* [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/libraries/index.html)
  + [Benchmarking Class](http://docs.google.com/libraries/benchmark.html)
  + [Caching Driver](http://docs.google.com/libraries/caching.html)
  + [Calendaring Class](http://docs.google.com/libraries/calendar.html)
  + [Shopping Cart Class](http://docs.google.com/libraries/cart.html)
  + [Config Class](http://docs.google.com/libraries/config.html)
  + [Email Class](http://docs.google.com/libraries/email.html)
  + [Encrypt Class](http://docs.google.com/libraries/encrypt.html)
  + [Encryption Library](http://docs.google.com/libraries/encryption.html)
  + [File Uploading Class](http://docs.google.com/libraries/file_uploading.html)
  + [Form Validation](http://docs.google.com/libraries/form_validation.html)
  + [FTP Class](http://docs.google.com/libraries/ftp.html)
  + [Image Manipulation Class](http://docs.google.com/libraries/image_lib.html)
  + [Input Class](http://docs.google.com/libraries/input.html)
  + [Javascript Class](http://docs.google.com/libraries/javascript.html)
  + [Language Class](http://docs.google.com/libraries/language.html)
  + [Loader Class](http://docs.google.com/libraries/loader.html)
  + [Migrations Class](http://docs.google.com/libraries/migration.html)
  + [Output Class](http://docs.google.com/libraries/output.html)
  + [Pagination Class](http://docs.google.com/libraries/pagination.html)
  + [Template Parser Class](http://docs.google.com/libraries/parser.html)
  + [Security Class](http://docs.google.com/libraries/security.html)
  + [Session Library](http://docs.google.com/libraries/sessions.html)
  + [HTML Table Class](http://docs.google.com/libraries/table.html)
  + [Trackback Class](http://docs.google.com/libraries/trackback.html)
  + [Typography Class](http://docs.google.com/libraries/typography.html)
  + [Unit Testing Class](http://docs.google.com/libraries/unit_testing.html)
  + [URI Class](http://docs.google.com/libraries/uri.html)
  + [User Agent Class](http://docs.google.com/libraries/user_agent.html)
  + [XML-RPC and XML-RPC Server Classes](http://docs.google.com/libraries/xmlrpc.html)
  + [Zip Encoding Class](http://docs.google.com/libraries/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/index.html)
  + [Array Helper](http://docs.google.com/array_helper.html)
  + [CAPTCHA Helper](http://docs.google.com/captcha_helper.html)
  + [Cookie Helper](http://docs.google.com/cookie_helper.html)
  + [Date Helper](http://docs.google.com/date_helper.html)
  + [Directory Helper](http://docs.google.com/directory_helper.html)
  + [Download Helper](http://docs.google.com/download_helper.html)
  + [Email Helper](http://docs.google.com/email_helper.html)
  + [File Helper](http://docs.google.com/file_helper.html)
  + [Form Helper](http://docs.google.com/form_helper.html)
  + [HTML Helper](http://docs.google.com/html_helper.html)
  + [Inflector Helper](http://docs.google.com/inflector_helper.html)
  + [Language Helper](http://docs.google.com/language_helper.html)
  + [Number Helper](http://docs.google.com/number_helper.html)
  + [Path Helper](http://docs.google.com/path_helper.html)
  + [Security Helper](http://docs.google.com/security_helper.html)
  + [Smiley Helper](http://docs.google.com/smiley_helper.html)
  + [String Helper](http://docs.google.com/string_helper.html)
  + [Text Helper](http://docs.google.com/text_helper.html)
  + [Typography Helper](http://docs.google.com/typography_helper.html)
  + URL Helper
  + [XML Helper](http://docs.google.com/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/libraries/index.html)
  + [Benchmarking Class](http://docs.google.com/libraries/benchmark.html)
  + [Caching Driver](http://docs.google.com/libraries/caching.html)
  + [Calendaring Class](http://docs.google.com/libraries/calendar.html)
  + [Shopping Cart Class](http://docs.google.com/libraries/cart.html)
  + [Config Class](http://docs.google.com/libraries/config.html)
  + [Email Class](http://docs.google.com/libraries/email.html)
  + [Encrypt Class](http://docs.google.com/libraries/encrypt.html)
  + [Encryption Library](http://docs.google.com/libraries/encryption.html)
  + [File Uploading Class](http://docs.google.com/libraries/file_uploading.html)
  + [Form Validation](http://docs.google.com/libraries/form_validation.html)
  + [FTP Class](http://docs.google.com/libraries/ftp.html)
  + [Image Manipulation Class](http://docs.google.com/libraries/image_lib.html)
  + [Input Class](http://docs.google.com/libraries/input.html)
  + [Javascript Class](http://docs.google.com/libraries/javascript.html)
  + [Language Class](http://docs.google.com/libraries/language.html)
  + [Loader Class](http://docs.google.com/libraries/loader.html)
  + [Migrations Class](http://docs.google.com/libraries/migration.html)
  + [Output Class](http://docs.google.com/libraries/output.html)
  + [Pagination Class](http://docs.google.com/libraries/pagination.html)
  + [Template Parser Class](http://docs.google.com/libraries/parser.html)
  + [Security Class](http://docs.google.com/libraries/security.html)
  + [Session Library](http://docs.google.com/libraries/sessions.html)
  + [HTML Table Class](http://docs.google.com/libraries/table.html)
  + [Trackback Class](http://docs.google.com/libraries/trackback.html)
  + [Typography Class](http://docs.google.com/libraries/typography.html)
  + [Unit Testing Class](http://docs.google.com/libraries/unit_testing.html)
  + [URI Class](http://docs.google.com/libraries/uri.html)
  + [User Agent Class](http://docs.google.com/libraries/user_agent.html)
  + [XML-RPC and XML-RPC Server Classes](http://docs.google.com/libraries/xmlrpc.html)
  + [Zip Encoding Class](http://docs.google.com/libraries/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/index.html)
  + [Array Helper](http://docs.google.com/array_helper.html)
  + [CAPTCHA Helper](http://docs.google.com/captcha_helper.html)
  + [Cookie Helper](http://docs.google.com/cookie_helper.html)
  + [Date Helper](http://docs.google.com/date_helper.html)
  + [Directory Helper](http://docs.google.com/directory_helper.html)
  + [Download Helper](http://docs.google.com/download_helper.html)
  + [Email Helper](http://docs.google.com/email_helper.html)
  + [File Helper](http://docs.google.com/file_helper.html)
  + [Form Helper](http://docs.google.com/form_helper.html)
  + [HTML Helper](http://docs.google.com/html_helper.html)
  + [Inflector Helper](http://docs.google.com/inflector_helper.html)
  + [Language Helper](http://docs.google.com/language_helper.html)
  + [Number Helper](http://docs.google.com/number_helper.html)
  + [Path Helper](http://docs.google.com/path_helper.html)
  + [Security Helper](http://docs.google.com/security_helper.html)
  + [Smiley Helper](http://docs.google.com/smiley_helper.html)
  + [String Helper](http://docs.google.com/string_helper.html)
  + [Text Helper](http://docs.google.com/text_helper.html)
  + [Typography Helper](http://docs.google.com/typography_helper.html)
  + URL Helper
  + [XML Helper](http://docs.google.com/xml_helper.html)

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

* [Docs](http://docs.google.com/index.html) »
* [Helpers](http://docs.google.com/index.html) »
* URL Helper

classic layout

URL Helper[**¶**](#gjdgxs)

The URL Helper file contains functions that assist in working with URLs.

* [Loading this Helper](#30j0zll)
* [Available Functions](#1fob9te)

[**Loading this Helper**](#3znysh7)[**¶**](#30j0zll)

This helper is loaded using the following code:

$this->load->helper('url');

[**Available Functions**](#2et92p0)[**¶**](#1fob9te)

The following functions are available:

site\_url([*$uri = ''*[, *$protocol = NULL*]])[¶](#tyjcwt)

| Parameters: | * **$uri** (*string*) – URI string * **$protocol** (*string*) – Protocol, e.g. ‘http’ or ‘https’ |
| --- | --- |
| Returns: | Site URL |
| Return type: | string |

Returns your site URL, as specified in your config file. The index.php file (or whatever you have set as your site **index\_page** in your config file) will be added to the URL, as will any URI segments you pass to the function, plus the **url\_suffix** as set in your config file.

You are encouraged to use this function any time you need to generate a local URL so that your pages become more portable in the event your URL changes.

Segments can be optionally passed to the function as a string or an array. Here is a string example:

echo site\_url('news/local/123');

The above example would return something like: *http://example.com/index.php/news/local/123*

Here is an example of segments passed as an array:

$segments = array('news', 'local', '123');  
echo site\_url($segments);

This function is an alias for CI\_Config::site\_url(). For more info, please see the [*Config Library*](http://docs.google.com/libraries/config.html) documentation.

base\_url(*$uri = ''*, *$protocol = NULL*)[¶](#3dy6vkm)

| Parameters: | * **$uri** (*string*) – URI string * **$protocol** (*string*) – Protocol, e.g. ‘http’ or ‘https’ |
| --- | --- |
| Returns: | Base URL |
| Return type: | string |

Returns your site base URL, as specified in your config file. Example:

echo base\_url();

This function returns the same thing as [site\_url()](#tyjcwt), without the *index\_page* or *url\_suffix* being appended.

Also like [site\_url()](#tyjcwt), you can supply segments as a string or an array. Here is a string example:

echo base\_url("blog/post/123");

The above example would return something like: *http://example.com/blog/post/123*

This is useful because unlike [site\_url()](#tyjcwt), you can supply a string to a file, such as an image or stylesheet. For example:

echo base\_url("images/icons/edit.png");

This would give you something like: *http://example.com/images/icons/edit.png*

This function is an alias for CI\_Config::base\_url(). For more info, please see the [*Config Library*](http://docs.google.com/libraries/config.html) documentation.

current\_url()[¶](#1t3h5sf)

| Returns: | The current URL |
| --- | --- |
| Return type: | string |

Returns the full URL (including segments) of the page being currently viewed.

Note

Calling this function is the same as doing this: | | site\_url(uri\_string());

uri\_string()[¶](#4d34og8)

| Returns: | An URI string |
| --- | --- |
| Return type: | string |

Returns the URI segments of any page that contains this function. For example, if your URL was this:

http://some-site.com/blog/comments/123

The function would return:

blog/comments/123

This function is an alias for CI\_Config::uri\_string(). For more info, please see the [*Config Library*](http://docs.google.com/libraries/config.html) documentation.

index\_page()[¶](#2s8eyo1)

| Returns: | ‘index\_page’ value |
| --- | --- |
| Return type: | mixed |

Returns your site **index\_page**, as specified in your config file. Example:

echo index\_page();

anchor(*$uri = ''*, *$title = ''*, *$attributes = ''*)[¶](#17dp8vu)

| Parameters: | * **$uri** (*string*) – URI string * **$title** (*string*) – Anchor title * **$attributes** (*mixed*) – HTML attributes |
| --- | --- |
| Returns: | HTML hyperlink (anchor tag) |
| Return type: | string |

Creates a standard HTML anchor link based on your local site URL.

The first parameter can contain any segments you wish appended to the URL. As with the [site\_url()](#tyjcwt) function above, segments can be a string or an array.

Note

If you are building links that are internal to your application do not include the base URL (http://...). This will be added automatically from the information specified in your config file. Include only the URI segments you wish appended to the URL.

The second segment is the text you would like the link to say. If you leave it blank, the URL will be used.

The third parameter can contain a list of attributes you would like added to the link. The attributes can be a simple string or an associative array.

Here are some examples:

echo anchor('news/local/123', 'My News', 'title="News title"');  
// Prints: <a href="http://example.com/index.php/news/local/123" title="News title">My News</a>  
  
echo anchor('news/local/123', 'My News', array('title' => 'The best news!'));  
// Prints: <a href="http://example.com/index.php/news/local/123" title="The best news!">My News</a>  
  
echo anchor('', 'Click here');  
// Prints: <a href="http://example.com">Click Here</a>

anchor\_popup(*$uri = ''*, *$title = ''*, *$attributes = FALSE*)[¶](#3rdcrjn)

| Parameters: | * **$uri** (*string*) – URI string * **$title** (*string*) – Anchor title * **$attributes** (*mixed*) – HTML attributes |
| --- | --- |
| Returns: | Pop-up hyperlink |
| Return type: | string |

Nearly identical to the [anchor()](#17dp8vu) function except that it opens the URL in a new window. You can specify JavaScript window attributes in the third parameter to control how the window is opened. If the third parameter is not set it will simply open a new window with your own browser settings.

Here is an example with attributes:

$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 Me!', $atts);

Note

The above attributes are the function defaults so you only need to set the ones that are different from what you need. If you want the function to use all of its defaults simply pass an empty array in the third parameter: | | echo anchor\_popup(‘news/local/123’, ‘Click Me!’, array());

Note

The **window\_name** is not really an attribute, but an argument to the JavaScript window.open() <http://www.w3schools.com/jsref/met\_win\_open.asp> method, which accepts either a window name or a window target.

Note

Any other attribute than the listed above will be parsed as an HTML attribute to the anchor tag.

mailto(*$email*, *$title = ''*, *$attributes = ''*)[¶](#26in1rg)

| Parameters: | * **$email** (*string*) – E-mail address * **$title** (*string*) – Anchor title * **$attributes** (*mixed*) – HTML attributes |
| --- | --- |
| Returns: | A “mail to” hyperlink |
| Return type: | string |

Creates a standard HTML e-mail link. Usage example:

echo mailto('me@my-site.com', 'Click Here to Contact Me');

As with the [anchor()](#17dp8vu) tab above, you can set attributes using the third parameter:

$attributes = array('title' => 'Mail me');  
echo mailto('me@my-site.com', 'Contact Me', $attributes);

safe\_mailto(*$email*, *$title = ''*, *$attributes = ''*)[¶](#lnxbz9)

| Parameters: | * **$email** (*string*) – E-mail address * **$title** (*string*) – Anchor title * **$attributes** (*mixed*) – HTML attributes |
| --- | --- |
| Returns: | A spam-safe “mail to” hyperlink |
| Return type: | string |

Identical to the [mailto()](#26in1rg) function except it writes an obfuscated version of the *mailto* tag using ordinal numbers written with JavaScript to help prevent the e-mail address from being harvested by spam bots.

auto\_link(*$str*, *$type = 'both'*, *$popup = FALSE*)[¶](#35nkun2)

| Parameters: | * **$str** (*string*) – Input string * **$type** (*string*) – Link type (‘email’, ‘url’ or ‘both’) * **$popup** (*bool*) – Whether to create popup links |
| --- | --- |
| Returns: | Linkified string |
| Return type: | string |

Automatically turns URLs and e-mail addresses contained in a string into links. Example:

$string = auto\_link($string);

The second parameter determines whether URLs and e-mails are converted or just one or the other. Default behavior is both if the parameter is not specified. E-mail links are encoded as [safe\_mailto()](#lnxbz9) as shown above.

Converts only URLs:

$string = auto\_link($string, 'url');

Converts only e-mail addresses:

$string = auto\_link($string, 'email');

The third parameter determines whether links are shown in a new window. The value can be TRUE or FALSE (boolean):

$string = auto\_link($string, 'both', TRUE);

url\_title(*$str*, *$separator = '-'*, *$lowercase = FALSE*)[¶](#1ksv4uv)

| Parameters: | * **$str** (*string*) – Input string * **$separator** (*string*) – Word separator * **$lowercase** (*bool*) – Whether to transform the output string to lower-case |
| --- | --- |
| Returns: | URL-formatted string |
| Return type: | string |

Takes a string as input and creates a human-friendly URL string. This is useful if, for example, you have a blog in which you’d like to use the title of your entries in the URL. Example:

$title = "What's wrong with CSS?";  
$url\_title = url\_title($title);  
// Produces: Whats-wrong-with-CSS

The second parameter determines the word delimiter. By default dashes are used. Preferred options are: **-** (dash) or **\_** (underscore)

Example:

$title = "What's wrong with CSS?";  
$url\_title = url\_title($title, 'underscore');  
// Produces: Whats\_wrong\_with\_CSS

Note

Old usage of ‘dash’ and ‘underscore’ as the second parameter is DEPRECATED.

The third parameter determines whether or not lowercase characters are forced. By default they are not. Options are boolean TRUE/FALSE.

Example:

$title = "What's wrong with CSS?";  
$url\_title = url\_title($title, 'underscore', TRUE);  
// Produces: whats\_wrong\_with\_css

prep\_url(*$str = ''*)[¶](#44sinio)

| Parameters: | * **$str** (*string*) – URL string |
| --- | --- |
| Returns: | Protocol-prefixed URL string |
| Return type: | string |

This function will add http:// in the event that a protocol prefix is missing from a URL.

Pass the URL string to the function like this:

$url = prep\_url('example.com');

redirect(*$uri = ''*, *$method = 'auto'*, *$code = NULL*)[¶](#2jxsxqh)

| Parameters: | * **$uri** (*string*) – URI string * **$method** (*string*) – Redirect method (‘auto’, ‘location’ or ‘refresh’) * **$code** (*string*) – HTTP Response code (usually 302 or 303) |
| --- | --- |
| Return type: | void |

Does a “header redirect” to the URI specified. If you specify the full site URL that link will be built, but for local links simply providing the URI segments to the controller you want to direct to will create the link. The function will build the URL based on your config file values.

The optional second parameter allows you to force a particular redirection method. The available methods are **auto**, **location** and **refresh**, with location being faster but less reliable on IIS servers. The default is **auto**, which will attempt to intelligently choose the method based on the server environment.

The optional third parameter allows you to send a specific HTTP Response Code - this could be used for example to create 301 redirects for search engine purposes. The default Response Code is 302. The third parameter is *only* available with **location** redirects, and not *refresh*. Examples:

if ($logged\_in == FALSE)  
{  
 redirect('/login/form/');  
}  
  
// with 301 redirect  
redirect('/article/13', 'location', 301);

Note

In order for this function to work it must be used before anything is outputted to the browser since it utilizes server headers.

Note

For very fine grained control over headers, you should use the [*Output Library*](http://docs.google.com/libraries/output.html) set\_header() method.

Note

To IIS users: if you hide the Server HTTP header, the *auto* method won’t detect IIS, in that case it is advised you explicitly use the **refresh** method.

Note

When the **location** method is used, an HTTP status code of 303 will *automatically* be selected when the page is currently accessed via POST and HTTP/1.1 is used.

Important

This function will terminate script execution.

[Next](http://docs.google.com/xml_helper.html)   [Previous](http://docs.google.com/typography_helper.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).