* [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](#gjdgxs)
  + [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](http://docs.google.com/url_helper.html)
  + [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](#gjdgxs)
  + [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](http://docs.google.com/url_helper.html)
  + [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) »
* HTML Helper

classic layout

HTML Helper[**¶**](#30j0zll)

The HTML Helper file contains functions that assist in working with HTML.

* [Loading this Helper](#1fob9te)
* [Available Functions](#3znysh7)

[**Loading this Helper**](#2et92p0)[**¶**](#1fob9te)

This helper is loaded using the following code:

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

[**Available Functions**](#tyjcwt)[**¶**](#3znysh7)

The following functions are available:

heading([*$data = ''*[, *$h = '1'*[, *$attributes = ''*]]])[¶](#3dy6vkm)

|  |  |
| --- | --- |
| Parameters: | * **$data** (*string*) – Content * **$h** (*string*) – Heading level * **$attributes** (*mixed*) – HTML attributes |
| Returns: | HTML heading tag |
| Return type: | string |

Lets you create HTML heading tags. The first parameter will contain the data, the second the size of the heading. Example:

echo heading('Welcome!', 3);

The above would produce: <h3>Welcome!</h3>

Additionally, in order to add attributes to the heading tag such as HTML classes, ids or inline styles, a third parameter accepts either a string or an array:

echo heading('Welcome!', 3, 'class="pink"');  
echo heading('How are you?', 4, array('id' => 'question', 'class' => 'green'));

The above code produces:

<h3 class="pink">Welcome!<h3>  
<h4 id="question" class="green">How are you?</h4>

img([*$src = ''*[, *$index\_page = FALSE*[, *$attributes = ''*]]])[¶](#1t3h5sf)

|  |  |
| --- | --- |
| Parameters: | * **$src** (*string*) – Image source data * **$index\_page** (*bool*) – Whether to treat $src as a routed URI string * **$attributes** (*array*) – HTML attributes |
| Returns: | HTML image tag |
| Return type: | string |

Lets you create HTML <img /> tags. The first parameter contains the image source. Example:

echo img('images/picture.jpg'); // gives <img src="http://site.com/images/picture.jpg" />

There is an optional second parameter that is a TRUE/FALSE value that specifics if the *src* should have the page specified by $config['index\_page'] added to the address it creates. Presumably, this would be if you were using a media controller:

echo img('images/picture.jpg', TRUE); // gives <img src="http://site.com/index.php/images/picture.jpg" alt="" />

Additionally, an associative array can be passed to the img() function for complete control over all attributes and values. If an *alt* attribute is not provided, CodeIgniter will generate an empty string.

Example:

$image\_properties = array(  
 'src' => 'images/picture.jpg',  
 'alt' => 'Me, demonstrating how to eat 4 slices of pizza at one time',  
 'class' => 'post\_images',  
 'width' => '200',  
 'height'=> '200',  
 'title' => 'That was quite a night',  
 'rel' => 'lightbox'  
);  
  
img($image\_properties);  
// <img src="http://site.com/index.php/images/picture.jpg" alt="Me, demonstrating how to eat 4 slices of pizza at one time" class="post\_images" width="200" height="200" title="That was quite a night" rel="lightbox" />

link\_tag([*$href = ''*[, *$rel = 'stylesheet'*[, *$type = 'text/css'*[, *$title = ''*[, *$media = ''*[, *$index\_page = FALSE*]]]]]])[¶](#4d34og8)

|  |  |
| --- | --- |
| Parameters: | * **$href** (*string*) – What are we linking to * **$rel** (*string*) – Relation type * **$type** (*string*) – Type of the related document * **$title** (*string*) – Link title * **$media** (*string*) – Media type * **$index\_page** (*bool*) – Whether to treat $src as a routed URI string |
| Returns: | HTML link tag |
| Return type: | string |

Lets you create HTML <link /> tags. This is useful for stylesheet links, as well as other links. The parameters are *href*, with optional *rel*, *type*, *title*, *media* and *index\_page*.

*index\_page* is a boolean value that specifies if the *href* should have the page specified by $config['index\_page'] added to the address it creates.

Example:

echo link\_tag('css/mystyles.css');  
// gives <link href="http://site.com/css/mystyles.css" rel="stylesheet" type="text/css" />

Further examples:

echo link\_tag('favicon.ico', 'shortcut icon', 'image/ico');  
// <link href="http://site.com/favicon.ico" rel="shortcut icon" type="image/ico" />  
  
echo link\_tag('feed', 'alternate', 'application/rss+xml', 'My RSS Feed');  
// <link href="http://site.com/feed" rel="alternate" type="application/rss+xml" title="My RSS Feed" />

Additionally, an associative array can be passed to the link() function for complete control over all attributes and values:

$link = array(  
 'href' => 'css/printer.css',  
 'rel' => 'stylesheet',  
 'type' => 'text/css',  
 'media' => 'print'  
);  
  
echo link\_tag($link);  
// <link href="http://site.com/css/printer.css" rel="stylesheet" type="text/css" media="print" />

ul(*$list*[, *$attributes = ''*])[¶](#2s8eyo1)

|  |  |
| --- | --- |
| Parameters: | * **$list** (*array*) – List entries * **$attributes** (*array*) – HTML attributes |
| Returns: | HTML-formatted unordered list |
| Return type: | string |

Permits you to generate unordered HTML lists from simple or multi-dimensional arrays. Example:

$list = array(  
 'red',  
 'blue',  
 'green',  
 'yellow'  
);  
  
$attributes = array(  
 'class' => 'boldlist',  
 'id' => 'mylist'  
);  
  
echo ul($list, $attributes);

The above code will produce this:

<ul class="boldlist" id="mylist">  
 <li>red</li>  
 <li>blue</li>  
 <li>green</li>  
 <li>yellow</li>  
</ul>

Here is a more complex example, using a multi-dimensional array:

$attributes = array(  
 'class' => 'boldlist',  
 'id' => 'mylist'  
);  
  
$list = array(  
 'colors' => array(  
 'red',  
 'blue',  
 'green'  
 ),  
 'shapes' => array(  
 'round',  
 'square',  
 'circles' => array(  
 'ellipse',  
 'oval',  
 'sphere'  
 )  
 ),  
 'moods' => array(  
 'happy',  
 'upset' => array(  
 'defeated' => array(  
 'dejected',  
 'disheartened',  
 'depressed'  
 ),  
 'annoyed',  
 'cross',  
 'angry'  
 )  
 )  
);  
  
echo ul($list, $attributes);

The above code will produce this:

<ul class="boldlist" id="mylist">  
 <li>colors  
 <ul>  
 <li>red</li>  
 <li>blue</li>  
 <li>green</li>  
 </ul>  
 </li>  
 <li>shapes  
 <ul>  
 <li>round</li>  
 <li>suare</li>  
 <li>circles  
 <ul>  
 <li>elipse</li>  
 <li>oval</li>  
 <li>sphere</li>  
 </ul>  
 </li>  
 </ul>  
 </li>  
 <li>moods  
 <ul>  
 <li>happy</li>  
 <li>upset  
 <ul>  
 <li>defeated  
 <ul>  
 <li>dejected</li>  
 <li>disheartened</li>  
 <li>depressed</li>  
 </ul>  
 </li>  
 <li>annoyed</li>  
 <li>cross</li>  
 <li>angry</li>  
 </ul>  
 </li>  
 </ul>  
 </li>  
</ul>

ol(*$list*, *$attributes = ''*)[¶](#17dp8vu)

|  |  |
| --- | --- |
| Parameters: | * **$list** (*array*) – List entries * **$attributes** (*array*) – HTML attributes |
| Returns: | HTML-formatted ordered list |
| Return type: | string |

Identical to [ul()](#2s8eyo1), only it produces the <ol> tag for ordered lists instead of <ul>.

meta([*$name = ''*[, *$content = ''*[, *$type = 'name'*[, *$newline = "n"*]]]])[¶](#3rdcrjn)

|  |  |
| --- | --- |
| Parameters: | * **$name** (*string*) – Meta name * **$content** (*string*) – Meta content * **$type** (*string*) – Meta type * **$newline** (*string*) – Newline character |
| Returns: | HTML meta tag |
| Return type: | string |

Helps you generate meta tags. You can pass strings to the function, or simple arrays, or multidimensional ones.

Examples:

echo meta('description', 'My Great site');  
// Generates: <meta name="description" content="My Great Site" />  
  
echo meta('Content-type', 'text/html; charset=utf-8', 'equiv');  
// Note the third parameter. Can be "equiv" or "name"  
// Generates: <meta http-equiv="Content-type" content="text/html; charset=utf-8" />  
  
echo meta(array('name' => 'robots', 'content' => 'no-cache'));  
// Generates: <meta name="robots" content="no-cache" />  
  
$meta = array(  
 array(  
 'name' => 'robots',  
 'content' => 'no-cache'  
 ),  
 array(  
 'name' => 'description',  
 'content' => 'My Great Site'  
 ),  
 array(  
 'name' => 'keywords',  
 'content' => 'love, passion, intrigue, deception'  
 ),  
 array(  
 'name' => 'robots',  
 'content' => 'no-cache'  
 ),  
 array(  
 'name' => 'Content-type',  
 'content' => 'text/html; charset=utf-8', 'type' => 'equiv'  
 )  
);  
  
echo meta($meta);  
// Generates:  
// <meta name="robots" content="no-cache" />  
// <meta name="description" content="My Great Site" />  
// <meta name="keywords" content="love, passion, intrigue, deception" />  
// <meta name="robots" content="no-cache" />  
// <meta http-equiv="Content-type" content="text/html; charset=utf-8" />

doctype([*$type = 'xhtml1-strict'*])[¶](#26in1rg)

|  |  |
| --- | --- |
| Parameters: | * **$type** (*string*) – Doctype name |
| Returns: | HTML DocType tag |
| Return type: | string |

Helps you generate document type declarations, or DTD’s. XHTML 1.0 Strict is used by default, but many doctypes are available.

Example:

echo doctype(); // <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">  
  
echo doctype('html4-trans'); // <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">

The following is a list of doctype choices. These are configurable, and pulled from application/config/doctypes.php

|  |  |  |
| --- | --- | --- |
| Document type | Option | Result |
| XHTML 1.1 | xhtml11 | <!DOCTYPE html PUBLIC “-//W3C//DTD XHTML 1.1//EN” “<http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd>”> |
| XHTML 1.0 Strict | xhtml1-strict | <!DOCTYPE html PUBLIC “-//W3C//DTD XHTML 1.0 Strict//EN” “<http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd>”> |
| XHTML 1.0 Transitional | xhtml1-trans | <!DOCTYPE html PUBLIC “-//W3C//DTD XHTML 1.0 Transitional//EN” “<http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd>”> |
| XHTML 1.0 Frameset | xhtml1-frame | <!DOCTYPE html PUBLIC “-//W3C//DTD XHTML 1.0 Frameset//EN” “<http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd>”> |
| XHTML Basic 1.1 | xhtml-basic11 | <!DOCTYPE html PUBLIC “-//W3C//DTD XHTML Basic 1.1//EN” “<http://www.w3.org/TR/xhtml-basic/xhtml-basic11.dtd>”> |
| HTML 5 | html5 | <!DOCTYPE html> |
| HTML 4 Strict | html4-strict | <!DOCTYPE HTML PUBLIC “-//W3C//DTD HTML 4.01//EN” “<http://www.w3.org/TR/html4/strict.dtd>”> |
| HTML 4 Transitional | html4-trans | <!DOCTYPE HTML PUBLIC “-//W3C//DTD HTML 4.01 Transitional//EN” “<http://www.w3.org/TR/html4/loose.dtd>”> |
| HTML 4 Frameset | html4-frame | <!DOCTYPE HTML PUBLIC “-//W3C//DTD HTML 4.01 Frameset//EN” “<http://www.w3.org/TR/html4/frameset.dtd>”> |
| MathML 1.01 | mathml1 | <!DOCTYPE math SYSTEM “<http://www.w3.org/Math/DTD/mathml1/mathml.dtd>”> |
| MathML 2.0 | mathml2 | <!DOCTYPE math PUBLIC “-//W3C//DTD MathML 2.0//EN” “<http://www.w3.org/Math/DTD/mathml2/mathml2.dtd>”> |
| SVG 1.0 | svg10 | <!DOCTYPE svg PUBLIC “-//W3C//DTD SVG 1.0//EN” “<http://www.w3.org/TR/2001/REC-SVG-20010904/DTD/svg10.dtd>”> |
| SVG 1.1 Full | svg11 | <!DOCTYPE svg PUBLIC “-//W3C//DTD SVG 1.1//EN” “<http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd>”> |
| SVG 1.1 Basic | svg11-basic | <!DOCTYPE svg PUBLIC “-//W3C//DTD SVG 1.1 Basic//EN” “<http://www.w3.org/Graphics/SVG/1.1/DTD/svg11-basic.dtd>”> |
| SVG 1.1 Tiny | svg11-tiny | <!DOCTYPE svg PUBLIC “-//W3C//DTD SVG 1.1 Tiny//EN” “<http://www.w3.org/Graphics/SVG/1.1/DTD/svg11-tiny.dtd>”> |
| XHTML+MathML+SVG (XHTML host) | xhtml-math-svg-xh | <!DOCTYPE html PUBLIC “-//W3C//DTD XHTML 1.1 plus MathML 2.0 plus SVG 1.1//EN” “<http://www.w3.org/2002/04/xhtml-math-svg/xhtml-math-svg.dtd>”> |
| XHTML+MathML+SVG (SVG host) | xhtml-math-svg-sh | <!DOCTYPE svg:svg PUBLIC “-//W3C//DTD XHTML 1.1 plus MathML 2.0 plus SVG 1.1//EN” “http://www.w3.org/2002/04/xhtml-math-svg/xhtml-math-svg.dtd”> |
| XHTML+RDFa 1.0 | xhtml-rdfa-1 | <!DOCTYPE html PUBLIC “-//W3C//DTD XHTML+RDFa 1.0//EN” “<http://www.w3.org/MarkUp/DTD/xhtml-rdfa-1.dtd>”> |
| XHTML+RDFa 1.1 | xhtml-rdfa-2 | <!DOCTYPE html PUBLIC “-//W3C//DTD XHTML+RDFa 1.1//EN” “<http://www.w3.org/MarkUp/DTD/xhtml-rdfa-2.dtd>”> |

br([*$count = 1*])[¶](#lnxbz9)

|  |  |
| --- | --- |
| Parameters: | * **$count** (*int*) – Number of times to repeat the tag |
| Returns: | HTML line break tag |
| Return type: | string |

Generates line break tags (<br />) based on the number you submit. Example:

echo br(3);

The above would produce:

<br /><br /><br />

Note

This function is DEPRECATED. Use the native str\_repeat() in combination with <br /> instead.

nbs([*$num = 1*])[¶](#35nkun2)

|  |  |
| --- | --- |
| Parameters: | * **$num** (*int*) – Number of space entities to produce |
| Returns: | A sequence of non-breaking space HTML entities |
| Return type: | string |

Generates non-breaking spaces ( ) based on the number you submit. Example:

echo nbs(3);

The above would produce:

Note

This function is DEPRECATED. Use the native str\_repeat() in combination with   instead.

[Next](http://docs.google.com/inflector_helper.html)   [Previous](http://docs.google.com/form_helper.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).