

Training: Rogue Wave  
 PHP Fundamentals II


FORM OF TRAINING	MATERIALS	PRICE	DURATION
Traditional	Digital materials	870 EUR	3 days
Traditional	CTAB Tablet	970 EUR	3 days
Distance learning	Digital materials	870 EUR	3 days
Distance learning	CTAB Tablet	870 EUR	3 days

### LOCATIONS

Krakow - 5 Tatarska Street, II floor, hours: 9:00 am - 4:00 pm  
 Warsaw - 17 Bielska Street, hours: 9:00 am - 4:00 pm

### TRAINING TERMS

2019-09-30 | 3 days | Warszawa

## TRAINING GOALS:

Designed for intermediate PHP developers, PHP Fundamentals II enables developers to further advance their professional skills in the PHP language and adopt industry best practices. It utilizes a hands-on approach led by an experienced instructor with numerous examples and practical exercises based on real life software development practices to enhance the hands on learning experience.

What will I learn:

- PHP Fundamentals II is designed to teach the intermediate PHP developer the higher level concepts, including a detailed introduction to object-oriented programming. You will be able to use advanced language and web development methodologies to creating more advanced web-based PHP and mobile backend applications. This class provides you with the skills and knowledge required to advance to modern PHP enterprise grade Frameworks development.

What will I be able to achieve:

- Utilize new PHP 7 concepts and language constructs
- Create database-driven web applications similar to the course applications
- Leverage object-oriented programming (OOP) techniques in your applications
- Use built-in objects to interface a database
- Analyze input data and learn to filter and validate it, and why
- Request a web service and process the return data

- Learn how to better recognize inefficient coding practices, and improve them
- Learn best practices
- Learn a few concepts called software design patterns
- How to throw and handle exception objects

#### Audience:

- This course is designed for PHP developers who have at least 12 months of active development/experience with PHP. An understanding of object-oriented programming in PHP is helpful.

#### CONSPECT:

- INTRODUCTIONS
  - Course introduction
  - Prerequisite knowledge
  - Course exercises
  - Overall objective
- PHP CONFIGURATION
  - Configuration directives
  - Run-time settable directives
  - Web server PHP and CLI configuration
  - Target environment configuration
  - Technology stack configuration for the course VM
- OBJECT-ORIENTED PHP - CONSTRUCTS
  - The class construct
  - Class constants, properties and methods
  - Object instances
  - Member visibility
  - Static properties and methods
  - Inheritance
  - Overrides
  - Magic methods
  - Class and method abstraction
  - Object interface
  - Type hinting
  - Traits
  - Namespace

- Object cloning and comparison
- Functions that work with classes
- Exceptions
- OBJECT-ORIENTED PHP - IMPLEMENTATION
  - Course application OOP implementation
  - File structure organization
  - Architectural thought of the course application
  - Model/View/Controller design pattern implementation
  - Controllers and input classes
  - Core classes and their role
  - Domain classes, services and business logic
  - Form and input classes and abstraction
  - Model classes and abstraction, database access and patterns
  - View classes and layouts
- PHP STANDARDS
  - Basic coding standard
  - Logger interface and standard
  - Autoloading and autoloading standard
  - Caching interface and standard
  - Messaging interface and standard
- DATABASE
  - Databases
  - The relational database engine
  - Structured query language (SQL)
  - Data modeling
  - PhpMyAdmin MySQL admin
  - Terminal MySQL
  - PHP data objects (PDO) API
  - Prepared statements
  - Stored procedure
  - Transactions
  - Patterns in database design
- INTERNET COMMUNICATION
  - Protocols and ports
  - Client/Server
  - HTTP/HTTPS

- Client side
- HTML web forms
- Server side
- Server side input validation
- Cookies
- Sessions
- Output control
- Headers and browser caching
- COMPOSER
  - Composer
  - Composer files
  - composer.phar
  - composer.json
  - Important composer commands
  - The "vendor" directory
  - Composer API docs and packages
- WEB SERVICES
  - Web services
  - Data formats
  - Parsing APIs
  - Web service requests
  - Soap services
  - RESTful services
  - Streams
- FINAL BINDINGS
  - Regular expressions
  - Software unit testing
  - PHP documenter
  - Web security

## REQUIREMENTS:

- INTRODUCTIONS
  - Course introduction
  - Prerequisite knowledge
  - Course exercises

- Overall objective
- PHP CONFIGURATION
  - Configuration directives
  - Run-time settable directives
  - Web server PHP and CLI configuration
  - Target environment configuration
  - Technology stack configuration for the course VM
- OBJECT-ORIENTED PHP - CONSTRUCTS
  - The class construct
  - Class constants, properties and methods
  - Object instances
  - Member visibility
  - Static properties and methods
  - Inheritance
  - Overrides
  - Magic methods
  - Class and method abstraction
  - Object interface
  - Type hinting
  - Traits
  - Namespace
  - Object cloning and comparison
  - Functions that work with classes
  - Exceptions
- OBJECT-ORIENTED PHP - IMPLEMENTATION
  - Course application OOP implementation
  - File structure organization
  - Architectural thought of the course application
  - Model/View/Controller design pattern implementation
  - Controllers and input classes
  - Core classes and their role
  - Domain classes, services and business logic
  - Form and input classes and abstraction
  - Model classes and abstraction, database access and patterns
  - View classes and layouts
- PHP STANDARDS

- Basic coding standard
- Logger interface and standard
- Autoloading and autoloading standard
- Caching interface and standard
- Messaging interface and standard
- DATABASE
  - Databases
  - The relational database engine
  - Structured query language (SQL)
  - Data modeling
  - PhpMyAdmin MySQL admin
  - Terminal MySQL
  - PHP data objects (PDO) API
  - Prepared statements
  - Stored procedure
  - Transactions
  - Patterns in database design
- INTERNET COMMUNICATION
  - Protocols and ports
  - Client/Server
  - HTTP/HTTPS
  - Client side
  - HTML web forms
  - Server side
  - Server side input validation
  - Cookies
  - Sessions
  - Output control
  - Headers and browser caching
- COMPOSER
  - Composer
  - Composer files
  - composer.phar
  - composer.json
  - Important composer commands
  - The "vendor" directory

- Composer API docs and packages
- WEB SERVICES
  - Web services
  - Data formats
  - Parsing APIs
  - Web service requests
  - Soap services
  - RESTful services
  - Streams
- FINAL BINDINGS
  - Regular expressions
  - Software unit testing
  - PHP documenter
  - Web security

## Difficulty level



## CERTIFICATE:

This course assumes the completion of the PHP Fundamentals I or PHP Foundations for IBM i Programmers courses or equivalent knowledge and development experience. This course requires a foundation-level knowledge of the PHP language and syntax.

## TRAINER:

Rogue Wave Zend Certified Trainer.

## ADDITIONAL INFORMATION:

The participants will obtain certificates signed by Rogue Wave Zend.

This course is also intended to help participants prepare for the ZCE PHP certification exam (Zend Certified Engineer PHP). ZCE PHP certification exams are offered at Pearson VUE test centers worldwide. More information about ZCE PHP certification on the <http://www.zend.com/en/services/certification/php-certification>