# wordpress主题开发

# 一、创立文件

/wp-content/themes/主题

新建index.php和style.css文件

# 二、设置主题

框架会自动识别主题，外观->主题

# 函数备用

wp\_trim\_words( $content, 66 ); //截取字符窜

get\_template\_directory() //获取主题路径，不直接输出

require\_once( );       // 引入文件

//按照这种方式所写的调用函数会判断模板文件夹下是否有 diy-one.php 如果没有则调用diy.php。

**get\_template\_part('diy','one')。**

# BUG

## 后台设置文章居中，前端页面没有效果

添加CSS样式，前端

.aligncenter img{

display: block;

margin-left: auto;

margin-right: auto;

}

.alignright img{

float: right;

margin-left: 10px;

}

.alignleft img{

float: left;

margin-right: 10px;

}

# 功能开发

## 基本功能

### （1）页面

/wp-content/themes/主题目录/page.php

页面模板默认为page.php ，

1. 创建自己的模板文件page-my.php,以page开头，可以放在主题目录下的任何位置
2. 添加注释，这个注释会解析成**模板名称**

/\*\*

\* Template Name: 成功案例

\*

\*\*/

1. 后台添加新页面，选择对应模板

### （2）文章

#### 首选：使用模板区分

/wp-content/themes/主题目录/single.php

文章会使用single.php做为默认模板

1. 创建自己的文章模板single-my.php
2. 添加注释，可以解析成模板名称

/\*

\* Template Name: 案例模板

\* Template Post Type: post

\*/

1. 后台添加新文章，选择对应为文章模板

#### 方案一：使用标签区分文章

/wp-content/themes/innofit/single.php

文章内容默认模板为content.php

1. 找到single.php中代码

//get\_post\_format()会返回使用的标签

//最终拼接的文件名为content-my.php

get\_template\_part( 'content', get\_post\_format());

2、复制content.php，重命名为content-my.php

3、修改content-my.php内容为想要的内容

4、**指定文章标签为指定标签**

**---------------------**

**通过在single.php文件中调用不同的模板，实现加载不同的模板**

#### 方案二：使用标签区分

/wp-content/themes/主题目录/funcion.php

1. 在文件最后添加

add\_action('template\_include', 'load\_single\_template');

function load\_single\_template($template) {

    $new\_template = '';

    // single post template 5

    if( is\_single() ) {

        //global $post;

        // 'wordpress' is category slugs 8

        // if( has\_term('wordpress', 'category', $post) ) {

        if( has\_post\_format('gallery')) { **//这里添加的是标签名**

            // 这里决定要加载的文件

            $new\_template = locate\_template(array('single-mylist.php' ));

        }

    }

    return ('' != $new\_template) ? $new\_template : $template;

}

1. 创建指定文件的标签

#### 启动文章标签

/wp-content/themes/主题目录/funcion.php

add\_theme\_support( 'post-formats',

array( 'aside', 'chat','gallery','image','link', 'quote', 'status', 'video', 'audio' )

);

显示所有的tab

### （3）分类目录

**使用别名区分分类**

/wp-content/themes/主题目录/category.php

分类目录模板默认为category.php，没有自己创建。

1、创建自己的模板，复制category.php，命名为category-my.php

2、修改category-my.php代码

//这里是加载同级目录下的content-addus.php文件

get\_template\_part( 'content-addus','' )

3、复制content.php，重命名为content-addus.php

4、修改content-addus.php内容为想要的内容

5、创建分类目录mylist(自定义)

6、**指定为目录别名为my(这里是明确使用哪一个分类模板，my名称需要与category-my.php的my一样**

**-------------**

**通过这个方法可以重新定义分类页面，直接重写categort-xxxx.php页面中的代码即可。**

### （4）查找主题加载的模板

/wp-includ/template-loader.php

if ( $template = apply\_filters( 'template\_include', $template ) ) {

    echo $template;exit(); //打印模板路径

    //include( $template );

} elseif ( current\_user\_can( 'switch\_themes' ) ) {

    $theme = wp\_get\_theme();

    if ( $theme->errors() ) {

        wp\_die( $theme->errors() );

    }

}

## 使用Ajax

### 后台注册

/wp-content/themes/主题目录/function.php

/\*\*

\* 处理请求数据

\*/

function content\_info() {

$post\_id = $\_POST['id'];

// $data = get\_post($post\_id, OBJECT)->post\_content; //根据ID获取文章

// 输出响应

header( "Content-Type: application/json" );

$data = array(

'info'=>'this is my test',

'success' => true,

);

echo json\_encode($data);

exit;//这个停止一定要写

}

//wp\_ajax\_nopriv\_ 是在没有登录的处理流程;wp\_ajax\_是登录后的处理流程  
//add\_action( ‘wp\_ajax\_nopriv\_\*\*\*’, ‘dosome’ );  
//add\_action( ‘wp\_ajax\_\*\*\*’, ‘dosome’ );

//函数名对应添加上，第一个表示用户没有登录时，这里全部都一样处理

add\_action( 'wp\_ajax\_nopriv\_content\_info', 'content\_info' ); //用户没登陆

add\_action( 'wp\_ajax\_content\_info', 'content\_info' ); //用户已登录

### 前台请求

//获取ajax处理地址，统一

$ajaxurl = "<?php echo admin\_url( 'admin-ajax.php' ); ?>"

var data={

"id": $id,

"action" : 'content\_info', //必填，action的参数要和请求的函数名一致

}

//正常的ajax

$.post($ajaxurl, data, function(response) {

console.log(response['info'])

});

## 导航栏自定义

### 注册导航栏

if(function\_exists('register\_nav\_menus')){

register\_nav\_menus( //注册菜单栏

array(

'header-menu' => \_\_( '导航自定义菜单' ), //显示的名称

// 'footer-menu' => \_\_( '页角自定义菜单' ),

// 'sider-menu' => \_\_('侧边栏菜单')

)

);

}

### 在生成导航栏的位置调用方法

<?php

echo wp\_nav\_menu(

array(

'theme\_location' => 'header-menu', //使用注册的菜单，function中注册

'menu' => 'two', //加载不同的菜单，默认为空

'container' => 'div', //ul最外面的div，父类节点 container => false去掉父类节点

'container\_class' => 'con', //导航栏最外面

'container\_id' => '', //导航栏ID

'menu\_class' => 'head-nav', //ul的class

'menu\_id' => '', //ul的id

'echo' => true, //直接显示

'fallback\_cb' => 'wp\_page\_menu',

'before' => '', //在li里输出内容

'after' => '', //在li里输出内容

'link\_before' => '', //在li里输出内容

'link\_after' => '', //在li里输出内容

'items\_wrap' => '<a class="logo"></a><nav class="%2$s">%3$s</nav>',

// 'items\_wrap' => '%3$s', //ul

'depth' => 0, //显示深度

'walker' => new wp\_bootstrap\_navwalker //自定义标题栏

)

);

?>

### 自定义导航栏

walker参数自定义导航栏，详细查看wp\_navwalker.php查看

## 后台添加指定菜单

/\*\*

\* 后台侧边栏添加自定义菜单

\* 在后台菜单生成时，挂载一个动作

\* add\_menu\_page菜单注册功能

\*\*/

add\_action('admin\_menu', 'register\_menu');

function register\_menu() {

add\_menu\_page(

'标题', //页面名称

'留言信息', //菜单名称

'edit\_theme\_options', //权限编辑主题选项

\_\_FILE\_\_, //打开文件的路径

'ruikeedu\_com\_toplevel\_page' //调用的方法，一般是返回一个页面

);}