

PHP 7 et 5, industrialisation des développements les bonnes pratiques



SII-298 3 Jours (21 Heures)

Description

Ce cours de perfectionnement est axé sur trois thèmes : l'optimisation du code au niveau algorithmique, l'adaptation de l'architecture logicielle aux nécessités du travail en équipes pluridisciplinaires ou multisites, et la rationalisation des développements orientés maintenabilité et réutilisabilité.

À qui s'adresse cette formation ?

Pour qui

Développeurs désirant améliorer la pérennité et la maintenabilité de leurs développements.

Prérequis

Aucune

Les objectifs de la formation

- Analyser les performances d'une application et optimiser le code PHP associé
- Améliorer la productivité du développement d'une application web PHP
- Mettre en oeuvre une architecture logicielle multicouche
- Introduire les Design Patterns dans la conception d'une application web PHP

Programme de la formation

Introduction

- Problèmes classiques rencontrés en développement.
- Culture d'entreprise : impacts à long et moyen termes.
- Inflation non contrôlée du code et théorie de l'avalanche.
- Réutilisabilité : stratégies et enjeux.

Optimisation de code

- Outils : de mesure (Pear), de simulation de charge client/serveur, d'optimisation.
- Compression à la volée.
- Principe de la compression/décompression des pages.
- Précautions et limites.
- Optimisation des algorithmes.
- Optimisation des boucles, des requêtes SQL.
- Fichiers texte et bases de données.
- Gestion des flux de sortie.
- Cache serveur.
- Réalisation d'un cache.
- Les classes existantes.
- Travaux pratiques Optimisation graduelle d'une page type, avec mesures comparées des performances obtenues.

Productivité

- La POO : bienfaits et limites.
- Etude de cas, pertinence de l'approche POO vs procédurale.
- La solution mixte.
- IDE et documentation.
- Les principaux IDE du marché.
- Outils de documentation et conventions.
- Configuration.
- Serveurs à balancement de charge.
- Approche matérielle.
- Impact sur le développement.
- Travaux pratiques Analyse de performances et consommation mémoire.
- Utilisation d'un IDE et d'un générateur de documentation.
- Etude d'une classe "session" pour la conservation des données en environnement multiserveurs.

Rationalisation des développements

- Architecture multicouche.
- Programmation " spaghetti " et maintenabilité.
- Approche pluridisciplinaire et sous-traitance.
- MVC : diviser pour mieux régner.
- MVC sans l'objet.
- Concepts avancés de la POO : classes abstraites, interfaces.
- Méthodes automatiques, surcharge, sauvegarde, clonage.
- Frameworks (cadres).
- Philosophie.
- Etude d'un framework " maison " .
- Principaux frameworks du marché.
- Design patterns (motifs de conception).
- Principe de base et avantages.
- Structure des patterns Usine.
- Singleton.
- Chaîne de commande.
- Observateur.
- Stratégie.
- Encapsulation des concepts variables.
- Travaux pratiques Passage d'une page " spaghetti " à un modèle MVC.
- Réalisation d'un moteur de recherche avec un framework.
- Implémentation d'une classe de connexion pour la conservation des ressources.