PHP2

python reformatterCanvas.py 18661 52532 https://www.roc.ovh/books/2024/page/php2

Inleiding Arrays

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.

Snap je alles, ga dan de opgave maken.

Opgave

Een array heeft een naam, een key en een value (waarde).

Stel de naam='mijnArray', de key=3 en de waarde=0.

Hoe zet je dit om in PHP code?

Inleveren

De PHP code (zie opgave).

Print 2de array element

Gebruik array `$myArray=[15,25,35,45,55,65]` en druk het tweede element af (dit is dus 25).

Lever de code in en gebruik de naam php201a-jouw-naam.php

Find the right element

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

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


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

Changing array value

Gebruik array

`$myArray=[10,20,30,40,50,60];`

Stap 1

Verander de eerste drie elementen één-voor-één in 3,2 en 1. Gebruik hiervoor code.

Dus **niet** de getallen 10, 20 en 30 vervangen in de regel `$myArray=[10,20,30,40,50,60])` , maar een index (key) gebruiken en in drie afzonderlijke stappen de eerste drie waardes veranderen.

Stap 2

Druk alleen het 2de en 3de element af.

De output ziet er dan als volgt uit:

```
2
1
```

Vergeet de <br> niet!

Inleveren

1. Lever de php code in.

Inleiding foreach

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

In het programmaatje dat hierboven staat worden alle werkdagen van de week afgedrukt. Bereid dit uit zodat alle dagen van de week op de jusite volgorde wordne afgedrukt.

Inleveren

Aangepast programma met foreach loop.

Klasgenoten

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. Gebruik een loop om de namen af te drukken (uitleg op vorige pagina) .

Inleveren

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

Jouw naam in lijst, vet!

Maak de vorige opdracht en zorg ervoor dat jouw eigen naam ook ergens in het array staat. We hebben dus nog steeds een array met minstens tien namen.

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

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.

Uitdaging (optioneel voor 2 bonuspunten): kun je je naam met uitroepteken ook vetgedrukt of misschien zelf rood-vergedruk afdrukken?

Inleveren

Lever de code in en gebruik de naam php202c-jouw-naam.php

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.

image-1654442163332.png

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.

image-1654445242685.png

Opdracht

Je hebt u een paar opdrachten van functies waarbij input wordt omgezet in output:

1. pasje + pincode -> geld-automaat -> cash
2. geld (+keuze) -> snoep automaat -> snoep
3. koe -> slachthuis -> vlees

input	functie	output
pasje + pincode	geldautomaat	cash
geld (+keuze)	snoepautomaat	snoep
koe	slachthuis	vlees

Inleveren

Bedenk zelf een extra regel in de tabel hierboven.

Bedenk zelf een functie waarbij input wordt omgezet in output. Omschrijf de input, functie en output.

Bereken functie

Berekenen met een function()

Zorg dat je de theorie hebt doorgelezen.

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

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

De output van de functie is alleen het resultaat (één getal dus). De output wordt met een *return* teruggeven.

Noem de function *myVoornaam* (vervang *voornaam* door jouw eigen voornaam) en test of je code werkt.
Test minimaal twee gevallen en laat de code die je hebt gebruikt om te testen in je code staan.

Inleveren

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

Temperatuur

Temperatuur

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 je 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.

De twee hoogste (bonus)

Deze opdracht is optioneel en verplicht voor eventuele "studieversnellers"

Maak een functie die één array als input krijgt en die de twee hoogste waarden van het array bij elkaar optelt.

Input	Output
[12,3,21,5,3,20]	41
[12,3,21,5,3,20,21,3,2,1,-4,0]	42
[2,1]	3
[0,0]	0
[2]	foutmelding: "array has only one value"

Laat in je code zien dat je al deze input scenario's test.

Inleveren

De code in met de naam php203c-optioneel-jouw-naam.php

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 - metingen

Laten we 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.

De code heeft een complex array met metingen van de temperatuur. Elke dag heeft drie meetwaarden; 's morgens, 's middags en 's avonds.

Zo is de meetwaarde van de **ochtend** van de **2de dag** 6 graden.

Check of je dit terug kan vinden in het array $metingenWeek in de code.

Test de code 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 juiste 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

De hoogste meetwaarde

Je kunt deze opdracht alleen maken als je de vorige opdracht (goed) hebt gemaakt.

Je gaat het programmaatje dat je hebt gemaakt uitbreiden.

Je gaat je vorige programma aanpassen zodat de output er als volgt, komt uit te zien:

```
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!

Inleveren

De aangepaste en geteste code in php02-04b-jouw-naam.php

Nog meer loops

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

Verander de code zodat je alle getallen van 2 tot en met 9 afdrukt.

Inleveren

De aangepaste code

Loops voorspellen

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: een pdf bestand waarin je de stappen die hierboven staan beschreven bent doorgelopen. Het voorbeeld hieronder laat zien wat we als uitwerking ongeveer van je verwachten:

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

PDF, php205a-jouw-naam.pdf met de uitgewerkte opgaven.

foreach versus for

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>";
}
```

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

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

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

Inleveren

php205b-jouw-naam.php met de uitgewerkte code met de twee soorten loops.

Oneindig optellen

Opdracht

Maak een for-loop waarin alle getallen van 1 tot en met 100 worden opgeteld.

Jouw code laat de optelsom zien van 1+2+3+4+5+6+7+.....99+100.

Uitleg

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.

Dus **stel** dat je dit doet voor de getallen 1 t/m 5 dan zou er 15 uitkomen (1+2+3+4+5=15).

Dus voor de duidelijkheid de code laat één getal zien en dat getal is dus de som van alle cijfers 1 t/m 100 .

Inleveren

php205c-jouw-naam.php met de uitgewerkte opgaven.

Nog meer functies

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

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','Raine','Diego','Olmer','Wessel','Jari','Maxey','Brian','
Kikiya'];
```

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.

```
foreach($mijnArray as $name) {
    // roep functie aan met welkom-bookdschap
    // roep functie aan met Tot-ziens-boodschap
}
```

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
```

Inleveren

Drie bestanden:

1. php16a-jouw-naam.php
   (vanuit dit bestand wordt de functie gebruikt die in mijnFuncties.php is gedefinieerd)

2. mijnFuncties.php

3. een screenshot van je browser inclusief URL waarin je het resultaat laat zien.

...

!Copy 6

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

Opgave 1

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).

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 twee getallen (dus de twee 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

Revision #13
Created 2024-06-08 15:38:04 UTC by Max
Updated 2024-09-29 16:07:02 UTC by Max