

1.2 Associative Arrays

Je gaat leren wat een associative array is en hoe je de inhoud hiervan kunt afdrukken in een tabel.

Associative arrays zijn belangrijk want resultaten van een query uit de database worden door PDO vaak als associative array terug gegeven. Hier gaan we later (in PHP PDO) ook mee oefenen. Ook zal dit terug komen in Laravel.

We hebben in de vorige les 'gewone' arrays gezien, een 'gewoon' array kun je aanspreken via bijvoorbeeld:

```
$array[0]
```

Dit is het eerste element van het \$array. De 0 is de index, dat geeft aan welk element je aanspreekt.

Bij een associative array heb je ook een index, maar de index is geen nummer maar een naam, we noemen dat een sleutel. Bijvoorbeeld:

```
$array['PHP']
```

Dit is het element in het \$array met de index 'PHP'. De index in een associative array wordt ook wel key genoemd.

Een index in een 'gewoon' array heet **key** in een associative array.

We kunnen op deze manier een heel array opbouwen, bijvoorbeeld:

```
<?php

$array['PHP']=5.5;
$array['Java']=6.5;
$array['Engels']=7.0;
$array['Nederlands']=6.5;
$array['Veilig Programmeren']=8.0;
```

Dit zouden bijvoorbeeld de cijfers voor de verschillende vakken kunnen zijn.

Deze kun je ook op een andere manier toekennen:

```
<?php
```

```
$array=array("PHP" => 5.5, "Java"=>6.5, ....);
```

Vraag 1: Maak nu de onderstaande code af zodat alle vakken, PHP, Java, Engels, Nederlands, en Veilig programmeren allemaal afgedrukt worden.

```
<?php
```

```
$array=array("PHP" => 5.5, "Java"=>6.5, MAAK HIER DE CODE AF);
```

```
echo "PHP:" . $array["PHP"]."<br>";
```

```
echo "Java:". $array["Java"]."<br>";
```

```
MAAK HIER DE CODE AF
```

```
?>
```

Nu werkt dit best een beetje onhandig want je zou deze vakken het liefst in een loop afdrukken. Net zoals de tafels die we in de vorige hoofdstukken in een loop maakte.

Maar we kunnen ook door een associative array met een loop. Hoe werkt dat?

De standaard manier is als volgt:

```
<?php
```

```
foreach ($array as $key => $value) {
```

```
    ...
```

```
}
```

In dit voorbeeld is \$array het associative array, en is \$key de key (zeg maar de index) en \$value de waarde.

Opdracht 1

Druk met behulp van een loop (foreach zoals hierboven is uitgelegd) het associatieve [array](#) helemaal af.

Als dt klaar is dan hebben we dus code gemaakt die het volgende afdrukt:

PHP:5.5
Java:6.5
Engels:7.0
Nederlands:6.5
Veilig Programmeren:8.0

Verander deze code zodat de gegevens in een tabel worden afgedrukt. Dus ongeveer zo:

Vak	Cijfer
PHP	5.5
Engels	7.0
Nederlands	6.5
Veilig Programmeren	8.0

PDO

Je gaat leren hoe PDO de inhoud van een tabel aan jou terug geeft en hoe je deze in een HTML tabel kan afdrukken.

Kijk eens naar de volgende code, dit is een voorbeeld van ene resultset zoals je die van PDO zou kunnen krijgen als je een query op de database uitvoerd.

```
<?php
$resultSet = array(
    array(
        'id' => "1",
        'name' => "Mo"
    ),
    array(
        'id'  => "2",
        'name' => "Angel"
    ),
    array(
        'id'  => "3",
        'name' => "Do"
    ),
    array(
        'id'  => "4",
        'name' => "Anouar"
```

```

    ),
    array(
        'id' => "5",
        'name' => "Keyana"
    )
)

```

Wat je ziet is een 'gewoon' array van associative arrays. Weet je nog dat we arrays van arrays konden maken? Dat doen we nu ook in de volgende vorm (schematisch):

```
[ [associative array], [associative array], [associative array], [associative array], [associative array] ]
```

We hebben dus een array genaamd, `$resultSet` en die heeft 5 elementen, `$resultSet[0]`, `$resultSet[1]`, ..., `$resultSet[5]`.

Opdracht 1: maak een for-loop en doe een `print_r` van alle 5 de elementen van de `$resultSet`.

Opdracht 2

Vervang nu de `print_r` zodat je nu ook de resultaten netjes in een tabel krijgt, dus de output moet er als volgt uit zien.

Id	Naam
1	Mo
2	Angel
3	Do
4	Anouar
5	Keyana

Dit zou dus de inhoud van een tabel kunnen zijn.

We hebben nu geleerd hoe we de return value van een PDO object in een tabel kunnen weergeven!

--

Besproken in de les ... to be continued

```
<?php
```

```
echo "<pre>";
```

```
$a = [1, 3, 5];
```

```
$b = [2, 4, 6];
```

```
$combine=[ $a, $b];
```

```
print_r($combine);
```

```
echo $combine[1][0];
```

```
echo $combine[1][1];
```

```
echo $combine[1][2];
```

```
$deref=$combine[1];
```

```
echo $deref[1];
```

```
echo "<hr>";
```

```
for($i=0; $i<3; $i++) {
```

```
    echo $combine[1][$i]."<br>";
```

```
}
```

```
echo "<hr>";
```

```
$array1=array('PHP' => 5.5, 'Java'=>6.5, 'Engels'=>7.0, 'Nederlands'=>6.5, 'Veilig Programmeren'=>8.0);
```

```
$array2=array('PHP' => 6.0, 'Java'=>7.0, 'Engels'=>7.5, 'Nederlands'=>7.0, 'Veilig Programmeren'=>8.5);
```

```
$combine=[ $array1, $array2, $array2 ];
```

```
print_r($combine);
```

```
echo $combine[1]['PHP'];
```

```
echo $combine[1]['Java'];
```

```
echo $combine[1]['Engels'];
```

```
echo "<hr>";
```

```
for($i=0; $i<count($combine); $i++) {
```

```
    echo "Cijfers persoon $i:<br>";
```

```
    foreach( $combine[$i] as $key => $value ) {
```

```
        echo "Vak: ".$key." cijfer: ".$value."<br>";
```

```
    }
```

```
}
```

Revision #7

Created 28 January 2020 17:54:34 by Admin

Updated 13 February 2020 09:54:02 by Max