

# Referência Rápida do WordPress

Temas, hooks, consultas, REST API, tipos de post personalizados

## Fundamentos do Tema

### Arquivos Obrigatórios

<b>style.css</b>	Cabeçalho de metadados do tema e estilos
<b>index.php</b>	Template de fallback (obrigatório)
<b>functions.php</b>	Configuração do tema, hooks, enqueue
<b>screenshot.png</b>	Prévia do tema (1200x900)

### Cabeçalho do style.css

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

### Configuração do functions.php

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

## Hierarquia de Templates

### Templates de Página

<b>front-page.php</b>	Página inicial estática
<b>home.php</b>	Índice de posts do blog
<b>single.php</b>	Post individual
<b>page.php</b>	Página individual
<b>archive.php</b>	Arquivos de posts (data, categoria)
<b>category.php</b>	Arquivo de categoria
<b>search.php</b>	Resultados de busca
<b>404.php</b>	Não encontrado
<b>index.php</b>	Fallback final

### Partes de Template

```
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');
```

### Ordem de Busca (Post Individual)

- 1 single-{post\_type}-{slug}.php
- 2 single-{post\_type}.php
- 3 single.php
- 4 singular.php
- 5 index.php

## O Loop

### Loop Padrão

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

## Tags de Template

<b>the_title()</b>	Exibe o título do post
<b>the_content()</b>	Exibe o conteúdo do post
<b>the_excerpt()</b>	Exibe o resumo do post
<b>the_permalink()</b>	Exibe a URL do post
<b>the_post_thumbnail()</b>	Exibe a imagem destacada
<b>the_date()</b>	Exibe a data de publicação
<b>the_author()</b>	Exibe o nome do autor
<b>the_category()</b>	Exibe os links de categoria

### Getters (Retornam, Não Exibem)

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

## Tipos de Post Personalizados

### Registrar CPT

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

### Taxonomia Personalizada

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

### Opções de Supports

<b>title</b>	Campo de título do post
<b>editor</b>	Editor de conteúdo
<b>thumbnail</b>	Imagem destacada
<b>excerpt</b>	Campo de resumo
<b>custom-fields</b>	Meta box para campos personalizados
<b>page-attributes</b>	Ordem de menu, pai
<b>revisions</b>	Histórico de revisões

## Hooks e Filtros

### Actions (Fazer Algo)

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

### Filters (Modificar Dados)

```
// Modificar o tamanho do resumo
add_filter('excerpt_length', fn() => 30);
// Adicionar texto a cada post
add_filter('the_content', function($c) {
    return is_single() ? $c . '<p>Thanks for reading!</p>' : $c;
});
```

## Hooks Principais

<b>init</b>	WP inicializado; registrar CPTs aqui
<b>wp_enqueue_scripts</b>	Enfileirar CSS/JS
<b>wp_head</b>	Saída em <head>
<b>wp_footer</b>	Saída antes de </body>
<b>save_post</b>	Após salvar um post
<b>pre_get_posts</b>	Modificar a consulta principal antes da execução
<b>the_content</b>	Filtrar saída do conteúdo do post
<b>the_title</b>	Filtrar saída do título do post

## WP\_Query

### Consulta Personalizada

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

### Parâmetros Comuns

<b>post_type</b>	post, page ou tipo personalizado
<b>posts_per_page</b>	Número de resultados (-1 para todos)
<b>category_name</b>	Filtrar por slug de categoria
<b>tag</b>	Filtrar por slug de tag
<b>s</b>	Palavra-chave de busca
<b>meta_key / meta_value</b>	Filtrar por campo personalizado
<b>tax_query</b>	Filtragem avançada de taxonomias
<b>date_query</b>	Filtrar por intervalo de datas

### Modificar Consulta Principal

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

## REST API

### Endpoints Integrados

<b>/wp-json/wp/v2/posts</b>	Listar / criar posts
<b>/wp-json/wp/v2/posts/{id}</b>	Obter / atualizar / excluir post
<b>/wp-json/wp/v2/pages</b>	Endpoint de páginas
<b>/wp-json/wp/v2/categories</b>	Endpoint de categorias
<b>/wp-json/wp/v2/media</b>	Biblioteca de mídia
<b>/wp-json/wp/v2/users</b>	Endpoint de usuários

### Endpoint Personalizado

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

### Buscar via 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));
```

# Referência Rápida do WordPress

## Funções

### Funções de Post

<code>get_posts(\$args)</code>	Recuperar array de posts
<code>wp_insert_post(\$arr)</code>	Criar ou atualizar um post
<code>wp_delete_post(\$id)</code>	Mover para lixeira ou excluir um post
<code>get_post_meta(\$id, \$key)</code>	Obter valor de campo personalizado
<code>update_post_meta(\$id, \$k, \$v)</code>	Definir valor de campo personalizado

### Usuário e Autenticação

<code>wp_get_current_user()</code>	Usuário atualmente conectado
<code>is_user_logged_in()</code>	Verificar status de login
<code>current_user_can(\$cap)</code>	Verificar capacidade
<code>wp_create_nonce(\$action)</code>	Gerar nonce de segurança
<code>wp_verify_nonce(\$nonce, \$a)</code>	Verificar nonce

### Funções Utilitárias

<code>esc_html(\$str)</code>	Escapar para contexto HTML
<code>esc_attr(\$str)</code>	Escapar para contexto de atributo
<code>esc_url(\$url)</code>	Sanitizar URL
<code>wp_kses_post(\$html)</code>	Permitir apenas HTML seguro para posts
<code>sanitize_text_field(\$s)</code>	Remover tags e espaços em branco extras
<code>absint(\$val)</code>	Inteiro absoluto

## Enqueue

### Enfileirar Estilos e 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);
});
```

### Funções de Enqueue

<code>wp_enqueue_style(\$h, \$src)</code>	Registrar + enfileirar CSS
<code>wp_enqueue_script(\$h, \$src)</code>	Registrar + enfileirar JS
<code>wp_localize_script(\$h, \$n, \$d)</code>	Passar dados PHP para JS
<code>wp_dequeue_style(\$handle)</code>	Remover estilo enfileirado
<code>wp_dequeue_script(\$handle)</code>	Remover script enfileirado

### Passar Dados para 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)
```

## Padrões Comuns

### Template de Página Personalizado

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

## Handler AJAX

```
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>';
});
// Uso: [greeting name="Zhi"]
```