs:

Array  
(  
 [month] => '01'  
 [year] => '2015'  
)

get\_total\_days(*$month*, *$year*)[¶](#1ci93xb)

| Parameters: | * **$month** (*int*) – Month * **$year** (*int*) – Year |
| --- | --- |
| Returns: | Count of days in the specified month |
| Return type: | int |

Total days in a given month:

echo $this->calendar->get\_total\_days(2, 2012);  
// 29

Note

This method is an alias for [*Date Helper*](http://docs.google.com/helpers/date_helper.html) function [days\_in\_month()](http://docs.google.com/helpers/date_helper.html#days_in_month).

default\_template()[¶](#3whwml4)

| Returns: | An array of template values |
| --- | --- |
| Return type: | array |

Sets the default template. This method is used when you have not created your own template.

parse\_template()[¶](#2bn6wsx)

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

Harvests the data within the template {pseudo-variables} used to display the calendar.

[Next](http://docs.google.com/cart.html)   [Previous](http://docs.google.com/caching.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).