REFERÊNCIA RÁPIDA DO WORDPRESS

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

Fundamentos do Tema

Arquivos Obrigatórios

`style.css`	Cabeçalho de metadados do tema e estilos
`index.php`	Template de fallback (obrigatório)
`functions.php`	Configuração do tema, hooks, enqueue
`screenshot.png`	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

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

`the_title()`	Exibe o título do post
`the_content()`	Exibe o conteúdo do post
`the_excerpt()`	Exibe o resumo do post
`the_permalink()`	Exibe a URL do post
`the_post_thumbnail()`	Exibe a imagem destacada
`the_date()`	Exibe a data de publicação
`the_author()`	Exibe o nome do autor
`the_category()`	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

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

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

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

`/wp-json/wp/v2/posts`	Listar / criar posts
`/wp-json/wp/v2/posts/{id}`	Obter / atualizar / excluir post
`/wp-json/wp/v2/pages`	Endpoint de páginas
`/wp-json/wp/v2/categories`	Endpoint de categorias
`/wp-json/wp/v2/media`	Biblioteca de mídia
`/wp-json/wp/v2/users`	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));
```

Funções

Funções de Post

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

Usuário e Autenticação

`wp_get_current_user()`	Usuário atualmente conectado
`is_user_logged_in()`	Verificar status de login
`current_user_can($cap)`	Verificar capacidade
`wp_create_nonce($action)`	Gerar nonce de segurança
`wp_verify_nonce($nonce, $a)`	Verificar nonce

Funções Utilitárias

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

`wp_enqueue_style($h, $src)`	Registrar + enfileirar CSS
`wp_enqueue_script($h, $src)`	Registrar + enfileirar JS
`wp_localize_script($h, $n, $d)`	Passar dados PHP para JS
`wp_dequeue_style($handle)`	Remover estilo enfileirado
`wp_dequeue_script($handle)`	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"]
```