

WordPress Quick Reference

Themes, hooks, queries, REST API, custom post types

Theme Basics

Required Files

style.css	Theme metadata header + styles
index.php	Fallback template (required)
functions.php	Theme setup, hooks, enqueue
screenshot.png	Theme preview (1200x900)

style.css Header

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

functions.php Setup

```
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']);
});
```

Template Hierarchy

Page Templates

front-page.php	Static front page
home.php	Blog posts index
single.php	Single post
page.php	Single page
archive.php	Post archives (date, category)
category.php	Category archive
search.php	Search results
404.php	Not found
index.php	Ultimate fallback

Template Parts

```
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');
```

Lookup Order (Single Post)

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

The Loop

Standard Loop

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

Template Tags

the_title()	Print post title
the_content()	Print post content
the_excerpt()	Print post excerpt
the_permalink()	Print post URL
the_post_thumbnail()	Print featured image
the_date()	Print publish date
the_author()	Print author name
the_category()	Print category links

Getters (Return, Not Print)

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

Custom Post Types

Register CPT

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

Custom Taxonomy

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

Supports Options

title	Post title field
editor	Content editor
thumbnail	Featured image
excerpt	Excerpt field
custom-fields	Meta box for custom fields
page-attributes	Menu order, parent
revisions	Revision history

Hooks & Filters

Actions (Do Something)

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

Filters (Modify Data)

```
// 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;
});
```

Key Hooks

init	WP initialized; register CPTs here
wp_enqueue_scripts	Enqueue CSS/JS
wp_head	Output in <head>
wp_footer	Output before </body>
save_post	After a post is saved
pre_get_posts	Modify main query before execution
the_content	Filter post content output
the_title	Filter post title output

WP Query

Custom Query

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

Common Parameters

post_type	post, page, or custom type
posts_per_page	Number of results (-1 for all)
category_name	Filter by category slug
tag	Filter by tag slug
s	Search keyword
meta_key / meta_value	Filter by custom field
tax_query	Advanced taxonomy filtering
date_query	Filter by date range

Modify Main Query

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

REST API

Built-in Endpoints

/wp-json/wp/v2/posts	List / create posts
/wp-json/wp/v2/posts/{id}	Get / update / delete post
/wp-json/wp/v2/pages	Pages endpoint
/wp-json/wp/v2/categories	Categories endpoint
/wp-json/wp/v2/media	Media library
/wp-json/wp/v2/users	Users endpoint

Custom Endpoint

```
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',
    ]);
});
```

Fetch from JavaScript

```
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 Quick Reference

Functions

Post Functions

<code>get_posts(\$args)</code>	Retrieve array of posts
<code>wp_insert_post(\$arr)</code>	Create or update a post
<code>wp_delete_post(\$id)</code>	Trash or delete a post
<code>get_post_meta(\$id, \$key)</code>	Get custom field value
<code>update_post_meta(\$id, \$k, \$v)</code>	Set custom field value

User & Auth

<code>wp_get_current_user()</code>	Current logged-in user
<code>is_user_logged_in()</code>	Check login status
<code>current_user_can(\$cap)</code>	Check capability
<code>wp_create_nonce(\$action)</code>	Generate security nonce
<code>wp_verify_nonce(\$nonce, \$a)</code>	Verify nonce

Utility Functions

<code>esc_html(\$str)</code>	Escape for HTML context
<code>esc_attr(\$str)</code>	Escape for attribute context
<code>esc_url(\$url)</code>	Sanitize URL
<code>wp_kses_post(\$html)</code>	Allow only safe post HTML
<code>sanitize_text_field(\$s)</code>	Strip tags and extra whitespace
<code>absint(\$val)</code>	Absolute integer

Enqueue

Enqueue Styles & Scripts

```
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);
});
```

Enqueue Functions

<code>wp_enqueue_style(\$h, \$src)</code>	Register + enqueue CSS
<code>wp_enqueue_script(\$h, \$src)</code>	Register + enqueue JS
<code>wp_localize_script(\$h, \$n, \$d)</code>	Pass PHP data to JS
<code>wp_dequeue_style(\$handle)</code>	Remove enqueued style
<code>wp_dequeue_script(\$handle)</code>	Remove enqueued script

Pass Data to JavaScript

```
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)
```

Common Patterns

Custom Page Template

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

AJAX Handler

```
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"]
```