

RIFERIMENTO RAPIDO WORDPRESS

Temi, hook, query, REST API, custom post type

Basi del Tema

File Richiesti

style.css	Intestazione metadati del tema + stili
index.php	Template di fallback (obbligatorio)
functions.php	Configurazione tema, hook, enqueue
screenshot.png	Anteprima del tema (1200x900)

Intestazione style.css

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

Configurazione 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']);
});
```

Gerarchia del Template

Template di Pagina

front-page.php	Pagina iniziale statica
home.php	Indice dei post del blog
single.php	Singolo post
page.php	Singola pagina
archive.php	Archivi post (data, categoria)
category.php	Archivio categorie
search.php	Risultati di ricerca
404.php	Pagina non trovata
index.php	Fallback definitivo

Parti di 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');
```

Ordine di Ricerca (Post Singolo)

- single-{post_type}-{slug}.php
- single-{post_type}.php
- single.php
- singular.php
- index.php

Il Loop

Loop Standard

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

Tag di Template

the_title()	Stampa il titolo del post
the_content()	Stampa il contenuto del post
the_excerpt()	Stampa il riassunto del post
the_permalink()	Stampa l'URL del post
the_post_thumbnail()	Stampa l'immagine in evidenza
the_date()	Stampa la data di pubblicazione
the_author()	Stampa il nome dell'autore
the_category()	Stampa i link alle categorie

Getter (Ritornano, Non Stampano)

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

Custom Post Type

Registrare un CPT

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

Tassonomia Personalizzata

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

Opzioni Supports

title	Campo titolo del post
editor	Editor del contenuto
thumbnail	Immagine in evidenza
excerpt	Campo riassunto
custom-fields	Meta box per campi personalizzati
page-attributes	Ordine menu, genitore
revisions	Cronologia revisioni

Hook e Filtri

Action (Esegui Qualcosa)

```
// Registra un hook
add_action('wp_head', 'my_analytics', 20);
function my_analytics() {
    echo <Script>? tracking </script>;
}
```

Filter (Modifica Dati)

```
// Modifica la lunghezza del riassunto
add_filter('excerpt_length', fn() => 30);
// Aggiungi testo a ogni post
add_filter('the_content', function($c) {
    return is_single() ? $c . '<p>Grazie per la lettura!</p>' :
    $c;
});
```

Hook Principali

init	WP inizializzato; registra i CPT qui
wp_enqueue_scripts	Aggiunge CSS/JS
wp_head	Output in <head>
wp_footer	Output prima di </body>
save_post	Dopo il salvataggio di un post
pre_get_posts	Modifica la query principale prima dell'esecuzione
the_content	Filtra l'output del contenuto del post
the_title	Filtra l'output del titolo del post

WP Query

Query Personalizzata

```
$q = new WP_Query([
    'post_type' => 'product',
    'posts_per_page' => 10,
    'orderby' => 'date',
    'order' => 'DESC',
]);
while ($q->have_posts()) : $q->the_post();
    the_title('<h2>', '</h2>');
endwhile; wp_reset_postdata();
```

Parametri Comuni

post_type	post, page, o tipo personalizzato
posts_per_page	Numero di risultati (-1 per tutti)
category_name	Filtra per slug categoria
tag	Filtra per slug tag
s	Parola chiave di ricerca
meta_key / meta_value	Filtra per campo personalizzato
tax_query	Filtro avanzato per tassonomia
date_query	Filtra per intervallo di date

Modifica la Query Principale

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

REST API

Endpoint Predefiniti

/wp-json/wp/v2/posts	Elenca / crea post
/wp-json/wp/v2/posts/{id}	Ottieni / aggiorna / elimina post
/wp-json/wp/v2/pages	Endpoint pagine
/wp-json/wp/v2/categories	Endpoint categoria
/wp-json/wp/v2/media	Libreria media
/wp-json/wp/v2/users	Endpoint utenti

Endpoint Personalizzato

```
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 da 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));
```

Funzioni

Funzioni Post

get_posts(\$args)	Recupera un array di post
wp_insert_post(\$arr)	Crea o aggiorna un post
wp_delete_post(\$id)	Manda nel cestino o elimina un post
get_post_meta(\$id, \$key)	Ottieni il valore di un campo personalizzato
update_post_meta(\$id, \$k, \$v)	Imposta il valore di un campo personalizzato

Utenti e Autenticazione

wp_get_current_user()	Utente attualmente connesso
is_user_logged_in()	Verifica lo stato di accesso
current_user_can(\$cap)	Verifica le capacità
wp_create_nonce(\$action)	Genera un nonce di sicurezza
wp_verify_nonce(\$nonce, \$a)	Verifica un nonce

Funzioni di Utilità

esc_html(\$str)	Escape per contesto HTML
esc_attr(\$str)	Escape per contesto attributo
esc_url(\$url)	Sanifica URL
wp_kses_post(\$html)	Permette solo HTML sicuro per i post
sanitize_text_field(\$s)	Rimuove tag e spazi extra
absint(\$val)	Intero assoluto

Enqueue

Aggiungere Stili e Script

```
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);
});
```

Funzioni Enqueue

wp_enqueue_style(\$h, \$src)	Registra + aggiunge CSS
wp_enqueue_script(\$h, \$src)	Registra + aggiunge JS
wp_localize_script(\$h, \$n, \$d)	Passa dati PHP a JS
wp_dequeue_style(\$handle)	Rimuove lo stile aggiunto
wp_dequeue_script(\$handle)	Rimuove lo script aggiunto

Passare Dati a 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)
```

Pattern Comuni

Template di Pagina Personalizzata

```
<?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'], $atts);
    return '<p>Ciao, ' . esc_html($a['name']) . '!</p>';
});
// Utilizzo: [greeting name="Zhi"]
```