

RÉFÉRENCE RAPIDE WORDPRESS

Thèmes, hooks, requêtes, REST API, types de contenu personnalisés

Bases des thèmes

Fichiers requis

style.css	En-tête de métadonnées du thème et styles
index.php	Template de repli (obligatoire)
functions.php	Configuration du thème, hooks, enqueue
screenshot.png	Aperçu du thème (1200x900)

En-tête de style.css

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

Configuration de 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']);
});
```

Hierarchie des templates

Templates de page

front-page.php	Page d'accueil statique
home.php	Index des articles du blog
single.php	Article individuel
page.php	Page individuelle
archive.php	Archives d'articles (date, catégorie)
category.php	Archive de catégorie
search.php	Résultats de recherche
404.php	Non trouvé
index.php	Repli final

Parties 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');
```

Ordre de recherche (article individuel)

- 1 single-[post_type]-[slug].php
- 2 single-[post_type].php
- 3 single.php
- 4 singular.php
- 5 index.php

La Boucle

Boucle standard

```
<?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()	Affiche le titre de l'article
the_content()	Affiche le contenu de l'article
the_excerpt()	Affiche l'extrait de l'article
the_permalink()	Affiche l'URL de l'article
the_post_thumbnail()	Affiche l'image à la une
the_date()	Affiche la date de publication
the_author()	Affiche le nom de l'auteur
the_category()	Affiche les liens de catégorie

Getters (retourment, n'affichent pas)

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

Types de contenu personnalisés

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

Taxonomie personnalisée

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

Options de supports

title	Champ de titre de l'article
editor	Éditeur de contenu
thumbnail	Image à la une
excerpt	Champ d'extrait
custom-fields	Meta box pour les champs personnalisés
page-attributes	Ordre du menu, parent
revisions	Historique des révisions

Hooks et filtres

Actions (faire quelque chose)

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

Filtres (modifier des données)

```
// Modifier la longueur de l'extrait
add_filter('excerpt_length', fn() => 30);
// Ajouter du texte à chaque article
add_filter('the_content', function($c) {
    return is_single() ? $c . '<p>Thanks for reading!</p>' : $c;
});
```

Hooks clés

init	WP initialisé, enregistrer les CPT ici
wp_enqueue_scripts	Mettre en file CSS/JS
wp_head	Sortie dans <head>
wp_footer	Sortie avant </body>
save_post	Après la sauvegarde d'un article
pre_get_posts	Modifier la requête principale avant exécution
the_content	Filter la sortie du contenu
the_title	Filter la sortie du titre

WP_Query

Requête personnalisée

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

Paramètres courants

post_type	post, page ou type personnalisé
posts_per_page	Nombre de résultats (-1 pour tous)
category_name	Filterer par slug de catégorie
tag	Filterer par slug d'étiquette
s	Mot-clé de recherche
meta_key / meta_value	Filterer par champ personnalisé
tax_query	Filterage avancé par taxonomie
date_query	Filterer par plage de dates

Modifier la requête 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

Points de terminaison intégrés

/wp-json/wp/v2/posts	Lister / créer des articles
/wp-json/wp/v2/posts/{id}	Obtenir / modifier / supprimer un article
/wp-json/wp/v2/pages	Point de terminaison des pages
/wp-json/wp/v2/categories	Point de terminaison des catégories
/wp-json/wp/v2/media	Médiathèque
/wp-json/wp/v2/users	Point de terminaison des utilisateurs

Point de terminaison personnalisé

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

Récupérer depuis 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));
```

Fonctions

Fonctions d'article

get_posts(\$args)	Récupérer un tableau d'articles
wp_insert_post(\$arr)	Créer ou mettre à jour un article
wp_delete_post(\$id)	Mettre à la corbeille ou supprimer un article
get_post_meta(\$id, \$key)	Obtenir la valeur d'un champ personnalisé
update_post_meta(\$id, \$k, \$v)	Définir la valeur d'un champ personnalisé

Utilisateur et authentification

wp_get_current_user()	Utilisateur actuellement connecté
is_user_logged_in()	Vérifier l'état de connexion
current_user_can(\$cap)	Vérifier une capacité
wp_create_nonce(\$action)	Générer un nonce de sécurité
wp_verify_nonce(\$nonce, \$a)	Vérifier un nonce

Fonctions utilitaires

esc_html(\$str)	Échapper pour le contexte HTML
esc_attr(\$str)	Échapper pour le contexte d'attribut
esc_url(\$url)	Assainir une URL
wp_kses_post(\$html)	Autoriser uniquement le HTML sûr
sanitize_text_field(\$s)	Cour les articles Supprimer les balises et les espaces superflus
absint(\$val)	Entier absolu

Enqueue

Mettre en file styles et 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);
});
```

Fonctions d'enqueue

wp_enqueue_style(\$h, \$src)	Enregistrer + mettre en file CSS
wp_enqueue_script(\$h, \$src)	Enregistrer + mettre en file JS
wp_localize_script(\$h, \$n, \$d)	Passer des données PHP à JS
wp_dequeue_style(\$handle)	Retirer un style de la file
wp_dequeue_script(\$handle)	Retirer un script de la file

Passer des données à 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)
```

Modèles courants

Template de page personnalisée

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

Gestionnaire 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>;
});
// Utilisation : [greeting name="Zhi"]
```