

WordPress 快速参考

主题、钩子、查询、REST API、自定义文章类型

主题基础

必要文件

style.css	主题元数据头部 + 样式
index.php	兜底模板 (必需)
functions.php	主题初始化、钩子、脚本入队
screenshot.png	主题预览图 (1200x900)

style.css 头部

```
/*
Theme Name: My Theme
Theme URI: https://example.com
Version: 1.0
Requires at least: 6.0
*/
```

functions.php 初始化

```
add_action('after_setup_theme', function() {
    add_theme_support('title-tag');
    add_theme_support('post-thumbnails');
    add_theme_support('html5', ['search-form', 'gallery']);
    register_nav_menus(['primary' => 'Main Nav']);
});
```

模板层级

页面模板

front-page.php	静态首页
home.php	文章索引页
single.php	单篇文章
page.php	单个页面
archive.php	文章归档 (日期、分类)
category.php	分类归档
search.php	搜索结果
404.php	页面未找到
index.php	终极兜底模板

模板部件

```
get_header(); // header.php
get_footer(); // footer.php
get_sidebar(); // sidebar.php
get_template_part('parts/card'); // parts/card.php
get_template_part('parts/card', 'featured');
```

单篇文章查找顺序

- 1 single-{post_type}-{slug}.php
- 2 single-{post_type}.php
- 3 single.php
- 4 singular.php
- 5 index.php

The Loop

标准循环

```
<?php if (have_posts()) : while (have_posts()) : the_post(); ?>
<h2><?php the_title(); ?></h2>
<?php the_content(); ?>
<?php endwhile; endif; ?>
```

模板标签

the_title()	输出文章标题
the_content()	输出文章内容
the_excerpt()	输出文章摘要
the_permalink()	输出文章 URL
the_post_thumbnail()	输出特色图片
the_date()	输出发布日期
the_author()	输出作者名
the_category()	输出分类链接

Getter (返回值, 不直接输出)

```
$title = get_the_title();
$url = get_permalink();
$id = get_the_ID();
$thumb = get_the_post_thumbnail_url(null, 'medium');
```

自定义文章类型

注册 CPT

```
add_action('init', function() {
    register_post_type('product', [
        'labels' => ['name' => 'Products', 'singular_name' =>
            'Product'],
        'public' => true,
        'has_archive' => true,
        'supports' => ['title', 'editor', 'thumbnail'],
    ]);
});
```

自定义分类法

```
register_taxonomy('brand', 'product', [
    'labels' => ['name' => 'Brands'],
    'hierarchical' => true,
    'rewrite' => ['slug' => 'brand'],
]);
```

supports 选项

title	文章标题字段
editor	内容编辑器
thumbnail	特色图片
excerpt	摘要字段
custom-fields	自定义字段 meta box
page-attributes	菜单顺序、父级
revisions	版本历史

钩子与过滤器

Action (执行操作)

```
// Register a hook
add_action('wp_head', 'my_analytics', 20);
function my_analytics() {
    echo '<script>/* tracking */</script>';
}
```

Filter (修改数据)

```
// Modify the excerpt length
add_filter('excerpt_length', fn() => 30);
// Append text to every post
add_filter('the_content', function($c) {
    _return is_single() ? $c . '<p>Thanks for reading!</p>' : $c;
});
```

常用钩子

init	WP 初始化完成; 在此注册 CPT
wp_enqueue_scripts	入队 CSS/JS
wp_head	在 <head> 中输出内容
wp_footer	在 </body> 前输出内容
save_post	文章保存后触发
pre_get_posts	在执行前修改主查询
the_content	过滤文章内容输出
the_title	过滤文章标题输出

WP Query

自定义查询

```
$q = new WP_Query([
    'post_type' => 'product',
    'posts_per_page' => 10,
    'orderby' => 'date',
    'order' => 'DESC',
]);
while ($q->have_posts()) : $q->the_post();
    the_title('<h3>', '</h3>');
endwhile; wp_reset_postdata();
```

常用参数

post_type	post、page 或自定义类型
posts_per_page	结果数量 (-1 返回全部)
category_name	按分类 slug 过滤
tag	按标签 slug 过滤
s	搜索关键词
meta_key / meta_value	按自定义字段过滤
tax_query	高级分类法过滤
date_query	按日期范围过滤

修改主查询

```
add_action('pre_get_posts', function($q) {
    if (!is_admin() && $q->is_main_query() && is_home()) {
        $q->set('posts_per_page', 5);
    }
});
```

REST API

内置端点

/wp-json/wp/v2/posts	列出/创建文章
/wp-json/wp/v2/posts/{id}	获取/更新/删除文章
/wp-json/wp/v2/pages	页面端点
/wp-json/wp/v2/categories	分类端点
/wp-json/wp/v2/media	媒体库
/wp-json/wp/v2/users	用户端点

自定义端点

```
add_action('rest_api_init', function() {
    register_rest_route('myplugin/v1', '/items', [
        'methods' => 'GET',
        'callback' => fn($req) => new WP_REST_Response(
            get_posts(['post_type' => 'item', 'numberposts' => 20])
        ),
        'permission_callback' => '__return_true',
    ]);
});
```

从 JavaScript 调用

```
const res = await fetch('/wp-json/wp/v2/posts?per_page=5');
const posts = await res.json();
posts.forEach(p => console.log(p.title.rendered));
```

WordPress 快速参考

函数

文章函数

<code>get_posts(\$args)</code>	获取文章数组
<code>wp_insert_post(\$arr)</code>	创建或更新文章
<code>wp_delete_post(\$id)</code>	将文章移入回收站或永久删除
<code>get_post_meta(\$id, \$key)</code>	获取自定义字段值
<code>update_post_meta(\$id, \$k, \$v)</code>	设置自定义字段值

用户与权限

<code>wp_get_current_user()</code>	当前登录用户
<code>is_user_logged_in()</code>	检查登录状态
<code>current_user_can(\$cap)</code>	检查用户能力
<code>wp_create_nonce(\$action)</code>	生成安全 nonce
<code>wp_verify_nonce(\$nonce, \$a)</code>	验证 nonce

工具函数

<code>esc_html(\$str)</code>	转义 HTML 上下文
<code>esc_attr(\$str)</code>	转义属性上下文
<code>esc_url(\$url)</code>	清理 URL
<code>wp_kses_post(\$html)</code>	仅允许安全的文章 HTML
<code>sanitize_text_field(\$s)</code>	去除标签与多余空白
<code>absint(\$val)</code>	取绝对整数值

入队

入队样式与脚本

```
add_action('wp_enqueue_scripts', function() {
    wp_enqueue_style('theme-style', get_stylesheet_uri(), [],
        '1.0');
    wp_enqueue_script('theme-js', get_theme_file_uri('/js/app.js'),
        ['jquery'], '1.0', true);
});
```

入队函数

<code>wp_enqueue_style(\$h, \$src)</code>	注册并入队 CSS
<code>wp_enqueue_script(\$h, \$src)</code>	注册并入队 JS
<code>wp_localize_script(\$h, \$n, \$d)</code>	将 PHP 数据传递给 JS
<code>wp_dequeue_style(\$handle)</code>	移除已入队的样式
<code>wp_dequeue_script(\$handle)</code>	移除已入队的脚本

向 JavaScript 传递数据

```
wp_enqueue_script('my-app', get_theme_file_uri('/js/app.js'));
wp_localize_script('my-app', 'myData', [
    'ajaxUrl' => admin_url('admin-ajax.php'),
    'nonce' => wp_create_nonce('my_action'),
]);
// JS: console.log(myData.ajaxUrl)
```

常用模式

自定义页面模板

```
<?php
/* Template Name: Full Width */
get_header(); ?>
<main class="full-width">
    <?php the_content(); ?>
</main>
<?php get_footer(); ?>
```

AJAX 处理器

```
add_action('wp_ajax_my_action', 'handle_my_action');
add_action('wp_ajax_nopriv_my_action', 'handle_my_action');
function handle_my_action() {
    check_ajax_referer('my_action', 'nonce');
    wp_send_json_success(['msg' => 'OK']);
}
```

Shortcode

```
add_shortcode('greeting', function($atts) {
    $a = shortcode_atts(['name' => 'World'], $atts);
    return '<p>Hello, ' . esc_html($a['name']) . '!</p>';
});
// Usage: [greeting name="Zhi"]
```