

Programming PHP

Santosh Kalwar, 18.03.2024

PHP Topics

General Discussions

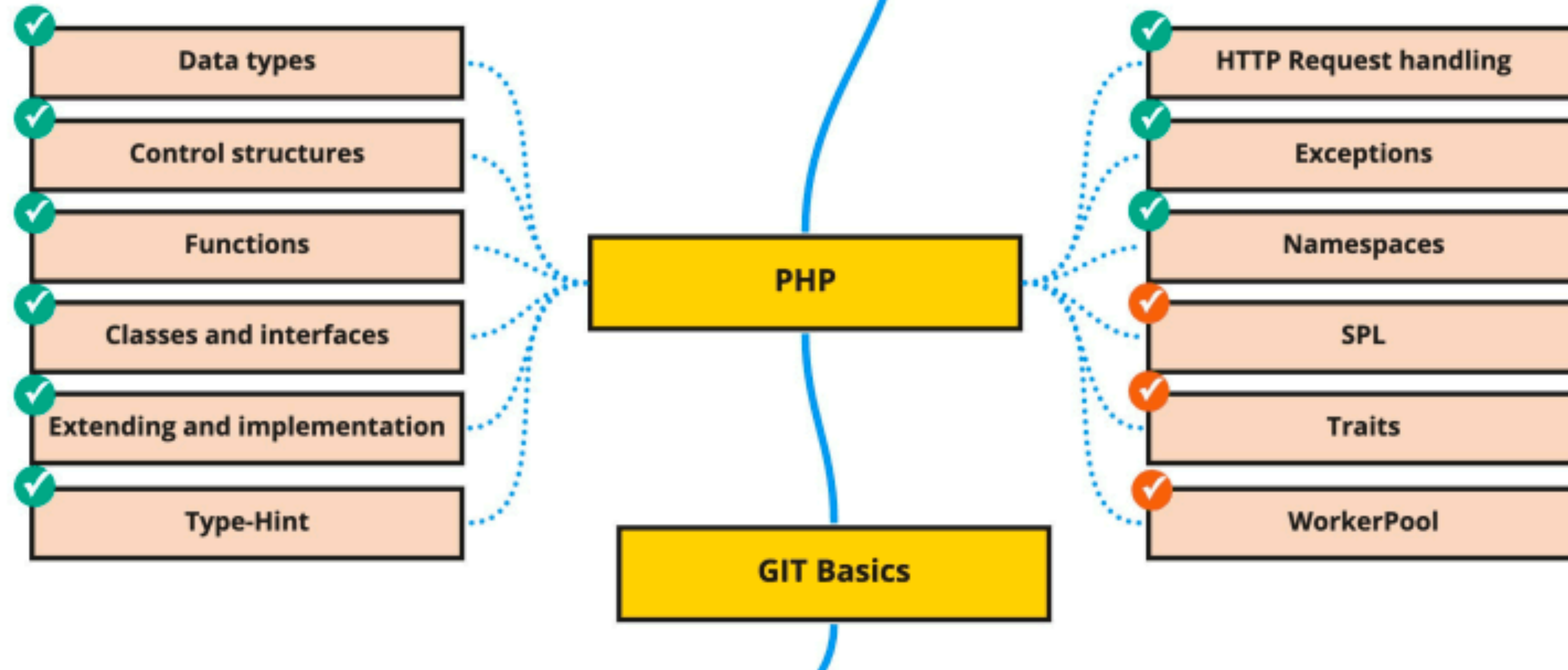
Recap of basic knowledge - Quiz

PHP

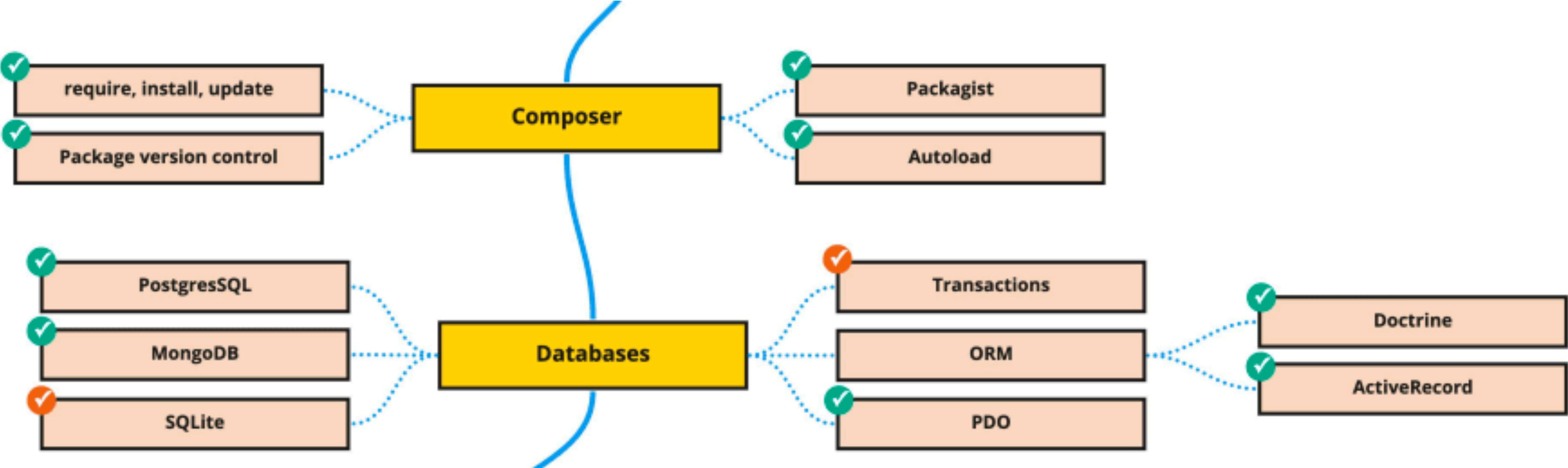
- PHP Developer Roadmap
 - <https://github.com/thecodeholic/php-developer-roadmap>
- VS Code Extentions
- PHP Installation
- You already went through PHP Basics (array, variables, operators, comparison, session, cookies)
- Local LAMP setup using Docker
- Recap Practices / Classroom coding

PHP Developer Roadmap

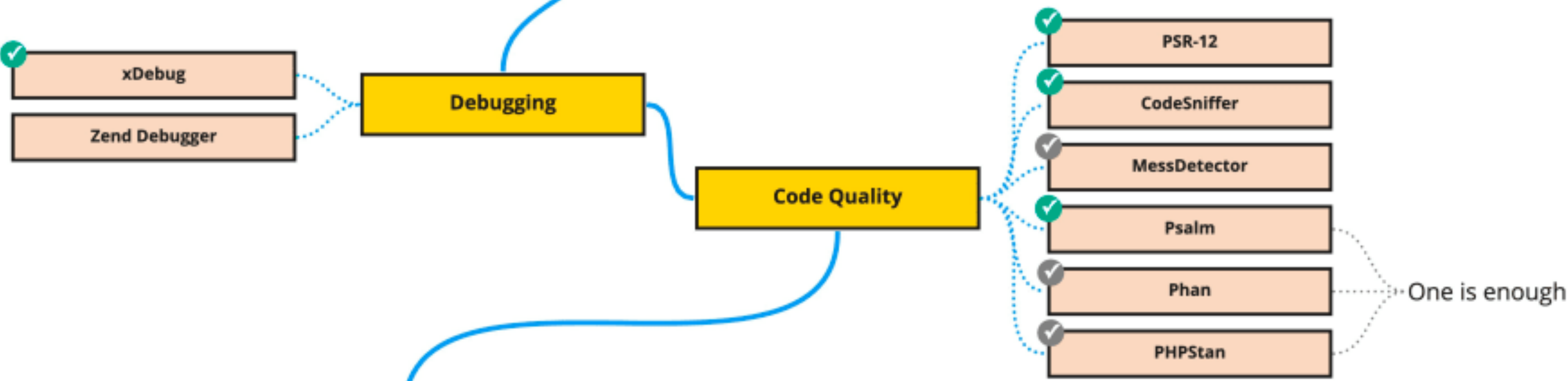
- ✔ Should be learned first of all
- ✔ Important, but not a priority
- ✔ Might be useful



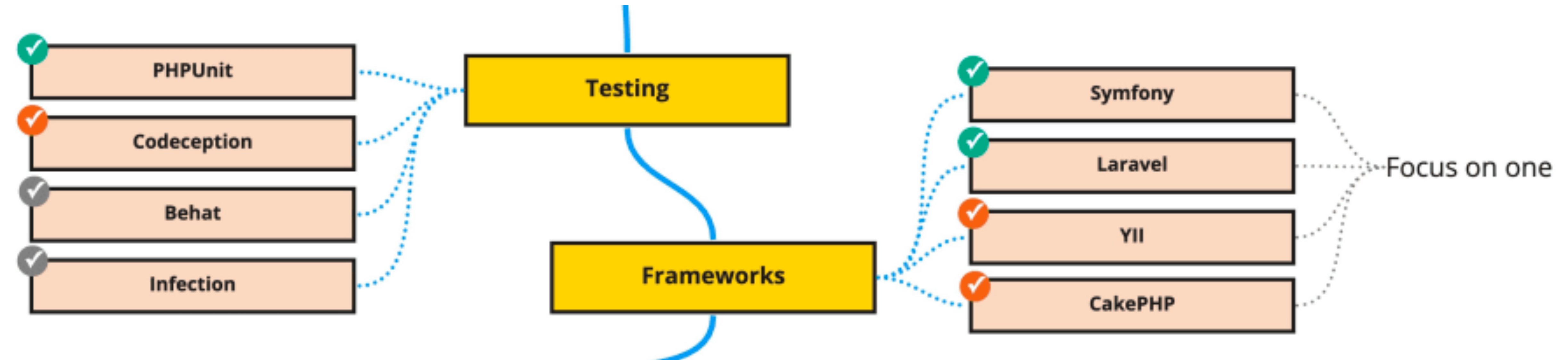
PHP Developer Roadmap



PHP Developer Roadmap



PHP Developer Roadmap



PHP Code Editors

- PhpStorm <https://www.jetbrains.com/phpstorm/>
- IntelliJ Idea <https://www.jetbrains.com/idea/>
- Netbeans <https://netbeans.apache.org/>
- VS Code <https://code.visualstudio.com/>
- Atom <https://atom.io/>
- Brackets <https://brackets.io/>

<https://kinsta.com/blog/php-editor/>

VS Code Extensions

Some PHP Code Extensions:

- PHP IntelliSense
<https://marketplace.visualstudio.com/items?fitemname=felixfbecker.php-intellisense>
- PHP Intelephense
<https://marketplace.visualstudio.com/items?fitemname=bmewburn.vscode-intelephense-client>
- phpfmt - PHP formatter
<https://marketplace.visualstudio.com/items?fitemname=kokororin.vscode-phpfmt>
- PHP Debug
<https://marketplace.visualstudio.com/items?fitemname=xdebug.php-debug>
- Add YAML extension
<https://marketplace.visualstudio.com/items?fitemname=redhat.vscode-yaml>

PHP Mac Installation

Using homebrew

```
brew install php  
php -v
```

```
php -S localhost:8008 helloWorld.php
```

<https://www.php.net/manual/en/install.macosx.packages.php>

Test Local PHP Mac Installation

```
<?php echo '<p>Hello PHP</p>'; ?>
```

What is PHP Stack?

- Linux
- Apache
- MySQL
- PHP

<https://www.zend.com/blog/choosing-right-php-stack>

PHP stack Installation

Traditional way to install PHP stack

<https://www.mamp.info/en/mamp/mac/>



LAMP



WAMP



MAMP

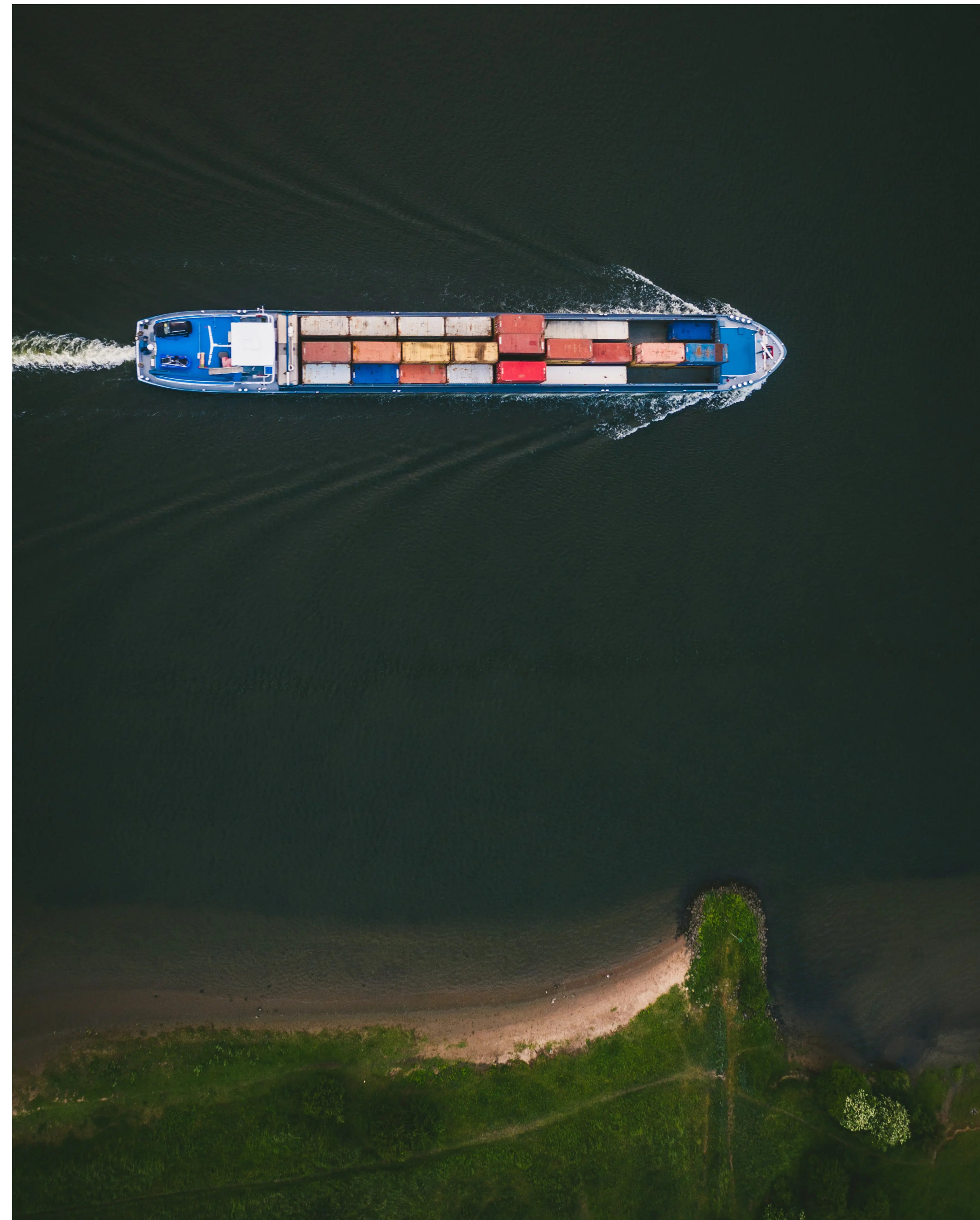


Why Docker?

Developing apps today requires so much more than writing code. Multiple languages, frameworks, architectures, and discontinuous interfaces between tools for each lifecycle stage creates enormous complexity.

Docker simplifies and accelerates your workflow, while giving developers the freedom to innovate with their choice of tools, application stacks, and deployment environments for each project.

<https://www.docker.com/why-docker/>

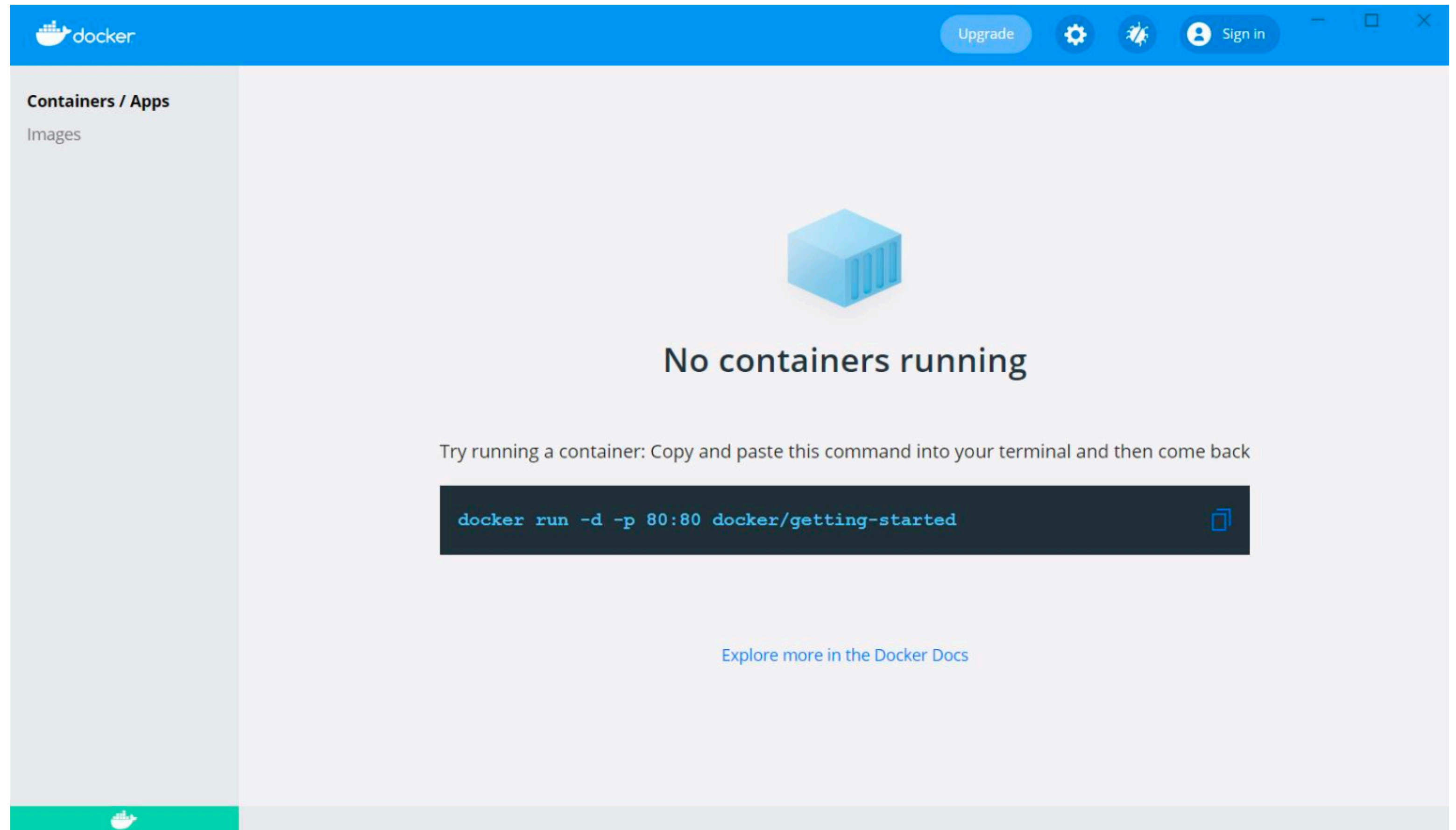


PHP stack Installation

Modern way to use PHP stack

- Docker Desktop

<https://www.docker.com/products/docker-desktop/>



Test PHP stack Installation

- PHP-MAMP should connect with MySQL
- phpMyAdmin
- Apache

Steps

1. git clone <https://github.com/kalwar/PHP-MAMP>
2. Make sure docker is up and running in your computer by typing: **docker ps** in terminal
3. Run **docker-compose up** in project root folder of PHP-MAMP
4. Check Docker Desktop
5. Open the Apache application in [**http://localhost:8005**](http://localhost:8005)
6. Copy or drag and drop **PHP_practice01** and **PHP_practice01** from **itsLearning** Practice folder to **src/** folder of PHP-MAMP
7. phpMyadmin in [**http://localhost:9080**](http://localhost:9080)

And mySQL database should show it is connected

PHP recap practice steps

- Go to http://localhost:8005//PHP_practice01/index.php
- For the folder **PHP_practice01**: you should see all the Practice from 0 - 10
- Now, your task is to read comment from 0.php till 5.php and write code and see results in browser
- Navigate to http://localhost:8005//PHP_practice01/index.php
- Check results by clicking on Practice Section 0, Practice Section 1 ... Practice Section 5

Recap Practice / Classroom Coding

- **PHP_practice01/1.php:**
 - Use the Echo Function to say hello with html h1 tags embedded inside php
 - Write a comment above echo function and explain what function is doing
- **PHP_practice01/2.php:**
 - Make 2 variables called number1 and number2 and set 1 to value of 10 and other to value of 20
 - Add these two variables and display the sum with echo
 - Make 2 arrays with same values, one regular and other associative
- **PHP_practice01/3.php:**
 - Make an if statement with elseif and else to finally display a string saying, I love PHP
 - Make a for loop that displays 10 numbers
 - Make a switch statement that tests against one condition with 5 cases
- **PHP_practice01/4.php:**
 - Define a function and make it return a calculation of two numbers
 - Make a function that passes parameters and call it using parameter values
- **PHP_practice01/5.php:**
 - Use a pre-built math function and echo it
 - Use a pre-built string function and echo it
 - Use a pre-built array function and echo it

Recap Practice / Classroom Coding

- **PHP_practice02/section_a/c01/variables.php:** Write a PHP code to assign name and price for the cost of the candy.
- **PHP_practice02/section_a/updating-variables.php:** Write a PHP code to update a value in a variable so that the cost of candy per pack is shown and the initial name variable is changed to something else. For example, if initial name is “Guest”, you can update it to “Your Name”
- **PHP_practice02/section_a/associative-arrays.php:** Write a PHP code to create and store array in a variable called \$nutrition with fat, sugar and salt and display the contents of Nutrition (per 100G) in percentage

Recap Practice / Classroom Coding

- **PHP_practice02/section_a/indexed-arrays.php:** Write a PHP code to create and store array for \$best_sellers where it holds list of best-selling items but display only three best-selling items on the page. The list of best-sellers could be e.g. Chocolate, Mints, Fudge, Bubble gum, Toffee, Jelly Beans etc
- **PHP_practice02/section_a/updating-arrays.php:** Write a PHP code to create and store array in \$nutrition, the value that is stored for the fat content should be updated and add a new element e.g. fiber and write those values out to the page.
- **PHP_practice02/section_a/multidimensional-arrays.php:** Write a PHP code to store indexed arrays in an array or multidimensional array with variable called \$offers. Each element in the array stores associated array holding name, price, and stock level of an item that is on offer. Print out the name and price of all the products.
- **PHP_practice02/section_a/echo-shorthand.php:** Write a PHP code to display name and favorites candy using echo shorthand.

Recap Practice / Classroom Coding

- **PHP_practice02/section_a/arithmetic-operators.php:** Write a PHP code to calculate the cost of an order. Let us say there are three candy items, and the cost of per pack is \$5. Calculate the subtotal with tax of 20% and total amount
- **PHP_practice02/section_a/string-operator.php:** Write a PHP code to concatenate greeting e.g. “Thank you” to customer who bought his candy order. Customer name can be anything for example “Mr. James”. The page should show:

Mr. James's Order
Thank you, Mr. James

- **PHP_practice02/section_a/comparison-operators.php:** Write a PHP code to compare and check if the quantity wanted is less than or equal to quantity in stock. If the user can buy based on comparison and if value is true, page should show 1 and if false, the page should show nothing.
- **PHP_practice02/section_a/logical-operators.php:** Write a PHP code to check if the customer only wants to buy limited packs of candy. Check if there are enough items in stock and secondly check that the item can be delivered. You can put imaginary number to do the comparison.

Recap Practice / Classroom Coding

- **PHP_practice03/section_a/c02/if-statement.php**
 - Create a simple if statement to greet user if the name is not empty.
- **PHP_practice03/section_a/c02/if-else-statement.php**
 - Create a simple if else statement to check if candy is in stock or not. If candy is not in stock, then print the message “Sold Out”; if it is available in stock, print the message “In Stock.”

Recap Practice / Classroom Coding

- **PHP_practice03/section_a/c02/ternary-operator.php**
- Refactor using the ternary operator: Create a simple *if else statement* to check whether candy is in stock. If candy is not in stock, then print the message “Sold Out”; if it is available in stock, print the message “In Stock.”
- **PHP_practice03/section_a/c02/if-else-if-statement.php**

Create a simple if else if statement to check if candy is in the stock or is coming soon or sold out. If candy is not in stock then print message “Sold Out”, if it is available in stock, print message “In Stock”

Recap Practice / Classroom Coding

- **PHP_practice03/section_a/c02/switch-statement.php**

Create a simple *switch* statement to get 20% off chocolates on Monday and 20% off mints on Tuesday, and in all other cases, it should show “Buy three packs, get one free.”

- **PHP_practice03/section_a/c02/while-loop.php**

Create a simple *while* loop to find prices for multiple packs of candy. For example, if one pack costs \$1.99 how much would 5 pack costs? Display the prices for all 5 packs of candy.

Recap Practice / Classroom Coding

- **PHP_practice03/section_a/c02/for-loop.php**

Create a simple *for* loop to find the prices of multiple candy packs. Let us assume one pack of candy costs \$1.99. How much did ten packs cost? Display each pack's costs on the web page.

- **PHP_practice03/section_a/c02/for-loop-higher-counter.php**

Create a simple *for* loop to find prices of multiple higher packs of candy. Let us assume the customer wants 10 packs to 100 packs of candies. How much do 10 to 100 packs cost? Display from 10 packs to 100 packs cost on the web page.

Recap Practice / Classroom Coding

- **PHP_practice03/section_a/c02/foreach-loop.php**

Create a simple *foreach* loop for candy products, e.g. Toffee costs 2.99, Mints costs 1.99, and Fudge costs 3.49 and show it in the web page table.

- **PHP_practice03/section_a/c02/include-and-require-files.php**

Write a PHP Code to include header.php and footer.php and check the candy stock. Let us assume you have 25 stock of candy, so check if you have “Good availability”, you have “low stock”, or you are running “Out of stock.”

Practice / Classroom Coding

- **PHP_practice04/section_a/c03/basicfunction.php**
 - Create three functions to generate the values as shown in this table. Price for Toffee is 3, Mints is 2 and Fudge is 8.
 - The first function should look at stock levels and create a message indicating whether or not more stock should be ordered. If the stock is less than 10 no Re-Order necessary.
 - The second function should find the total value of stock for each item that is sold.
 - And finally the third function should calculate how much tax will be due when all of the remaining stock has been sold.

| The Candy Store | | | | |
|-----------------|-------|----------|-------------|---------|
| STOCK CONTROL | | | | |
| PRODUCT | STOCK | RE-ORDER | TOTAL VALUE | TAX DUE |
| Toffee | 12 | No | \$36 | \$7.2 |
| Mints | 26 | No | \$52 | \$10.4 |
| Fudge | 8 | Yes | \$32 | \$6.4 |

Recap Practice / Classroom Coding

- **PHP_practice05/section_b/c05/case-and-character-count.php**

Write PHP Code to convert case in lowercase, uppercase, count number of characters and word count

- **PHP_practice05/section_b/c05/array-functions.php**

Write PHP Code to create

- array of greetings (i.e. “Hi”, “Howdy”, “Hello”, “Hola”, “Cia”, “Moi”, “Namaste”, “Welcome”)
- then display random greeting
- find array of best sellers of items (i.e. “notebook”, “pencil”, “ink”)
- count items and display top items
- create an array holding customer details (e.g firstname, lastname, email).
- and if you have customer first name add it to greeting

Recap Practice / Classroom Coding

- **PHP_practice05/section_b/c05/array-updating-functions.php**

Write PHP Code to

- create array of items (items could be e.g. “notebook”, “pencil”, “eraser”) being ordered
- add a new item (i.e. “scissors”) to start of an array
- remove the last item from array

- **PHP_practice05/section_b/c05/files.php**

Write PHP Code to show file informations of “logo.png”
e.g.

- file name
- file size
- MIME type and
- Folder
- If no such file exists display “There is no such file”

Note: You can use PHP in-built function e.g. pathinfo, filesize, mime_content_type

Getting Data from Browsers

The PHP superglobals **\$_GET** and **\$_POST** are used to collect form-data.

PHP - A Simple Form

The example below displays a simple HTML form with two input fields and a submit button:

Example

```
<html>
<body>

<form action="welcome.php" method="post">
Name: <input type="text" name="name"><br>
E-mail: <input type="text" name="email"><br>
<input type="submit">
</form>

</body>
</html>
```

1. Go and open form.php in PHP_practice05/section_b/c05/form.php
2. Insert some data e.g. your name and email address
3. See the results.

Getting Data from Browsers

The same result could also be achieved using the HTTP GET method:

```
<!DOCTYPE HTML>
<html>
<body>

<form action="welcome_get.php" method="get">
Name: <input type="text" name="name"><br>
E-mail: <input type="text" name="email"><br>
<input type="submit">
</form>

</body>
</html>
```

1. Go and open form_get.php in http://localhost:8005/PHP_practice05/section_b/c05/form_get.php
2. Insert some data e.g. your name and email address
3. See the results.

GET vs. POST

Both GET and POST create an array (e.g. `array(key1 => value1, key2 => value2, key3 => value3, ...)`). This array holds key/value pairs, where keys are the names of the form controls and values are the input data from the user.

Both GET and POST are treated as `$_GET` and `$_POST`. These are superglobals, which means that they are always accessible, regardless of scope - and you can access them from any function, class or file without having to do anything special.

`$_GET` is an array of variables passed to the current script via the URL parameters.

`$_POST` is an array of variables passed to the current script via the HTTP POST method.

When to use GET?

Information sent from a form with the GET method is **visible to everyone** (all variable names and values are displayed in the URL).

```
$city = isset($_GET['city']) ? $_GET['city'] : ' ';
```

```
$city = $_GET['city'] ?? ' ' ;    // if it does not exist: store blank string
```

GET also has limits on the amount of information to send. The limitation is about 2000 characters. However, because the variables are displayed in the URL, it is possible to bookmark the page.

GET may be used for sending non-sensitive data.

Note: NEVER use GET for sending passwords or other sensitive information!

<https://www.php.net/manual/en/function.isset.php>

When to use POST?

Information sent from a form with the POST method is **invisible to others** (all names/values are embedded within the body of the HTTP request) and has **no limits** on the amount of information to send.

```
$email = $_POST ['email'];  
$age = $_POST['age'] ?? false;
```

Moreover POST supports advanced functionality such as support for multi-part binary input while uploading files to server.

However, because the variables are not displayed in the URL, it is not possible to bookmark the page.

Note: Developers should prefer POST for sending form data.

Chapter Summary

- Data sent via query strings and forms is added to the `$_GET` and `$_POST` superglobals arrays, which store all the data they receive as strings.
- If a value may be missing from a super global array use the `isset()` function to check if it is present or supply a default with the null-coalescing operator `??`.
- Before processing data, validate it. Check required data was supplied and that it is in the right format.
- Before showing user data, sanitize it to prevent XSS attacks. Replace reserved characters with entities.
- Use validation filters or sanitization filters to filter and validate the data types.

Recap Practice / Classroom Coding

- **PHP_practice01/6.php:**
 - Make a form that submits one value to POST super global

- **PHP_practice06/section_b/c06/get-1.php**

Run PHP Code and visualise how the browser URL parameter changes.
Read the steps inside the code.

For example

‘Helsinki’ has address ‘Kaivokatu 1, 00100 Helsinki’,
‘London’ has for example ‘48 Store Street’, WC1E, 7BS and
‘Sydney’ has address e.g. ‘1243 7th Street, 10212’

Note: For more info and instructions, check the practice folder with steps

Recap Practice / Classroom Coding

- **PHP_practice06/section_b/c06/get-2.php**

Write PHP Code

- to validate the query string data from previous example
- and use validation to check if the query string holds a valid location.
- For example, if query string contains a city, it should store in a variable and if not, the variable e.g. \$city can hold a blank string.

Hint: Use the \$_GET

You may reference it with <a href="get-1.php?city=<?= \$key ?>"><?= \$key ?>

- **OPTIONAL: PHP_practice06/section_b/c06/get-3.php**

Write PHP Code based on previous one to validate and check if query string holds a valid location. For example, in the URL insert different city which should redirect to "page-not-found.php"

Recap Practice / Classroom Coding

- **PHP_practice06/section_b/c06/collecting-form-data.php**

Fix this PHP form to collect form data and display in the page below.

- **PHP_practice06/section_b/c06/check-for-http-post.php**

Write a PHP code for checking a form has been submitted.

You can use `$_SERVER` superglobal array for specific `REQUEST_METHOD` which stores the HTTP method used to request the page.

Whatever user inserts in input box should be displayed e.g.

"You searched for ..." (replace ... with term user searched for)

- **PHP_practice06/section_b/c06/check-for-http-get.php**

Write a PHP code for checking a form has been submitted.

You can use `$_SERVER` superglobal array for specific `REQUEST_METHOD` which stores the HTTP method used to request the page.

Whatever user inserts in input box should be displayed e.g.

"You searched for ..." (replace ... with term user searched for)

Recap Practice / Classroom Coding

- **PHP_practice06/section_b/c06/validate-number-range.php**

Write PHP code to check if the given number is valid.

- The age is valid if this is between 16-65 years old.
- Display the message “Age is valid” otherwise display “You must be 16-65”

- **PHP_practice06/section_b/c06/validate-password.php**

Write PHP code to validate password and also check password strength. Following conditions should be checked

- Password length must contain ≥ 8 characters
- Check if it contains combination of uppercase characters, lowercase characters
- And numbers

Display “Password is valid” once above conditions are fulfilled else display “Password not strong enough”