4 Complexe arrays

We hebben al naar arrays gekeken. In de terugblik hieronder kijken we nog een keer naar deze 'eenvoudige' arrays. Deze arrays zijn 'eenvoudig' omdat ze uit één array bestaan. Je hebt ook arrays die bestaan uit arrays. Een array van arrays dus. Dit zijn *multidimensionale arrays* of *complexe arrays.* We praten vanaf nu over complexe arrays maar:

> complex array = multidimensionale array

Een complex array is dus een array van een array. Je hebt zelfs arrays van arrays van arrays van arrays,.... maar voor nu houden we het bij arrays van arrays.

In deze les leren hoe een array van een array er uit ziet en hoe je er met en zonder gebruikt te maken van een loop gegevens uit kan lezen.

Terugblik

Zoals uitgelegd in voorgaande lessen kan een array meerdere waarden bevatten. Iedere waarde is gekoppeld aan een voor de array unieke *sleutel*. Deze sleutels worden gebruikt om een specifieke waarde uit de array af te kunnen lezen of te kunnen bewerken.

Bijvoorbeeld

```
$fruit = ['appel', 'banaan', 'citroen'];
$fruit = array('appel', 'banaan', 'citroen');
```

Beide regels zijn hetzelfde en eigenlijk gebruiken we altijd de eerste variant. De tweede staat erbij omdat je dit soms kan tegenkomen in code.

Stel je wilt het eerste element uit het $fruit array afdrukken, dat kan met:

```
echo $fruit[0];
```

Het eerste element heeft dus key 0, het tweede key 1, het derde key 2, enzovoorts.

Het array $fruit is een 'eenvoudig' array. Er bestaan ook complexe arrays. Dit zijn arrays in arrays. Als je later leert om gegevens uit een database te halen dan krijg je te maken met complexe arrays.

Complex array

Kijk naar de volgende code.

```
$eten = [ 'fruit', 'groent', 'brood'];
```

We hebben een array van drie elementen en elk element heeft één string waarde.

We zouden elk element ook meerdere waarden kunnen geven. We kunnen namelijk van elk element weer een array maken. We krijgen dan een array van een array.

```
$eten = [ [],
          [],
          [] ];
```

We hebben nu een array dat bestaat uit drie lege arrays. Nu gaan we de lege arrays vullen.

```
<?php

$eten = [ ['appel','peer','druif','banaan','perzik','bosbes',],
          ['prei', 'aardappel', 'wortel', 'sla'],
          [ 'croisant', 'wit brood', 'bruin brood', 'volkoren']
        ];
```

Om een element uit het array af te drukken moeten we nu twee keys (indexes) megeven. Stel je wilt "prei" uit het array afdrukken, dan moet je van het 2de array het 1ste element afdrukken. Omdat we vanaf 0 beginen met tellen, wordt dat dan .

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

Opdracht 4a

Gebruik het array eten en druk *appel*, *bosbes*, *sla* en *croisant* af. Gebruik daarbij het array en $eten en gebruik de juiste indexes (zoals in het voorbeeld).

Complex array met loop

Laten we nog een complex array bouwen.

Stel je wilt elke dat de temperatuur drie maal opmeten. Een keer 's morgens, één keer 's middags en één keer 's avonds.

Je kunt dit dan opslaan in een array met drie elementen.

```
$metingen = [ 7, 12, 11 ];
```

Stel dat je van meerdere dagen de metingen wilt opslaan. Dan kun je een array van een array maken. Je maakt dan een array waarin arrays zitten die dan elk drie elementen hebben, bijvoorbeeld:

```
$metingenWeek=[ [14,11,12], [ 6,11,11], [7,12,14], [9,14,13], [10,15,13] , [11,15,14 ],
[13,16,14] ];
```

Je hebt nu array met metingen van 7 dagen en elke dag heb je een array met drie metingen. Zo kun je zien dat de eerste dag de warmste ochtend had en dat de laatste dag de warmste middag had.

print_r()

Met de PHP function print_r() kan je een array afdrukken.

Stel voor dat je de temperatuur van de eerste dag (element 0!) van de ochtend wil afdrukken, hoe doe je dat dan?

```
echo $metingenWeek[0][0];
```

Stel wel willen het volgende afdrukken:

```
Dag 1:
  dagdeel 14
  dagdeel 11
  dagdeel 12

 Dag 2:
   dagdeel 6
   dagdeel 11
   dagdeel 11

 etc., etc.
```

We maken een loop en hebben het volgende programmaatje gemaakt.

```
$metingenWeek=[ [14,11,12], [ 6,11,11], [7,12,14], [9,14,13], [10,15,13] , [11,15,14 ],
[13,16,14] ];
```

```
$dagNummer=0;

foreach($metingenWeek as $dag) {
    echo "Dag ".$dagNummer."<br>";
    echo "'s ochtends: ".$metingenWeek[0][0]."<br>";
    echo "'s middags : ".$metingenWeek[0][1]."<br>";
    echo "'s avonds  : ".$metingenWeek[0][2]."<br>";
}
```

Helaas er zitten twee fouten in het programmaatje. Deze fouten moet je oplossen in de opdracht.

Opdracht 4b

Test de code uit het voorbeeld uit.

Probleem 1

Het dagnummer blijft op 0 staan. Los dit op en zorg ervoor dat de 7 dagen allemaal worden afgedrukt en dat het juisten dagnummer wordt afgedrukt. De dagnummers beginnen te tellen bij 1.

Pas de code aan zodat de dagnummers correct worden afgedrukt.

Probleem 2

Voor 's morgens, 's middags en 's avonds worden telkens dezelfde getallen afgedrukt. Dat klopt niet. In het voorbeeld staan voor dag 1 en dag 2 de juiste waarden afgedrukt. De programmeur heeft een foutje (of eigenlijk twee) gemaakt. Spoor de fout op en zorg ervoor dat de output klopt en dat de juisten getallen (temperaturen) worden afgedrukt.

Vergeet je code niet te testen!

Inleveren de aangepaste en geteste code in php02-04a-jouw-naam.php

Opdracht 4c

Bereid het programmaatje van opdracht 4b uit zodat de output als volgt wordt.

```
Dag 1:
  dagdeel 14
  dagdeel 11
  dagdeel 12
  hoogste waarde is 14
```

```
Dag 2:
  dagdeel 6
  dagdeel 11
  dagdeel 11
  hoogste waarde is 11

etc., etc.
```

Je drukt dus telkens per dag de hoogste temperatuur af. Dat doe je voor alle dagen.

Tip: kijk naar de PHP functie max() op https://www.w3schools.com/php/func_math_max.asp

Deze functie kan handig van pas komen. Je kan en mag de opdracht ook op een andere manier oplossen.

Gaat het jouw lukken? Succes!

--

Revision #13
Created 2022-06-03 15:14:24 UTC by Max
Updated 2022-12-18 12:35:34 UTC by Max