



The Handbook

Codex

[http://codex.wordpress.org/Site Architecture 1.5](http://codex.wordpress.org/Site_Architecture_1.5)

Version Date

8 August 2005

Site Architecture 1.5

The following is a description of the general site architecture for WordPress v1.5. WordPress Theme authors are encouraged to maintain much of the core site architecture of XHTML tags and CSS selectors, but they are not required to. Therefore, this is just a general outline and your Theme may be different.

Template Driven Pages

Before we get to the [Core Structure](#) of the WordPress page architecture, you need to understand that WordPress uses [template files](#) to generate the final page "look" and content. For example, when viewing the front page of your WordPress site, you are actually viewing several template files:

- `index.php`
- `header.php`
- `sidebar.php`
- `footer.php`

When you view a single post page, you might be viewing the following template files:

- `single.php`
- `header.php`
- `sidebar.php`
- `footer.php`
- `comments.php`

On a multi-post page like categories, archives, and search, you might be viewing any combination of the following template files:

- `index.php`
- `category.php`
- `404.php`
- `search.php`
- `header.php`
- `sidebar.php`
- `footer.php`

We've specified which CSS selectors belong in which template files as much as possible in the following architecture specifications.

Core Structure

The core structure of a WordPress site represents the main containers which hold the page's content. The core structure of a WordPress site features, at a minimum, are:

- Header
- Sidebar/Menu
- Content

- Footer

These are the main containers in which the most important parts of the page are "contained". Remember, the core structure is like building blocks. They are dependent upon each other. If you change one, you have to change the others.

Classic Theme

```
<body>
<div id="rap">
<h1 id="header"></h1>
<div id="content"></div>
<div id="menu"></div>
<p class="credit"></p>
</div>
</body>
```

Default Theme

```
<body>
<div id="page">
<div id="header"></div>
<div id="content" class="narrowcolumn"></div>
<div id="sidebar"></div>
<div id="footer"></div>
</div><!-- end page -->
</body>
```

While one calls their sidebar `sidebar` and the other `menu`, the main difference between the two Theme's core structures is the use of the header and footer. For the Classic Theme, the header is in an `h1` tag and the footer is in a paragraph tag. In the Default Theme, the header is in a `div` called `header` and the footer is in the `footer` `div`.

Both Themes feature a container that encompasses or "wraps" itself around the entire page. This wrapping container allows for more definitive control over the overall structure, often used in combination with the `body` tag. In different WordPress Themes, these are also found by these names:

- container
- page
- wrap
- rap

Some Themes may add a second, third, or even fourth sidebar, creating a column effect. Or they may include additional wrappers around the entire page or specific containers, but this is the core structure.

The Modular Template Files

Based upon the premise of building blocks, WordPress Themes divide up the core structure between blocks called [template files](#). These are the template files:

- Header - `header.php`
- Sidebar/Menu - `sidebar.php`
- Content - `index.php`, `single.php`, `page.php`, `category.php`, `author.php`, `search.php`, etc.
- Footer - `footer.php`

As you can see, the content container can be found across many other template files. These are generated dependent upon the user's request. If they click on a category, the category template is displayed. If they choose a [Page](#), the page template is used. And so on.

Combined with the [WordPress Loop](#) and queries, a variety of templates can be generated, and the web page designer can style all of these differently and independently from each other.

Interior Structures

Within these core structural containers are smaller building blocks that hold the specific content within the parent container. WordPress Themes can feature a variety of these, but we are only going to concentrate on the two Themes that come with WordPress. It is from these two Themes that most WordPress Themes are based.

Header

The header is the structure that traditionally sits at the top of a web page. It contains the title of the website. It may also be referred to as the *masthead*, *head*, *title*, and *banner*. In all WordPress Themes, the `header` is found within the `header.php` template file.

The Classic Theme features the simplest header code:

```
<h1 id="header"></h1>
```

The Default Theme has a more complex header code:

```
<div id="header">
  <div id="headerimg">
    <h1></h1>
    <div class="description"></div>
  </div>
</div>
```

While the styles for the Classic Theme are found within the Theme's `style.css` style sheet file, styles for the Default Theme are found within the `style.css` and the `<head>` of the `header.php` [template file](#). Working with these styles is extensively covered in [Designing Headers](#).

Content

The content container in WordPress plays the most important role. It holds the [WordPress Loop](#) which dictates the generation of content on the page depending upon the request by the user.

The Classic Theme has the simplest content structure:

```
<div id="content">
  <h2>Date</h2>
  <div class="post" id="post-1">
    <h3 class="storytitle">Post Title</h3>
    <div class="meta">Post Meta Data</div>
    <div class="storycontent">
```

```

        <p>Welcome to WordPress.</p>
    </div>
    <div class="feedback">Comments (2)</div>
</div>

```

The Classic Theme hosts containers for the Date, Title, Post Meta Data, Post Content, and Feedback (number of comments). It also showcases a powerful feature. The ability to individually style a single post's look.

```
<div class="post" id="post-1">
```

The `post` CSS class selector applies the `post` styles to this container, but it also has an ID which is generated automatically by WordPress. The code looks like this:

```
<div class="post" id="post-<?php the_ID(); ?>">
```

The use of the template tag `the_ID()` generates the ID number of the post. This provides a unique identifier for internal page links as well as for styles. This post could have a style for `post-1`, as could `post-2`. While it is a bit excessive to feature a style for every post, there may be a post or two you need to have look a little different. Some plugins may use this identifier to automatically change the look of different posts, too.

The Default Theme content container features a different look for **multi-post views** like the front page, categories, archives, and searches, and a different **single post view** for single posts. The multi-post view looks like this:

```

<div id="content" class="narrowcolumn">
    <div class="post" id="post-1">
        <h2>Post Title</h2>
        <small>Date</small>
        <div class="entry">
            <p>Post Content.</p>
        </div>
    <p class="postmetadata">Post Meta Data Section</p>
    </div>
    <div class="navigation">
        <div class="alignleft">Previous Post</div>
        <div class="alignright">Next Post</div>
    </div>
</div>

```

There is a lot going on here. Let's break it down.

```
<div id="content" class="narrowcolumn">
```

In the **multi-post views**, it features a `class="narrowcolumn"` and in the **single post views**, it features `class="widecolumn"` and the sidebar is not generated on that page, allowing the post to be viewed "wide" across the width of the content area.

```
<div class="post" id="post-1">
```

Like the Class Theme, this division sets up the style for `post` and the identifier for `post-X`, with `X` representing the post's unique ID number. This allows for customizing the specific post's look.

```
<h2>Post Title</h2>
```

This encompasses the post's title code, styled by the `<h2>` tag.

```
<small>Date</small>
```

The date code is surrounded and styled by the `small` tag.

```
<div class="entry">
```

The post content is styled by a combination of the styles within the `entry` CSS selectors and the paragraph tag.

```
<p class="postmetadata">Post Meta Data Section</p>
```

The [Post Meta Data Section](#) contains the data details about the post such as the date, time, and categories the post belongs to.

```
<div class="navigation">
```

The [Next and Previous Links](#) are styled in the `navigation`. They also include classes for `alignleft` for the Previous Post and `alignright` for the Next Post in chronological order.

These elements are shifted around in the **single post view** content structure:

```
<div id="content" class="widecolumn">
  <div class="navigation">
    <div class="alignleft"></div>
    <div class="alignright"></div>
  </div>
  <div class="post" id="post-1">
    <h2>Post Title</h2>
    <div class="entrytext">
      <p>Post content.</p>
      <p class="postmetadata alt">
        <small>Post Meta Data</small>
      </p>
    </div>
  </div>
</div>
```

The `widecolumn` class is featured to stretch the content across the page to fill in the absence of the sidebar. The `navigation` has been moved up to the top. And the Post Meta Data is now incorporated into the `entrytext` parent container and styled differently with an `alt` style added.

These two examples from the Default Theme give you just a glimpse into the myriad ways your WordPress site can be customized.

Comments

Comments may be featured on the single post view or in a popup window. The overall styles for the two sets of comments remain basically the same. The two template files are `comments.php` and `comments-popup.php`

Classic Theme Comments

```
<h2 id="comments">1 Comment
  <a href="#postcomment" title="Leave a comment">>></a></h2>
  <ol id="commentlist">
    <li id="comment-1">
      <p>Hi, this is a comment.</p>
      <p><cite>Comment by Person</cite> </p>
    </li>
  </ol>
<p>
  <a href='http://example.com/archives/name-of-post/feed/'>
  <abbr title="Really Simple Syndication">RSS</abbr>
  feed for comments on this post.</a>
```

```

<a href="http://example.com/name-of-post/trackback/" rel="trackback">
TrackBack <abbr title="Uniform Resource Identifier">URI</abbr>
</a>
</p>
<h2 id="postcomment">Leave a comment</h2>
<form action="http://example.com/blog/wp-comments-post.php"
method="post" id="commentform">
<p>
<input type="text" name="author" id="author" value="" size="22"
tabindex="1" />
<label for="author">
<small>Name (required)</small>
</label>
</p>
<p>
<input type="text" name="email" id="email" value="" size="22"
tabindex="2" />
<label for="email">
<small>Mail (will not be published) required</small>
</label>
</p>
<p>
<input type="text" name="url" id="url" value="" size="22" tabindex="3"
/>
<label for="url">
<small>Website</small>
</label>
</p>
<p>
<small><strong>XHTML:</strong> List of Tags you
can use in comments</small>
</p>
<p>
<textarea name="comment" id="comment" cols="100%" rows="10"
tabindex="4">
</textarea>
</p>
<p>
<input name="submit" type="submit" id="submit" tabindex="5"
value="Submit Comment" />
<input type="hidden" name="comment_post_ID" value="1" />
</p>
</form>
</div>

```

While individual sections of the comments feature styling reference, the Classic Theme has no general comment division or group style reference, one could be easily added.

#comments h2

Styles the title at the top of the comments list which says "Comments 4 Leave a Comment", with the latter part of the sentence in a link that jumps to `<h2 id="postcomment">Leave a comment</h2>`.

#comment-n

Comments are given a unique ID number, signified here by the letter n. This allows them to be styled individually or through the use of [Comment Plugins](#).

#comments ol

This begins the **ordered list** of the comments, counting down from one, and sets the overall style of the comments list.

#comments li

Style reference for each comment on the list.

#comments p

This paragraph tag styles the actual comments on the comment list.

#comment cite

This use of the `cite` controls the look of the commentor's name. It usually states "Name says:" in the comments list.

#comments h2 or #postcomment

The `h2` heading can be styled two ways, as `#comments h2` or `#postcomment`. The latter is used by the "Leave a Comment" link from the top of the comments section, too.

#commentform

Style reference for the overall "form" for inputting comments. Each input area has it's own ID.

#author

ID reference for the comment author's input area.

#comments small

The `<small>` tag is used in several places in the Classic Theme. This usage surrounds the text in the **comment submit form** and the text for the **list of tags** that can be used in the comment.

#email

ID reference for the comment author's email.

#url

ID reference for the comment author's URL.

#comment

ID reference for the comment input textarea. It does not style the final generated comment, just the input box.

#comment #submit

There are two submit buttons in the Classic Theme, for search and comment submissions. This is the submit comment button.

Default Theme Comments

The Default Theme comments feature a loop query within the `comments.php` and `comments-popup.php` which changes some of the information depending upon if comments are open, closed, and any present. If the comments are open or closed and no comments have been made, this information will be displayed within the `<h3 id="comments">` tag.

```
<h3 id="comments">One Response to "Hello world!"</h3>
<ol class="commentlist">
  <li class="alt" id="comment-1">
    <cite>
      <a href="http://example.org/" rel="external nofollow">Mr WordPress</a>
    </cite> Says:<br />
    <small class="commentmetadata">
      <a href="#comment-1" title="">Date and Time</a>
    </small>
    <p>Hi, this is a comment.</p>
  </li>
</ol>
<h3 id="respond">Leave a Reply</h3>
<form action="http://example.com/blog/wp-comments-post.php" method="post"
id="commentform">
  <p>
    <input name="author" id="author" value="" size="22" tabindex="1"
```



```

type="text">
    <label for="author">
        <small>Name (required)</small>
    </label>
</p>
<p>
    <input name="email" id="email" value="" size="22" tabindex="2"
type="text">
        <label for="email">
            <small>Mail (will not be published) required</small>
        </label>
</p>
<p>
    <input name="url" id="url" value="" size="22" tabindex="3" type="text">
        <label for="url">
            <small>Website</small>
        </label>
</p>
<p>
    <small><strong>XHTML:</strong> You can use these
tags:....</small>
</p>
<p>
    <textarea name="comment" id="comment" cols="100" rows="10" tabindex="4">
</textarea>
</p>
<p>
    <input name="submit" id="submit" tabindex="5" value="Submit Comment"
type="submit">
    <input name="comment_post_ID" value="1" type="hidden">
</p>
</form>
</div>

```

While individual sections of the comments feature styling reference, the Default Theme has no general comment division or group style reference, though one could be easily added.

#comments

Styles the `<h3>` tag for the "number of responses to the post" heading.

#commentlist ol

Styles the "ordered list" of the comments list.

.alt li and #comment-n

The comment list items have two style references. The first one is the class `alt` and the second is the comment ID number signified here by the letter `n`. This allows them to be styled individually or through the use of [Comment Plugins](#).

cite

The tag `cite` frames the "Name says:" and link to the comment author's URL.

.commentmetadata small

The `<small>` tag has a class of `commentmetadata` which allows the date and time of the post to be styled.

ol #commentlist p

Styles the paragraph within the ordered list of comments.

#respond h3

Styles the heading for "Leave a Reply".

#commentform

Style reference for the overall "form" for inputting comments. Each input area has it's own ID.

#author

ID reference for the comment author's input area.

#comments small

The `<small>` tag is used in several places in the Classic Theme. This usage surrounds the text in the **comment submit form** *and the text for the list of tags* that can be used in the comment.

#email

ID reference for the comment author's email.

#url

ID reference for the comment author's URL.

#comment

ID reference for the comment input textarea. It does not style the final generated comment, just the input box.

#comment #submit

There are two submit buttons in the Classic Theme, for search and comment submissions. This is the submit comment button.

Popup Comments

The Classic and Default Themes' `comments-popup.php` template file is essentially the same. They use the layout for the [Classic Theme comment structure](#). While the Classic Theme uses `<h2>` headings and the Default Theme uses `<h3>` headings for the title headings in their comments, in the `comments-popup.php` template file, they both use the `<h2>` heading tag.

```
<body id="commentpopup">
<h1 id="header"></h1>
<h2 id="comments">Comments</h2>
....Classic Theme comment section.....
...Classic Theme footer....
```

The `body` tag sets the style for the overall page with `#commentpopup`. The `h2` heading begins the comments section.

If you make modifications to the structure of the tags within the header and footer of the overall Theme, ensure those structural changes are applied to the comments popup template, especially if you will be [releasing the Theme to the public](#).

Sidebar

As you saw with the Default Theme, the sidebar can be visible or not, depending upon the template file in use. The sidebar, in general, can be simple or complex. WordPress Themes often feature information within the sidebar in **nested lists**. There is a step-by-step guide for the sidebar at [Customizing Your Sidebar](#) and more information on [Styling Lists with CSS](#), too.

In general, the WordPress sidebar features titles of the various sections within a list, with the section items in a nested list below the title.

The Classic Theme sidebar looks like this, with the links removed for simplification:

```
<div id="menu">
  <ul>
```

```

<li class="pagenav">Pages
  <ul>
    <li class="page_item">Contact</li>
    <li class="page_item">About</li>
  </ul>
</li>
<li id="linkcat-1"><h2>Blogroll</h2>
  <ul>
    <li>Blogroll Link 1</li>
    <li>Blogroll Link 1</li>
    <li>Blogroll Link 1</li>
  </ul>
</li>
<li id="categories">Categories:
  <ul>
    <li>Category Link 1</li>
    <li>Category Link 2</li>
  </ul>
</li>
<li id="search">
  <label for="s">Search:</label>
  <form id="searchform" method="get" action="/index.php">
<div>
  <input type="text" name="s" id="s" size="15" /><br />
  <input type="submit" value="Search" />
</div>
</form>
</li>
<li id="archives">Archives:
  <ul>
    <li>Archives Month Link 1</li>
    <li>Archives Month Link 2</li>
  </ul>
</li>
<li id="meta">Meta:
  <ul>
    <li>RSS Feed Link</li>
    <li>RSS Comments Feed Link</li>
    <li>XHTML Validator</li>
    <li>XFN Link</li>
    <li>WordPress Link</li>
  </ul>
</li>
</ul>
</div>

```

Most of these are self-explanatory. Each set of links has its own CSS selector: [Pages](#), categories, archives, search, and meta.

Pages and Link Categories

The [Pages](#) and [Links](#) category, labeled "Blogroll", uses the `<?php get_links_list(); ?>` and `<?php wp_list_pages(); ?>` template tags which automatically generates a heading.

For the **Links** category, it generates an `h2` heading for that set of links. This means you can style the `menu h2` heading to look differently from the rest of the headings, or, if you want them to all look the same, make sure that the `menu h2` style *matches* the rest of the category styles which are not automatically generated.

The **Pages** template tag generates `pagenav` as the heading and then identifies the pages in a new way. As a general list viewed on multi-post and single post views, the Page list items feature a `class="page_item"` to style those links. When viewing an individual Page, that Page's link will change to `class="current_page_item"`, which can then be styled to look differently from the rest of the Page links.

Categories, Archives, and Meta

The other sidebar section titles, *categories*, *archives*, *meta*, and others, do not use template tags which generate their own titles. These are set inside of PHP statements which "print" the text on the page. While these could be put inside of [heading tags](#), WordPress uses the `_e()` function to display or "echo" the text titles while also marking the text as a possible target for language translation. If you will be [developing your theme](#) for [public release](#), using the echo functions is highly recommended.

You can style these individually or all the same. Some Themes, like the Default Theme, put all these in `<h2>` headings so the list headings will all look the same. Therefore, they may or may not use style references for each section. You may add them if you need them to change the look of each section of links.

Search Form

The search form is found within the `searchform.php`. It may be found in different locations within the sidebar. To style the overall search form, use the `search` ID. Here is a list of the individual areas of the search form which may be styled by default. You may add style classes to gain more control over the look of your search form.

```
<li id="search">
  <label for="s">Search:</label>
  <form id="searchform" method="get" action="/index.php">
    <div>
      <input type="text" name="s" id="s" size="15" /><br />
      <input type="submit" value="Search" />
    </div>
  </form>
```

`#search`

The overall style for the search form.

`#search label`

Used to style the `label` tag, if necessary.

`#searchform`

Used to style the form itself.

`#search div`

This unlabeled `div` is a child container of the parent container `search` and maybe styled from within that selector.

`#searchform input`

To style the input area for the search, this selector combination will work.

`#searchsubmit`

Used by the Default Theme, this selector may be used to style the **search** or **submit** button.

The search form area, input, and button can be styled in many ways, or left with the default input and "button" look.

Meta Feed Links

The Meta links may be shown as text or icons representing the various links. The XHTML and CSS validation links may use the W3 icons. The various Feeds can also be represented as icons. Or left as text. It's up to you. Use of the feeds within your sidebar with text or icons is covered by the article [WordPress Feeds](#).

Footer

The footer is found within the `footer.php` template file. In both the Default and Classic Themes, the footer contains little information.

Classic Theme

```
<p class="credit">
  <!--15 queries. 0.152 seconds. -->
  <cite>
    Powered by <a href='http://wordpress.org'
      title='Powered by WordPress, state-of-the-art
      semantic personal publishing platform.'>
      <strong>WordPress</strong></a>
  </cite>
</p>
</div>
```

The footer's content is styled with the `credit` class and the paragraph and `cite` tags.

The tag displays the number of mysql queries used on the page and the time it took for the page to load, in HTML commented code. It is there for the administrator's convenience and use. It is only visible within the page's source code. If you would like to display this visible on the page, remove the [comments](#). It's look will be influenced by the `credit` class style of the paragraph tag. On the template file, it looks like this:

```
<!--<?php echo $wpdb->num_queries; ?> queries.
<?php timer_stop(1); ?> seconds. -->
```

Default Theme

```
<div id="footer">
  <p>Blogging in the WordPress World
    is proudly powered by
    <a href="http://wordpress.org/">WordPress</a><br />
    <a href="feed:http://example.com/feed/">Entries (RSS)</a>
    and
    <a href="feed:http://example.com/comments/feed/">
    Comments (RSS)</a>.
  <!-- 18 queries. 0.186 seconds. -->
</p>
</div>
```

The Default Theme's footer is styled by the `footer` ID and the paragraph tag. While the footer area itself maybe styled by the `footer`, the paragraph tag controls the text within it. To style the paragraph tag differently within the `footer` than the rest of your page:

```
#footer p {styles}
```

Resources

- [CSS](#)
- [Finding Your CSS Styles](#)
- [CSS Troubleshooting](#)
- [Using Themes](#)
- [Theme Development](#)
- [Designing Themes for Public Release](#)
- [WordPress Lessons](#)
- [Blog Design and Layout](#)
- [Stepping Into Template Tags](#)