PHP 2

Deze lessen zijn inhaallessen die tijdens de 3de priode PHP zijn tussengevoegd.

- Overview
- PHP en HTML
- Variabelen en Data Types
- Arrays
- Loops
- Opgaven Arrays en Loops
- If en condities
- Associative Arrays - 1
- Assiociative Arrays - 2
- Opgaven Arrays/Sort
- Debugging
- Test

Overview

Intro

- PHP en HTML
- PHP Comments

Data types

- Variabelen
- Type Casting
- Math
- Arrays
- Associative Arrays

Control Structures

- it - then -else
- comparison and logic operators
- switch
- while
- for
- foreach

Functions

- functions
- parameters
- return values
- Scope (Global v Local)

- Constants

Build in functions

- math
- string
- array
- date

Forms

- Database PDO

PHP Web

- post and get super global
- $_server
- Cookies
- Sessions

OOP

...

PHP en HTML

In deze les keer je hoe je HTML en PHP combineert

PHP in HTML en HTML in PHP

```
<?php
// de variable $today de (Engelse) dag van de week
    $today = strftime("%A");
 ?>

 <h1>Today it is <?php echo $today ?></h1>

<?php echo "<h1>Today it is $today </h1>"; ?>

<?php echo "<h1>Today it is ".$today."</h1>"; ?>
```

Voer de code uit. Zie je dat je drie keer hetzelfde resultaat hebt? Wat zou jouw vookeur hebben en waarom?

Quotes

```
<?php

$name="Max";

echo "\"$name\" Laptop\nHoera!";
echo "<br>";
echo 'Max\'s Laptop\nHoera!';
echo "<br>";
echo '$name Laptop';

?>
```

- Tussen dubbel quotes worden variabelen geparsed.
- Tussen singel quotes worden variabelen niet geparsed, een $ is gewoon een dollar.

- Tussen dubbel quotes kun je verschillen escape sequences gebruiken

```
\’ – To escape ‘ within single quoted string.
\” – To escape “ within double quoted string.
\\ – To escape the backslash.
\$ – To escape $.
\n – To add line breaks between string.
\t – To add tab space.
\r – For carriage return.
```

- Tussen '' kun je alleen de singel quote escapen \'

Heredoc

```
<?php
$name="Max";

print <<< PINDA
<p>
Dit is een lang verhaal dat gaat over een ....<br/>
...het gaat ook over een andere....<br/>
en tot slot gaat het ook over het slot.<br/>
Ondertekend: \"$name\"  <br>
<hr>
</p>
PINDA;
?>
```

Een *here document* (heredoc) is eigenlijk een hele lange echo waarbij alleen variabelen worden gerenderd.

> Het teken dat het einde van een heredoc markeert (in dit voorbeeld PINDA) moet altijd aan het begin van de regel staan en eindigen met een ; Je mag deze regel dus niet inspringen!

Opgaven

1. Schrijf een regel code die het volgende in de browser afdrukt: `Are you sure to delete c:\*.*?`

2. Nu heb je een variabele met jouw naam (zoals in het voorbeeld de voorbeelden hierboven) en je wilt afdrukken:
   `Max, weet je zeker dat je c:\*.* wilt weggooien?`
   De naam Max wordt daarbij vervangen door jouw eigen naam met een variabele!

3. Maak een heredoc en gebruik weer een variabele $naam. Druk het volgende in een heredoc af:
   `"In de variabele $naam staat een waarde Max"`
   Het woord Max aan het einde van de zin komt dus uit een variable met je eigen naam.

4. Bestudeer op phpnet op welke drie manieren je commentaar aan je code kunt toevoegen. Gebruik alle drie de manieren op code toe te voegen aan de code die je bij vraag 3 hebt gemaakt.

Variabelen en Data Types

In deze les gaan we kijken naar wat variabelen en datatypen zijn en hoe PHP hier mee omgaat.

Variabelen

Variablen zijn 'doosjes' waarin we gegevens in op kunnen slaan. We kunnen verschillende soorten gegevens in zo'n 'doosje' stoppen; getallen in de vorm van integers, floats en ook woorden die we strings noemen. Ten slotte onderscheiden we booleans, 1 of 0, true of false (waar of niet-waar).

```
<?php
$tekst = 'Hello World!'; // String
$leeftijd = 20; // Integer
$prijs = 135.75; // Float
$check = true; // Boolean
?>
```

Loosely typed

In somigen programmeertalen moet je variabelen eerst declareren. Je geeft daarmee van te voren aan welke variablen je gaat gebruiken en welk type gegevens je in deze variabele gaat opslaan. In PHP hoeft dat niet. PHP noemen we daarom een "*loosely typed language*". We hoeven dus niet van te voren aan te geven voor soort gegevens we in een variabele gaan opslaan. PHP probeert zo goed mogelijk zelf te bepalen welk soort gegevens er in een variabale staan. Meestal gaat dat goed maar we zullen zien dat dat niet altijd zo is.

Type Casting

We kunnen bewerkingen doen met variabelen, bijvoorbeeld optellen.

```
<?php
$a="12";
$b=13;
echo $a+$b;
?>
```

> Vraag 1: Welk type hebben de variable $a en $b?

```
<?php
$a=12;
$b="13";

echo 'A='.$a;
echo '<br>';
echo 'B='.$b;
echo '<br>';
echo $a+$b;
echo '<br>';
echo 'A='.$a+$b;
?>
```

> Vraag 2: Voer de bovenstaande code uit. Wat gebeurt er? Wat is er mis met regel 11?

Je kunt dus PHP forceren om een variabele, of een uitkomst van een berekening een bepaald type te geven. Bijvoorbeeld:

```
echo 'A='.(string)($a+$b);
```

Hiermee vertel je PHP dat het resultaat van $a+$b moet worden omgezet in een type string. En om die manier kun je de uitkomt via echo afdrukken. Dit noem je *type casting*.

> Tip: met de echo gettype($a); kun je het type dat PHP aan een variabele toekent laten zien.

PHP zet dus zelf het type van een variable om. Maar in het voorbeel dhebben we gezien dat dat niet altijd goed gaat en dan moeten wij in PHP aangeven wat er moete gebeuren.

Stel je wilt 13/2 berekenen. De uitkost is 6.5 en dat is een float. Maar stel dat je altijd een intereger wilt hebben dan zal je dat moeten aangeven met een type cast:

```
echo (int)($b/$a);
```

Vraag: wat gebeurt er als de tweede set haakjes weg laat dus: `echo (int)$b/$a;`

> Vraag 3: Leg uit wat er gebeurt als je dit laatste statement uitvoerd.

Math

Zoek op en beschrijf van de volgende statements wat ze doen en wat de uitkomst is.

abs(-3.4);	
max(3,4,2,0,-2)	
min(3,4,2,0,-2)	
rand(1,25)	
echo 3+2*4	
echo 3**2	

$a++ neemt de variable $a en verhoogd hem. Stel $a=3 en je doet echo $a++ wat verwacht je en wat zie je?

Voer deze code uit.

```
<?php
$i=3;
echo $i++;
echo $i;
?>
```

> Vraag 4: leg uit hoe deze code werkt.

```
<?php
$i=3;
echo ++$i;
echo $i;
?>
```

> Vraag 5: leg uit hoe deze code werkt.

In plaats van $a++ kun je ook $a-- gebruiken.

> Vraag 6: stel je vervangt ++ in -- in de vorige code. Voorspel wat er gebeurt en test of het klopt.

Arrays

In deze les ga je leren wat een array is en hoe je multi dimentionale arrays kunt gebruiken.

TODO: http://www.phpboek.net/phpboek-6-arrays.php

Arrays

Een array is een type variable waarin meerdere gegevens zitten. Een array bestaat uit elementen. Elk element is eigenlijk een aparte variabele. **Het eerste element is element 0**, het tweede element is element 1, ...en het 10de element is element 9. In een array mag je verschillende datatypes door elkaar gebruiken. Je kunt een array op veschillende manieren declareren.

Arrays worden veel gebruikt. Als resultaten uit een database worden gehaald kunnen die in een array worden gezet.

```
<?php
$arrayA = [1,2,3,4,5,'a'];
$arrayB = array(1,2,3,4,5,'a');
?>
```

Je kunt een array element gebruiken door de array variabele met een index: `$arrayA[0], of $arrayA[1]`

Er zijn veel PHP functies die iets met een array doen, een paar voorbeelden

count($array)	return het aantal array elementen.
array_shift($array)	return het eerste element uit het array en haal die eruit.
array_pop($array)	return het laatste element uit het array en haal die eruit.
array_push($array, ...)	voeg een of meer alementen toe aan het eind van het $array.
in_array('abc', $array)	Zoek of 'abc' in het array staat.

Opdracht

Wat doet de volgende code? Voorspel eerst welke vier maanden er gaan worden afgedrukt en controleer het dan of dat inderdaad zo is.

```
<?php
$maanden = array ('Januari', 'Februari', 'Maart', 'April', 'Mei', 'Juni', 'Juli',
'Augustus','September', 'Oktober', 'November', 'December');

echo $maanden[1];
echo $maanden[count($maanden)-1];
echo array_shift($maanden);
echo array_shift($maanden);
?>
```

Arrays van Arrays

Zoals uitgelegd kan er van alles in een array zitten; integers, strings, maar ook arrays. Een element van een arrays kan dus op zich zelf weer arrays zijn. Hoe werkt dat?

Stel je hebt vier arrays; $zomer, $herfst, $winter en $lente. Wat er in deze arrays staat is eigenlijk niet zo van belang. Stel we willen een nieuw array maken waarin deze vier arrays staan, dan kan dat als volgt:

`$jaar_2014=[ $zomer, $herfst, $winter, $lente ]`

In elke element zit nu een array want de variable $zomer $hefst, $winter en $lente zijn immers op zichzelf allen ook een array.

Stel we hebben een aantal jaren op deze manier gedefinieerd dus $jaar_2010 en $jaar_2011 zijn allen arrays met elk vier arrays. Nu kunnen we deze jaren op zichzelf ook weer in een array zetten:

`$jaren = [ $jaar_2010, $jaar_2011 ];`

We hebben nu een array van een array van een array. Drie arrays in elkaar.

We hebben het nog niet gehad maar er in het array $zomer, $herfst, $winter en $lente staan telkens 3 getallen die de gemiddelde temeratuur van de betreffende maand voorstellen.

Dus we hebben een array met 5 jaren en elk jaar is een array met vier seizoenen en elk seizoen heeft vier temperaturen.

Schematisch:

Jaar array	Seizoen	Temperatuur
2010	zomer	[21, 22, 24, 20]
	herfst	[20,19,19,17]
	winter	[15,12,8,5]

	lente	[7,10,18,18]
2011	zomer	[20, 22, 23, 20]
	herfst	[20,19,21,17]
	winter	[15,12,9,8]
	lente	[7,10,14,18]
etc. etc.		

Opdracht

Maak dit beschreven $jaren array in PHP.

Als je dit array hebt gemaakt en je wilt de temperatuur uit de eerste maand van de winter uit 2010 weten, hoe doe je dit?

Er zijn twee manieren:

> Allereerst kun je een arrays uit het array halen, met
>
> `$ditjaar = $jaren[0]`
>
> $ditjaar heeft nu de waarde van het eerste element van $jaren en dat is jaar 2010. Nu willen we de winter hebben:
>
> `$ditseizoen = $ditjaar[2]`
>
> $ditseizoen wordt nu gevuld met het 3de element van het array $ditjaar (en $ditjaar was 2010).
>
> Als laatste kunnen we nu de temperatuur van de eerste maand uit de winter halen:
>
> `$temperatuur = $ditseizoen[0]`

Je kan het ook in één stap doen en dat is de tweede manier:

> `$jaren[0][2][0]`

Herken je de nummers? Wat hier staat is: neem uit het array jaren het 1ste element en daarvan het 3de element en daarvan het 1ste element.

Oefeningen Arrays

1. Hoe druk je het 2de element af van het array `$myArray=[10,20,30,40,50,60]` ?
2. Wat is de waarde van het 1ste element van het array `$myArray=[1,2,3,4,5,6,7]` ?

3. Wat wordt er afgedrukt `echo count([1,2,3])` ?
4. Hoe druk je het laatste element af van het array `myArray=[1,2,3,4,5]` ?

Loops

In deze les ga je de for-loop, de while-loop en de do-while loop leren.

Loops komen in alle programmeertalen voor. Zij zorgen ervoor dat je een bepaalde handeling tig keer kunt uitvoeren zonder dat je hele lange en bijna dezelfde code moet intypen.

For-loop

Stel je wil 10x een bepaalde regel printen. Daarvoor kun je de regel in een loop zetten:

```
<?php
for($i=0; $i<10; $i++) {
echo "De waarde van \$i = $i<br>";
}
?>
```

Voor deze code uit en kijk wat er gebeurt.

De for-loop wordt een bepaald aantal keren uitgevoerd. Dit aantal keer stel jij als programmeur vast. De loop kent 3 fases: begin, de uitvoering en het eind.

Begin

De loop begint met $i=0 (dit het eerste argument van de for-loop).
Voordat de loop mag worden uitgevoerd wordt de vergelijking in het tweede argument uitgevoerd; alleen als deze vergelijking true oplevert mag de eerste *iteratie* (ronde) beginnen.

Uitvoering

De loop gaat lopen en bij elke *iteratie* (=volgende ronde) wordt eerst het derde argument uitgevoerd. In dit geval dus $i++. Daarna wordt gekeken of de vergelijking in het tweede argument nog true oplevert. Als dat zo is dan gaat de loop de volgende iteratie in.

Eind

Zodra de vergelijking in het tweede argument false oplevert stopt de loop.

Opdrachten

Schrijf van alle volgende loops op hoe vaak die wordt uitgevoerd, bedenk ook waarom.

1. `for($i=5; $i<=10; $i++){...`
2. `for($i=100; $i<102; $i++){...`
3. `for($i=10; $i>0; $i--){...`
4. `for($i=0; $i<10; $i=$i+2){...`
5. `for($x=0; $i<10; $X--){...`

While-Loop

De while loop is eigenlijk eenvoudiger uit te leggen. Hij kent één vergelijking en zolang die true is blijft de loop doorlopen.

```
$i=0;
while($i<10){
$i++;
    ...
}
```

Met deze while-loop doe je eigenlijk hetzelfde als in de for loop: `for($i=0; $i<10; $i++)`

Deze loop is iets 'gevaarlijker' omdat de drie fases van de loop niet per sé bij elkaar staan. Dat is ' gevaarlijk' omdat je dan eerder het overzicht verliest en misschien het ophogen (in het voorbeeld regel3) vergeet. Ook kan het minder overzichtelijk zijn.

Bij een while-loop moet je net als bij een for-loop altijd nadenken over de 3 fasen:

1. hoe begint de loop (in het voorbeeld $i=0);
2. wat gebeurt er bij elke iteratie en;
3. hoe stop de loop?

Stel je hebt een functie die een row uit de database haalt, deze functie heet getRow() en deze functie returned de regel. Als er geen regels meer zijn dat wordt er 0 teruggegeven. Je kunt nu een while-loop gebruiken om alle regels af te drukken:

```
<?php
$dezeRegel = getRow();

while($dezeRegel<>0){
echo $dezeRegel;
    $dezeRegel=getRow();
}
?>
```

Het is gebruikelijk om dit op deze manier uit te voeren, maar het kan ook in een for-loop. Welke methode vind jij beter leesbaar?

```
<?php
for( $dezeRegel = getRow(); $dezeRegel<>0; $dezeRegel=getRow();){
echo $dezeRegel;
}
?>
```

In sommige gevallen wil je eindeloze loop maken. Een while loop kun je eenvoudig eindeloos maken: `while(true){...`

Do-While

De do-while loop is een variant op de while-loop. In deze loop staat de vergelijking aan het eind in plaats van in het begin zoals bij de while-loop en de for-loop.

```
<?php
$i = 0;
do {
    echo $i;
} while ($i > 0);
?>
```

Wat denk je dat deze code doet? Probeer deze code. Als je deze code uitvoert, dan zie je dat de loop 1x uitgevoerd wordt en dat terwijl $i nooit groter dan 0 is (en dus de vergelijking nooit true oplevert). Een Do-While loop wordt dus altijd minimaal 1x uitgevoerd. Dat geldt niet voor de ' gewone' while-loop of voor de for-loop.

Stroomdiagrammen

Hieronder zie je twee stroom-diagrammen. Welke hoort bij de while en welke bij de do-while? Welke zou het beste passen bij een for-loop?

Image result for do while loop php	Image result for while loop in php

Break

Met een break statement spring je uit de huidige loop. Je gaat direct naar het eind en begint met het statement dat vlak na de } staat aan het einde van de loop.

Continue

Met het continue statement stop je met de huidige iteratie en ga je naar het begin van de loop om de volgende iteratie te beginnen.

Opgaven Arrays en Loops

We gaan nu oefenen met arrays en loops.

We hebben nu arrays en loops leren kennen. Met deze combinatie kun je veel handige dingen coderen.

Eerst een voorbeeld:

```
<?php
$directors = array( "Alfred Hitchcock", "Stanley Kubrick", "Martin Scorsese", "Fritz Lang" );
for($i=0; $i<count($directors); $i++){
echo $directors[$i]."<br>";
}
?>
```

Een paar zaken zijn belangrijk:

1. Het eerste array element is element 0, de loop begint dus met 0!
2. De count() van het array is het aantal elementen, in dit voorbeeld 4.
3. De loop stopt bij $i=4 dus de laatste iteratie is met $i=3. Even checken....bestaat element $directors[3]? En is het inderdaad het laatste element?

Nu jij:

1. Maak een array met de dagen van de week: maandag, dinsdag,... etc. Maak een loop en druk de dagen onder elkaar af.
2. Gebruik het array uit de vorige opdracht maar druk nu het volgende af: "Dag 1 is maandag", "Dag 2 is dinsdag", etc.
3. Druk nu de dagen van de week af in omgekeerde volgorde: dus begin met zondag, dan zaterdag, etc.
4. Gebruik nu een loop om de tafel van 7 af te drukken, dus 1 X 7 = 7, 2 X 7 = 14, etc etc. Vervang de 7 voor een variabele en zet deze boven in je code. Druk nu de tafel van 127 af.

If en condities

In deze les leer je met condities om gaan.

Als je code conditioneel wilt uitvoeren kun je een if statement gebruiken. Stel je wilt alleen iets printen als de waarde van $i 0 is.

Dan doe je:

```
<?php
if($i == 0) {
echo "Iets";
}
```

je ziet dat de vergelijking tussen ( ) staat. En je ziet dat er niet = maar == staat. In PHP gebruik je = voor het toekennen van een warde aan een variabele, zoals $a=1. Bij een vergleijking gebruik je altijd ==.

De vergleijking resulteert in waar of niet waar; *true* of *false*.

Naast == kun je ook andere vergelijkingen maken. De belangrijkste zijn.

==	is gelijk aan
>	groter dan
<	kleiner dan
>=	groter of gelijk
<=	kleiner of gelijk
<> of !=	ongelijk

Je kunt ook een conditie/vergelijking maken en meerdere zaken testen. Je wilt bijvoorbeeld code uitvoeren als twee variabelen beide 0 zijn. Dan doe je:

`if ( $a==0 and $b==0) { ....}`

Je kunt ook een or gebruiken en dat betekent dan dat één van de vergelijking waar moet zijn, bijvoorbeeld:

`if ( $a==0 or $b==0) { ....}`

Een vergelijking levert altijd `true` of `false` op. Bij true wordt het code blok achter de if uitgevoerd en bij false wordt het code blok overgeslagen.

Soms worden true en false ook als 1 of 0 weergegeven.

Dus `($a==$a)` en `true` en `1` zijn in een vergelijking allemaal hetzelfde.

> Vraag 1: stel $a=0 en $b=1 en je hebt een if($a*$b), levert dit true of false op en wordt het code blok van de if dan wel of niet uigevoerd?

Een if kan ook een else hebben:

```
<?php
if ( $a==0 ) {
echo "true";
} else {
echo "false";
}
```

Het code blok in de else wordt uitgevoerd als de vergelijking in de if, false is.

Ten slotte heb je ook nog een elseif.

```
<?php
$t = date("H");

if ($t < "10") {
    echo "Have a good morning!";
} elseif ($t < "20") {
    echo "Have a good day!";
} else {
    echo "Have a good night!";
}
```

In deze code wordt $t het uur (van de dag). Als de eerste vergelijking true is dan wordt de eerste echo uitgevoerd en is alles klaar. Als de eerste vergelijking false is dna wordt de tweede vergelijking uitevoerd, als deze ook false is dan zou er een tweede elseif kunnen staan daar waar nu de else staat. In dit geval staat er gewoon een else.

De bovenstaande code kan ook anders worden geschreven. Zie de code hieronder. Meestal is de code met de eslsif beter leesbaar.

```
<?php
$t = date("H");
```

```
if ($t < "10") {
    echo "Have a good morning!";
} else {
if ($t < "20") {
echo "Have a good day!";
} else {
echo "Have a good night!";
    }
}
```

Opgave

Bepaal van de volgende condities of ze true of false zijn. Dus als het een if statement zou zijn zou dan het code blok na de if worden uitgevoerd (true) of niet (false)?

$a=0 en $b=1

Opgavee	Vergelijking	true of false?
1	($a==1)	
2	($b==1)	
3	($a==1 or $b==1)	
4	($a==0 and $b<>1)	
5	($a<>1 and $b<>0)	
6	($a+$b==1 and $a*$b==0)	
7	($a+$b==1 or $a*$b==0)	
8	($a)	
9	($b)	
10	($a and $b)	
11	($a or $b)	
12	($a or $b*$a)	
13	(($a+2)*10>5 and $a)	

--

Associative Arrays - 1

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

Associative arrays zijn belanrijk want resultaten van een query uit de database worden door PDO vaak als associcative 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 assiociative array wordt ook wel key genoemd.

> Een index in een 'gewoon' array heet **key** in een assiociative 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 assiociative array, en is $key de key (zeg maar de index) en $value de waarde.

Opdracht

We hebben code gemaakt die het volgende afdrukte:

```
PHP:5.5
Java:6.5
Engels:7.0
Nederlands:6.5
```

```
Veilig Programmeren:8.0
```

Verander deze code zodat deze met de loop `foreach ($array as $key => $value)` werkt en zodat de code in een tabel wordt afgedrukt. Dus ongeveer zo:

Vak	Cijfer
PHP	5.5
Engels	7.0
Nederlands	6.5
Veilig Programmeren	8.0

--

Assiociative Arrays - 2

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 assiociative arrays. Weet je nog dat we arrays van arrays konden maken? Dat doen we nu ook in de volgende vorm (schematisch):

```
[ [assiociative array], [assiociative array], [assiociative array], [assiociative array],
[assiociative array] ]
```

We hebben dus array, $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

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 ziet er zou dus de inhoud van een tabel kunnen zijn.

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

--

Opgaven Arrays/Sort

Opgave 1

We gaan oefenen met loops, arrays assiociative arrays en de verschillende array sort function.

Maak het volgende array

```
<?
$ceu = array( "Italy"=>"Rome", "Luxembourg"=>"Luxembourg", "Belgium"=> "Brussels",
"Denmark"=>"Copenhagen", "Finland"=>"Helsinki", "France" => "Paris", "Slovakia"=>"Bratislava",
"Slovenia"=>"Ljubljana", "Germany" => "Berlin", "Greece" => "Athens", "Ireland"=>"Dublin",
"Netherlands"=>"Amsterdam", "Portugal"=>"Lisbon", "Spain"=>"Madrid", "Sweden"=>"Stockholm",
"United Kingdom"=>"London", "Cyprus"=>"Nicosia", "Lithuania"=>"Vilnius", "Czech
Republic"=>"Prague", "Estonia"=>"Tallin", "Hungary"=>"Budapest", "Latvia"=>"Riga",
"Malta"=>"Valetta", "Austria" => "Vienna", "Poland"=>"Warsaw") ;
```

en maak een PHP script dat de volgende output genereert:

```
The capital of Netherlands is Amsterdam
The capital of Greece is Athens
The capital of Germany is Berlin
```

Let op dat je het assiociative array sorteerd op 'value' (Amsterdam, Athens, Berlin,.....).

Zoek hiervoor de juiste functie op: w3schools

Opgave 2a

We hebben een lijst getallen: 78, 60, 62, 68, 71, 68, 73, 85, 66, 64, 76, 63, 75, 76, 73, 68, 62, 73, 72, 65, 74, 62, 62, 65, 64, 68, 73, 75, 79, 73

Stel je wilt het gemiddelde van de vijf hoogste waarden van deze reeks berekenen. Hoe doe je dat?

Eerst maken we het probleem kleiner door het in stapjes op te delen:

1. Bepaal de vijf hoogste waarden.
2. Bereken het gemiddelde van deze vijf waarden.

We hebben nu dus twee problemen, laten eerst naar probleem 1 kijken en dat weer onderverdelen in nog kleiner stapjes.

De vijf hoogste waarden kunnen we bepalen door het array eerst te sorteren, dus als we probleem 1 verdeer opdelen dan hebben we de stappen:

1.1 Sorteer het array aflopend (hoogste waarden vooraan).
1.2 Neem de eerste 5 elementen van het array.

1.1 kunnen we doen met de juiste sorteer functie, zie w3schools
1.2 kunnen we doen met een for-loop; we itereren door element 0,1,2,3 en 4.

Doe dit en druk de hoogste 5 waarden af om te controleren of we inderdaad de vijf hoogste waarden krijgen.

Nu wordt stap 2 (het gemiddelde bepalen) niet meer zo moeilijk. In plaats van het uitprinten van de vijf hoogdte waarden tellen we deze bij elkaar op en als we dat gedaan hebben dan delen we de uitkomt door 5.

Waarom? Omdat je het gemiddelde van vijf getallen als volgent berekent. Stel de hoogste waarden zijn 20,19,18,17,15,10 dan bereken je het gemiddelde door de berekening (20+19+18+17+15+10)/5.

Maak het hele PHP programma af. Zorg nu dat de 5 een variabele is ($aantal) zodat je ook eenvoudig het gemiddelde kan berekenen van de 4 hoogste waarden, of de 6 hoogste waarden of de 7, of 8.,....

Opgave 2b

Wat gebeurt er als je $aantal= 30 maakt?

Beveilig het PHP programma zodat als je de variabele $aantal niet zo groot kan worden dat het PHP programma een foutmelding geeft.

Opgave 3

Pas de code aan die je hebt gemaakt bij opgave 2b zodat je het gemiddelde van de ***laagste*** vijf waarden berekend.

--

Debugging

In deze les gaan we code debuggen

In de onderstaande blokjes code zitten meerdere fouten, haal deze eruit en maak de code werkend.

Opgave 1

```
<?php
for($i=0; $i<10; $i++)
    echo $i."<br>"
}
```

Opgave 2

```
<?php
$a=1;
if ($a) {
echo "Hello darkness my old friend!"
else {
echo "There is no way this will go around";
}
```

Opgave 3

```
<?php


$string="Hallo";
echo <<< TEKST
    $string allemaal
TEXT;

echo "<br>";
echo 'Nog een keer $string';
```

Opgave 4

```php
<?php
$myArray=["a","b","c","d","e"];
array_push($myArray,k,l,m,n,o);
echo strtoupper(array_shift($myArray));
echo strtoupper(array_pop($myArray));
```

Opgave 5

```php
<?php
$myArray=[ [ titel => 'The Adventures of Olm', 'price' => 9.89 ],
        [ titel => 'Speak Louder', 'price' => 12.49 ],
        [ titel => 'Advanced PHP', 'price' => 34.99 ] ];

foreach ($myArray as $item) {
    foreach($item as $key => $value) {
        echo $key.':'.$value."<br>"
    }
    echo <hr>;
}
```

Opgave 6

Deze code geeft geen foutmelding, maar de browser blijft hangen. Waarom? Verbeter de code.

```php
<?php
for ($i=0; $i<10; $i++) {
    if ($i=1) {
         echo $i."<br>";
    }
}
```

Opgave 7

Deze code geeft foutmelding, en de browser blijft (mogleijk) hangen. Waarom? Verbeter de code.

```php
<?php
for($nummer=0; $nummer<10; $number++)
□if ($number%2==0) {
□□echo "Just antoher line....";
   □□echo "...and the even number is....".$number."<br>";
```

```
}
```

Opgave 8

Als output moet er een goede tabel komen met twee kolommen. In kolom 1 staat 1 t/m 10 en in kolom 2 staat 2,4,6,8,10

```
<?php
echo "<table><br>";
for($i=1;$i<10;$i=$i+1)
    echo "<tr>";
    echo "<td>".$i."</td><td>".$i*2."</td>";
    echo "</tr>";
}
echo "</table><br>"

?>
```

Opgave 9

```
function setMyCookie() {
    echo "<pre>";
    print_r($_COOKIE);

    echo "Set cookie";
    if ( isset( $_COOKIE['counter'] ) )
        $counter = $_COOKIE['counter'] + 1;
    } else {
        $counter = 0;
    }
    setcookie('counter',$counter,time()+10);

    print_r($_COOKIE);
    echo "</pre>"
}

?>
<?php setMyCookie(); ?>
```

Opgave 10

```php
<?php
$i=0;
$total=-3;
while($i++<30) {
    if ($total<100 && $total<>0 ) {
        $total+=$i;
    } else {
        $total-=$i;
    }
}
echo $total;

// Uitkomst moet 88 zijn!
```

Test

Herhaling van de lessen tot nu toe.

Voorbeeld

In de opgaven hieronder moet je op de plaats van de <XXX1>, <XXX2>,.... de code zelf aanvullen. Bijvoorbeeld:

Gegeven:

```
for($i=0; $< <XXX1> ; <XXX2>) {
```

Gevraagd wordt nu op de plaats van <XXX1> en <XXX2> de code aan te vullen zodat je loop hebt die 10x itereert; wordt uitgevoerd.

Er zijn meerdere antwoorden mogelijk maar de eenvoudigste is

```
<XXX1>= 10 en <XXX2>= $i++
```

Opgaven

Opgave 1

```
<?php
for($i=3; <XX1>; $i++){
```

Wat moet er op de plaats van<XX1> staan om de loop precies 3 keer te laten uitvoeren?

Opgave 2

```
<?php
$i=<XX1>;
while($i>0) {
$i--;
```

Wat moet er om de plaats van <XXX1> staan om deze loop precies 10 keer te laten uitvoeren?

Opgave 3

```
<?php
$aantal=10;
do {
print "Aantal is nog ".$aantal.""<br>";
    if ($aantal == 2) {
    $aantal = 0;
    }
    $aantal--;
} while ($aantal > <XXX1> );
```

Je wilt dat de getallen 10 tot en met 2 worden afgedrukt. Wat moet je op de plaats van <XXX1> invullen?

Opgave 4

```
<?
$a=1;
$j=0;
while($a) {
  <XXX1>
  for($i=0;$i<10;$i++){
    echo "Hi!"."<br>";
  }
  if ($j==5) {
    $a=<XXX2>;
  }
}
```

Je wilt dat er 50x ""Hi!" wordt afgedrukt. Wat komt er dan te staan voor <XXX1> en <XXX2> ?

Opgave 5

```
<?
for($i=0; $i<10; $i++){
for($i=0; $i<10; $i++){
    echo "Hello"."<br>";
    }
}
```

Deze code zou 10 x 10 X Hello moeten afdrukken, maar het werkt niet. Waarom werkt de code niet, wat is er fout?

Let op dat dit een veel gemaakte fout is bij een loop in een loop.

Opgave 6

```
<?php
for($i=0; $i<10; $i++){
    if ($i == 5) {
    <XXX1>
    }
echo $i."<br>";
}
```

Deze loop drukt de getallen 0, 1, 2, ... etc onder elkaar af. Wat moet er op de plaats van de <XXX1> komen om ervoor te zorgen dat het getal 5 niet wordt afgedrukt?

Opgave 7

```
<?php
$persons = array("Mary" => "Female", "John" => "Male", "Mirriam" => "Female");
echo "Mary is a " . <XXX1>;
?>
```

Wat moet er komen in plaats van <xxx1> zodat het geslacht ('Female') van mary wordt afgdrukt?

Opgave 8

```
$myAraay = array ( "color" => array ( "a" => "Red", "b" => "Green", "c" => "White"));
```

Gebruik het array $myArray en druk Red, Green en White af.

Opgave 9

```
<?php
$movies =array(
"comedy" => array("Pink Panther", "John English", "See no evil hear no evil"),
"action" => array("Die Hard", "Expendables"),
"epic" => array("The Lord of the rings"),
"Romance" => array("Romeo and Juliet")
);
```

Dit is het resultaat van een query. Je ziet nu de uitkomst maar er kunnen ook meer of minder films zijn. Er kunnen bijvoorbeeld ook 12 comedy's zijn, of 5 maar ook 0. Maak en test nu code zodat je

alle comedy films afdrukt. Houd daarbij rekening met het feit dat je dus niet weet hoeveel comedy films er zijn.

Dus zoek eerst in het array $movies naar de comedy en druk dan het array af waarin de titels staan. In dit geval zijn het er drie, maar dat weet je niet van te voren.

Opgave 10

```
<?php
<?
$ceu = array("Countries"=>array( "Italy"=>"Rome", "Luxembourg"=>"Luxembourg", "Belgium"=>
"Brussels", "Denmark"=>"Copenhagen", "Finland"=>"Helsinki", "France" => "Paris",
"Slovakia"=>"Bratislava", "Slovenia"=>"Ljubljana", "Germany" => "Berlin", "Greece" =>
"Athens", "Ireland"=>"Dublin", "Netherlands"=>"Amsterdam", "Portugal"=>"Lisbon",
"Spain"=>"Madrid", "Sweden"=>"Stockholm", "United Kingdom"=>"London", "Cyprus"=>"Nicosia",
"Lithuania"=>"Vilnius", "Czech Republic"=>"Prague", "Estonia"=>"Tallin",
"Hungary"=>"Budapest", "Latvia"=>"Riga", "Malta"=>"Valetta", "Austria" => "Vienna",
"Poland"=>"Warsaw")) ;
```

1. Druk uit dit array de de hoofdstad van Duitsland (Germany) af.
2. Maak code die alle landen uit het $ceu array afdrukt.
3. Maak code dit alle landen uit het $ceu array gesorteerd afdrukt.
4. Maak code die alle hoofdsteden uit het $ceu array gesorteerd afdrukt.

Opgave 11

Maak gebruik van loops om onderstaand patroon in je web browser af te drukken. Op de eerste regel staat 1 sterretje, op de tweede 2,op de derde 3, ... en op de 10de regels staan 10 sterretjes.

```
*
**
***
****
*****
******
*******
********
*********
**********
```

Tip: maak eerst een 'gewone' loop van 10 (for($i=0.....) en druk $i af. Wat zie je? Welke getallen zie je en hoe verhouden die zich tot het aantal sterretjes dat je moet afdrukken?