PHP L3 Voetbal

PHP Level 3 met de voetbal challenge

- 0 Inleiding
- 01 De Som
- 02 Optellen en de nul
- 03 Tussen 0 en 10?
- 04 Array optellen
- 05 Voetbal
- 06 Swap valules
- 07 Druk Ass. array af
- 08 Key and value
- 09 Maak een ass. array
- 10 Arry from database
- 11 Thuisploeg
- 12 Challenge Voetbal

0 Inleiding

Verderop krijg je de *PHP voetbal challenge*.

Om de PHP Voetbal challenge te kunnen maken moeten we de stof uit PHP level-2 goed beheersen en bovendien moeten we nog kennismaken met één nieuw type array, het assiociatieve array.

In deze module gaan we dus eerst nog een paar dingen bespreken voordat we met de challange aan de slag gaan.

Deze zaken zijn een ***noodzakelijke*** voorbereiding om de voetbal-challenge te kunnen maken.

Verder is het leren werken met assiociative arrays een goede voorbereiding op de lessen over PHP Frameworks Yii en Laravel.

Lees de theorie goed door en bekijk de filmpjes. Ga niet direct naar de opdrachten want dat kost uiteindelijk meer tijd. Begrijp je de theorie niet, vraag dan een docent om hulp.

Succes!

Terugblik

We hebben al heel wat over arrays gesproken.

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'];

--

01 De Som

De Som

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).

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

Kijk naar dit schema en kijk of je begrijpt wat er gebeurt. Telkens wordt de input getransformeerd (verandert). De input 3,2,1 wordt bijvoorbeeld verandert in 6, de input 1,12,3 in 16 etc.

Maak een functie die dit doet.

En test met alle input waarden zoals hierboven aangegeven. Gebruik de volgende template:

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

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

```
echo "<br>";
echo "mijnFunctie(7,0,3): ";
echo mijnFunctie(7,0,3);
echo "<br>";
echo "mijnFunctie(0,0,0): ";
echo mijnFunctie(0,0,0);
```

Klopt je output?

Je output moet er dus als volgt uit zien:

image-1666512677545.png

Inleveren

1. de code in en gebruik de naam php71-jouw-naam.php
2. een schermafdruk van de test waarbij alle juiste waarden worden afgedrukt zoals in het voorbeeld.

--

02 Optellen en de nul

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 en plaats testcode in je code die alle bovenstaande situaties test. Gebruik eenzelfde soort template als in de vorige opgave.

De output ziet er dan als volgt uit:

image-1666512923240.png

Tip: gebruik voor de situatie met een 0, een conditional statement en een extra return.

Inleveren

1. Lever de code in en gebruik de naam php72-jouw-naam.php
2. een schermafdruk van de test waarbij alle juiste waarden worden afgedrukt zoals in het voorbeeld.

03 Tussen 0 en 10?

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. Test alle input waarden weer net als in de vorige opgave.

De output ziet er als volgt uit:

image-1666512997262.png

Inleveren

1. Lever de code in en gebruik de naam php73-jouw-naam.php
2. Een schermafdruk van je gehele browser waarin je laat zien dat je code werkt (zoals in het voorbeeld)

--

04 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 ge-returned.

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

1. Lever de code in en gebruik de naam php74-jouw-naam.php
2. een schermafdruk van de test waarbij alle juiste 5 waarden worden afgedrukt zoals in het voorbeeld.

--

05 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

In de eerste regel zijn er 5 wedstrijden gespeeld. De eerste wedstrijd is verloren (1-3), de tweede is gewonnen (4-0), de derde is gelijk gespeeld (0-0), de vierde is ook gelijkgespeeld (1,1) en de vijfde is verloren (0,2).
Dus er is één wedstrijd gewonnen en er zijn twee wedstrijden gelijk gespeeld. Dat geeft dan dus 3+1+1 5 punten.

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

1. Lever de code in en gebruik de naam php75-jouw-naam.php
2. Een schermafdruk waarin je laat zien dat je je code test met de bovenstaande test cases.

--

06 Swap valules

Stap 1

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]

De input en output zijn beiden een array.

Template

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


  return $swap;
}

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

Output

Je ziet als het goed is dat je een array terug krijgt waarbij de waarden zijn omgedraaid. De output ziet er dan als volgt uit:

```
Array
(
    [0] => 2
    [1] => 1
)

Array
(
    [0] => 2
    [1] => 3
)

Array
(
    [0] => 1
    [1] => 1
)
```

Stap 2

Je hebt de vorige opgave uitgevoerd? Begin dan pas aan deze stap.

Nu heb je een array van arrays met uitslagen.

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

Je gebruikt de code van stap 1, maar zet er een **loop** omheen.

De output wordt dus:

```
Array
(
    [0] => 3
    [1] => 1
)

Array
(
    [0] => 0
    [1] => 4
)
```

```
Array
(
    [0] => 0
    [1] => 0
)

Array
(
    [0] => 1
    [1] => 1
)

Array
(
    [0] => 2
    [1] => 0
)
```

Succes!

Inleveren

1. Lever de code van stap 1 in en gebruik de naam php761-jouw-naam.php
2. Lever de code van stap 2 in en gebruik de naam php762-jouw-naam.php
3. een schermafdruk van de test van stap 2, waarbij alle juiste waarden worden afgedrukt (zoals in het voorbeeld bij stap 2).

--

07 Druk Ass. array af

Neem deze template en vul de code aan zodat je het hele associatieve array afbeeldt.

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

De output moet er als volgt uit komen te zien (dus elke waarde op een nieuwe regel).

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

Inleveren

1. Lever de code in en gebruik de naam php77-jouw-naam.php
2. Een schermafdruk met de juiste output (mag in ander font als in het voorbeeld)

08 Key and value

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.aspLinks to an external site.

Gebruik de foreach loop om alle keys en values af te drukken. De output moet er precies zo uitzien als in deze opdracht staat (let dus op de komma's en de regels).

Inleveren

1. Lever de code in en gebruik de naam php78-jouw-naam.php
2. Lever een schermafdruk van de output in.

09 Maak een ass. array

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).

Eis: gebruik een loop voor het afdrukken van de waarden. Druk alleen de waardes netjes leesbaar af (druk de keys niet af).

Voorbeeld output:

```
Danish
Brown
16 november 2004
Amsterdam
```

Inleveren

1. Lever de code in en gebruik de naam php79-jouw-naam.php.
2. Een schermafdruk van de output in de browser.

10 Arry from database

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 element (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

--

11 Thuisploeg

Gebruik het assiociative array (van de vorige opdracht);

```
$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 uitslagen 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

--

12 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.

image-1657872023212.png

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.