8 Assiociative Arrays (PHP L3)

We hebben al heel wat over arrays gesproken.

Terugblik

Tot nu toe hebben we twee soorten arrays gezien.

1. Eenvoudige arrays, en;
2. complexe arrays (multidimensionale arrays).

Voorbeelden

```
$array1=['waarde 1', 'waarde 2', 'waarde 3', 'waarde 4'];
$array2=[ ['waarde 1a', 'waarde 2a', 'waarde 3a', 'waarde 4a'],
          ['waarde 1b', 'waarde 2b', 'waarde 3b', 'waarde 4b'],
          ['waarde 1c', 'waarde 2c', 'waarde 3c', 'waarde 4c']
        ];
```

Herken je het complexe array?

$array2 is een array van 3 arrays en elk van die drie arrays heeft 4 elementen.

Assiociative array

In deze les gaan we kennis maken met nog een type array, het *assiociative array*.

Er bestaan verschillende namen voor een assiociative array. In python heet het een dictionairy en soms wordt ook wel de meer algemenere term key-value array gebruikt. Je zult aan het eind van de les wel begrijpen waarom.

> assiociative array = dictionary = key-value array

Assiociative arrays komen ook voor in Python (daar heten dit Dictionaries).

Het verschil tussen een 'gewoon' (indexed) array en een assiociative array is de key (sleutel).

Snap je dit nog niet helemaal, of wil je nog meer voorbeelden zien, kijk dan deze video.

https://studio.youtube.com/video/5lJLecI0BTA/edit

De key

Met een key bepaal je welke elemenent je wilt zien in een array. Bijvoorbeeld:

```
$array=['Amsterdam','Rotterdam','Den Haag','Almere'];
echo $array[1];
```

Kijk naar regel 2, daar staat een [1]. Dat is de key . De key is 1 en dat betekent dat je het tweede element van het array afdrukt. In dit geval '*Rotterdam*' dus. Nee niet de '*Amsterdam*' want een array begint te tellen bij 0.

Stel je wilt de plaats 'Almere' overschrijven? Dat kan met:

```
$array[3]='Utrecht';
```

Bij een assiociative array moet jij zelf voor elk element de key bepalen. Dat ziet er bijvoorbeeld zo uit:

```
$array[ 'naam' => 'Nike Chui' , 'datum' => '12 november 2020' , 'project' => 'PHP Project'];
```

Om nu het tweede element van dit array af te drukken gebruik je:

```
echo $array['datum'];
```

Het maakt niet uit op welke plaats de datum staat, de key is datum. De waarde die bij deze key hoort wordt afgedrukt.

Stel je wilt de datum aanpassen dan kun je dat als volgt doen:

```
$array['datum']='1 december 2020';
```

Samengevat

	Andere namen	Hoe benader je een element?	Voorbeeld
Indexed Array	gewoon array, 1 dimensionaal array, eenvoudig array	via de key, die is altijd 0,1,2,3,4....	$myArray[13]

Assiociative Array	key-value array, Dictionary (Python), hash table	via de key, dit moet een unieke string zijn waarvan je zelf de waarde bepaald.	$myArray['datum'];

Opdracht 1

Neem deze template en vul de code aan zodat je het hele assiociative array afdrukt.

```
$array[ 'naam' => 'Nike Chui' , 'datum' => '12 november 2020' , 'project' => 'PHP Project'];
```

De output moet er als volgt uit komen te zien.

```
Nike Chui
12 november 2020
PHP Project
```

Inleveren

Lever de code in en gebruik de naam php81-jouw-naam.php

Opdracht 2

In de vorige opdracht drukte we alle waarden af. We kunnen ook de keys en de waarden afdrukken met een loop. Dat is vooral handig als we niet weten hoe groot het array. We maken en loop en drukken gewoon het hele assiociative array af.

We gebruiken hetzelfde array, maar de output moet anders worden, namelijk:

```
key=naam, value=Nike Chui
key=datum, value=12 november 2020
key=project, value=PHP Project
```

Zoek zelf op hoe je dat doet, bijvoorbeeld op:

https://www.w3schools.com/php/php_arrays_associative.asp

Gebruik de foreach loop om alle keys en values af te drukken. De output moet er precies zo uitzien als in deze opdracht staat.

Inleveren

Lever de code in en gebruik de naam php82-jouw-naam.php

Opdracht 3

Maak zelf een assioiciative array. Maak de elementen:

```
voornaam, achternaam, geboortedatum, woonplaats
```

Zet de juiste waarden, dus jouw voornaam, jouw achternaam, jouw geboortedatum en jouw woonplaats.

Druk alle waarden uit dit assiociatieve array af (hoe, met of zonder loop, dat mag je zelf weten).

Inleveren

Lever de code in en gebruik de naam php83-jouw-naam.php

Opdracht 4

Wat je in de praktijk vaak tegenkomt zijn arrays van assioicative arrays. Als je (later) gegevens uit de database haalt dan krijg je ook vaak een array van assiociative arrays terug. Hoe ziet dat eruit?

```
$uitslagen=[
[ 'thuis' => 'FC Twente', 'uit' => 'FC Utrecht', 'uitslag' => [0,1] ]
[ 'thuis' => 'FC Twente', 'uit' => 'FC Volendam', 'uitslag' => [3,1] ]
[ 'thuis' => 'FC Emmen', 'uit' => 'Feyenoord', 'uitslag' => [0,3] ]
[ 'thuis' => 'Vitesse', 'uit' => 'FC Twente', 'uitslag' => [1,1] ]
]
```

Elke regel is eigenlijk een assioicative array. Er staan 4 assiociative arrays in het array $uitlsagen.

Probeer maar eens:

```
echo "<pre>";
print_r($uitslagen[0]);
```

Met print_r() kun je een array afdrukken.

Je kunt op deze manier dus $uitslagen[0], $uitslagen[1], $uitslagen[2] en $uitslagen[3] afdrukken.

Stel je wilt de thuisploeg afdrukken van de tweede uitslag. Hoe gaat dat?

```
echo $uitslagen[1]['thuis'];
```

En Stel je wilt weten hoeveel de thuisploeg van de derde wedstrijd heeft gescoord.

```
echo $uitslagen[3]['uitslag'][0]
```

Je begint dus met de vierde regel (3) dan neem je de 'uitslag' en van de uitslag neem je dan het eerste elemenet (0).

OK nu jij. Gebruik het array $uitslagen en druk de volgende gegevens af:

Het aantal doelpunten dat FC Twente heeft gescoord.

De output moet er als volg uit zien.

```
FC Twente scoort 0 punten
FC Twente scoort 3 punten
FC Twente scoort 1 punten
```

De waarden 0, 3 en 1 moet je uit het assiociative array halen.

Inleveren

1. Lever de code in en gebruik de naam php84-jouw-naam.php
2. Lever een schermafdruk van je gehele browser en laat daarbij de output zien, gebruik de naam php84-jouw-naam.png

Opdracht 5

Gebruik het assiociative array van opdracht 4.

```
$uitslagen=[
[ 'thuis' => 'FC Twente', 'uit' => 'FC Utrecht', 'uitslag' => [0,1] ]
[ 'thuis' => 'FC Twente', 'uit' => 'FC Volendam', 'uitslag' => [3,1] ]
[ 'thuis' => 'FC Emmen', 'uit' => 'Feyenoord', 'uitslag' => [0,3] ]
[ 'thuis' => 'Vitesse', 'uit' => 'FC Twente', 'uitslag' => [1,1] ]
]
```

Je gaat nu in een loop de punten van thuisploeg uitrekenen.

Je krijgt 3 punten als je de wedstrijd wint, 1 bij een gelijkspel en 0 als je de wedstrijd verliest.

In de eerste wedstrijd heeft de thuisploeg 0 punten gescoord, in de tweede wedstrijd 3, in de derde wedstrijd 0 en in de vierde wedstrijd 1.

De output van jouw code ziet er als volgt uit:

```
Wedstrijd 1: de thuisploeg krijgt 0 punten
Wedstrijd 2: de thuisploeg krijgt 3 punten
Wedstrijd 3: de thuisploeg krijgt 0 punten
Wedstrijd 4: de thuisploeg krijgt 1 punten
```

Lastig? Probeer dit in stapjes te doen:

Stap 1, maak eerst een loop waarin je de uitlslagen afdrukt (gebruik print_r).

Stap 2, druk nu de uitslagen niet af met print_r maar druk de afzonderlijke cores af, dus 0 1 en dan 3 1 enzovoorts.

Stap 3 gebruik nu een if en test of het eerste getal groter is dan het tweede. Als dat zo is dan druk je 3 af omdat de thuisploeg dan 3 punten krijgt.

Stap 4 maak nog een if en test of het eerste en tweede getal gelijk zijn. Als dat zo is dan druk je 0 af.

Stap 5 maak nog een if en test of het tweede getal groter is dan het eerste. als dat zo is dan druk je 1 af.

Stap 6 je bent bijna klaar je moet nu alleen nog text om de uitslag heen printen. Dus druk nu niet 0 af, maar 'de thuisploeg krijgt 0 punten.

Stap 7, nu nog het eerste gedeelte. Maak een variabele $teller en zet de waarde op 1. Druk af Wedstrijd $teller en hoog daarna de $teller op met 1 ($teller = $teller + 1)

Inleveren

1. Lever de code in en gebruik de naam php85-jouw-naam.php
2. Lever een schermafdruk van je gehele browser en laat daarbij de output zien, gebruik de naam php85-jouw-naam.png

--

Revision #18
Created 2022-06-10 17:59:24 UTC by Max
Updated 2022-10-23 08:07:31 UTC by Max