5.1 CRUD - Read

Wat moet goed je weten voor deze les:

- Hoe werken arrays.
- Hoe werken assiociative arrays.
- Hoe werken tables.

Wat ga je leren deze les?

- Je gaat werken met PHP objecten
- Je gaat leren werken met methods en properties van een object
- Je gaat inheritance toepassen

CRUD

CRUD is een groep functionaliteiten die je voor je examen moet kunnen

CRUD staat voor Create, Read, Update en Delete. We gaan een pagina maken waar je deze functies kunt uitvoeren. We gaan dit testen op de airlines table in de database flights. Dat betekent dat webeginnen met het maken van een overzicht van alle flights (R), Dit is de R(ead) van de CRUD.

Voordat we beginnen moeten we controleren of we alles uit de vorige lessen goed hebben.

Controleer of je de volgende structuur hebt:

```
/includes/db.php
crud.php
view.html
```

In /incudes/db.php heb je een db class gemaakt. Deze zorgt ervoor dat er een verbinding wordt gemaakt met de SQL database.

crud.php en view.html zijn nu nog leeg.

Je hebt een tabel `airlines` gemaakt in een nieuwe databse `flights`.

Remember: *een object is een variabele in een object!*

In de `crud.php` file maken we een class airlines. In de __construct() maken we een DB object. Het airlines object heeft nu property die verwijst naar een DB object.

> Vraag 1: Hoe heet de property (variabele) in crud.php die het object DB bevat?

```
<?php

include_once "includes/db.php";

class airlines .... {
    public function getAllAirlines() {
        return $this->executeSQL("select * from airlines");
    }
}
?>
```

Bekijk bovensstaande code en let vooral op de regel waar de class airlines wordt gemaakt. Zie je de puntjes? Daar moet nog wat komen te staan. De class airlines maken we namelijk een 'kind' van de class 'DB'. Hierbij erft het kind, alle eigenschappen van de moeder.

Weet je nog hoe dat zat met classes en ínherentence'?

Als je het nog eens wilt bekijken dan kun je het volgende video filmpje bekijken:

Inheritance in PHP eplained

Vul nu de juiste code in op de plaats van de puntjes zodat de airlines class alle eigenschappen van de DB class erft.

> Vraag 2: welke statement gebruik je in PHP om een child class te maken?

Remember: *een method is een functie in een object!*

Nu gaan we onze code testen en dat doen we door vanuit view de code aan te roepen.

Zet dan de volgende code in view.php

```
<?php
include "crud.php";

$airlines = new Airlines;
echo $airlines->getAllAirlines();
```

```
?>
```

Bekijk wat er gebeurt. Welk getal wordt er afgedrukt en waarom?

Rows

In de class DB staat een method getRow. Roep deze aan vanuit de view.php.

`$row=$airlines->getRow();`

Druk de regel af,wat zie je? Waarom wertk dat niet? Gebruik de PHP functie `var_dump($row)` om de regel af te drukken. Zie je nu wat er wordt afgedrukt?

We gaan nu in driestappen alle regels netjes afdrukken.

Stap 1, we gaan in een loop alle regels afdruken met var_dump

Dat ziet er ongeveel zo uit:

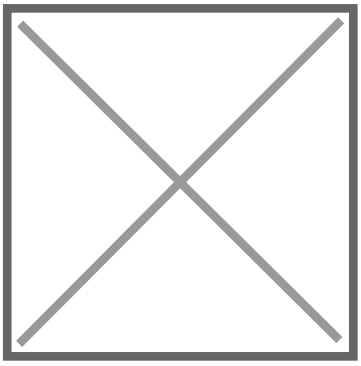
```
array(2) { ["code"]=> string(2) "AA" ["airline"]=> string(22) "American Airlines Inc." }
array(2) { ["code"]=> string(2) "AS" ["airline"]=> string(20) "Alaska Airlines Inc." }
array(2) { ["code"]=> string(2) "B6" ["airline"]=> string(15) "JetBlue Airways" }
array(2) { ["code"]=> string(2) "DL" ["airline"]=> string(20) "Delta Air Lines Inc." }
....
....
```

Om de regels één voor één op te halen moet je net zo lang de method getRow() aanroepen todat je ..... totdat je wat? Kijk in de code voor het antwoord.

Stap 2, druk de regels af als twee losse variabele

```
AA American Airlines Inc.
AS Alaska Airlines Inc.
B6 JetBlue Airways
DL Delta Air Lines Inc.
....
....
```

Stap 3, zet nu alles netjes in een table:



Eventueel kun je de table met CSS nog wat mooier maken. Je kunt nog een nette kopregel toevoegen.

Op zich zijn het opfraien van de GUI geen keiharde voor de back-end programmeur, maar als je met een paar eenvoudige stappen de output wat kunt 'opleuken' dat toont dat toch beter.

De R van CRUD is nu klaar.

Revision #9
Created 2019-09-16 20:17:40 UTC by Admin
Updated 2020-10-05 11:17:56 UTC by Max