

Web Expert 2 – Oefeningen PHP

Oefening 1

Maak een nieuwe map “oefeningen_php” aan en open deze map in jouw editor.

1. composer

Maak in jouw editor, in jouw projectmap, een composer.json bestand aan en zorg ervoor dat je het volgende pakket toevoegt aan je project:

<https://packagist.org/packages/guzzlehttp/guzzle>.

Guzzle is een populaire library om op een eenvoudige manier HTTP requests uit te voeren.

Wanneer je dit pakket hebt toegevoegd en composer hebt geïntialiseerd, zou je de volgende structuur in jouw projectmap moeten hebben:

- vendor
 - ...
- composer.json
- composer.lock

2. .gitignore

Zorg ook meteen voor een .gitignore bestand zodat je, indien je dit wenst, je oefening naar een git repository kan pushen. Momenteel moet er enkel een referentie naar de vendor map in het .gitignore bestand.

Wil je meteen een volledige PHP .gitignore inladen, dan kan je hier een voorbeeld terug vinden: <https://gist.github.com/Yousha/a5514afd6cda8afba800f5af9f7115b4>

3. De main app

Maak in de root van je projectmap het bestand app.php aan. Dit gaat het bestand zijn dat we gaan gebruiken om gegevens terug te geven naar de console.

In dit bestand gaan we aangeven dat we de libraries in de vendor map gaan gebruiken. Dit doe je door de volgende lijn toe te voegen: `require_once("../vendor/autoload.php")`

4. De src map

Maak in de root van je project een map src aan. Hierin gaat alle code van het project komen.

In PhpStorm kan je deze map aanduiden als sources root, waarmee je aangeeft dat al jouw broncode zich in deze map bevindt. Dit doe je op de volgende manier:

1. *Rechtermuisknop op de src map*
2. *Mark Directory as*
3. *Mark as Sources root*

Je zal merken dat de map nu een andere (blauwe) kleur krijgt.

De Activity klasse

Maak in de src map een nieuwe klasse Activity aan. Deze klasse heeft:

- Een private variabele `$client` van het type `Client` (`GuzzleHttp`). Om het type te gebruiken dien je eerst aan te geven dat je deze library gaat gebruiken. Dit doe je door bovenaan in je bestand, boven de klasse gebruik te maken van een use statement: `use GuzzleHttp\Client`. Hierdoor gaat er gebruik gemaakt worden van de library die zich in de vendor map bevindt.
- Een constructor zonder argumenten. Deze constructor initialiseert de klassevariabele `$client`.
- Een publieke functie `getActivity` met een string als return type.
 - o De functie maakt gebruik van de request functie van het `$client` object (zie documentatie `GuzzleHttp`).

Het is de bedoeling dat je een GET request uitvoert naar <https://www.boredapi.com/api/activity> en dat de methode, de activiteit terug geeft.

- Zet jouw request in de try van een try – catch blok.
- Indien in de data van je object een activiteit zit, geef je de activiteit terug. Zit er geen activiteit in, dan geef je de string “No activity available” terug.
- Kom je in de catch uit, geef dan de string “Something went wrong: exception” + de message van de exception terug.

Test jouw werk nu door in app.php een nieuwe instantie van de Activity klasse aan te maken en gebruik te maken van de getActivity methode op deze instantie om een activiteit te printen naar de console.

Extra

Gebruik het package `tdtrung17693/php-chalk` om het resultaat van de methode op een andere manier weer te geven.

- Bij succes, geef je de activiteit terug in het groen.
 - Bij het niet vinden van een activiteit geef je het resultaat terug in het blauw en onderlijnd.
 - Bij een error geef je het resultaat terug in het rood en in het vet.
-

Oefening 2

Maak in de src map een nieuwe klasse Triangle aan. Deze klasse heeft:

- Een private variabele `$lines` van het type `int`.
- Een private variabele `$input` van het type `string`.
- Een constructor met 2 argumenten:
 - `$lines` van het type `int`.
 - `$input` van het type `string`.

- In de constructor:
 - Wordt de klassevariabele \$lines geïnitieerd op basis van de private methode setLines.
 - Deze methode heeft een argument \$lines van het type int.
 - Indien \$lines null is of kleiner of gelijk aan 1 is, dan wordt de klassevariabele \$lines op 5 gezet, zo niet dan krijgt deze de waarde van het argument \$lines.
 - Wordt de klassevariabele \$input geïnitieerd op basis van de variabele \$input.
 - Indien de string \$input leeg is of de lengte van de string groter is dan 1, dan wordt “#” gebruikt als de standaardwaarde voor \$input.
- Een publieke methode rightAngledTriangle met een string als return type. Deze methode geeft een rechthoekige driehoek terug en maakt hierbij gebruik van de klassevariabelen \$lines en \$input.

Bv: \$lines = 7 - \$input = #

```
#
##
###
####
#####
#####
#####
#####
```

- Een publieke methode `equilateralTriangle` met een string als return type.

Deze methode geeft een gelijkzijdige driehoek terug en maakt hierbij gebruik van de klassevariabelen `$lines` en `$input`.

Bv: `$lines = 7 - $input = #`

```

#
###
#####
#####
#####
#####
#####
#####
#####

```

Test jouw werk door de methodes aan te roepen in `app.php` en je resultaat te printen naar de console.

Extra

- Maak een publieke methode `inverseRightAngledTriangle` met een string als return type. Deze methode geeft een inverse rechthoekige driehoek terug en maakt hierbij gebruik van de klassevariabelen `$lines` en `$input`.

Bv: `$lines = 7 - $input = #`

```

#
##
###
####
#####
#####
#####
#####

```

- Maak een functie `intervalEquilateralTriangle` met een string als return type en een string `$interval` als argument. Zorg ervoor dat in de driehoek op de eerste positie en vervolgens op elk interval een "@" afgedrukt wordt in plaats van de input string.

Bv: `$lines = 7 - $input = # - $interval=5`

```

      @
    ###
  #@###
#@#####@
#####@#####
@#####@#####@
#####@#####@###

```

Oefening 3

Maak in de src map een nieuwe klasse Row aan. Deze klasse heeft:

1. Een publieke functie `createRow`. De functie geeft een array terug met de gehele getallen gelegen tussen min en max (beide inclusief). Voorzie de argumenten van een type en definieer het return type.
Indien max kleiner is dan min wordt er een lege array terug gegeven.
2. Een publieke functie `createRowStep` met een optioneel argument `step`, dit argument heeft de standaard waarde 1. Stap bevat de tussenstap tussen 2 waarden in de teruggegeven array.
 - `createRowStep(1,5,2)` geeft `[1,3,5]` terug
 - `createRowStep(5,1,-2)` geeft `[5,3,1]` terug.

Test jouw werk door de functies aan te roepen in `app.php` en het resultaat te printen naar de console.

Extra:

Roep een rij met getallen op via de functie `createRow`. Gebruik vervolgens de PHP functie `array_map` om een nieuwe array aan te maken waarbij elk getal van de eerste rij vermenigvuldigd wordt met 3.
