1、 WordPress 执行顺序

输入网址:

调用 index.php 文件

调用 wp-blog-header.php 文件

调用 wp-load.php 文件

wp-load.php 中调用 wp-config.php 文

wp-config.php 内调用 wp-settings.php

wp-settings.php 内包含所有需要包含的类与函数，并定义一个全局对象 $wp_query 与数据库交互。

调用 template-loader.php 文件，根据 $wp_query 执行的结果输出主题中对应的模板文件。在模板文件中可以直接引用$wp_query 及其相关变量值与方法。

注：可以使用的常量：

网站文件根目录绝对路径： ABSPATH;

网站文件的 wp-content 目录绝对路径：WP_CONTENT_DIR；

网站文件的语言文件目录绝对路径 WP_LANG_DIR

网站文件的插件目录绝对路径 WP_PLUGIN_DIR

网站的 wp-content 目录的 url：WP_CONTENT_URL；

网站的插件 plugins 目录的 url: WP_PLUGIN_URL;

网站的 wp-includes 目录绝对路径：ABSPATH . WPINC

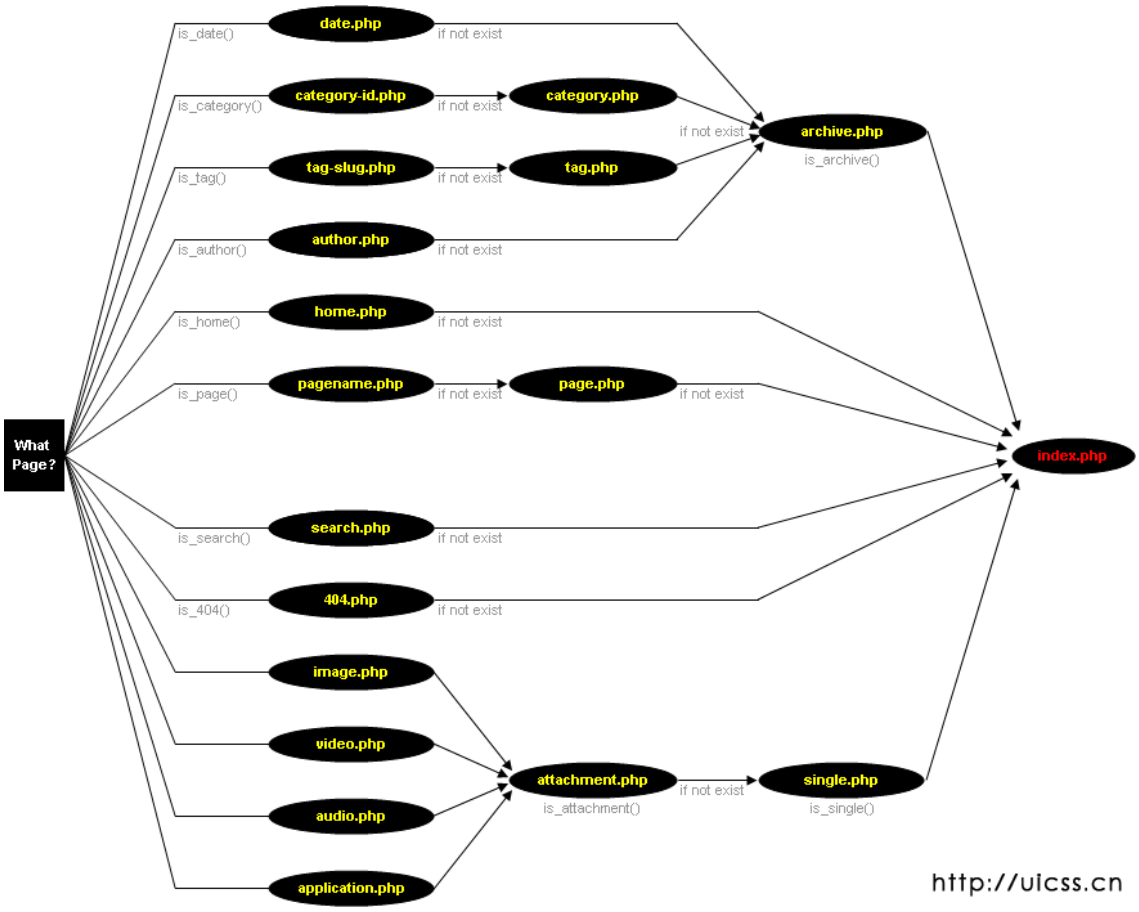
当前正在使用的主题的模板所在目录路径：TEMPLATEPATH

当前正在使用的模板样式所在目录路径：STYLESHEETPATH

（其他常量参考：wp-settings.php）

2、 资源：www.wordpress.la 、www.wordpress.org

3、 WordPress 的模板体系



上图是 wordpress 内置调用模板体系。

wordpress 根据调用链接的参数，在 template_loader.php 利用上述条件判断函数来自动判断出将要调用哪类模板。调用顺序如上图，即不存在的话就调用后面的模板。

在每个调用的模板文件内，存在许多的模板标签函数，用于输出该查询类别下的文章（wordpress 数据库中自定义页面与一般博文数据格式完全相同）。和其它一些参数如目录标题、链接等。这些模板标签函数都是基于$wp_query 对象对应的类对数据库的查询结果而存在。

4、 利用模板标签在模板中输出相应数据。

4.1 主题模板页面中文章循环输出（the loop）

Wordpress 用 the loop 来输出获取的文章列表（posts）中的每一篇文章。在 loop 内部利用相应的模板标签可以输出一篇博文相关的任何东西，比如标题，作者，发表时间，内容，摘要，标签(tag)，所属类别（栏目，category）等等。同时可以利用任何 html,css,php 来格式化这些数据。

the loop 可以用在任何模板文件中用于输出具体的文章和页面。比如 index.php, category.php, tag.php 等等。（下面的所用介绍都是基于使用内置模板文件）

4.2 主题模板中循环输出的一般语句格式：（模板页面内默认采用$wp_query 根据访问链接参数获得的查询结果）

```
<?php if ( have_posts() ) : while ( have_posts() ) : the_post(); ?>//循环开始，1.5-2.8 版本
<!—此处输出每篇文章信息 ， 是循环主题–>
<?php endwhile; else: ?>
<p><?php _e('Sorry, no posts matched your criteria.'); ?></p>//表示没有文章时的结果
<?php endif; ?>
```

用在循环主体之间的标签主要有：

the_title( );//输出此时的文章标题。有三个参数，$before,$after,$display,可以为空。

get_the_title();//返回给定 ID 的文章标题字符串。一个参数，$id.可以在循环体外用。

the_ID( );//输出此时的文章的 ID。没有参数。

get_the_ID();//返回此时的文章的 ID，是一个数值型，没有参数。

the_title_attribute();//输出此时文章的标题，但经过净化，包括去掉 html 标签等。有三个参数，$before,$after,$echo,可以为空，默认$echo 为 1，即输出。

the_content();//输出此时的文章内容.有三个参数。默认不填。

the_excerpt();//输出此时的文章摘要。没有参数。

the_tags();//输出这篇文章附加的标签（tags），每个标签带有链接，点击就会显示该标签所包含的所有文章。有三个参数，$before, $separator, $after.可以为空。

the_category();//输出这篇文章所属的分类（可能有多项）,同时带有链接，点击就会显示该分类下的所有文章。有二个参数，$separator, $parents.，可以为空。

the_author();//输出这篇文章的作者名，该作者档案中的显示名。没有参数。

the_author_link();//输出这篇文章的作者显示名，同时链接到该作者的主页。没有参数。

the_author_posts_link();//输出这篇文章的作者显示名，同时加有链接，点击就会显示该作者发表的所有文章。没有参数。

the_author_posts();//输出这篇文章的作者共发表文章的总数（草稿与私有文章不算）

the_time();//输出篇文章的发表时间，需要匹配时间格式参数。l, F j, Y,s, g:i a 等

the_date();//输出文章的发表日期，只是循环体中的第一篇文章的日期。四个参数，$format, $before, $after, $echo.

the_date_xml();//以 YYYY-MM-DD 的格式输出这篇文章的发表时间，无参数。

get_permalink();//得到文章单个页面显示的固定链接。在循环体内不带参数时，返回当前文章的固定链接,在循环体外不带参数时,返回循环体最后一篇文章的固定链接，带有参数$id 时，返回对应文章的固定链接。

the_permalink();//输出当前文章单个页面显示的固定链接。没有参数。

4.3 自定义查询结果

Wordpress 在主题模板中提供了至少三种自定义查询方式，用于在输出的模板页面中直接从数据库查询相关文章，构建自定义的循环输出（the loop）。

wordpress 每一次通过 url 访问都会实例化 WP_query 类，根据给定的参数获得查询结果。流程一般为：提供参数——>调用 WP_query 类实例化对象——>获得查询结果为博文列表。

自定义循环输出是为了获得与通过链接得到的默认查询结果不一样的博文列表。一般有三种方式：

1、query_posts();函数

此函数用来控制应该显示哪些类别的博文。实际上是先取消系统已有的全局变量$wp_query,然后重新实例化 WP_query 类，新的对象仍命名为$wp_query,根据新的参数从新查询，获得新的查询结果。可以说它是更新了$wp_query 的值，得到我们想要的自定义查询结果。

用处：

在主页上显示一篇文章；

显示一个特定时间、特定分类的所有文章。

在主页上显示最近发表的文章。

改变文章列表的显示排序。

排除一个或多个类别的文章。

可以再已有的单个页面中执行二级循环，已获得另外的文章列表等。

…

参数：两种方式给该函数传递参数：

（1） 字符串形式，可以在运行该函数之前，可以直接在括号中添加。

如：$query= “page_id=7”;query_posts($query);

或者：query_post(‘page_id=7”);//多个参数用&连接入”cat=22&order=ASC”;

(2) 数组形式：$query=array(‘cat’=>22,’order’=>’ASC’); query_posts($query);

参数总类：

（1） 类别参数（category）:显示某种类别的文章需设定的参数。

cat – 文章类别 ID。ID 为正时表示显示该类别文章，为负时，表示排除该类别，可以为多个 ID。

category_name：文章类别名字。

category__and ：文章同时属于那几个类别，需用 ID，同时参数用数组，如 array(2,3);

category__in:文章属于哪一个类别，之一即可。需用 ID，参数用数组，同上。

category__not_in ： 文章不属于哪几个类别。需用 ID，参数用数组，同上。

(2) 标签参数：（tags）

（3） 作者参数（author）

author:作者 ID。显示该作者的所有文章。为负数时表示排除该作者的文章。
author_name=作者名。显示该作者的所有文章。

（4） 文章与页面参数（page and post）

'p' => 27 ： 文章 ID
'name' => 'about-my-life' ： 请求文章别名（slug）为此的文章。
'page_id' => 7 ： 请求页面 ID 为 7 的页面。
'pagename' => 'about' ： note that this is not the page's title, but the page's path
'posts_per_page' => 1： 使用'posts_per_page' => 3 显示 3 篇文章. 使用 'posts_per_page' => -1 显示所用文章。
'showposts' => 1： 使用'showposts' => 3 显示 3 篇文章. 使用 'showposts' => -1 显示所有文章。不支持 posts_per_page。
'post__in' => array(5,12,2,14,7) ： 显示数组中这些 ID 对应的文章。
'post__not_in' => array(6,2,8) ： 排除数组中 ID 对应的这些文章。
'post_type' => 'page' ： 返回页面; 默认是 post;能够符的值： attachment, page, post, or revision.不能得到文章的修订版（revisions）。.
'post_status' => 'publish' – 返回状态为 publish 的作品. 也能够使用 pending, draft, future, private, trash.
'post_parent' => 93 –返回该 ID 对应的页面的子页面。.

（5） 时间参数：

hour= 小时 (from 0 to 23)
minute= 分钟 (from 0 to 60)
second= 秒钟 (0 to 60)
day= 一月的第几天 (from 1 to 31)
monthnum= 几月 (from 1 to 12)
year=4 位数年份 (e.g. 2009)
w= 一年的第几周(from 0 to 53) and uses the MySQL WEEK command Mode=1.

（6） 偏移参数与排序参数，联合参数符：

query_posts()函数与循环循环显示的一般用法：

```
<?php
//The Query
query_posts('posts_per_page=5');//设置查询参数，要保留已有的链接参数加上
        global $query_string;把它与新的查询参数合在一起。
//The Loop
if ( have_posts() ) : while ( have_posts() ) : the_post();
    <!—循环体，用前面已有的模板标签，与默认情况一致。-->
endwhile; else:
    <!—没有文章该怎么做-->
endif;
//Reset Query
wp_reset_query();//重置查询。破坏先前的查询，建立新的查询，避免再次使用
        query_posts()进行新的查询时出错。其实是恢复到链接参数的查询结果状
        态。
?>
```

2、**get_posts()函数**

这个函数是利用输入的参数，重新实例化一个 WP_query 对象，得到查询结果。完全与 $wp_query 无关。

查询结果返回一个数组，存放每一个文章对象。

一般用法：

```
<?php
 $posts = get_posts(’category=4&numberposts=5′ );//直接向数据库查询 POST 表。
if( $posts ) {
foreach( $posts as $post ) {//返回一个数组$posts,里面存放每一条文章对象。
//this is now the Loop
setup_postdata( $post );//将对象数据进行处理之后可以使用前面的那些 loop 之内
        的模板标签函数。也可以直接使用$post 对象输出数据，如$post->ID。
// print out the post using the_title(), the_link(), etc
}
}
?>
```

如何传递参数：

2.6 版本以上的参数传递方式与参数类型与 query_posts()函数完全一致。

3、直接使用 WP_Query()类来创建数据交互对象。

```
<?php
my_query = new WP_Query(’ category_name=featured&showposts=5′ );//新建一个查询对象，参数在后。参数设置与前面差别不大。
if ($my_query->have_posts()) {
while ($my_query->have_posts()) {
$my_query->the_post();
// output the post data.与前面的 loop 内模板标签一样的用法。
}
}
```