



DÉVELOPPEMENT BACKEND

WordPress Développeur : thèmes sur mesure, PHP et architecture

Maîtrisez WordPress côté développeur. Thème enfant, thème sur mesure from scratch, template hierarchy, hooks, Custom Post Types et REST API : trois jours pour sortir du no-code et prendre le contrôle total de WordPress.

| DURÉE | TARIF HT | NIVEAU | LANGUE | GROUPE | FORMAT |
|----------|---------------------------|----------------------|-----------------|------------|------------------|
| 3j (21h) | 1490.00 € 894.00 € | Intermédiaire | Français | 3-6 | Formation |

1 PUBLIC VISÉ

- Développeurs web junior ou intermédiaire maîtrisant HTML, CSS, JavaScript et les bases de PHP souhaitant intégrer WordPress dans leur stack.
- Intégrateurs front-end voulant produire des thèmes WordPress sur mesure sans dépendre de constructeurs de pages.
- Freelances techniques cherchant à proposer des développements WordPress personnalisés à leurs clients.
- Personnes ayant suivi la formation **WordPress Essentiel** et souhaitant passer du no-code au développement.

2 PRÉREQUIS

Maîtrise de HTML et CSS indispensable. Bases solides en PHP : variables, fonctions, tableaux, conditions, boucles, includes. Notions de JavaScript ES6+. Savoir utiliser le terminal, FTP et un éditeur de code (VS Code recommandé). La formation **WordPress Essentiel** ou une pratique autonome de WordPress (tableau de bord, thèmes, plugins) est fortement recommandée en prérequis.

3 OBJECTIFS PÉDAGOGIQUES

- Comprendre l'architecture interne de WordPress : template hierarchy, boucle, hooks et filtres
- Créer un thème enfant complet avec surcharge CSS et PHP ciblée
- Développer un thème WordPress from scratch structuré et maintenable
- Enregistrer et afficher des Custom Post Types et des Custom Fields
- Étendre WordPress via les hooks : add_action(), add_filter() et création de shortcodes
- Consommer et exposer des données via la REST API WordPress
- Appliquer les bonnes pratiques de sécurité et de performance côté développeur

**4 PROGRAMME DÉTAILLÉ****Jour 1 (7h) - Architecture WordPress et thème enfant****Module 1 - WordPress sous le capot (2h)**

Architecture interne (1h)

- Cycle de vie d'une requête WordPress : de l'URL à l'affichage HTML
- Le fichier wp-config.php : constantes, préfixes, modes debug
- L'arborescence WordPress : wp-content, wp-includes, wp-admin
- Base de données WordPress : structure des tables, rôle de chaque table

Template Hierarchy (1h)

- Comment WordPress choisit le fichier template à charger
- Hiérarchie complète : single.php, page.php, archive.php, taxonomy.php, 404.php
- Fichiers spéciaux : functions.php, style.css, screenshot.png
- TD guidé : mapper une maquette sur la template hierarchy

Module 2 - La boucle WordPress et les fonctions template (2h)

- The Loop : fonctionnement, have_posts(), the_post()
- Fonctions template essentielles : the_title(), the_content(), the_excerpt(), the_permalink(), the_ID()
- Fonctions conditionnelles : is_single(), is_page(), is_archive(), is_home()
- WP_Query : requêtes personnalisées, arguments, reset de boucle
- TD guidé : créer une boucle personnalisée avec filtres par catégorie et limite de résultats
- TP corrigé : page d'archive custom affichant articles filtrés avec pagination

Module 3 - Thème enfant : surcharge CSS et PHP (3h)

Créer un thème enfant (1h)

- Structure minimale : dossier, style.css avec en-tête de déclaration, functions.php
- Charger la feuille de style parent avec wp_enqueue_scripts
- Activer le thème enfant et vérifier l'héritage

Surcharge CSS et templates (2h)

- Inspecter les styles du thème parent et cibler les sélecteurs à surcharger
- Spécificité CSS, !important et bonnes pratiques
- Copier un template du thème parent dans le thème enfant pour le modifier
- Modifier header.php, footer.php et single.php
- TD guidé : thème enfant avec CSS personnalisé + single.php modifié
- TP corrigé : thème enfant livrable avec 3 templates surchargés

Jour 2 (7h) - Thème from scratch et système de hooks



Module 4 - Développer un thème WordPress from scratch (4h)

Structure du thème (1h)

- Créer le thème : style.css, index.php, functions.php
- Déclarer les fonctionnalités du thème : add_theme_support() (post-thumbnails, title-tag, menus)
- Enregistrer des menus et des zones de widgets
- Enqueue correct des scripts et styles avec wp_enqueue_style() et wp_enqueue_script()

Construire les templates (3h)

- header.php et footer.php : structure sémantique, wp_head(), wp_footer()
- index.php : boucle principale avec la template hierarchy
- single.php : affichage d'un article avec métadonnées
- page.php : template de page avec zones de contenu
- archive.php : liste d'articles par catégorie ou taxonomie
- 404.php : page d'erreur personnalisée
- Fichiers partiels avec get_template_part()
- TD guidé : thème minimaliste fonctionnel from scratch
- TP corrigé : thème complet avec 6 templates opérationnels

Module 5 - Hooks : actions, filtres et shortcodes (3h)

Le système de hooks (1h30)

- Actions vs filtres : différence fondamentale et cas d'usage
- add_action() : hook sur un événement WordPress
- add_filter() : modifier une valeur retournée par WordPress
- Hooks courants : init, wp_enqueue_scripts, the_content, save_post
- Créer ses propres hooks avec do_action() et apply_filters()
- TD guidé : modifier l'extrait automatique, ajouter du contenu après chaque article

Shortcodes et widgets (1h30)

- Créer un shortcode simple et un shortcode avec attributs
- Shortcode encapsulant du contenu : \$content
- Créer un widget personnalisé en étendant WP_Widget
- TP corrigé : shortcode de mise en avant + widget de liste personnalisée

Jour 3 (7h) - Custom Post Types, ACF et REST API

Module 6 - Custom Post Types et taxonomies (2h30)

- Pourquoi les CPT : aller au-delà des pages et des articles
- Enregistrer un CPT avec register_post_type() : labels, supports, capabilities
- Enregistrer une taxonomie personnalisée avec register_taxonomy()



Afficher les CPT dans la template hierarchy

- Ajouter des métadonnées avec `add_meta_box()` ou ACF (Advanced Custom Fields)
- Sauvegarder et récupérer les métadonnées avec `update_post_meta()` et `get_post_meta()`
- TD guidé : CPT "Formations" avec taxonomie "Catégorie" et champs personnalisés
- TP corrigé : archive et single d'un CPT avec métadonnées affichées

Module 7 - REST API WordPress (2h30)

- Qu'est-ce que la REST API WordPress ? Endpoints natifs et structure des réponses
- Consommer la REST API avec `fetch` en JavaScript : GET articles, pages, CPT
- Authentification : nonces pour les requêtes front-end, Application Passwords pour les accès tiers
- Enregistrer un endpoint personnalisé avec `register_rest_route()`
- Exposer les métadonnées d'un CPT via la REST API
- Cas d'usage : headless WordPress, application JavaScript consommant les données WP
- TD guidé : afficher dynamiquement des CPT en JavaScript via la REST API
- TP corrigé : endpoint custom exposant les formations avec filtres par catégorie

Module 8 - Sécurité, performance et bonnes pratiques (2h)

Sécurité côté développeur (1h)

- Nonces WordPress : valider les actions et protéger les formulaires
- Échapper les sorties : `esc_html()`, `esc_attr()`, `esc_url()`, `wp_kses()`
- Valider et assainir les entrées : `sanitize_text_field()`, `absint()`, `wp_unslash()`
- Capacités et rôles : vérifier les permissions avant toute action sensible

Performance et bilan (1h)

- Transients API : mettre en cache des requêtes coûteuses
- Éviter les requêtes N+1 avec `WP_Query` et `update_post_meta_cache`
- Bonnes pratiques : préfixer ses fonctions, ne pas modifier le core, documenter son code
- Bilan de formation : revue de code collective, questions ouvertes, retour formateur individualisé

5 COMPÉTENCES VISÉES

- Lire et appliquer la template hierarchy WordPress pour cibler le bon fichier template
- Créer un thème enfant fonctionnel : `style.css`, `functions.php`, surcharge de templates
- Développer un thème WordPress from scratch : `index.php`, `header.php`, `footer.php`, `single.php`, `page.php`, `archive.php`
- Enregistrer des Custom Post Types et des taxonomies personnalisées avec `register_post_type()`
- Ajouter des métadonnées avec ACF ou `add_meta_box()`
- Créer des shortcodes et des widgets personnalisés
- Interroger la REST API WordPress avec `fetch` et exposer des endpoints personnalisés



6 MODALITÉS PÉDAGOGIQUES

Formation délivrée en présentiel ou distanciel (visioconférence). Le formateur alterne entre méthode démonstrative (live coding commenté et progressif), méthode interrogative (analyse et critique de code WordPress existant) et méthode active (TD guidés et TP corrigés). Chaque participant travaille sur son propre environnement WordPress tout au long de la formation. L'accent est mis sur la compréhension des mécanismes internes de WordPress pour produire un code maintenable et sécurisé.

7 MOYENS PÉDAGOGIQUES ET TECHNIQUES

- Support de cours numérique mis à disposition des apprenants
- Dépôt GitHub de démonstration avec squelettes de thème, snippets PHP commentés et corrections par module
- Environnement de développement préconfiguré : VS Code + LocalWP + Query Monitor + PHP_CodeSniffer
- Pour le distanciel : visioconférence (Zoom ou équivalent), partage d'écran, chat en direct
- Accès à la plateforme pédagogique LaPolaris (supports, ressources, émargement)

8 MODALITÉS D'ÉVALUATION

- En cours de formation : TD guidés et TP corrigés à chaque module avec feedback formateur immédiat
- En fin de formation : thème WordPress from scratch avec CPT, hooks personnalisés et endpoint REST API
- Questionnaire d'auto-évaluation des acquis en fin de parcours

9 CRITÈRES D'ÉVALUATION

- Thème WordPress from scratch fonctionnel avec 6 templates respectant la template hierarchy
- Thème enfant avec CSS surchargé et au moins 2 templates PHP modifiés
- Custom Post Type enregistré avec taxonomie et métadonnées affichées en front-end
- Endpoint REST API personnalisé fonctionnel et consommé en JavaScript
- Code sécurisé : sorties échappées, entrées sanitisées, nonces utilisés sur les formulaires

10 MODALITÉS DE VALIDATION

Attestation de fin de formation délivrée à l'issue du parcours, conditionnée à une assiduité d'au moins 80 % et à la présentation du thème WordPress développé en formation. L'attestation précise les objectifs atteints et les compétences acquises.

11 SUIVI ET ACCOMPAGNEMENT

- Feuilles d'émargement signées par demi-journée (présentiel) ou émargement numérique (distanciel)
- Traçabilité des activités pédagogiques réalisées
- Attestation d'assiduité délivrée en fin de formation
- Suivi individuel via les TP corrigés et le thème produit en formation



12 CONDITIONS D'ACCÈS

Formation accessible sur inscription directe, sans prérequis administratif particulier. Le financement peut être pris en charge par l'employeur dans le cadre d'un plan de développement des compétences, ou en autofinancement avec possibilité de paiement en plusieurs fois.

13 DÉLAIS D'ACCÈS

Inscription possible jusqu'à **5 jours ouvrés** avant le début de la session. Pour toute demande urgente, nous contacter directement.

S PROCHAINES SESSIONS

27/05/2026 → 29/05/2026

À distance

6 places restantes

01/06/2026 → 03/06/2026

À distance

6 places restantes

03/06/2026 → 05/06/2026

À distance

6 places restantes

08/06/2026 → 10/06/2026

À distance

6 places restantes

10/06/2026 → 12/06/2026

À distance

6 places restantes

15/06/2026 → 17/06/2026

À distance

6 places restantes

ACCESSIBILITÉ · HANDICAP

Nos formations sont accessibles aux personnes en situation de handicap. Pour toute situation nécessitant un aménagement (matériel, temporel ou pédagogique), nous vous invitons à nous contacter avant l'inscription afin d'étudier les adaptations possibles.

Référent handicap : contact@lapolaris.fr

LaPolaris

TÉL. +33762584798

EMAIL contact@lapolaris.fr

WEB lapolaris.fr