Programming PHP

Santosh Kalwar, 23.03.2022

PHP Topics

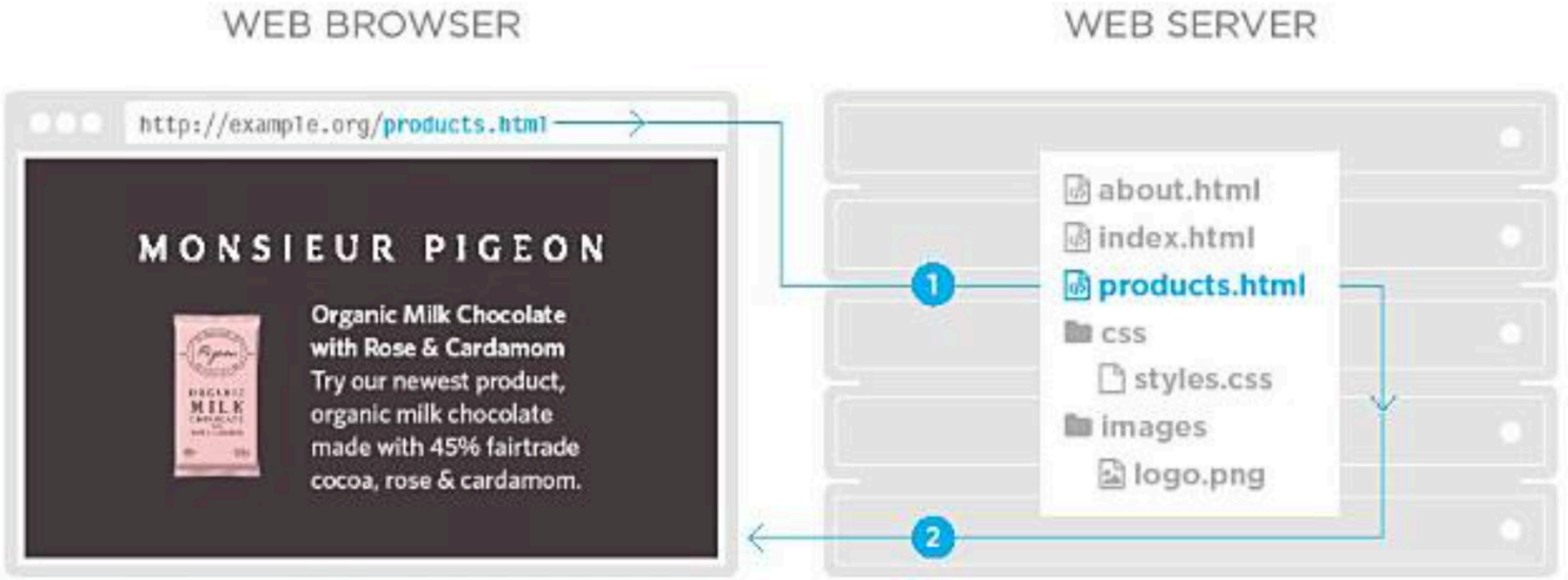
General Discussions

PHP
- What is PHP?
- PHP Developer Roadmap
  - https://github.com/thecodeholic/php-developer-roadmap
- VS Code Extentions
- PHP Installation

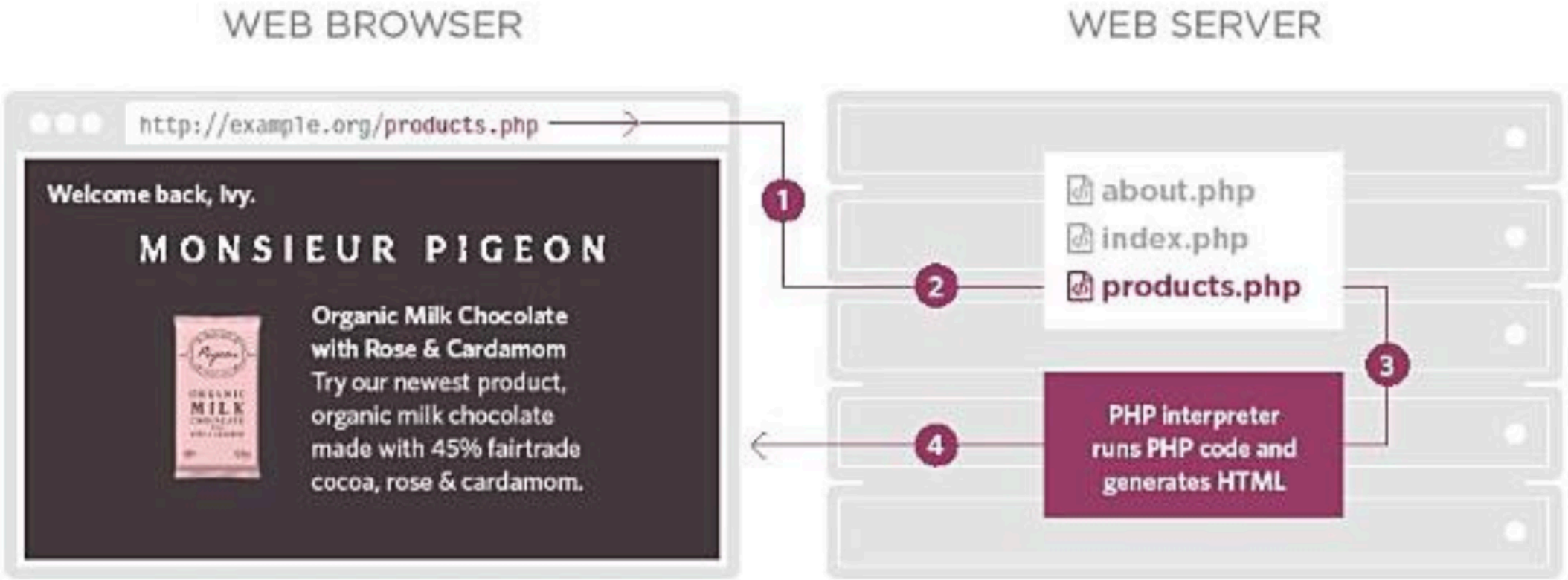
PHP Basic Operations
- Variables and Types
- Numbers
- Strings
- Arrays and Operators

- Practice / Classroom coding

Static Vs Dynamic sites



Static Vs Dynamic sites



What is PHP?

Personal Home Page
- but nowadays called Hypertext Preprocessor

Multi-purpose

Large community of developers

>75% of world websites use PHP

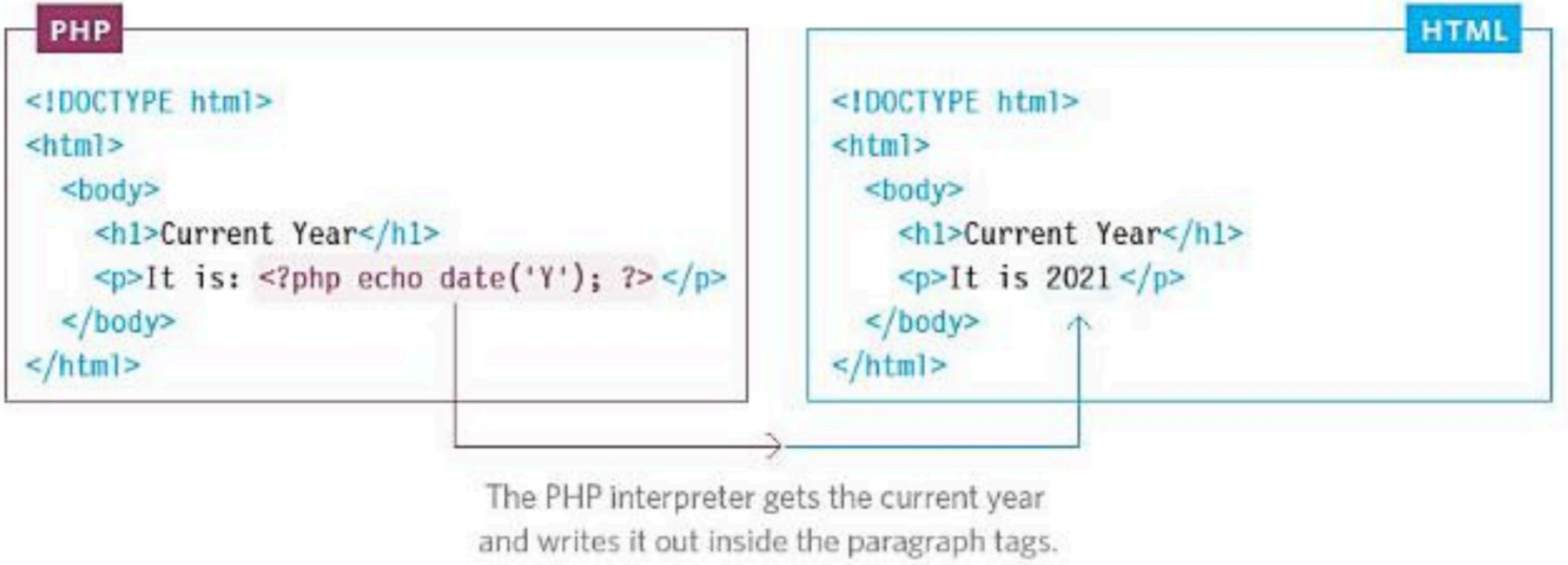
Popular CMSs and Frameworks are build using PHP
- Wordpress
- Symfony (we will learn this after PHP lessons)
- Laravel

```php
<?php
    echo 'Hello, PHP!';
?>
```

PHP Basic Syntax

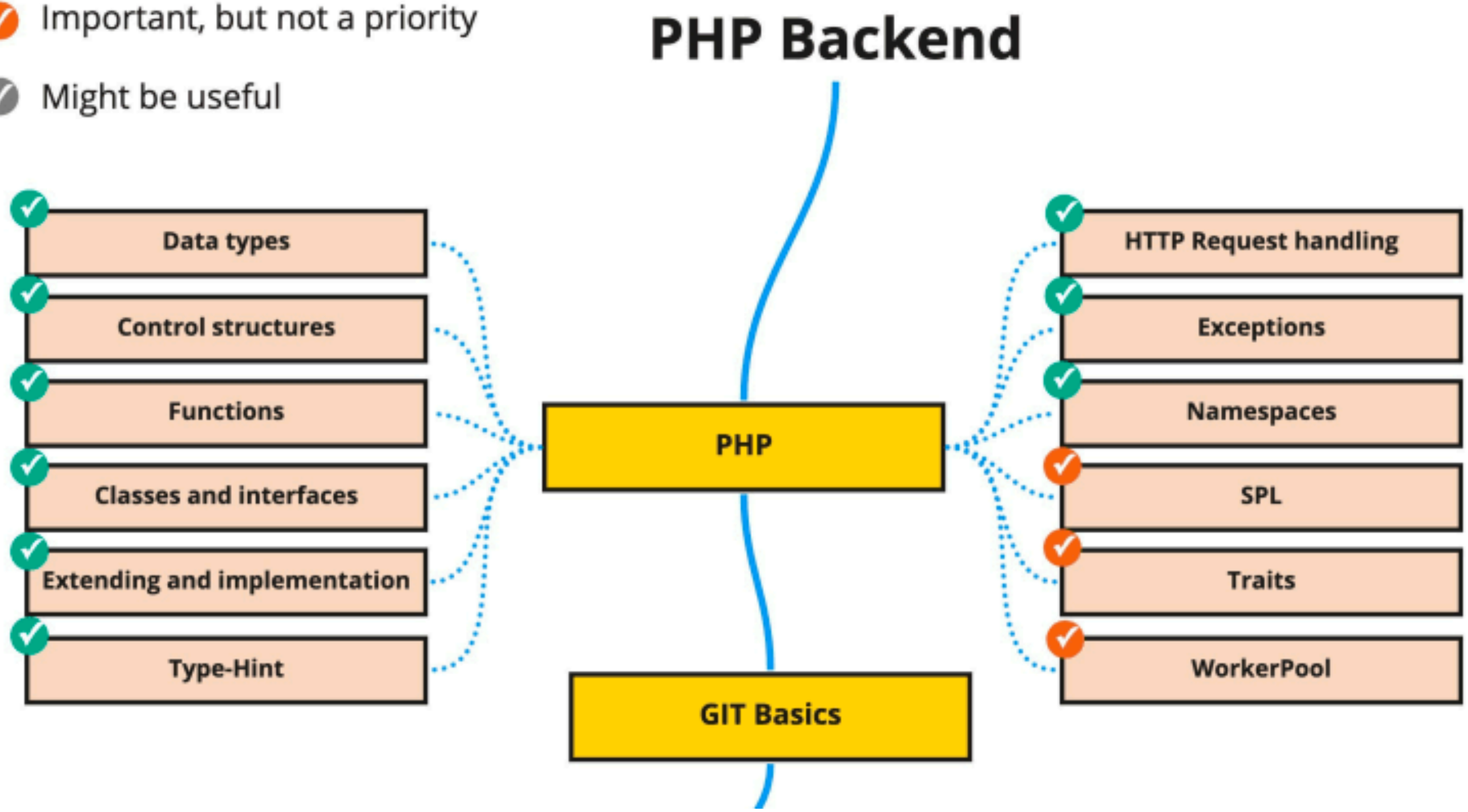
- Files with PHP content are suffixed with .php extension
- PHP scripts are wrapped within <?php ….. ?> blocks
- Content of PHP files are often a mixture of HTML and PHP scripting code
- PHP statements end with semicolon **;**

What is PHP page?

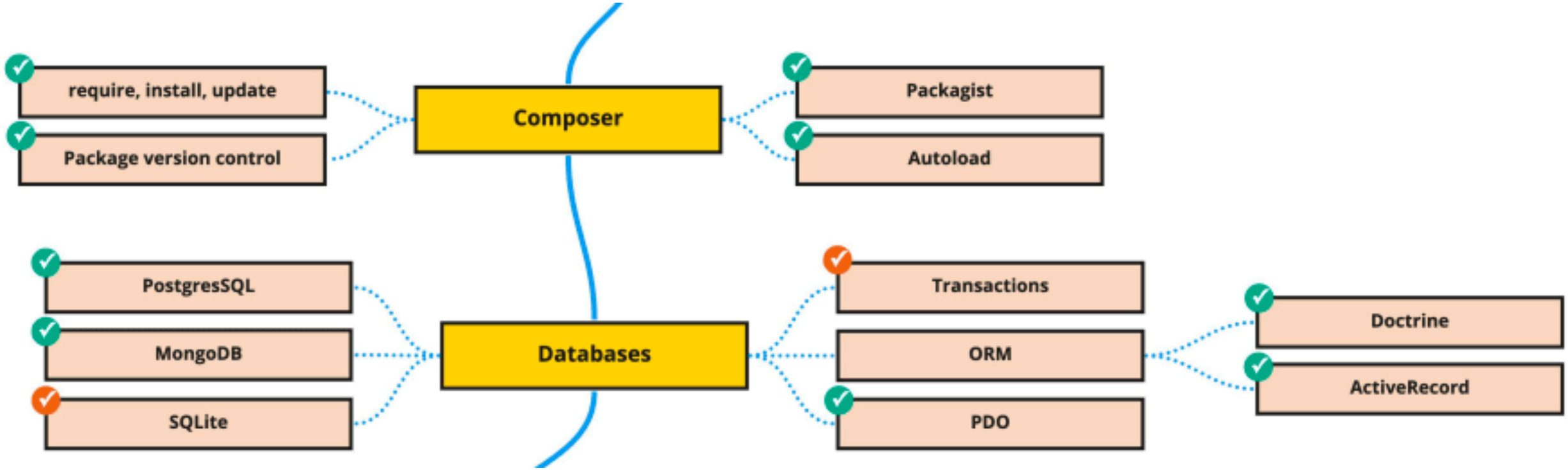


PHP Developer Roadmap

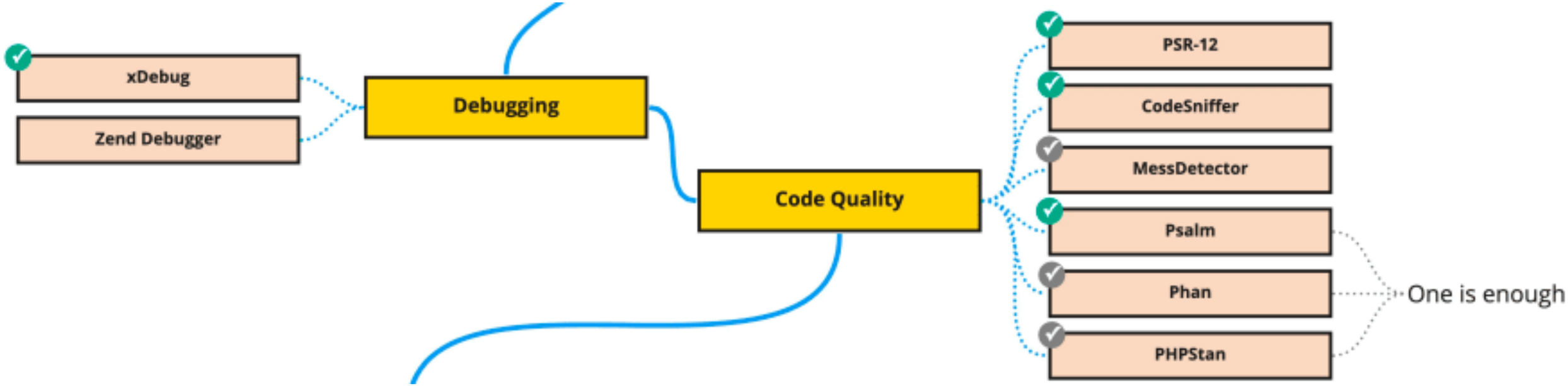
- Should be learned first of all
- Important, but not a priority
- Might be useful



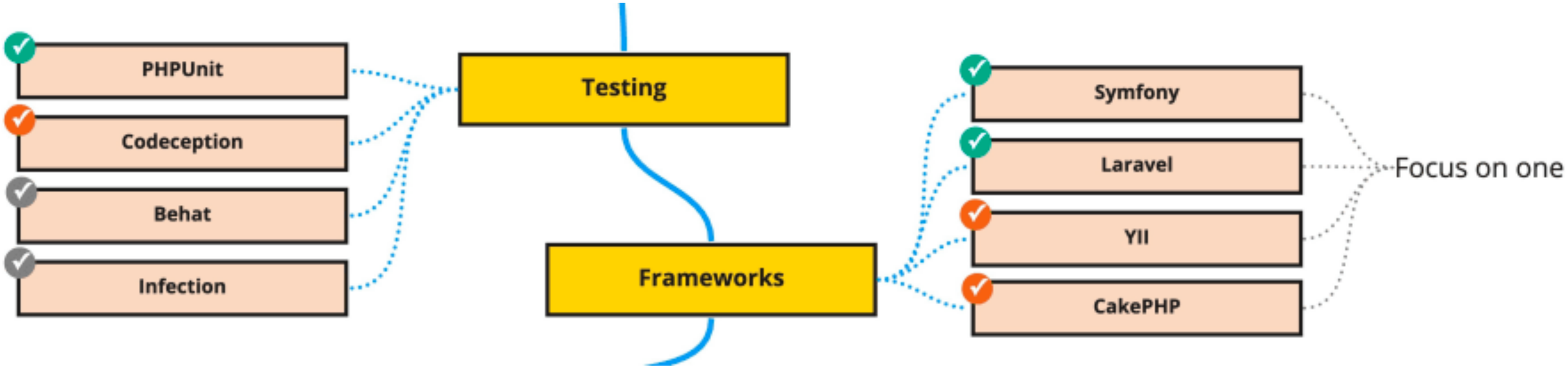
PHP Developer Roadmap



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?itemName=felixfbecker.php-intellisense

- PHP Intelephense
  https://marketplace.visualstudio.com/items?itemName=bmewburn.vscode-intelephense-client

- phpfmt - PHP formatter
  https://marketplace.visualstudio.com/items?itemName=kokororin.vscode-phpfmt

- PHP Debug
  https://marketplace.visualstudio.com/items?itemName=xdebug.php-debug

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?

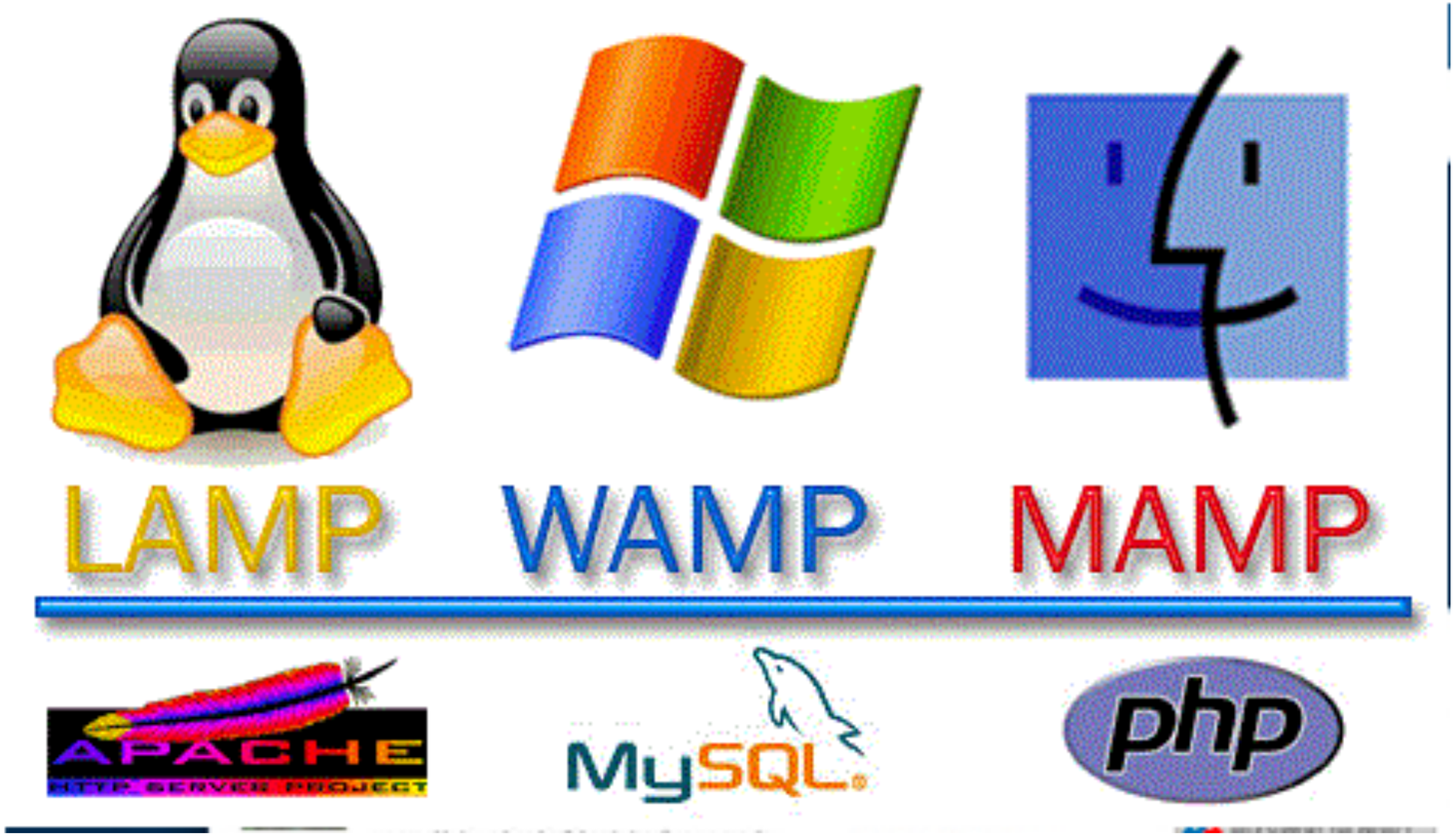
- Linux
- Apache
- MySQL
- PHP

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/

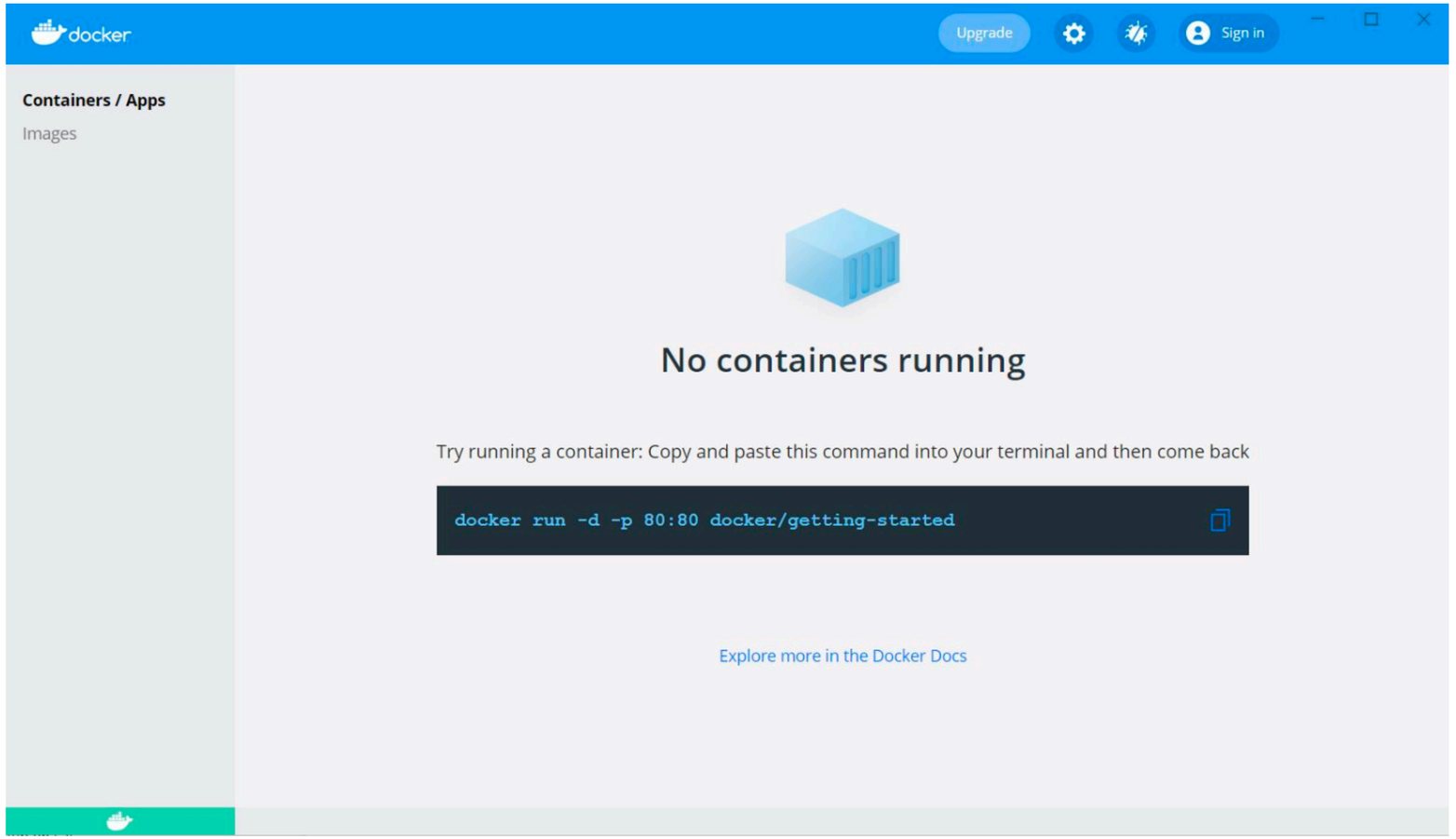


PHP stack Installation

Mordern way to install PHP

- 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. Run docker-compose up

3. Check Docker Desktop

4. Open the Apache application in http://localhost:8005

5. phpMyadmin in http://localhost:9080

And mySQL should show it is connected

PHP Intro

- PHP Test Page

- Echo Statement

- Comments

PHP Variables

- PHP Variables must always start with $ sign e.g.
  - $name = ‘James’; $price = 5;
  - echo $name; echo $price;

- PHP distinguishes between different types of values you can store in a variable (such as text and numbers) and these are know as *data types*
  - A piece of text is called a **string**
  - A whole number is called an **integer**
  - A fractional number is called a **float**
  - A value of true and false is called **boolean**
  - A series of related names and values can be stored in an **array**

How to name PHP variables

1

Begin with a dollar sign ($).

✓ `$greeting`

✗ `greeting`

2

Follow it with a letter or an underscore (not a number).

✓ `$greeting`

✗ `$2_greeting`

3

Then use any combination of letters A-z (uppercase and lowercase), numbers, and underscores. (No dashes or periods are allowed.)

✓ `$greeting_2`

✗ `$greeting-2`

✗ `$greeting.2`

Note: `$this` has special meaning. Do not use it as a variable name.

✗ `$this`

PHP Basic Data Types (Scalar - String, Number, Boolean)

STRING DATA TYPE

Programmers call a piece of text a **string**. The `string` data type can consist of letters, numbers and other characters, but they are used to represent text.

```
$name = 'Ivy';
```

Strings are always enclosed in either single or double quotes. The opening quote must match the closing quote.

- ✓ `$name = 'Ivy';`
- ✓ `$name = "Ivy";`
- ✗ `$name = "Ivy';`
- ✗ `$name = 'Ivy";`

NUMERIC DATA TYPES

Numeric data types let you perform mathematical operations with the values that they hold, such as addition or multiplication.

```
$price = 5;
```

Numbers are not written in quotes. If you place numbers in quotes, they can be treated as strings rather than numbers.

PHP has two numeric data types:

`int` represents integers, which are whole numbers (e.g., 275)

`float` holds floating point numbers, which represent fractions (e.g., 2.75)

BOOLEAN DATA TYPE

The `boolean` data type can only have one of two values: `true` or `false`. These values are commonly in most programming languages.

```
$logged_in = true;
```

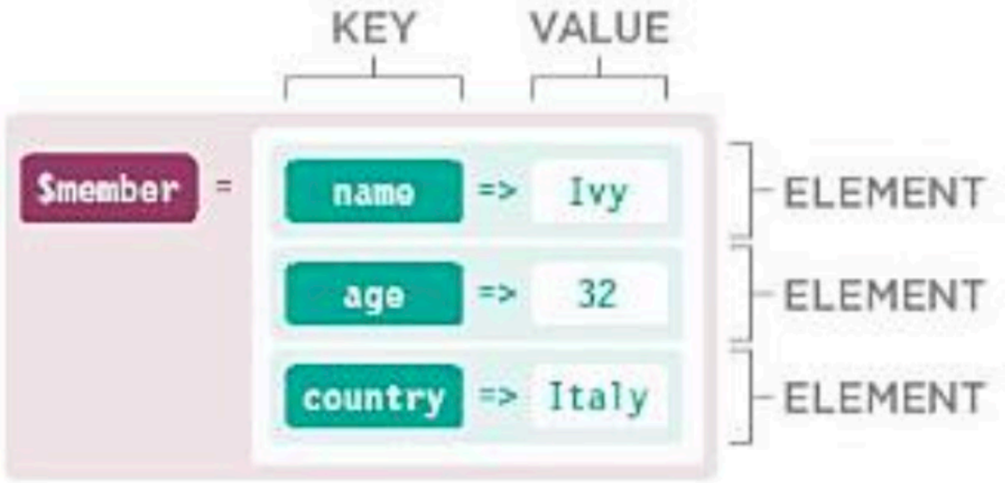
`true` and `false` should be written in lowercase, and are not placed in quotes. At first, boolean values might seem abstract, but many things can be represented using true or false, such as:

- Is a visitor logged in?
- Have they agreed to terms and conditions?
- Does a product qualify for free shipping?

PHP Data Types - Arrays

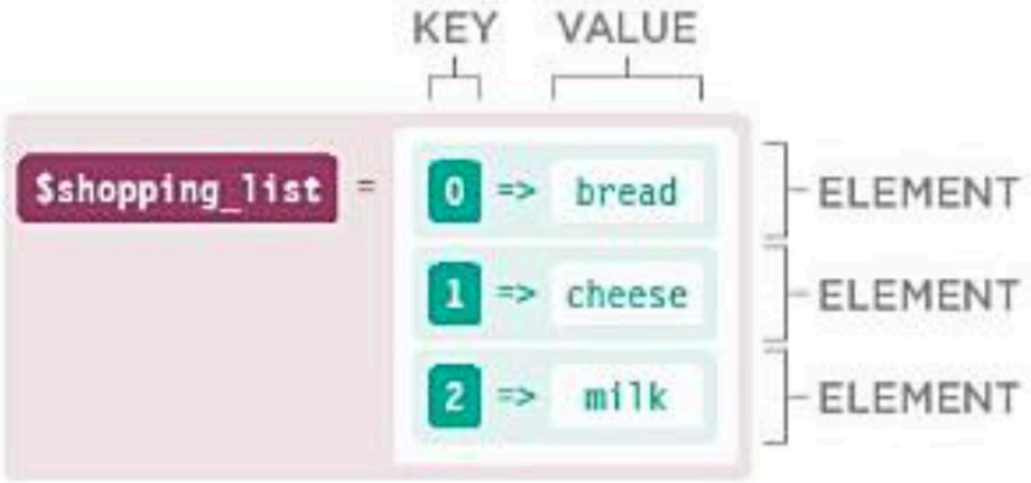
ASSOCIATIVE ARRAY

The array below is designed to hold data that represents a member of a website. Each time the array is used, the names used in the keys (that describe the data stored in each element of the array) will remain the same.

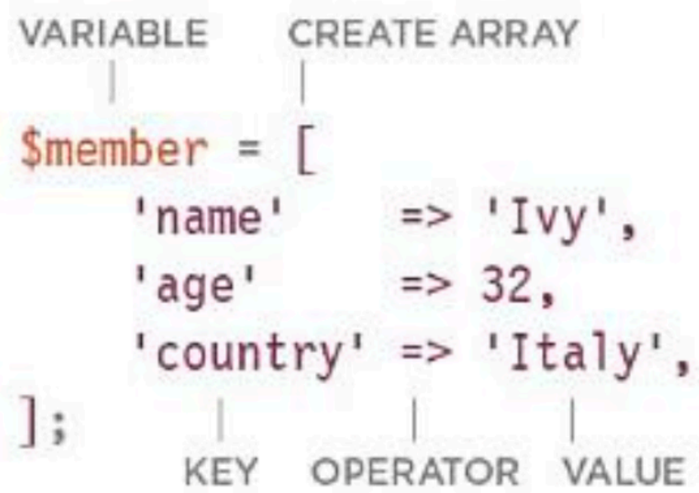


INDEXED ARRAY

The array below is designed to hold a shopping list. Lists like this can hold a different number of elements each time they are used. The key does not use a name to describe each item in the list; it uses an index number (which is an integer and starts at 0).



PHP Data Types - Associative Arrays

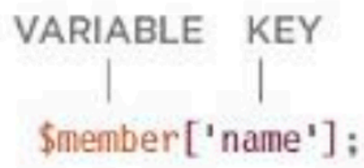


An associative array can also be created using the syntax shown below, with the word array followed by parentheses (instead of square brackets).

```
$member = array(
  'name'    => 'Ivy',
  'age'     => 32,
  'country' => 'Italy',
);
```

To access an element in an associative array, use:

- The name of the variable holding the array
- Followed by square brackets and quote marks
- The key for the element you want to retrieve



PHP Data Types - Indexed Arrays

Above, bread would be assigned the index number 0, cheese 1, and milk 2. Index numbers are often used to indicate the order of the items listed in the array.

An indexed array can also be created using the syntax shown below, with the word `array` followed by parentheses (instead of square brackets).

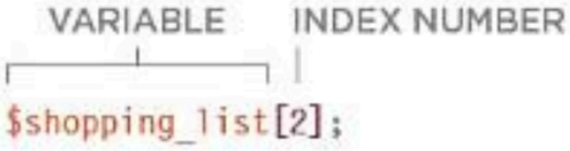
```
$shopping_list = array('bread',
                       'cheese',
                       'milk');
```

Each value that is being added to the array can be on the same line or on a new line (as shown above).

To access the items in an indexed array, use:

- The name of the variable holding the array
- Followed by square brackets (no quote marks)
- The index number of the item you want to access (in the square brackets)

The code below gets the third item in the array, so in this example it would get the value `milk`.



PHP Data Types - Updating Arrays

To update a value stored in an associative array, use:

- The name of the variable holding the array
- Followed by square brackets
- Then the key name in quotes
- An assignment operator
- The new value it should hold

```
$member['name'] = 'Tom';
```

VARIABLE KEY NEW VALUE

To add a new item to an associative array, do exactly the same as above, but use a new key name (not one that has already been used in the array).

The quotes go around the key name when it is a string because quotes indicate a string data type.

To update a value stored in an indexed array, use:

- The name of the variable holding the array
- Followed by square brackets
- The index number (not in quotes)
- An assignment operator
- The new value it should hold

```
$shopping_list[2] = 'butter';
```

VARIABLE INDEX NUMBER NEW VALUE

You will see how to add items to indexed arrays on p220. The process is different because you can specify the position of the new item in the array.

Quotes are not placed around index numbers because number data types are not quoted.

PHP Data Types - Storing Arrays in an Array

```php
$members = [
    ['name' => 'Ivy',  'age' => 32, 'country' => 'UK',],
    ['name' => 'Emi',  'age' => 24, 'country' => 'Japan',],
    ['name' => 'Luke', 'age' => 47, 'country' => 'USA',],
];
```

To get the array that holds data about Emi, use the:

- Name of the variable that holds the indexed array
- Index number of the element you want to access in square brackets (remembering that indexed arrays start at 0, and that numbers are not placed in quotes).

```php
$members[1];
```

To get the age of Luke, use the:

- Name of the variable that holds the indexed array
- Index number of the element that holds the array of data about Luke, in square brackets
- Key of the element you want to access in the array about Luke in a second set of square brackets (because the key is a string, put it in quotes)

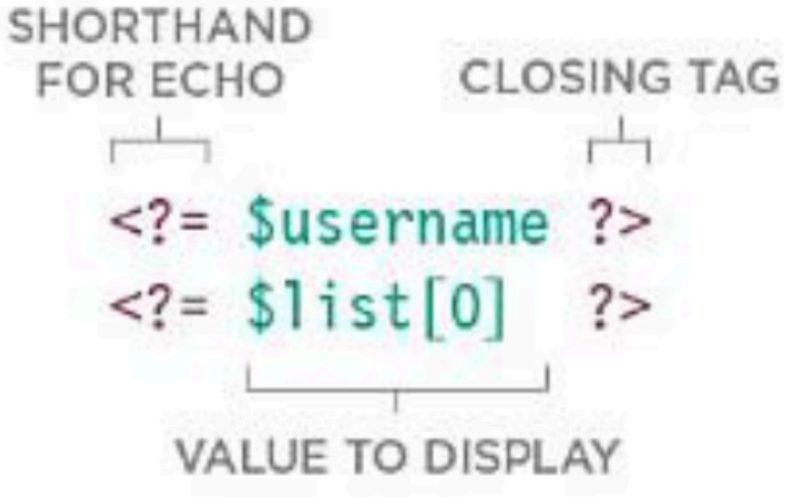
```php
$members[2]['age'];
```

Shorthand for ECHO

Instead of writing `<?php echo $name; ?>`
you can use the shorthand `<?= $name ?>`
this is the only time you do not need to
use the full opening `<?php` delimiter.

You do *not* need:

- The letters php in the opening tag
- The echo command
- A semicolon before the closing tag



String Operators

CONCATENATION OPERATOR

The concatenation operator is a period symbol. It joins the value in one string to the value in another. In the example below, the variable called `$name` would hold the string `'Ivy Stone'`:

```
$forename = 'Ivy';
$surname  = 'Stone';
$name     = $forename . ' ' . $surname;
```

Note that a space is added between the `$forename` and `$surname` variables; if the space was not there, the `$name` variable would hold the value `IvyStone`.

You can concatenate as many strings as you want in one statement providing the concatenation operator is used between every string.

CONCATENATING ASSIGNMENT OPERATOR

If you want to append some text to an existing variable, you can use the concatenating assignment operator. You can think of it as shorthand for creating an updated string:

```
$greeting = 'Hello ';
$greeting .= 'Ivy';
```

Here, the string `'Hello '` is stored in a variable called `$greeting`. On the next line, the concatenating assignment operator adds the string `'Ivy'` to the end of the value held by the variable called `$greeting`.

Now, the `$greeting` variable holds the value `'Hello Ivy'`. As you can see, it uses one less line of code than the example on the left hand side.

Comparision Operators

==

IS EQUAL TO

This operator compares two values to see if they are the same.

`'Hello' == 'Hello'` results in `true` because they *are* the same string.
`'Hello' == 'Goodbye'` results in `false` because they are *not* the same string.

!= OR <>

IS NOT EQUAL TO

These operators compare two values to see if they are *not* the same.

`'Hello' != 'Hello'` results in `false` because they *are* the same string.
`'Hello' != 'Goodbye'` results in `true` because they are *not* the same string.

Comparision Operators

IS IDENTICAL TO

This operator compares two values to check that both the value and data type are the same.

`'3' === 3` results in `false`
because they are *not* the same data type.
`'3' === '3'` results in `true`
because they *are* the same data type and value.

IS NOT IDENTICAL TO

This operator compares two values to check that both the value and the data type are *not* the same.

`3.0 !== 3` results in `true`
because they are *not* the same data type.
`3.0 !== 3.0` results in `false`
because they *are* the same data type and value.

Logical Operators

&&

LOGICAL AND

This operator tests more than one condition.

```
((2 < 5) && (3 >= 2))
```
This results in a value of `true`

If both expressions evaluate to `true`, the expression returns `true`. If one of these results in `false`, the expression results in `false`.

```
true  && true  results in true
true  && false results in false
false && true  results in false
false && false results in false
```

You can use the word **and** instead of two ampersands.

||

LOGICAL OR

This operator tests at least one condition.

```
((2 < 5) || (2 < 1))
```
This results in a value of `true`

If either expression evaluates to `true`, the expression returns `true`. If both expressions result in `false`, the expression results in `false`.

```
true  || true  results in true
true  || false results in true
false || true  results in true
false || false results in false
```

You can use the word **or** instead of two pipestem characters.

!

LOGICAL NOT

This operator takes a single boolean value and negates it.

```
!(2 < 1)
```
This results in a value of `true`

The `!` negates an expression. If it is `false` (without the `!` before it), it results in `true`. If the statement is `true`, it results in `false`.

```
!true  results in false
!false results in true
```

You *cannot* use the word **not** instead of the exclamation mark.

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_practice02/section_a/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

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 multidimentional 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.

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 contcatenate 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/comparision-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 imaginery number to do the comparision.