

# Référence rapide WordPress

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

## Bases des thèmes

### Fichiers requis

<b>style.css</b>	En-tête de métadonnées du thème et styles
<b>index.php</b>	Template de repli (obligatoire)
<b>functions.php</b>	Configuration du thème, hooks, enqueue
<b>screenshot.png</b>	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 fonctions.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']);
});
```

## Hiérarchie des templates

### Templates de page

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

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

### Getters (retournent, 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

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

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

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

<b>/wp-json/wp/v2/posts</b>	Lister / créer des articles
<b>/wp-json/wp/v2/posts/{id}</b>	Obtenir / modifier / supprimer un article
<b>/wp-json/wp/v2/pages</b>	Point de terminaison des pages
<b>/wp-json/wp/v2/categories</b>	Point de terminaison des catégories
<b>/wp-json/wp/v2/media</b>	Médiathèque
<b>/wp-json/wp/v2/users</b>	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));
```

# Référence rapide WordPress

## Fonctions

### Fonctions d'article

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

### Utilisateur et authentification

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

### Fonctions utilitaires

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

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

```
<?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"]
```