

WordPress クイックリファレンス

テーマ、フック、クエリ、REST API、カスタム投稿タイプ

テーマの基本

必須ファイル

style.css	テーマメタデータヘッダーとスタイル
index.php	フォールバックテンプレート (必須)
functions.php	テーマ設定、フック、エンキュー
screenshot.png	テーマプレビュー (1200x900)

style.css ヘッダー

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

functions.php のセットアップ

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

テンプレート階層

ページテンプレート

front-page.php	静的フロントページ
home.php	ブログ投稿インデックス
single.php	単一の投稿
page.php	単一のページ
archive.php	投稿アーカイブ (日付、カテゴリー)
category.php	カテゴリーアーカイブ
search.php	検索結果
404.php	見つかりません
index.php	最終フォールバック

テンプレートパーツ

```
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');
```

検索順序 (単一投稿)

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

ループ

標準ループ

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

テンプレートタグ

the_title()	投稿タイトルを出力
the_content()	投稿コンテンツを出力
the_excerpt()	投稿の抜粋を出力
the_permalink()	投稿 URL を出力
the_post_thumbnail()	アイキャッチ画像を出力
the_date()	公開日を出力
the_author()	著者名を出力
the_category()	カテゴリーリンクを出力

ゲッター (出力せず返す)

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

カスタム投稿タイプ

CPT の登録

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

カスタムタクソノミー

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

supports オプション

title	投稿タイトルフィールド
editor	コンテンツエディター
thumbnail	アイキャッチ画像
excerpt	抜粋フィールド
custom-fields	カスタムフィールドのメタボックス
page-attributes	メニュー順序、親
revisions	リビジョン履歴

フックとフィルター

アクション (何かを実行)

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

フィルター (データを変更)

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

主要フック

init	WP 初期化済み。ここで CPT を登録
wp_enqueue_scripts	CSS/JS をエンキュー
wp_head	<head> に出力
wp_footer	</body> の前に出力
save_post	投稿が保存された後
pre_get_posts	実行前にメインクエリを変更
the_content	投稿コンテンツ出力をフィルター
the_title	投稿タイトル出力をフィルター

WP Query

カスタムクエリ

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

一般的なパラメーター

post_type	post、page、またはカスタムタイプ
posts_per_page	結果数 (-1 ですべて)
category_name	カテゴリースラッグでフィルター
tag	タグスラッグでフィルター
s	検索キーワード
meta_key / meta_value	カスタムフィールドでフィルター
tax_query	高度なタクソノミーフィルタリング
date_query	日付範囲でフィルター

メインクエリの変更

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

REST API

組み込みエンドポイント

/wp-json/wp/v2/posts	投稿の一覧取得 / 作成
/wp-json/wp/v2/posts/{id}	投稿の取得 / 更新 / 削除
/wp-json/wp/v2/pages	ページエンドポイント
/wp-json/wp/v2/categories	カテゴリーエンドポイント
/wp-json/wp/v2/media	メディアライブラリ
/wp-json/wp/v2/users	ユーザーエンドポイント

カスタムエンドポイント

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

JavaScript からの取得

```
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 クイックリファレンス

関数

投稿関数

<code>get_posts(\$args)</code>	投稿の配列を取得
<code>wp_insert_post(\$arr)</code>	投稿を作成または更新
<code>wp_delete_post(\$id)</code>	投稿をゴミ箱または完全削除
<code>get_post_meta(\$id, \$key)</code>	カスタムフィールドの値を取得
<code>update_post_meta(\$id, \$k, \$v)</code>	カスタムフィールドの値を設定

ユーザーと認証

<code>wp_get_current_user()</code>	現在ログイン中のユーザー
<code>is_user_logged_in()</code>	ログイン状態を確認
<code>current_user_can(\$cap)</code>	権限を確認
<code>wp_create_nonce(\$action)</code>	セキュリティ nonce を生成
<code>wp_verify_nonce(\$nonce, \$a)</code>	nonce を検証

ユーティリティ関数

<code>esc_html(\$str)</code>	HTML コンテキスト用にエスケープ
<code>esc_attr(\$str)</code>	属性コンテキスト用にエスケープ
<code>esc_url(\$url)</code>	URL をサニタイズ
<code>wp_kses_post(\$html)</code>	安全な投稿 HTML のみ許可
<code>sanitize_text_field(\$s)</code>	タグと余分な空白を削除
<code>absint(\$val)</code>	絶対値の整数

エンキュー

スタイルとスクリプトのエンキュー

```
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);
});
```

エンキュー関数

<code>wp_enqueue_style(\$h, \$src)</code>	CSS を登録してエンキュー
<code>wp_enqueue_script(\$h, \$src)</code>	JS を登録してエンキュー
<code>wp_localize_script(\$h, \$n, \$d)</code>	PHP データを JS に渡す
<code>wp_dequeue_style(\$handle)</code>	エンキュー済みスタイルを削除
<code>wp_dequeue_script(\$handle)</code>	エンキュー済みスクリプトを削除

JavaScript にデータを渡す

```
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)
```

よく使うパターン

カスタムページテンプレート

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

AJAX ハンドラー

```
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']);
}
```

ショートコード

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