PHP L2

PHP Level 2 Loops en functions

- 1 Array variabelen
- 2 Foreach loop
- 3 Functions
- 4 Complexe arrays
- 5 Nog meer loops (for)
- 6 Nog meer functions
- 7 Opdracht
- 8 Assiociative Arrays (PHP L3)
- 10 Challenge Voetbal

1 Array variabelen

Wat ga je leren?

In deze les leer je wat een array is en hoe je een array kan gebruiken. Je leert dat een array bestaat uit elementen.

Je leert dat alle elementen van een array een nummer hebben en dat de computer begint te tellen bij 0. Het eerste element van een array heeft hierdoor de key 0.

In bijna alle talen heb je een speciaal soort variabele; het array. Een array is een verzameling van variabelen. Een soort lijst waarin allemaal variabelen zitten.

In Python heb je ook *arrays* en daar wordt het *list* genoemd. In PHP noemen we het een array.

In het Engels betekent een array, "*een groep mensen of dingen die op een bepaalde manier zijn geordend*".

Als je een variabele als een bladzijde van een boek ziet waar je een nummer op kan zetten of een string. Dan is een array een boek. Een boek heeft dan bladzijden en op elke bladzijde staan dan een variabele.

In een array begin je altijd op bladzijde 0. Op pagina 0 van het boek staat dus de eerste variabele.

Hoe ziet dat er uit in code?

```
$mijnArray=['maandag','dinsdag','woensdag','donderdag','vrijdag','zaterdag','zondag'];
```

Je hebt hier dus een array, op bladzijde 0 staat de string 'maandag', op bladzijde 1 staat de string 'dinsdag', enzovoorts.

```
$mijnArray=['maandag','dinsdag','woensdag','donderdag','vrijdag','zaterdag','zondag'];
echo $mijnArray[0];
echo "<br>";
echo $mijnArray[1];
```

Een "bladzijde" van een array heet een element. Op regel 2 van de code wordt dus het eerste element van het array $mijnArray afgedrukt., op regel 4 wordt het tweede element van $mijnArray afgedrukt.

Bekijk nu het filmpje tot aan 4:49 (foreach loop dus niet).

https://www.youtube.com/embed/mNcZG4-Mi9M

Als je het filmpje hebt bekeken dan weet je wat de key van een array is en hoe je een array waarde kan wijzigen.

Opgave 1a

Gebruik array `$myArray=[10,20,30,40,50,60]` en druk het tweede element af.

Lever de code in en gebruik de naam php201b-jouw-naam.php

Opgave 1b

Gebruik array `$myArray=[10,20,30,40,50,60]` verander de eerste drie elementen één-voor-één in 1,2 en 3.

Druk de elementen 2 en 3 af. De output ziet er als volgt uit:

```
3
40
```

Lever de code in en gebruik de naam php201b-jouw-naam.php

Opgave 1c

Vervang in de onderstaande code de xxx. Maak gebruik van het array $seizoenen en druk het juiste seizoen af.

```
$seizoenen=[ 'zomer', 'herfst', 'winter', 'lente' ];

echo "Het koudste seizoen is de ".xxx;
echo "Het warmste seizoen is de ".xxx;
```

Lever de code in en gebruik de naam php201c-jouw-naam.php

--

2 Foreach loop

Wat ga je leren?

In deze les gaan we leren wat een foreach loop is en hoe je daarmee makkelijk dingen met een array kan doen.

In de vorige les hebben we kennis gemaakt met arays. Als je iets met een array wilt doen dan kun je natuurlijk element voor element benaderen en (bijvoorbeeld) afdrukken:

```
echo myArray[0];
echo myArray[1];
echo myArray[2];
echo myArray[3];
echo myArray[4];
.....
```

Maar er zijn handigere methoden om iets met een array te doen. Daarvoor hebben we een loop.

Met een foreach loop kun je door een array heen lopen. Je kunt element voor element doorlopen.

Kijk naar het tweede gedeelte van de film (vanaf 4:49, het eerste gedeelte heb je in de vorige les al bekeken).

https://www.youtube.com/embed/mNcZG4-Mi9M

Er zijn meerdere soorten loops en daar gaan we nog een keer uitgebreid naar kijken. Voor nu kijken we eerst even naarde foreach loop. Deze is belangrijk om dat je die kunt gebruiken om door een array heen te lopen. Stel je wilt alle elementen van een array afdrukken.

Dat kan op deze manier:

```
$mijnArray=['maandag','dinsdag','woensdag','donderdag','vrijdag'];


echo $myArray[0];
echo "<br>";
echo $myArray[1];
echo "<br>";
echo $myArray[2];
```

```
echo "<br>";
echo $myArray[3];
echo "<br>";
echo $myArray[4];
echo "<br>";
```

Dit is alleen onhandig want stel dat we zaterdag en zondag toevoegen aan het array dan moeten we regels toevoegen om deze elementen af te drukken.

Dat kan dus handiger, kijk maar de volgende code.

```
$mijnArray=['maandag','dinsdag','woensdag','donderdag','vrijdag'];


foreach($mijnArray as $dag)
{
  echo $dag;
  echo "<br>";
}
```

Wow dat is korter, toch?

Hoe werkt dit?

- Op regel 3 staat: zet één-voor-één elk element uit het array $mijnArray in de variabele $day en voer de code uit die tussen de { en } staat.
- Op regel 4 begint het codeblok, dit is het begin van de code die voor elk element wordt uitgevoerd.
- Op regel 5 wordt de $dag afgedrukt. Dat wordt dus voor elk element van $mijnArray gedaan.
- Op regel 6 wordt er een nieuwe regel afgedrukt en ook dit wordt voor elk element van het array uitgevoerd.
- Op regel 7 wordt het codeblok afgesloten, dit is het einde van de loop.

Opdracht 2a

Maak een array met namen van klasgenoten. Zet in ieder geval 10 namen in het array. De namen zijn strings dus denk erom dat de namen tussen " moeten staan.

Druk de inhoud van het array af en zet op elke regel een naam. Gebruik dus een <br> na elke naam.

Inleveren

Lever de code in en gebruik de naam php202a-jouw-naam.php

Opdracht 2b

Zorg ervoor dat jouw eigen naam ook ergens in het array staat.

Zet in de foreach loop een if statement. Als de loop bij jouw naam is dan dan druk je een !-teken af.

Stel jij heet Omar, dan zou de if er zo uit zien:

```
if ($naam == "Omar") {
  echo "!";
}
```

Zet dit in de loop en controleer of er alleen achter jouw eigen naam een ! wordt afgedrukt.

Let er dus op dat de complete if in het code blok van de loop komt te staan. Let er ook op dat in het voorbeeld $naam wordt gebruikt, het kan zijn dat jij in jouw loop voor een andere variabele naam hebt gekozen.

Inleveren

Lever de code in en gebruik de naam php202b-jouw-naam.php

3 Functions

Wat ga je leren?

In alle programmeertalen heb je functies. Met functies kun je code opdelen in logsiche blokken en kan je veel voorkomende stukjes code hergebruiken.

In deze les leer je wat een functie is en hoe je die in PHP kan gebruiken.

Stel je hebt een stukje code waarin je een lastige berekening maakt en deze berekening wil je op verschillende plaatsen in jouw code uitvoeren dan zou het handig zijn dat je niet telkens opnieuw die (lastige) berekening moet maken. Dat hoeft ook niet als je functies gebruikt, Een functie is een stukje code dat je op elke moment kan uitvoeren.

PHP heeft zelf ook veel ingebouwde functies die jij kunt gebruiken, maar daarover in een andere les over functies meer.

Een functie is een stukje code. De code ontvangt input doet er wat mee en de function maakt dan output.



Geldautomaat

In het dagelijks leven zijn er nog veel meer dingen die je met een function kan vergelijken. Neem bijvoorbeeld een geldautomaat.



Je stopt er wat in (pasje, pincode) en de function gaat van alles doen zoals je saldo en pincode controleren. Als alle controles goed gaan krijg je als output geld.

Wiskunde

In de wiskunde heb je ook functies en doen hetzelfde:

f(x) = x * x + 3

Deze functie krijgt als input een waarde x en zet deze om in output. Als je in deze functie 2 stopt dan krijg je er 7 uit (2 * 2 + 3).

Voorbeeld

Stel we willen deze laatste wiskunde functie in PHP-code omzetten, dan ziet dat er zo uit.

```
function myFunction($x) {
  $output = $x * $x + 3;
  return($output);
}
```

Regel 1, begin je de functie. Dat doe je altijd met het woord function, gevolgd door de functienaam die je zelf mag verzinnen. Dan volgt een { om aan te geven dat alle code tussen { en } bij de function hoort. De function loopt dus door tot en met regel 4.

De function krijgt de waarde $x als input. In het volgende voorbeeld laat ik zien hoe je deze waarde aan de functie meegeeft.

Regel 2, we zetten het resultaat van de berekening in de variabele $output.

Regel 3, hier maken we de output. In het volgende voorbeeld laat ik zien hoe je die output kunt krijgen.

Regel 4, Dit is het einde van de function.

Hoe roepen we deze function aan en hoe ontvangen we de output?

```
$in=4;
$uit=myFunction($in);
echo $uit;

$uit=myFunction(6);
echo $uit;

echo myFunction(9);
```

Je ziet hier drie voorbeelden van hoe je de function kan aanroepen. In het eerste voorbeeld geven we de variabele $in de waarde 4, en we geven deze waarde mee als input van de function. Het

resultaat komt in $uit. En op regel 3 drukken we de waarde van $uit dan af.

Op regel 5 geven we gelijk de waarde 6 als input mee.

Op regel 7 geven we gelijk de waarde 9 mee en drukken de output gelijk af zonder deze eerst op te slaan in een variabele.

In het volgende filmpje laat een dialectische bekende programmeur zien hoe je een functie kan maken voor het omrekenen van liters naar gallons. Gallons is in Amerika de maat voor volume. Je koop daar dus bijvoorbeeld geen liter Cola maar een halve gallon.

https://www.youtube.com/embed/XfnH3AEF5Z8

PHP-functions

PHP heeft zelf een groot aantal ingebouwde functions. Hieronder een paar voorbeelden:

PHP Function	input	Output	Voorbeeld
strlen()	string	lengte van de string	echo strlen("Hello");
str_word_count()	string	aantal woorden van de string	echo str_word_count("Hello there!");

voorbeeld

```
$str="The quick brown fox jumps over the lazy dog";

$res1=strlen($str);
$res2=str_word_count($str);

echo "De string is %res1 letters en bestaat uit $res2 woorden.";
```

Op regel 3 en 4 worden twee ingebouwde PHP functions gebruikt.

Weet je overigens waar de zin ("The quick brown fox.....") op slaat?

Deze zin wordt soms gebruikt voor testen omdat elke letter van het alfabet er precies in keer in voorkomt.

Opdracht 3a

Bekijk het voorbeeld en maak een eigen functie. De berekening die de function moet maken is:

```
$output = $x * 2;
```

Noem de function *myVoornaam* (vervang *voornaam* door jouw eigen voornaam) en test of je code werkt.

Inleveren

php203a-jouw-naam.php met daarin de function.

Opdracht 3b

Maak een function waarmee je de temperatuur van graden in Fahrenheit kan omrekenen.

Het omrekenen doe je als volgt:

```
$farenheit = ($celcius * 9/5) + 32;
```

Maak een function die aan de volgende voorwaarden voldoet:

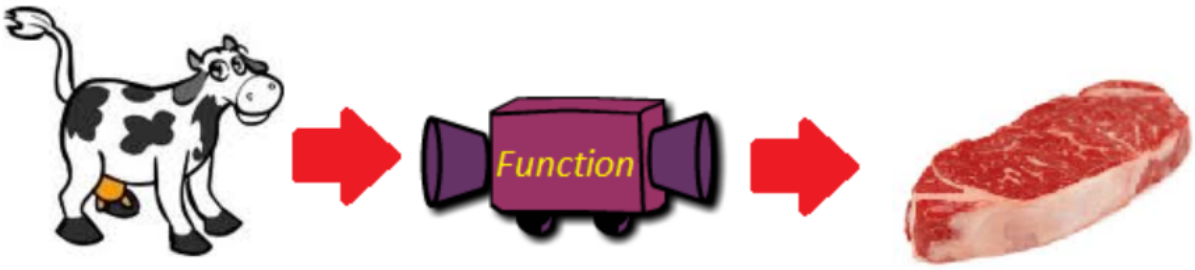
function naam	bedenk een logische naam
function input	een getal, de temperatuur in graden celcius
function output	een getal, de temperatuur omgerekend in graden fahrenheit

Test jouw functie. Roep de function op met tenminste 3 verschillende input waarden en druk de resultaten af. Laat deze testen in jouw code staan.

Op deze pagina kun je celcius in fahrenheit om laten rekenen. Gebruik deze site om te testen.

Inleveren

php203b-jouw-naam.php met daarin de function en tenminste drie testen waarin je laat zien dat de juisten waarden door de function laat berekenen.



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!

--

5 Nog meer loops (for)

We hebben eerder gezien hoe we met een foreach loop door een array kunnen heen lopen.

We hebben ook loops die we kunnen gebruiken zonder dat we een array hebben.

Stel we willen de getallen 1,2,3,4,5,6,7,8,9,10 op het scherm zetten, dan kan dat ook met een loop.

```
for($i=1; $i<=10, $i=$i+1) {
  echo $i.", ";
}
```

Wat gebeurt hier?

Op regel 1 worden loop gemaakt. We gebruiken de variabele $i en zetten die op 1 ($i=1), dan zeggen we doe de loop zolang $i kleiner of gelijk is aan 10 en elke keer dat de volgende keer in de loop komen dan doen we $i = $i + 1 en we verhogen $i dus met 1.

Op regel 2 drukken we het resultaat af en op regel 4 sluiten we het codeblok dat deel van de loop uitmaakt.

Het is gebruikelijk om voor een for-loop de variabele $i te gebruiken.

Opdracht 5a

Je hebt 5 loops (a t/m e). Bedenk *vooraf* van elke loop wat jij denkt dat die doet. Schrijf dat in eigen woorden op.

Test daarna of je antwoorden kloppen en geef bij elke loop weer of je antwoord klopt. Als je antwoord niet klopt dan verbeter je je antwoord.

```
//loop A
for($i=1; $i<=20, $i=$i+1) {
  echo $i.", ";
}

//loop b
for($i=20; $i<30, $i=$i+1) {
  echo $i.", ";
}
```

```
//loop c
for($i=2; $i<100, $i=$i+2) {
  echo $i.", ";
}

//loop d
for($i=10; $i>0, $i=$i-1) {
  echo $i.", ";
}

//loop e
for($i=100; $i>0, $i=$i-2) {
  echo $i.", ";
}
```

Inleveren MS Word file (docx) waarin je het volgende beschrijft (de ingevulde tekst in een voorbeeld!).

```
//loop a
Deze loop begint bij 3 en wordt telkens 1 opgehoogd, de loop stop bij 10. Dus de output is
3,4,5,6,7,8,9,10

Nadat ik heb getest blijkt dat de 10 niet wordt afgedrukt dat komt omdat er <10 staat en niet
<=10

//loop b

Deze loop .......

Ik heb getest en het antwoord klopt.

// loop c
 Deze loop.....

 Ik heb getest en ....

 // loop d
  Deze loop.....
```

```
 Ik heb getest en ....

 // loop e
  Deze loop .....

 Ik heb getest en ....
```

Inleveren php205a-jouw-naam.docx met de uitgewerkte opgaven.

Opdracht 5b

Je kunt een array ook met een for-loop doorlopen. Bestudeer de volgende code.

```
$myArray=[1,3,2];

foreach($myArray as $item) {
echo $item;
 echo "<br>";
}

for($i=0; $i<3; $i++) {
echo $myArray[$i];
    echo "<br>";
}
```

Probeer de code uit en probeer te verklaren wat je ziet.

Maak code die het volgende array af drukt één keer met een foreach-loop en daarna met een for-loop.

```
$myColors['rood','paars','oranje','blauw','wit'];
```

Inleveren php205b-jouw-naam.docx met de uitgewerkte opgaven.

Opdracht 5c

Maak een loop (je mag zelf kiezen welke loop) waarin je alle getallen van 1 tot en met 100 optelt.

Jouw code druk op een getal af de som van 1+2+3+4+5+6+7+.....99+100.

Je kunt natuurlijk de volgende code inleveren:

```
echo 1+2+3+4+5+6+7+8+9+10+11+12;
```

En dan tot 100. Dat kan, maar stel dat ik nu vraag om alle getallen van 1 tot en met 10 000 op te tellen? Dan is dat bijna niet meer te doen. Je moet voor deze opdracht dan ook een loop gebruiken.

Als je niet weet hoe je dat aanpakt dan maak je dit probleem eerst eenvoudiger; je drukt eerst alle getallen 1 tot en met 100 af. Als dat gelukt is dan ga je al deze getallen bij elkaar op tellen.

Inleveren php205c-jouw-naam.docx met de uitgewerkte opgaven.

6 Nog meer functions

Nog een voorbeeld van een function

In deze les wordt een iets complexere functie gemaakt. In deze functie worden de even getallen uit een array gefilterd.

https://www.youtube.com/embed/INHM0S_8GHA

Parameters

Dat wat je meegeeft aan een function heten parameters. In de vorige voorbeelden (filmpjes) zagen we dat we telkens precies één parameter meegaven. In het eerste voorbeeld was dat een getal en in het tweede voorbeeld was dat een array.

Je kunt ook minder of meer parameter meegeven, bijvoorbeeld nul:

```
function welkom() {
    $tekst = 'Welkom op mijn website';
    return $tekst;
}
```

En je kunt ook meer parameters meegeven, bijvoorbeeld drie:

```
function welkom2($tekst, $naam, $hoofdletters = FALSE ) {
    //bepaal tekst
    if ($tekst == 1) {
        $uitvoer = 'Welkom '. $naam;
    }
    elseif ($tekst == 2) {
        $uitvoer = 'Tot ziens, ',$naam;
    }
    //bepaal hoofdletters
    if ($hoofdletters == TRUE) {
        $uitvoer = strtoupper($uitvoer);
    }
    //geef resultaat
```

```
    return $uitvoer;
}
```

De derde parameter is wat vreemd want de waarde wordt al bepaald in de functie, althans zo lijkt het. Dit is een zogenaamde *optionele* parameter. Je hoeft hem niet mee te geven en als je hem niet meegeeft dan is die 'by default' false.

Dus

> **welkom2(2, 'Mark', False)**
> is hetzelfde als
> **welkom2(2, 'Mark')**

Apart bestand

Als je veel functies hebt dan kun je die in een apart bestand zetten dat houd de boel overzichtelijk en maakt samenwerking ook makkelijker, omdat iedereen aan zijn eigen set functions kan werken.

Stel je zet de functie uit het laatste voorbeeld in een apart bestand met de naam mijnFuncties.php, dan zou de volgende code gewoon werken.

```
<?php
include_once('mijnFuncties.php');
echo welkom2(1,'Ayoub',true);
?>
```

Opdracht 6a

Maak een php file, mijnFuncties.php met de functie welkom2() uit één van de voorbeelden van hierboven.

Maak een file php16a-jouw-naam.php en include het bestand mijnFuncties.php file.

Voeg de volgende regel toe aan php16a-jouw-naam.php

```
$mijnArray=[''Nouaman','Aart','Samil','Rainee','Diego','Omer','Wessel','Jari','Max','Brian','K
ikiya'];
```

Gebruik het bovenstaande array en roep voor ***elk*** van de namen de welkom2() functie twee maal aan, één keer om de welkom-boodschap af te drukken en één keer om de Tot-ziens-boodschap af te drukken.

Gebruik een ***loop*** en roep vanuit de loop de functie op de juiste manier aan.

Dus de output ziet er zo uit:

```
Welkom Nouaman
Tot ziens Nouaman

Welkom Aart
Tot ziens Aart

Welkom Samil
Tot ziens Samil

....

Welkom Kikiya
Tot ziens Kikiya
```

--

7 Opdracht

In deze les gaan we stof herhalen. We gaan met functies werken en je leert onderdelen die je later later nodig hebt voor de eindopdracht.

Opgave 1

We hebben een functie die het volgende doet. De functie krijgt drie waarden en returned de som van de drie getallen (dus de twee input variabelen opgeteld).

Input	Output
mijnFunctie(3,2,1)	6
mijnFunctie(1,12,3)	16
mijnFunctie(0,4,0)	4
mijnFunctie(7,0,3)	10
mijnFunctie(0,0,0)	0

Maak een functie die dit doet.

En test met alle input waarden zoals hierboven aangegeven. Je kunt de volgende template gebruiken.

```
function mijnFunctie($a, $b, $c) {
   // maak hier de code voor jouw functie
}

echo mijnFunctie(3,2,1);
echo "<br>";
echo mijnFunctie(1,12,3);
echo "<br>";
echo mijnFunctie(0,4,0);
echo "<br>";
echo mijnFunctie(7,0,3);
echo "<br>";
echo mijnFunctie(0,0,0);
```

Inleveren

Lever de code in en gebruik de naam php71-jouw-naam.php

Opgave 2

We hebben een functie die het volgende doet. De functie krijgt drie waarden en returned de som van de drie getallen (dus de drie input variabelen opgeteld). Er is nu een uitzondering. Zodra er een 0 in één van de input variabelen voorkomt dan wordt de return waarde 0.

Input	Output
mijnFunctie(3,2,1)	6
mijnFunctie(1,12,3)	16
mijnFunctie(0,4,0)	0
mijnFunctie(7,0,3)	0
mijnFunctie(0,0,0)	0

Maak een functie die dit doet.

Gebruik de template van opgave 1.

Inleveren

Lever de code in en gebruik de naam php72-jouw-naam.php

Opgave 3

We hebben een functie die het volgende doet. De functie krijgt drie waarden en returned de som van de twee getallen (dus de twee input variabelen opgeteld). Er is nu een uitzondering. Zodra alle input variabelen 0 zijn wordt de return waarde 0.

Input	Output

mijnFunctie(3,2,1)	6
mijnFunctie(1,12,3)	16
mijnFunctie(0,4,0)	4
mijnFunctie(7,0,3)	10
mijnFunctie(0,0,0)	0

Maak een functie die dit doet.

Gebruik de template van opgave 1.

Inleveren

Lever de code in en gebruik de naam php73-jouw-naam.php

Opgave 4

We hebben een functie die het volgende doet.

De functie telt de drie input waarden bij elkaar op.

Is de som groter of gelijk aan 10 dan wordt de return waarde 1.
Is de som kleiner dan 10 en groter dan 0 dan wordt de return waarde 0.
Is de som 0 dan wordt de return waarde -1.

Input	Output
mijnFunctie(3,2,1)	0
mijnFunctie(1,12,3)	1
mijnFunctie(0,4,0)	0
mijnFunctie(7,0,3)	1
mijnFunctie(0,0,0)	-1

Maak een functie die dit doet.

Gebruik de template van opgave 1.

Inleveren

Lever de code in en gebruik de naam php74-jouw-naam.php

Opgave 5 - array optellen

We hebben een functie die het volgende doet.

De functie telt alle waarden van een array bij elkaar op. De som alle getallen van het array wordt door de functie gereturned.

Let op het input array kan verschillende lengtes hebben.

Input	Output
mijnFunctie([3,2,1])	6
mijnFunctie([1,12,3,1,2])	19
mijnFunctie([1,12,3,1,2,1])	20
mijnFunctie([2,4])	6
mijnFunctie([4])	4

Maak een functie die dit doet.

Je kunt een array doorlopen met een foreach-loop zoals we in les 2 hebben gehad.

Je kunt de template van opgave 1 gebruiken, maar je moet deze wel aanpassen.

Inleveren

Lever de code in en gebruik de naam php74-jouw-naam.php

Opgave 7 - voetbal

De voetbaluitslagen staan in een array.

```
$uitslagen = [ [1,3], [4,0], [0,0] ,[1,1],[0,2] ];
```

In dit lijstje heeft de club FCR (Footbal Club Royal), vijf wedstrijden gespeeld. De uitslagen waren 1-3, 4-0, 0-0, 1-1 en 0-2.

Maak een functie die met deze uitslagen het aantal punten voor FCR uitrekent.

Voor een gewonnen wedstrijd krijgt FCR 3 punten, voor gelijkspel 1 punt en voor een verloren spel 0 punten.

Uitslag	Punten
1-3	0
4-0	3
0-0	1
1-1	1
0-2	0

FCR heeft dus in dit voorbeeld 5 punten.

Maak een functie die de punten aan de hand van de uitslagen berekend.

Input	Punten
[[1,3], [4,0], [0,0] ,[1,1],[0,2]]	5
[[1,1], [0,0], [0,2] ,[1,1],[0,2]]	3
[[1,1], [0,0], [0,0] ,[1,1],[2,2]]	5
[[1,0], [1,0], [2,0] ,[1,1],[0,2]]	10

Template

```
function berekenScore($uitslagen) {
  // bereken score


  return($score);
}

echo berekenScore([ [1,3], [4,0], [0,0] ,[1,1],[0,2] ]);
echo "<br>";
```

```
echo berekenScore([ [1,1], [0,0], [0,2] ,[1,1],[0,2] ]);
echo "<br>";
echo berekenScore([ [1,1], [0,0], [0,0] ,[1,1],[2,2] ]);
echo "<br>";
echo berekenScore([ [1,0], [1,0], [2,0] ,[1,1],[0,2] ]);
```

Inleveren

Lever de code in en gebruik de naam php77-jouw-naam.php

Opgave 8 - swap

In de vorige opgave hadden we het over de uitslag van een voetbalwedstrijd. Een uitslag 1,0 betekende dat de thuisploeg (FCR) 1 doelpunt heeft gemaakt en de uit spelende ploeg had 0 punten gescoord.

Soms staat de uitslag in de verkeerde volgorde. FCR heeft met 3-2 gewonnen en de uitslag is 2-3.

Nu moet jij een functie maken die twee getallen in een array omdraait.

Input	Output
swap([1,2])	[2,1]
swap([3,2])	[2,3]
swap([1,1])	[1,1]

Template

```
function swap($uitslag) {
  // swap de twee elementen van het array


  return($swap);
}
```

```
print_r( swap($uitslag([2,1])) );
echo "<br>";
print_r( swap($uitslag([2,3])) );
echo "<br>";
print_r( swap($uitslag([1,1])) );
```

Inleveren

Lever de code in en gebruik de naam php78-jouw-naam.php

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

--

10 Challenge Voetbal

Voor deze challenge heb je kennis nodig van:

- HTML en CSS basis kennis
- gebruik van variabelen in PHP
- loops in PHP
- indexed arrays, assiociative arrays en complexe arrays (datastructuren) in PHP
- gebruik van condities (if-then) in PHP

Data structuren

Data structuren zijn manieren waarop gegevens in de computer zijn opgeslagen. In deze challenge gaan we kijken naar een datastructuur van voetbaluitslagen. Uit deze uitslagen gaan we per ploeg berekenen hoeveel punten deze club in totaal heeft.

In deze challenge bestaat de datastructuur uit een array van gespeelde wedstrijden.

Elke wedstrijd is een asscociative array, bijvoorbeeld:

```
['thuis' => 'FC Twente', 'uit' => 'Fortuna Sittard', 'uitslag'=> [1,2] ]
```

Het assiociative array heeft drie elementen. De naam van de thuisclub, de naam van de uitclub en de uitslag. De uitslag zelf is weer een (indexed) array met twee elementen.

De uitdaging

Je hebt een datastructuur $uitslagen (zie regel 3 en volgende in code). Deze datastructuur is een array van acciociative arrays waarin van een aantal gespeelde wedstrijden de eindscore staat.

Je ziet op bijvoorbeeld regel 4 dat Feyenoord tegen FC Twente heeft gespeeld en dat de uitslag 1-2 was. FC Twente heeft dus gewonnen.

De opdracht is om een lijstje te maken waarin van alle voetbalclubs het aantal punten dat zij hebben behaald wordt berekend.

De punten worden als volgt berekend;

Uitslag	Punten

Gewonnen	3
Gelijkspel	1
Verloren	0

Voorbeeld, later we FC Twente nemen.

FC Twente heeft 3 x gespeeld (regel 4, 21 en 31). FC Twente heeft 2x gewonnen en 1x verloren. FC Twente heeft dus 6 punten uit 3 wedstrijden.

De output zou dan worden:

Club	Punten	Gespeeld
FC Twente	6	3
...	...	..

Op regel 47 t/m 53 worden er 2 assioiciative arrays gemaakt en worden alle waarden op 0 gezet. Je kunt deze arrays gebruiken om de output in vast te leggen. In de tabel eronder worden de waarden afgedrukt.

```
<?php

$uitslagen=[
['thuis' => 'Feyenoord', 'uit' => 'FC Twente', 'uitslag'=> [1,2] ],
['thuis' => 'AZ', 'uit' => 'RKC Waalwijk', 'uitslag'=> [1,3] ],
['thuis' => 'PEC Zwolle', 'uit' => 'PSV', 'uitslag'=> [1,2] ],
['thuis' => 'Heracles Almelo', 'uit' => 'Sparta Rotterdam', 'uitslag'=> [1,3] ],
['thuis' => 'sc Heerenveen', 'uit' => 'Go Ahead Eagles', 'uitslag'=> [3,1] ],
['thuis' => 'FC Groningen', 'uit' => 'SC Cambuur', 'uitslag'=> [2,3] ],
['thuis' => 'Vitesse', 'uit' => 'Ajax', 'uitslag'=> [2,2] ],
['thuis' => 'Willem II', 'uit' => 'FC Utrecht', 'uitslag'=> [3,0] ],
['thuis' => 'N.E.C.', 'uit' => 'Fortuna Sittard', 'uitslag'=> [0,1] ],

['thuis' => 'Ajax', 'uit' => 'sc Heerenveen', 'uitslag'=> [5,0] ],
['thuis' => 'RKC Waalwijk', 'uit' => 'Heracles Almelo', 'uitslag'=> [2,0] ],
['thuis' => 'Fortuna Sittard', 'uit' => 'Vitesse', 'uitslag'=> [1,2] ],
['thuis' => 'Sparta Rotterdam', 'uit' => 'PEC Zwolle', 'uitslag'=> [2,0] ],
['thuis' => 'Go Ahead Eagles', 'uit' => 'Feyenoord', 'uitslag'=> [0,1] ],
['thuis' => 'SC Cambuur', 'uit' => 'Willem II', 'uitslag'=> [1,1] ],
['thuis' => 'PSV', 'uit' => 'N.E.C.', 'uitslag'=> [3,2] ],
['thuis' => 'FC Twente', 'uit' => 'FC Groningen', 'uitslag'=> [3,0] ],
```

```
['thuis' => 'FC Utrecht', 'uit' => 'AZ', 'uitslag'=> [2,2] ],

['thuis' => 'Feyenoord', 'uit' => 'PSV', 'uitslag'=> [2,2] ],
['thuis' => 'AZ', 'uit' => 'Ajax', 'uitslag'=> [2,2] ],
['thuis' => 'Vitesse', 'uit' => 'sc Heerenveen', 'uitslag'=> [1,2] ],
['thuis' => 'N.E.C.', 'uit' => 'Go Ahead Eagles', 'uitslag'=> [1,0] ],
['thuis' => 'FC Groningen', 'uit' => 'Sparta Rotterdam', 'uitslag'=> [1,2] ],
['thuis' => 'PEC Zwolle', 'uit' => 'FC Utrecht', 'uitslag'=> [1,1] ],
['thuis' => 'Willem II', 'uit' => 'Heracles Almelo', 'uitslag'=> [2,0] ],
['thuis' => 'FC Twente', 'uit' => 'Fortuna Sittard', 'uitslag'=> [1,2] ],
['thuis' => 'SC Cambuur', 'uit' => 'RKC Waalwijk', 'uitslag'=> [1,1] ],

['thuis' => 'N.E.C.', 'uit' => 'Fortuna Sittard', 'uitslag'=> [0,0] ],
];

echo "<table border=1>";
echo "<tr><th>Thuis</th><th>Uit</th><th></th><th></th></tr>";
foreach ($uitslagen as $uitslag) {
  echo "<tr>";
  echo "<td>".$uitslag['thuis']."</td>";
  echo "<td>".$uitslag['uit']."</td>";
  echo "<td>".$uitslag['uitslag'][0]."</td>";
  echo "<td>".$uitslag['uitslag'][1]."</td>";
  echo "</tr>";
}
echo "</table>";

$punten=[];
$gespeeld=[];
foreach ($uitslagen as $uitslag) {
  $punten[$uitslag['thuis']]=0;
  $punten[$uitslag['uit']]=0;
  $gespeeld[$uitslag['thuis']]=0;
  $gespeeld[$uitslag['uit']]=0;
}

echo "<table border=1>";
echo "<tr><th>Club</th><th>Punten</th><th>Gespeeld</th></tr>";
foreach ($punten as $key => $value) {
```

```
  echo "<tr>";
  echo "<td>".$key."</td>";
  echo "<td>".$value."</td>";
  echo "<td>".$gespeeld[$key]."</td>";
  echo "</tr>";
}
echo "</table>";
```

Output

De output is gesorteerd op punten en laat een lijstje zien van alle voetbalclubs en hun behaalde punten. De tabel ziet er netjes uit.

Standen in de Eredivisie

Club	Punten	Gespeeld
Sparta Rotterdam	9	3
Fortuna Sittard	7	4
RKC Waalwijk	7	3
Willem II	7	3
PSV	7	3
sc Heerenveen	6	3
FC Twente	6	3
SC Cambuur	5	3
Ajax	5	3
N.E.C.	4	4
Vitesse	4	3
Feyenoord	4	3
FC Utrecht	2	3
AZ	2	3
PEC Zwolle	1	3
FC Groningen	0	3
Heracles Almelo	0	3
Go Ahead Eagles	0	3

Aanpak

Er zijn meerdere mogelijkheden om dit aan te pakken. Hieronder wordt een mogelijke oplossing beschreven. Je hoeft deze stappen niet te volgend.

1. We moeten door de uitslagen array heen lopen om per wedstrijd te bepalen welke ploeg hoeveel punten krijgt.
   Maak dus een loop die door de uitslagen loopt.
2. In de loop maak je twee variabelen $*thuisClub* en $*uitClub*. Je geeft deze de waarden van de thuis en uitclub.

   -> **Test** of je deze twee variabelen de juiste waarde heb gegeven door ze af te drukken.
3. In de loop maak je nog twee variabelen $*thuisScore* en $*uitScore*. Je geeft deze de waarden van het aantal doelpunten dat de thuis- en uitclub hebben gescoord.

   -> **Test** of je ook deze twee variabelen de juiste waarde heb gegeven door ze af te drukken.
4. Nu zijn er drie mogelijkheden:
   De *$thuisScore* > $*uitScore*, de $*thuisClub* krijgt 3 punten
   De $*uitScore* > $*thuisScore* , de $*uitClub* krijgt 3 punten
   De $*thuisScore* == $*uitScore*, de $*thuisClub* en $*uitClub* krijgen beiden 1 punt.

   Maak de juiste if's en vul het punten array.

   Denk eraan dat de key van het punten array de naam van de club is en die staat in $*thuisClub* en $*uitClulb*.

   -> **Test**, bedenk een manier om te testen of de club de juiste punten krijgt.
5. Voor de thuis- en uitclub moet je het aantal gespeelde wedstrijden met één ophogen. Hiervoor hoog je de juiste waarde op in het $*gespeeld* array. Ook in het $*gespeeld* array is de key de naam van de club.
6. Test de gehele berekening.

Eisen

1. Zorg ervoor dat alle uitkomsten kloppen (60 punten).
2. Zorg ervoor dat de output gestyled is (gebruik styles en classes). Maak er wat moois van, overtuig de klant dat jij iets moois kan maken (20 punten)

3. Zorg ervoor dat de HTML code klopt en volledig is. Dus gebruik <html>, <head> etc (10 punten).
4. De code klopt, het inspringen en uitlijnen is goed en je hebt duidelijk commentaar toegevoegd om uit te leggen wat je doet (10 punten).

Inleveren

1. Screendump van jouw gehele browser met het resultaat.
2. De PHP code.