PHP 7 and 5, industrialization of developments, good practices

★★★★★

SII-298 **3 Days (21 Hours)**

Description

This advanced course focuses on three themes: optimization of code at the algorithmic level, adaptation of software architecture to the needs of working in multidisciplinary or multisite teams, and rationalization of developments oriented towards maintainability and reusability.

Who is this training for ?

For whom

Developers wishing to improve the sustainability and maintainability of their developments.

Prerequisites

Aucune

Training objectives

- Analyze the performance of an application and optimize the associated PHP code
- Improve the productivity of developing a PHP web application
- Implement a multi-layer software architecture
- Introduce Design Patterns in the design of a PHP web application

Training program

Introduction

- Classic problems encountered in development.
- Corporate culture: long and medium term impacts.
- Uncontrolled code inflation and avalanche theory.
- Reusability: strategies and challenges.

Optimisation de code

- Tools: measurement (Pear), client/server load simulation, optimization.
- On-the-fly compression.
- Principle of compression /decompression of pages.
- Precautions and limits.
- Optimization of algorithms.
- Optimization of loops, SQL queries.
- Text files and databases.
- Output stream management.
- Server cache.
- Creating a cache.
- Existing classes.
- Practical work Gradual optimization of a typical page, with comparative measurements of the performances obtained.

Productivité

- OOP: benefits and limits.
- Case study, relevance of the OOP vs procedural approach.
- The mixed solution.
- IDE and documentation.
- The main IDEs on the market.
- Documentation tools and conventions.
- Configuration.
- Swinging servers load.
- Hardware approach.
- Impact on development.
- Practical work Performance analysis and memory consumption.
- Use of an IDE and a documentation generator.
- Study of a "session" class for data conservation in a multi-server environment.

Rationalisation des développements

- Multi-layer architecture.
- "Spaghetti" programming and maintainability.
- Multidisciplinary approach and outsourcing.
- MVC: divide to rule better.
- MVC without the object.
- Advanced OOP concepts: abstract classes, interfaces.
- Automatic methods, overloading, saving, cloning.
- Frameworks (frameworks).
- Philosophy.
- Study of an in-house framework.
- Main frameworks on the market .
- Design patterns.
- Basic principle and advantages.
- Structure of Factory patterns.
- Singleton.
- Command string.
- Observer.
- Strategy.
- Encapsulation of variable concepts.
- Practical work Transition from a "spaghetti" page to an MVC model.
- Creation of a search engine with a framework.
- Implementation of a connection class for the conservation of resources.