* [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](#gjdgxs)
  + [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](http://docs.google.com/calendar.html)
  + [Shopping Cart Class](#gjdgxs)
  + [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) »
* Shopping Cart Class

classic layout

Shopping Cart Class[**¶**](#30j0zll)

The Cart Class permits items to be added to a session that stays active while a user is browsing your site. These items can be retrieved and displayed in a standard “shopping cart” format, allowing the user to update the quantity or remove items from the cart.

Important

The Cart library is DEPRECATED and should not be used. It is currently only kept for backwards compatibility.

Please note that the Cart Class ONLY provides the core “cart” functionality. It does not provide shipping, credit card authorization, or other processing components.

* [Using the Cart Class](#1fob9te)
  + [Initializing the Shopping Cart Class](#3znysh7)
  + [Adding an Item to The Cart](#2et92p0)
  + [Adding Multiple Items to The Cart](#tyjcwt)
  + [Displaying the Cart](#3dy6vkm)
  + [Updating The Cart](#1t3h5sf)
    - [What is a Row ID?](#4d34og8)
* [Class Reference](#2s8eyo1)

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

[**Initializing the Shopping Cart Class**](#3rdcrjn)[**¶**](#3znysh7)

Important

The Cart class utilizes CodeIgniter’s [Session Class](http://docs.google.com/sessions.html) to save the cart information to a database, so before using the Cart class you must set up a database table as indicated in the [Session Documentation](http://docs.google.com/sessions.html), and set the session preferences in your application/config/config.php file to utilize a database.

To initialize the Shopping Cart Class in your controller constructor, use the $this->load->library() method:

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

Once loaded, the Cart object will be available using:

$this->cart

Note

The Cart Class will load and initialize the Session Class automatically, so unless you are using sessions elsewhere in your application, you do not need to load the Session class.

[**Adding an Item to The Cart**](#26in1rg)[**¶**](#2et92p0)

To add an item to the shopping cart, simply pass an array with the product information to the $this->cart->insert() method, as shown below:

$data = array(  
 'id' => 'sku\_123ABC',  
 'qty' => 1,  
 'price' => 39.95,  
 'name' => 'T-Shirt',  
 'options' => array('Size' => 'L', 'Color' => 'Red')  
);  
  
$this->cart->insert($data);

Important

The first four array indexes above (id, qty, price, and name) are **required**. If you omit any of them the data will not be saved to the cart. The fifth index (options) is optional. It is intended to be used in cases where your product has options associated with it. Use an array for options, as shown above.

The five reserved indexes are:

* **id** - Each product in your store must have a unique identifier. Typically this will be an “sku” or other such identifier.
* **qty** - The quantity being purchased.
* **price** - The price of the item.
* **name** - The name of the item.
* **options** - Any additional attributes that are needed to identify the product. These must be passed via an array.

In addition to the five indexes above, there are two reserved words: rowid and subtotal. These are used internally by the Cart class, so please do NOT use those words as index names when inserting data into the cart.

Your array may contain additional data. Anything you include in your array will be stored in the session. However, it is best to standardize your data among all your products in order to make displaying the information in a table easier.

$data = array(  
 'id' => 'sku\_123ABC',  
 'qty' => 1,  
 'price' => 39.95,  
 'name' => 'T-Shirt',  
 'coupon' => 'XMAS-50OFF'  
);  
  
$this->cart->insert($data);

The insert() method will return the $rowid if you successfully insert a single item.

[**Adding Multiple Items to The Cart**](#lnxbz9)[**¶**](#tyjcwt)

By using a multi-dimensional array, as shown below, it is possible to add multiple products to the cart in one action. This is useful in cases where you wish to allow people to select from among several items on the same page.

$data = array(  
 array(  
 'id' => 'sku\_123ABC',  
 'qty' => 1,  
 'price' => 39.95,  
 'name' => 'T-Shirt',  
 'options' => array('Size' => 'L', 'Color' => 'Red')  
 ),  
 array(  
 'id' => 'sku\_567ZYX',  
 'qty' => 1,  
 'price' => 9.95,  
 'name' => 'Coffee Mug'  
 ),  
 array(  
 'id' => 'sku\_965QRS',  
 'qty' => 1,  
 'price' => 29.95,  
 'name' => 'Shot Glass'  
 )  
);  
  
$this->cart->insert($data);

[**Displaying the Cart**](#35nkun2)[**¶**](#3dy6vkm)

To display the cart you will create a [view file](http://docs.google.com/general/views.html) with code similar to the one shown below.

Please note that this example uses the [form helper](http://docs.google.com/helpers/form_helper.html).

<?php echo form\_open('path/to/controller/update/method'); ?>  
  
<table cellpadding="6" cellspacing="1" style="width:100%" border="0">  
  
<tr>  
 <th>QTY</th>  
 <th>Item Description</th>  
 <th style="text-align:right">Item Price</th>  
 <th style="text-align:right">Sub-Total</th>  
</tr>  
  
<?php $i = 1; ?>  
  
<?php foreach ($this->cart->contents() as $items): ?>  
  
 <?php echo form\_hidden($i.'[rowid]', $items['rowid']); ?>  
  
 <tr>  
 <td><?php echo form\_input(array('name' => $i.'[qty]', 'value' => $items['qty'], 'maxlength' => '3', 'size' => '5')); ?></td>  
 <td>  
 <?php echo $items['name']; ?>  
  
 <?php if ($this->cart->has\_options($items['rowid']) == TRUE): ?>  
  
 <p>  
 <?php foreach ($this->cart->product\_options($items['rowid']) as $option\_name => $option\_value): ?>  
  
 <strong><?php echo $option\_name; ?>:</strong> <?php echo $option\_value; ?><br />  
  
 <?php endforeach; ?>  
 </p>  
  
 <?php endif; ?>  
  
 </td>  
 <td style="text-align:right"><?php echo $this->cart->format\_number($items['price']); ?></td>  
 <td style="text-align:right">$<?php echo $this->cart->format\_number($items['subtotal']); ?></td>  
 </tr>  
  
<?php $i++; ?>  
  
<?php endforeach; ?>  
  
<tr>  
 <td colspan="2"> </td>  
 <td class="right"><strong>Total</strong></td>  
 <td class="right">$<?php echo $this->cart->format\_number($this->cart->total()); ?></td>  
</tr>  
  
</table>  
  
<p><?php echo form\_submit('', 'Update your Cart'); ?></p>

[**Updating The Cart**](#1ksv4uv)[**¶**](#1t3h5sf)

To update the information in your cart, you must pass an array containing the Row ID and one or more pre-defined properties to the $this->cart->update() method.

Note

If the quantity is set to zero, the item will be removed from the cart.

$data = array(  
 'rowid' => 'b99ccdf16028f015540f341130b6d8ec',  
 'qty' => 3  
);  
  
$this->cart->update($data);  
  
// Or a multi-dimensional array  
  
$data = array(  
 array(  
 'rowid' => 'b99ccdf16028f015540f341130b6d8ec',  
 'qty' => 3  
 ),  
 array(  
 'rowid' => 'xw82g9q3r495893iajdh473990rikw23',  
 'qty' => 4  
 ),  
 array(  
 'rowid' => 'fh4kdkkkaoe30njgoe92rkdkkobec333',  
 'qty' => 2  
 )  
);  
  
$this->cart->update($data);

You may also update any property you have previously defined when inserting the item such as options, price or other custom fields.

$data = array(  
 'rowid' => 'b99ccdf16028f015540f341130b6d8ec',  
 'qty' => 1,  
 'price' => 49.95,  
 'coupon' => NULL  
);  
  
$this->cart->update($data);

[**What is a Row ID?**](#44sinio)[**¶**](#4d34og8)

The row ID is a unique identifier that is generated by the cart code when an item is added to the cart. The reason a unique ID is created is so that identical products with different options can be managed by the cart.

For example, let’s say someone buys two identical t-shirts (same product ID), but in different sizes. The product ID (and other attributes) will be identical for both sizes because it’s the same shirt. The only difference will be the size. The cart must therefore have a means of identifying this difference so that the two sizes of shirts can be managed independently. It does so by creating a unique “row ID” based on the product ID and any options associated with it.

In nearly all cases, updating the cart will be something the user does via the “view cart” page, so as a developer, it is unlikely that you will ever have to concern yourself with the “row ID”, other than making sure your “view cart” page contains this information in a hidden form field, and making sure it gets passed to the update() method when the update form is submitted. Please examine the construction of the “view cart” page above for more information.

[**Class Reference**](#2jxsxqh)[**¶**](#2s8eyo1)

*class* CI\_Cart[¶](#z337ya) $product\_id\_rules = '.a-z0-9\_-'

These are the regular expression rules that we use to validate the product ID - alpha-numeric, dashes, underscores, or periods by default

$product\_name\_rules = 'w -.:'

These are the regular expression rules that we use to validate the product ID and product name - alpha-numeric, dashes, underscores, colons or periods by default

$product\_name\_safe = TRUE

Whether or not to only allow safe product names. Default TRUE.

insert([*$items = array()*])[¶](#3j2qqm3)

|  |  |
| --- | --- |
| Parameters: | * **$items** (*array*) – Items to insert into the cart |
| Returns: | TRUE on success, FALSE on failure |
| Return type: | bool |

Insert items into the cart and save it to the session table. Returns TRUE on success and FALSE on failure.

update([*$items = array()*])[¶](#1y810tw)

|  |  |
| --- | --- |
| Parameters: | * **$items** (*array*) – Items to update in the cart |
| Returns: | TRUE on success, FALSE on failure |
| Return type: | bool |

This method permits changing the properties of a given item. Typically it is called from the “view cart” page if a user makes changes to the quantity before checkout. That array must contain the rowid for each item.

remove(*$rowid*)[¶](#4i7ojhp)

|  |  |
| --- | --- |
| Parameters: | * **$rowid** (*int*) – ID of the item to remove from the cart |
| Returns: | TRUE on success, FALSE on failure |
| Return type: | bool |

Allows you to remove an item from the shopping cart by passing it the $rowid.

total()[¶](#2xcytpi)

|  |  |
| --- | --- |
| Returns: | Total amount |
| Return type: | int |

Displays the total amount in the cart.

total\_items()[¶](#1ci93xb)

|  |  |
| --- | --- |
| Returns: | Total amount of items in the cart |
| Return type: | int |

Displays the total number of items in the cart.

contents([*$newest\_first = FALSE*])[¶](#3whwml4)

|  |  |
| --- | --- |
| Parameters: | * **$newest\_first** (*bool*) – Whether to order the array with newest items first |
| Returns: | An array of cart contents |
| Return type: | array |

Returns an array containing everything in the cart. You can sort the order by which the array is returned by passing it TRUE where the contents will be sorted from newest to oldest, otherwise it is sorted from oldest to newest.

get\_item(*$row\_id*)[¶](#2bn6wsx)

|  |  |
| --- | --- |
| Parameters: | * **$row\_id** (*int*) – Row ID to retrieve |
| Returns: | Array of item data |
| Return type: | array |

Returns an array containing data for the item matching the specified row ID, or FALSE if no such item exists.

has\_options(*$row\_id = ''*)[¶](#qsh70q)

|  |  |
| --- | --- |
| Parameters: | * **$row\_id** (*int*) – Row ID to inspect |
| Returns: | TRUE if options exist, FALSE otherwise |
| Return type: | bool |

Returns TRUE (boolean) if a particular row in the cart contains options. This method is designed to be used in a loop with contents(), since you must pass the rowid to this method, as shown in the Displaying the Cart example above.

product\_options([*$row\_id = ''*])[¶](#3as4poj)

|  |  |
| --- | --- |
| Parameters: | * **$row\_id** (*int*) – Row ID |
| Returns: | Array of product options |
| Return type: | array |

Returns an array of options for a particular product. This method is designed to be used in a loop with contents(), since you must pass the rowid to this method, as shown in the Displaying the Cart example above.

destroy()[¶](#1pxezwc)

|  |  |
| --- | --- |
| Return type: | void |

Permits you to destroy the cart. This method will likely be called when you are finished processing the customer’s order.

[Next](http://docs.google.com/config.html)   [Previous](http://docs.google.com/calendar.html)

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

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).