Usages des techniques numériques

Maitriser les technologies du web

Module 4 – Paramétrage avancé d'un CMS (WordPress)

Introduction

Comprendre le fonctionnement d’un CMS et de ses paramètres.

- WordPress est un système de gestion de contenus conçu pour s’adapter à des usages variés.
  Sa structure modulaire permet de créer aussi bien un **site vitrine** qu’une **boutique en ligne**, un **espace membre** ou un **portail d’information**.

- Ce module présente les éléments de configuration à mettre en place après l’installation technique de WordPress sur un hébergeur distant, dans une logique de déploiement professionnel.

- **Objectifs :**
  - sécuriser l’instance,
  - optimiser les performances,
  - structurer les contenus à venir,
  - anticiper les évolutions possibles.

Sommaire

1. Analyser les enjeux du paramétrage avancé
2. Configurer un environnement WordPress professionnel
3. Structurer un site pour un usage évolutif

4. Paramétrage d'un CMS - WordPress Enjeux

1. Analyser les enjeux du paramétrage avancé

a. Identifier les caractéristiques d'un CMS modulaire

- WordPress est un CMS dit "modulaire", c'est-à-dire construit autour d'une structure de base (core) à laquelle viennent s'ajouter des **modules d'extension** (plugins), des **thèmes de présentation**, et des **contenus dynamiques**.

- Cette architecture permet :
  - d'adapter l'interface aux besoins du projet (design, navigation, fonctionnalités),
  - de déléguer certaines tâches à des extensions spécialisées (référencement, sécurité, formulaires, etc.),
  - de faire évoluer un site sans repartir de zéro.

- La modularité implique néanmoins des **choix techniques structurants** dès l'installation :
  - éviter l'encombrement fonctionnel (plugins redondants ou inutiles),
  - sélectionner des composants fiables et maintenus,
  - construire une base propre, rapide, lisible dans la durée.

1. Analyser les enjeux du paramétrage avancé

b. Comprendre les bonnes pratiques de mise en production

- Un site WordPress prêt à être mis en ligne doit respecter un certain nombre de **normes techniques minimales** :
  - **Stabilité** : pas d'erreurs PHP, pas de conflits d'extensions.
  - **Sécurité** : accès restreints, protections anti-intrusion, mises à jour automatiques.
  - **Performance** : temps de chargement rapide, poids des pages contrôlé.
  - **Structure** : URL lisibles, arborescence logique, taxonomies adaptées.
  - **Accessibilité** : interface responsive, contrastes suffisants, navigation clavier.
  - **Respect des obligations légales** : mentions, RGPD, cookies.

- Ces éléments conditionnent à la fois l'expérience utilisateur, le référencement, la pérennité du site et sa capacité à évoluer.

1. Analyser les enjeux du paramétrage avancé

c. Organiser les conditions techniques d'un site évolutif

- Le paramétrage initial influence la capacité du site à :
  - accueillir de nouveaux contenus sans désorganisation,
  - intégrer des fonctionnalités supplémentaires (vente, newsletter, espace privé),
  - rester maintenable dans le temps (mise à jour des composants sans perte de données).

- Logique de production :
  - une **structuration anticipée** (menu, types de contenus, navigation),
  - une **organisation claire des contenus** (titres, catégories, médias),
  - une **structuration souple** pour permettre l'ajout ultérieur de contenus complexes (contenus personnalisés, champs dynamiques ou systèmes de filtrage).

2. Configurer un environnement WordPress professionnel

a. Réglages fondamentaux du système

- Les premiers réglages disponibles dans le back-office WordPress influencent directement l'organisation du site, la lisibilité de ses URL, le comportement de ses pages ou encore l'ouverture aux interactions.

Rubrique	Réglages recommandés
Général	Titre du site, slogan, fuseau horaire, langue
Écriture	Désactivation de la publication par email (rarement utilisée)
Lecture	Définir une page d'accueil statique (si site vitrine) et une page d'articles
Discussion	Désactiver les commentaires si non pertinents, éviter les rétroliens
Médias	Limiter la génération de tailles d'image inutiles
Permaliens	Choix d'une structure lisible : /%postname%/ (référencement + clarté)

2. Configurer un environnement WordPress professionnel

b. Gestion des utilisateurs et des rôles

- WordPress permet de gérer plusieurs utilisateurs avec des **rôles distincts**, ce qui facilite la délégation de tâches tout en maintenant un contrôle sur les accès.

Rôle	Capacités principales
Administrateur	Accès complet au site et à tous les réglages
Éditeur	Publication et gestion de tous les contenus
Auteur	Publication de ses propres contenus uniquement
Contributeur	Rédaction de contenus soumis à validation
Abonné	Accès restreint (lecture, commentaires, espace privé)

- Ce système peut être étendu par des extensions (ex. : *Members*, *User Role Editor, Woocommerce Products Vendor*) pour créer des rôles personnalisés ou restreindre certains accès.

2. Configurer un environnement WordPress professionnel

c. Sélection d'un constructeur (builder)

- Le choix du builder conditionne :
  - la structure visuelle du site,
  - les performances (vitesse de chargement, compatibilité mobile),
  - la flexibilité de mise en page,

- **Critères de sélection d'un constructeur de pages :**
  - Légèreté et rapidité (test via Google PageSpeed ou GTMetrix)
  - Support actif et mises à jour régulières
  - Documentation claire
  - Compatibilité avec WooCommerce

> Les constructeurs sont des thèmes WordPress nécessitant un child theme.

2. Configurer un environnement WordPress professionnel

c. Sélection d'un constructeur (builder)

- **Part de marché mondiale des constructeurs WordPress (2025)**

  (moyenne issue de plusieurs sources : W3Techs, Markentiefe, WPBeginner, BuiltWith, 2023–2025)

Rang	Constructeur / Builder	Part WP	Tendance 2025–2030	Commentaire synthétique
1	**Gutenberg (éditeur natif)**	**≈ 60–66 %**	En hausse	De plus en plus puissant (Full Site Editing).
2	**Elementor**	**≈ 20–30 %**	Stable à forte croissance	Toujours leader des builders tiers, fort écosystème.
3	**WPBakery Page Builder**	**≈ 10–12 %**	En déclin	Hérité d'anciennes installations, peu de nouveautés.
4	**Divi Builder (Elegant Themes)**	**≈ 2–4 %**	Stable (en refonte)	Licence à vie attractive, Divi 5 attendu.
5	**Beaver Builder**	**≈ 1 %**	En retrait	Marché pro réduit, peu de mise à jour majeure.
6	**Bricks Builder**	**≈ 0,3–0,8 %**	Croissance forte	Gagne les agences et développeurs, très léger.
7	**Breakdance Builder**	**≈ 0,1–0,4 %**	En forte expansion	Nouveau, rapide, conçu par Oxygen Team.

2. Configurer un environnement WordPress professionnel

d. Sélection d'un thème professionnel

- Le choix du thème conditionne :
  - la structure visuelle du site,
  - les performances (vitesse de chargement, compatibilité mobile),
  - la flexibilité de mise en page,
  - la compatibilité avec les constructeurs de pages.

- **Critères de sélection d'un thème fiable :**
  - Légèreté et rapidité (test via Google PageSpeed ou GTMetrix)
  - Bonne intégration avec Elementor, Divi ou Gutenberg
  - Support actif et mises à jour régulières
  - Documentation claire
  - Compatibilité avec WooCommerce si besoin

2. Configurer un environnement WordPress professionnel

d. Sélection d'un thème professionnel

- **Thème enfant (Child Theme)**
  - Dans une logique de production, il est recommandé d'utiliser un **thème enfant**, c'est-à-dire une copie du thème parent qui hérite de sa structure et de son style, tout en permettant de personnaliser le code sans risque lors des mises à jour.
  - Le thème enfant se compose généralement de deux fichiers :
    . style.css : qui référence le thème parent ;
    . functions.php : pour ajouter du code personnalisé ou des fonctions spécifiques.
  - Cela garantit une **personnalisation durable** du site tout en conservant la stabilité et la maintenabilité du projet.

> Exemple : "Hello Elementor Child" est le thème enfant le plus utilisé avec Elementor pour les projets sur mesure.

2. Configurer un environnement WordPress professionnel

e. Sécurisation avancée d'un site WordPress

- Un site WordPress bien configuré doit anticiper les risques liés à la sécurité : piratage, injection de code, vols de données, etc.
  Une stratégie de sécurité efficace combine **prévention, surveillance et réaction rapide.**

Étape	Objectif	Outils / Bonnes pratiques
1. Protection des accès	Empêcher les connexions non autorisées	Modifier l'URL d'accès à l'administration (/wp-admin), limiter les tentatives de connexion, activer la double authentification
2. Gestion des comptes utilisateurs	Réduire les risques liés aux rôles trop permissifs	Créer un compte administrateur distinct du compte rédacteur, utiliser des mots de passe robustes
3. Mises à jour régulières	Corriger les failles connues	Maintenir à jour le cœur WordPress, les thèmes et les plugins
4. Sauvegarde automatique	Pouvoir restaurer rapidement le site en cas d'attaque	Extensions recommandées : **UpdraftPlus**, Jetpack Backup
5. Sécurisation du serveur	Renforcer les défenses côté hébergeur	Utiliser HTTPS (certificat SSL), configurer le pare-feu, bloquer les IP suspectes
6. Surveillance continue	Détecter rapidement les anomalies	**Wordfence**, Sucuri Security, iThemes Security

2. Configurer un environnement WordPress professionnel

e. Sécurisation avancée d’un site WordPress

- **Paramétrage du fichier wp-config.php**
  - Le fichier wp-config.php contient les informations essentielles de connexion et certains réglages critiques.
  - Bonnes pratiques :
    . **Modifier le préfixe des tables** ($table_prefix = 'wp_';) pour réduire les risques d’attaque.

2. Configurer un environnement WordPress professionnel

f. Optimisation des performances et du référencement

- La performance d'un site WordPress influence directement **le confort de navigation, le référencement naturel (SEO)** et la conversion.
  Un site rapide et bien structuré inspire confiance et favorise la rétention des utilisateurs.

Domaine	Objectif	Outils / Méthodes
1. Vitesse de chargement	Réduire le temps d'affichage des pages	Utiliser un système de cache (WP Rocket, W3 Total Cache), activer la compression Gzip
2. Optimisation des médias	Alléger les fichiers lourds	Compresser les images (TinyPNG, Smush, Imagify), adopter le format Avif ou WebP
3. Nettoyage du code	Supprimer les scripts inutiles	Désactiver les plugins non utilisés, minifier CSS/JS
4. Référencement	Structurer les contenus pour les moteurs de recherche	Installer RankMath, rédiger des balises méta et titres clairs
5. Accessibilité mobile	Garantir une lecture fluide sur tous les écrans	Choisir un thème responsive, tester sur mobile via Google Mobile-Friendly Test
6. Suivi des performances	Analyser et améliorer continuellement	Google PageSpeed Insights, GTmetrix, Google Search Console

2. Configurer un environnement WordPress professionnel

g. Maintenance et sauvegarde régulière

- La maintenance d'un site WordPress est un **processus continu**, garantissant sa stabilité, sa sécurité et son efficacité dans le temps.

- **Bonnes pratiques de maintenance :**
  - Vérifier les mises à jour des extensions, thèmes et du core toutes les semaines.
  - Sauvegarder la base de données et les fichiers avant chaque modification majeure.
  - Tester les extensions sur un environnement de préproduction avant de les déployer.
  - Surveiller les liens brisés (Broken Link Checker) et corriger les erreurs 404.
  - Contrôler régulièrement les performances et le référencement.
  - Planifier un audit technique trimestriel.

- **Conseil professionnel :** documenter chaque intervention dans un journal de maintenance.
  Cela facilite la gestion collaborative et la continuité du projet.

2. Configurer un environnement WordPress professionnel

g. Maintenance et sauvegarde régulière

- **Environnement de préproduction (staging)**

  - Avant toute mise à jour ou ajout de fonctionnalité majeure, il est recommandé d'utiliser un **environnement de préproduction** (staging site).
  - Cet espace miroir du site de production permet de tester les nouvelles extensions, mises à jour ou réglages sans impacter la version en ligne.
  - De nombreux hébergeurs (o2switch, OVH, Kinsta…) proposent une fonction de staging intégrée.

  > Objectif : prévenir les erreurs de compatibilité et sécuriser le processus de maintenance.

3. Structurer un site WordPress pour un usage évolutif

a. Configurer les types de contenus personnalisés (Custom Post Types – CPT)

- WordPress propose par défaut deux types de contenus : *pages* et *articles*.
- Dans un contexte professionnel, il est souvent nécessaire de **créer des types de contenus adaptés au projet** : projets, témoignages, événements, ressources, équipes, etc.
- **Objectif :**
  Séparer logiquement les contenus pour simplifier leur gestion et leur affichage.
- **Méthodes possibles :**
  - Via une extension : → **Custom Post Type UI**, **ACF Pro**
  - Ou via un code personnalisé ajouté au fichier functions.php (ou mieux, ajouté à l'extension **Code Snippets**)

3. Structurer un site WordPress pour un usage évolutif

b. Définir des taxonomies personnalisées

- Les **taxonomies** (catégories, étiquettes) permettent de regrouper les contenus par thématique, auteur, niveau ou tout autre critère pertinent.
- Créer des taxonomies personnalisées facilite le filtrage, la recherche et l'affichage dynamique.
- **Exemple :**
  - Pour un site de portfolio : taxonomies "Type de projet", "Technologie utilisée".
  - Pour un site institutionnel : taxonomies "Département", "Public cible".
- **Extension recommandée :**
  - **Custom Post Type UI** (pour créer CPT + taxonomies visuellement)
  - **ACF** (pour ajouter des champs associés)

> Ces réglages renforcent la cohérence des contenus, et simplifient les futures intégrations avec des modules de recherche ou de filtrage.

3. Structurer un site WordPress pour un usage évolutif

c. Gérer les champs personnalisés (Custom Fields)

- Les **Custom Fields** (ou *champs personnalisés*) permettent d'ajouter des informations supplémentaires à un contenu : date d'événement, lien externe, durée, niveau, etc.
- **Extension de référence :**
  - **Advanced Custom Fields (ACF)** : intuitive, fiable, utilisée dans les environnements pro.
- **Exemples de champs possibles :**

Type de champ	Utilisation	Exemple
Texte	Lien, sous-titre, référence produit	https://monprojet.fr
Image	Illustration personnalisée	Logo client
Date	Date d'un événement ou d'une publication	12/05/2026
Sélecteur	Choix de catégorie ou d'état	"Publié / Archivé"
Relation	Lier plusieurs contenus entre eux	Projet / Témoignage client

3. Structurer un site WordPress pour un usage évolutif

d. Créer des modèles de pages dynamiques

- Une fois les contenus structurés (CPT, taxonomies, champs personnalisés), il est essentiel de définir **comment ces informations seront affichées sur le site**.
- Cette étape relie le **back-office (données)** au **front-end (affichage visuel)** à travers la hiérarchie de templates WordPress ou un constructeur de pages.

1. Hiérarchie des templates WordPress

- WordPress s'appuie sur une **hiérarchie de fichiers modèles** pour déterminer automatiquement quel fichier utiliser pour afficher un contenu.
- Cette logique permet de personnaliser l'affichage de chaque type de contenu sans dupliquer le code inutilement.

3. Structurer un site WordPress pour un usage évolutif

d. Créer des modèles de pages dynamiques

1. Hiérarchie des templates WordPress

- Exemple de hiérarchie WordPress :

Modèle	Usage
single.php	modèle par défaut pour un contenu unique
single-{posttype}.php	modèle spécifique à un Custom Post Type
archive.php	modèle pour les pages d'archives
archive-{taxonomy}.php	modèle pour une taxonomie particulière
page.php	modèle pour les pages statiques
front-page.php	modèle pour la page d'accueil

> **Astuce :** lors d'un développement sur mesure, créer ces fichiers dans un **thème enfant** permet de contrôler la structure sans risque de perte lors des mises à jour du constructeur.

3. Structurer un site WordPress pour un usage évolutif

d. Créer des modèles de pages dynamiques

2. Utilisation du thème enfant (Child Theme)

- Le **thème enfant** est indispensable pour toute personnalisation de modèle :
  il hérite des fichiers du thème parent, mais permet d'en modifier certains (ex. : single.php, header.php, footer.php) sans impacter le cœur du thème.
- **Structure minimale d'un thème enfant :**

  /wp-content/themes/hello-elementor-child/

  a. **style.css** : Fichier de déclaration du thème enfant.
  Il indique le nom du thème, son auteur, et surtout le **thème parent** dont il hérite.
  Il peut aussi contenir des **règles CSS spécifiques** pour personnaliser l'apparence du site sans modifier les fichiers du parent.
  Exemple de déclaration minimale :

```
/*
Theme Name: Hello Elementor Child
Template: hello-elementor
Version: 1.0
*/
```

3. Structurer un site WordPress pour un usage évolutif

d. Créer des modèles de pages dynamiques

2. Utilisation du thème enfant (Child Theme)

- **Structure minimale d'un thème enfant :**

  /wp-content/themes/hello-elementor-child/

  b. functions.php :

  - Permet d'ajouter les **fonctions PHP nécessaires à l'affichage dynamique des contenus** (titres, champs ACF, boucles personnalisées…).
  - Sert à **charger les styles et scripts du thème parent**, à **déclarer des menus**, des **zones de widgets**, ou à **enregistrer un Custom Post Type**.
  - C'est le cœur logique du thème enfant.

> À retenir :
Le **thème enfant** n'est pas toujours un espace de code complexe, mais plutôt une **structure de sécurité et d'évolution** : il protège les personnalisations lors des mises à jour du thème parent, tout en restant compatible avec les constructeurs visuels.

3. Structurer un site WordPress pour un usage évolutif

e. Gérer les menus, widgets et zones dynamiques

- La structure du site repose sur des **zones dynamiques** (menus, barres latérales, pieds de page). Une configuration claire favorise la maintenance et l'évolution.

- **Recommandations :**
  - Définir les menus principaux dès le départ (navigation, pied de page, menu contextuel).
  - Activer les zones de widgets dans le thème (souvent via Apparence → Widgets).
  - Créer des **zones personnalisées** si besoin via le functions.php ou Code Snippets

> **Astuce :** une structure claire de menus et widgets facilite la modularité du site et permet d'activer de nouvelles sections sans refonte.

3. Structurer un site WordPress pour un usage évolutif

f. Préparer les extensions pour l’évolution future

- Avant d’ajouter de nouvelles fonctionnalités (e-commerce, e-learning, espace privé), il est essentiel de vérifier la compatibilité technique des extensions.

- **Checklist avant installation d’un plugin :**
  - Vérifier la date de la dernière mise à jour.
  - Contrôler la compatibilité avec la version actuelle de WordPress.
  - Tester en environnement de préproduction.
  - Lire les avis et la documentation officielle.
  - Supprimer les plugins inutilisés pour alléger le site.

> Principe clé : “moins, mais mieux”.
Chaque extension doit répondre à un besoin précis et être maintenable dans le temps.

Synthèse

- Le site WordPress est désormais **paramétré pour la production** : chaque composant – contenus, modèles, extensions, sécurité – est configuré pour garantir la **stabilité, la maintenabilité et la pérennité** du projet.
- L'environnement repose sur une structure claire et évolutive : **types de contenus personnalisés, taxonomies, champs dynamiques, thèmes enfants et modèles hiérarchiques**, assurant la cohérence du front-end et la durabilité des personnalisations.
- Les bonnes pratiques de configuration : sécurité du wp-config.php, environnement de test, documentation des réglages > renforcent la fiabilité du site.
- Ce **socle technique complet** offre une base solide pour l'intégration de la **couche visuelle et interactive**, conformément aux standards UX/UI et aux exigences de performance web.