



DÉVELOPPEMENT BACKEND

PHP moderne : bases du développement côté serveur

Apprendre PHP 8, le langage serveur le plus utilisé du web. Variables, fonctions, connexion à une base de données MySQL : les bases pour construire des applications web dynamiques.

DURÉE	TARIF HT	NIVEAU	LANGUE	GROUPE	FORMAT
3j (21h)	790.00 €	Débutant	Français	3-12	Formation

1 PUBLIC VISÉ

- Débutants souhaitant apprendre un premier langage backend pour construire des applications web dynamiques.
- Personnes en reconversion vers le développement web ayant déjà des notions de [HTML et CSS](#).
- Développeurs front-end voulant comprendre ce qui se passe côté serveur et élargir leur profil vers le full-stack.
- Toute personne souhaitant créer des sites web dynamiques connectés à une base de données sans passer par un CMS.

2 PRÉREQUIS

- Notions de [HTML et CSS](#) : savoir créer une page web statique avec des formulaires.
- Aucune expérience en backend ou en programmation côté serveur n'est requise.
- Savoir utiliser un navigateur web et un éditeur de texte (VS Code recommandé).

3 OBJECTIFS PÉDAGOGIQUES

- Comprendre le fonctionnement de l'architecture client-serveur et le rôle de PHP dans une application web
- Maîtriser la syntaxe PHP 8 : variables, conditions, boucles, fonctions et tableaux
- Traiter des formulaires HTML avec PHP en sécurisant les données reçues
- Interagir avec une base de données MySQL via PDO : créer, lire, modifier et supprimer des données
- Gérer les sessions et les cookies pour maintenir l'état d'un utilisateur entre les pages
- Comprendre les bases de la sécurité web : injection SQL, XSS, CSRF, validation et filtrage des entrées
- Organiser un projet PHP avec une architecture claire et maintenable
- Construire une application web dynamique complète de bout en bout

**4 PROGRAMME DÉTAILLÉ****Jour 1 (7h) - Fondamentaux PHP et traitement des données****Module 1 - Découverte de PHP et de l'environnement (2h)**

Architecture client-serveur (1h)

- Comment fonctionne le web : navigateur, serveur, HTTP
- Le rôle de PHP : générer du HTML dynamiquement côté serveur
- PHP dans l'écosystème : WordPress, Symfony, Laravel — pourquoi apprendre PHP en 2025
- Cycle de vie d'une requête HTTP : de la saisie URL à l'affichage de la page

Mise en place de l'environnement (1h)

- Installer XAMPP ou Laragon : Apache, MySQL, PHP en local
- Structure d'un projet PHP : dossiers, fichiers, point d'entrée
- Premier script PHP : balises d'ouverture, echo, var_dump
- Activer l'affichage des erreurs en développement
- Cas pratique : afficher une page HTML générée par PHP avec la date du jour

Module 2 - Syntaxe PHP 8 : variables, conditions et boucles (3h)

Variables et types de données (1h)

- Déclarer une variable en PHP : la convention du dollar
- Les types natifs : string, int, float, bool, null, array
- Typage dynamique et inférence : comment PHP détermine le type
- Les nouveautés PHP 8 : types union, nullsafe operator, match expression
- Concaténation de chaînes et interpolation de variables
- Cas pratique : afficher une fiche produit avec des variables

Conditions et structures de contrôle (1h)

- if, elseif, else : syntaxe et opérateurs de comparaison
- L'opérateur ternaire et le null coalescing operator ??
- switch et match : choisir selon une valeur
- Cas pratique : afficher un message personnalisé selon l'heure de la journée

Boucles (1h)

- while, do...while : boucler tant qu'une condition est vraie
- for : boucler un nombre défini de fois
- foreach : parcourir un tableau
- break et continue : contrôler le flux d'une boucle
- Cas pratique : afficher une liste de produits depuis un tableau PHP

Module 3 - Fonctions et tableaux (2h)

Fonctions (1h)

- Déclarer et appeler une fonction
- Paramètres, valeurs par défaut et typage des paramètres en PHP 8
- Valeur de retour et type de retour déclaré
- Portée des variables : locale vs globale



Fonctions anonymes et closures

- Cas pratique : bibliothèque de fonctions utilitaires pour formater des prix et des dates

Tableaux (1h)

- Tableaux indexés et tableaux associatifs
- Tableaux multidimensionnels
- Fonctions essentielles : array_map, array_filter, array_reduce, usort
- Déstructuration de tableaux avec list() et la syntaxe courte []
- Cas pratique : filtrer et trier un catalogue de produits

Jour 2 (7h) - Formulaires, sécurité et base de données

Module 4 - Formulaires et traitement des données (2h)

Recevoir et traiter les données d'un formulaire (1h)

- Les superglobales PHP : \$_GET, \$_POST, \$_SERVER, \$_FILES
- Différence entre GET et POST : cas d'usage et sécurité
- Vérifier la présence et la validité des données : isset, empty, filter_input
- Afficher les erreurs de validation au-dessus du formulaire
- Cas pratique : formulaire de contact avec validation côté serveur

Sécuriser les données reçues (1h)

- Ne jamais faire confiance aux données utilisateur : le principe de base
- Échapper les sorties HTML avec htmlspecialchars pour prévenir le XSS
- Valider et filtrer avec filter_var : emails, URLs, entiers
- Préparer les données avant insertion en base de données
- Cas pratique : sécuriser un formulaire d'inscription complet

Module 5 - Base de données MySQL et PDO (4h)

Modéliser et créer une base de données (1h)

- Les bases de MySQL : tables, colonnes, types de données, clés primaires
- Créer une base de données et des tables avec phpMyAdmin
- Relations entre tables : clés étrangères, jointures
- Cas pratique : concevoir le schéma d'une application de gestion d'articles

Connexion et requêtes avec PDO (1h30)

- Pourquoi PDO et pas mysqli : portabilité et sécurité
- Créer une connexion PDO : DSN, gestion des exceptions
- Requêtes simples avec query() : lire des données
- Requêtes préparées avec prepare() et execute() : la règle absolue contre l'injection SQL
- Récupérer les résultats : fetch, fetchAll, fetchColumn
- Cas pratique : afficher la liste des articles depuis la base de données

Opérations CRUD complètes (1h30)

- CREATE : insérer un enregistrement avec lastInsertId
- READ : lire un enregistrement par son identifiant
- UPDATE : modifier un enregistrement existant
- DELETE : supprimer un enregistrement avec confirmation



Cas pratique : interface complète de gestion d'articles avec ajout, modification et suppression

Module 6 - Sessions, cookies et authentification (1h)

- Les cookies : créer, lire, modifier, supprimer avec `setcookie()`
- Les sessions PHP : `session_start()`, `$_SESSION`, `session_destroy()`
- Implémenter une authentification simple : connexion, déconnexion, accès protégé
- Sécuriser les mots de passe : `password_hash()` et `password_verify()`
- Cas pratique : page de connexion avec redirection selon le statut de la session

Jour 3 (7h) - Approfondissement et projet de synthèse

Module 7 - Organisation du code et bonnes pratiques (3h)

Structurer un projet PHP (1h30)

- Séparer la logique PHP du rendu HTML : le pattern MVC simplifié
- Les fichiers `include` et `require` : factoriser l'entête, le pied de page, la connexion BDD
- Organiser ses dossiers : `public/`, `includes/`, `config/`, `views/`
- Le fichier de configuration centralisé : constantes et paramètres de connexion
- Cas pratique : refactoriser le projet CRUD avec une architecture propre

Gestion des erreurs et débogage (1h)

- Les niveaux d'erreur PHP : `notice`, `warning`, `fatal error`
- `try`, `catch`, `finally` : gérer les exceptions proprement
- Journaliser les erreurs avec `error_log`
- Différencier environnement de développement et de production
- Cas pratique : ajouter une gestion d'erreur robuste à la connexion PDO

Sécurité web approfondie (30min)

- Les attaques CSRF : comprendre et implémenter un token de protection
- Upload de fichiers sécurisé : vérifier le type MIME, limiter la taille
- Les headers HTTP de sécurité : `Content-Security-Policy`, `X-Frame-Options`

Module 8 - Projet de synthèse (4h)

Réalisation d'une application PHP complète

- Cahier des charges : application de gestion de contenu avec authentification
- Fonctionnalités : inscription et connexion sécurisée avec `password_hash`, CRUD complet sur une ressource, sessions pour protéger les pages d'administration, validation et sécurisation des formulaires, affichage dynamique des données depuis MySQL, architecture organisée avec `include` et fichiers de configuration
- Étape 1 : mise en place de la base de données et de la connexion PDO
- Étape 2 : pages d'inscription et de connexion avec gestion de session
- Étape 3 : interface d'administration CRUD protégée
- Étape 4 : affichage public des données avec pagination
- Revue de code collective : sécurité, organisation, lisibilité
- Retour formateur individualisé sur le projet rendu



5 COMPÉTENCES VISÉES

- Écrire des scripts PHP 8 fonctionnels et lisibles respectant les bonnes pratiques
- Concevoir et interroger une base de données MySQL avec PDO de manière sécurisée
- Traiter et valider des formulaires HTML côté serveur
- Gérer les sessions utilisateur pour implémenter une authentification simple
- Structurer une application PHP avec une organisation de fichiers claire et maintenable
- Appliquer les bases de la sécurité web pour protéger une application contre les vulnérabilités courantes

6 MODALITÉS PÉDAGOGIQUES

Formation délivrée en présentiel ou distanciel (visioconférence). Le formateur alterne entre méthode démonstrative (live coding commenté et progressif), méthode interrogative (analyse de code existant et identification des erreurs classiques) et méthode active (exercices pratiques et mini-projets par module). L'accent est mis sur la compréhension du fonctionnement réel du serveur web plutôt que sur la mémorisation de la syntaxe.

7 MOYENS PÉDAGOGIQUES ET TECHNIQUES

- Support de cours numérique mis à disposition des apprenants
- Dépôt GitHub de démonstration avec exercices et corrections par module
- Environnement de développement : VS Code + XAMPP ou Laragon + phpMyAdmin
- 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 : exercices pratiques et mini-projets corrigés à chaque module
- En fin de formation : réalisation d'une application PHP complète avec authentification et base de données
- Questionnaire d'auto-évaluation des acquis en fin de parcours

9 CRITÈRES D'ÉVALUATION

- Écriture de scripts PHP 8 fonctionnels et lisibles sans erreurs de syntaxe
- Utilisation systématique des requêtes préparées PDO pour toute interaction avec la base de données
- Validation et sécurisation correcte des données reçues depuis les formulaires
- Implémentation fonctionnelle de la gestion de session pour protéger les pages d'administration
- Organisation du code avec une séparation claire entre logique métier et rendu HTML

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 réalisation du projet de synthèse. 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 exercices corrigés et le projet de synthèse

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.

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.
Réfèrent handicap : contact@lapolaris.fr

LaPolaris

TÉL. +33762584798

EMAIL contact@lapolaris.fr

WEB lapolaris.fr