

CAHIER TP APPRENANT — 8 SEMAINES

TP semaine 1 — Core, bootstrap, données et WP_Query

But : Diagnostiquer le cycle de vie d'une requête et manipuler la base avec les API natives.

1. Installer WordPress dans un environnement local avec une base dédiée.
2. Activer WP_DEBUG_LOG hors production et contrôler wp-content/debug.log.
3. Créer 5 articles : 3 publiés, 2 brouillons.
4. Exécuter WP-CLI pour afficher ID, titre et statut.
5. Écrire un script d'inspection qui liste trois articles et une métadonnée.
6. Documenter la différence entre la requête principale et une requête secondaire.

Preuves à remettre : lien commit Git, capture terminal, trace logs, fiche erreurs rencontrées, explication de la correction.

Barème : 20 points selon la grille du manuel.

TP semaine 2 — PHP orienté objet, Hooks, Composer, qualité

But : Structurer un socle PHP modulaire et mettre en place une discipline de qualité.

1. Créer le fichier composer.json et la section autoload PSR-4.
2. Créer une classe AssetManager sous src/Frontend/.
3. Initialiser l'application depuis functions.php et composer vendor/autoload.php.
4. Installer PHPCS/WPCS dans les dépendances de développement.
5. Créer un test PHPUnit sur une classe de formatage pure.
6. Pousser les changements vers un dépôt Git de formation.

Preuves à remettre : lien commit Git, capture terminal, trace logs, fiche erreurs rencontrées, explication de la correction.

Barème : 20 points selon la grille du manuel.

TP semaine 3 — Structure, header, footer, The Loop, responsive

But : Construire un thème classique activable et accessible sans constructeur tiers.

1. Créer style.css et index.php pour rendre le thème visible.
2. Configurer les supports post-thumbnails, title-tag et menu primary.
3. Créer header.php, footer.php, single.php, page.php et front-page.php.
4. Créer un composant article-card dans template-parts/.
5. Développer un CSS mobile-first et tester l'accès au menu au clavier.
6. Comparer rendu articles, pages et page d'accueil.

Preuves à remettre : lien commit Git, capture terminal, trace logs, fiche erreurs rencontrées, explication de la correction.

Barème : 20 points selon la grille du manuel.

TP semaine 4 — Archives, pagination, taxonomies, Gutenberg

But : Créer des templates réutilisables, une pagination fiable et maîtriser theme.json.

1. Créer le template archive-agency_project.php.
2. Ajouter au moins 13 réalisations pour tester trois pages.
3. Développer les Template Parts et l'état sans contenu.
4. Créer un template de taxonomie agency_sector.
5. Ajouter theme.json et contrôler les couleurs de l'éditeur.
6. Tester recherche sans résultats et URL 404.

Preuves à remettre : lien commit Git, capture terminal, trace logs, fiche erreurs rencontrées, explication de la correction.

Barème : 20 points selon la grille du manuel.

TP semaine 5 — Plugin, hooks, CPT, taxonomies, Settings API

But : Développer un plugin métier indépendant du thème et protégé.

1. Installer le plugin Agency Portfolio Manager dans staging.
2. Créer CPT Réalisations et Services.
3. Ajouter taxonomy Secteurs et des métadonnées sécurisées.
4. Créer une page Settings API et enregistrer un nombre compris entre 1 et 24.
5. Tester un utilisateur qui ne peut pas enregistrer le réglage.
6. Désactiver/réactiver et vérifier conservation des contenus.

Preuves à remettre : lien commit Git, capture terminal, trace logs, fiche erreurs rencontrées, explication de la correction.

Barème : 20 points selon la grille du manuel.

TP semaine 6 — REST CRUD, JavaScript, permissions et PHPUnit

But : Créer, lire, modifier et supprimer logiquement des contenus via des endpoints testés.

1. Ajouter route GET publique ne renvoyant que les projets publiés.
2. Déclarer arguments per_page bornés entre 1 et 24.
3. Créer une route POST qui crée un brouillon pour un utilisateur autorisé.
4. Ajouter PATCH/DELETE et vérifier edit_post/delete_post objet par objet.
5. Créer des tests REST de lecture anonyme, écriture anonyme et écriture autorisée.
6. Afficher les données via fetch dans un composant sans innerHTML non fiable.

Preuves à remettre : lien commit Git, capture terminal, trace logs, fiche erreurs rencontrées, explication de la correction.

Barème : 20 points selon la grille du manuel.

TP semaine 7 — Crawl, indexabilité, canonicals et JSON-LD

But : Auditer les signaux techniques SEO et publier des données structurées non dupliquées.

1. Créer un inventaire de 10 URL et mesurer statut/canonical.
2. Contrôler titres, méta description, H1, liens et alternatifs.
3. Vérifier la cohérence sitemap/robots/noindex.
4. Activer le JSON-LD uniquement s'il n'est pas déjà émis par le SEO plugin.
5. Valider BlogPosting et BreadcrumbList selon pertinence.
6. Rédiger un journal des anomalies classées par impact.

Preuves à remettre : lien commit Git, capture terminal, trace logs, fiche erreurs rencontrées, explication de la correction.

Barème : 20 points selon la grille du manuel.

TP semaine 8 — Core Web Vitals, SQL, CI/CD et rollback

But : Améliorer des métriques mesurées et livrer de façon reproductible.

1. Relever un état avant optimisation : URLs, timings, captures et configurations.
2. Identifier la ressource LCP et corriger si nécessaire le chargement.
3. Tester images responsive et réservations d'espace pour limiter CLS.
4. Réduire les ressources inutiles en préservant les fonctionnalités.
5. Exécuter la CI statique puis une matrice réelle PHP/WordPress en staging.
6. Créer et restaurer une sauvegarde de validation, puis documenter le rollback.

Preuves à remettre : lien commit Git, capture terminal, trace logs, fiche erreurs rencontrées, explication de la correction.

Barème : 20 points selon la grille du manuel.