MODERN PHP KICKOFF

Modern PHP Development Intro

Ramy Hakam

Our Steps

PHP In Deep

PHP Behind The Scenes

PHP Basics

Types, Functions, Classes,

OOP

OOP in PHP

Composer/PEAR

Composer and PEAR

PSR/PHP-FIG

PHP Standards

PHPUnit /Debuging

Xdebug in php

PHP Versions

what is new in PHP

PHP MVC

The famous Framworks

PHP Events

Event Driven & CQRS In PHP

DDD &TDD

Splitting Your business layer

Agnostic PHP

Agnostic PHP Application

PHP Docker

Dockerize your PHP Apps

Let's Dig in

PHP

PHP Behind The scenes

Is PHP compiled or interpreted?

PHP Is Compiled

TO

1-Convert the code to a bytecode

2-Resolve Functions and Classes

3-Creating a symbol table

PHP Is Interrupted

TO

1- Goes through the bytecode
Line by line and executes it

2- Handles runtime exceptions

```php
<?php
$show_value    = 123;
echo 'sing_quote'.$show_value;
echo "double_quote{$show_value}";
?>
```

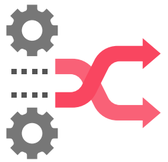
PHP Code

OPCode

```
1: <?php
2: $show_value   = 123;
        0  ASSIGN                !0, 123
3: echo 'sing_quote'.$show_value;
        1  CONCAT                'sing_quote', !0 =>RES[~1]
        2  ECHO                  ~1
4: echo "double_quote{$show_value}";
        3  ADD_STRING            'double_quote' =>RES[~2]
        4  ADD_VAR               ~2, !0 =>RES[~2]
```

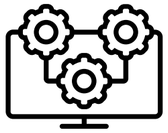
PHP Code

Your PHP files / Classes



OpCode

Virtual Machine Code



Interrupt OpCode

Execute Code Instructions

Send It To Web Server

Send Response To web Browser

PHP 101

PHP BASICS

PHP Basics Refreshment

PHP Basics

Types

String

Array

Integers

Boolean

Float

PHP 101

Loops

Classes

Functions

Callbacks

Anonymous

PHP SPL

Data Structures

Design Patterns

Heap/Queue

Stack /Linked List

Observer Pattern

PHP 101

PHP OOP

Object-oriented in PHP

OOP Concepts In PHP

- Classes / Objects
- Inheritance
- Interfaces
- Abstraction
- Traits

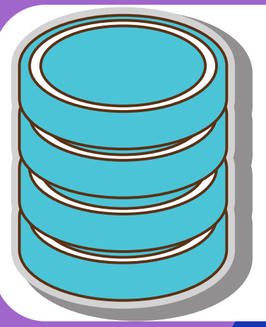
PHP 101

COMPOSER/ PEAR

PHP Package Managers

Composer

Your Favourite package manager



Packagist

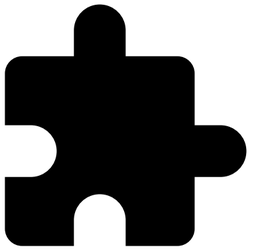
Autoloader

Scripts

PEAR

PHP Extention Manager

Packages



Extensions

Code Standard

What Is PECL?
The sisiter of PEAR?

PHP 101

PHP PSR/FIG

PHP Standard Recommendation

PHP CodeSniffer !

PHP 101

PHP UNIT

PHP Test Library

Unit Testing	Integration Testing	Functional Testing
Test One Class independently	Test More Than one class	Test A full Featrues
Mock All Other services	Inegrate Services together	Call external APIS
Check How internal Code Works	Slower and more complexity	Check all The business cases

PHP 101

PHP DEBUGGING

Xdebug For PHP

STOP Using PHP ECHO

PHP developers that don't use Xdebug for debugging are amateurs.

PHP 101

PHP VERSIONS

The Future of PHP

PHP Versions

PHP 5

Constructors

Interfaces

New MySQL Extension

XML

OPcache

PHP 7

Strict Types

Return Types

Array unpack

Anonymous Classes

Preloader

PHP 8

Attributes

Enumeration

Fibers

Union Types

JIT Compiler

HHVM For PHP and Hack

PHP 8 Can be used For

Machine Learning

Big Data

PHP 101

PHP MVC

PHP Frameworks

MVC Alternatives

Async PHP

React php

Swoole

PHP SWoole

```php
<?php
use Swoole\Http\Server;
use Swoole\Http\Request;
use Swoole\Http\Response;

$server = new Swoole\HTTP\Server("127.0.0.1", 9501);

$server->on("start", function (Server $server) {
    echo "Swoole http server is started at http://127.0.0.1:9501\n";
});

$server->on("request", function (Request $request, Response $response) {
    $response->header("Content-Type", "text/plain");
    $response->end("Hello World\n");
});

$server->start();
```

PHP 101

PHP EVNTS

Events and CQRS

Event and Listeners !

Command Query Responsibility Segregation (CQRS)

CQRS & Event Sourcing

Broadway

Prooph

PHP 101

PHP DDD & TDD

PHP Advanced Architecture

DDD in PHP

DDD

Strategic Design

Bounded contexts

Ubiquitous Language

Context Maps

Tactical design

Entities

Services

Repositories

Events

Factories

Architecture design

Hexagonal

Layered

CQRS

DDD Layers

Presentation Layer (Controller)

This layer is the part where interaction with external systems happens

Application Layer (CQRS/Services)

It is the layer where business process flows are handled

Domain Layer (Entities)

It is the layer where all business rules related to the problem to be solved

Infrastructure Layer (External Links)

This layer will be the layer that accesses external services such as database

DDD & CQRS

TDD in PHP

TDD Steps

1. Read, understand, and process the feature or bug request.

2. Translate the requirement by writing a unit test. If you have hot reloading set up, the unit test will run and fail as no code is implemented yet.

3. Write and implement the code that fulfills the requirement. Run all tests and they should pass, if not repeat this step

PHP 101

AGNOSTIC PHP

PHP Advanced Architecture

The Big Picture

DDD

CQRS

=

Agnostic

PHP 101

DOCKER & PHP

PHP Deployment

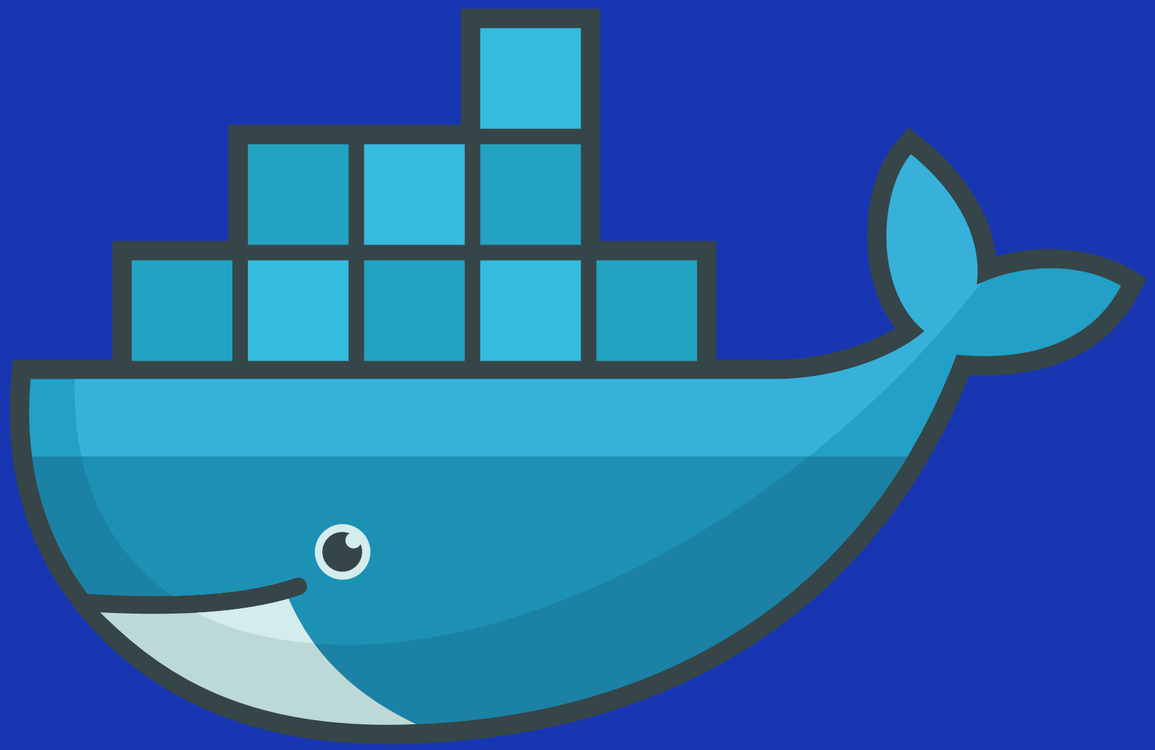
1- Check

2-Build

Phing Build Tool

3-TEST

4-Ship



PHP 101

CONCLUSION

PHP Modern Kickoff