PHP - refresh opgaven

Inleiding

Het is waarschijnlijk weer even geleden dat je met PHP heb gecodeerd. Om de kennis wat op te frissen zijn er in deze module 10 opgaven. In alle opgaven worden onderwerpen behandeld die in voorgaande modules al zijn behandeld.

Is er niets nieuws? Ja zeker wel. We leren wat HTML refresh is en we maken een begin met *gestructureerd* testen.

Wat leren we?

Opgave 1 gaan we naar de $_GET variabele kijken. Weet je nog wat dat was?

Opgave 2 gaan we kijken naar forms en de $_GET variabele.

Opgave 3 gaat over loops.

In Opgave 4 gaan we voor de afwisseling iets nieuws (html refresh) ontdekken.

Vanaf opgave 5 gaan we naar functies kijken. We maken steeds een stukje code in een functie en gaan ook leren hoe we deze functie moeten testen. Testen is een onderdeel van examen en je zult moeten aantonen dat je gestructureerd kan testen. Deze opgave zijn daar een aanzet toe.

Opgave 1, $_GET

In deze opgave kijken we terug hoe het ook alweer zat met $_GET. Deze variabele wordt gevuld als je een form invult met de method=GET.

```
<form action="/action_page.php" method="get">
    <input type="text" id="name" name="name">
```

Maar deze code zorgt ervoor dat de parameters via de URL (link, web adres) worden doorgegeven. Stel in dit voorbeeld voer je de naam *Ties* in. Dan zie je de URL:

```
?name=Ties
```

Maak een PHP pagina die de jouwe naam van de URL leest en deze afdrukt. Je hoeft geen form te maken.

Bijvoorbeeld, de URL is:

```
localhost/phprefresh/opgave1.php?naam=Prince
```

Op het scherm verschijnt nu:

```
Welkom Prince!
```

(let op de uitroepteken achter de naam).

Inleveren

- de php-code in php-01-jouwnaam.php

Opgave 2, forms

We hebben in de vorige opgave met de $_GET gewerkt. Nu gaan we met een form de $_GET variabele vullen.

Je mag ook *method*="*post*" gebruiken, maar dat is lastige om te zien wat er dan gebeurt omdat de form variabelen niet via de URL worden verstuurd.

Maak een eenvoudig HTML form en vraag de gebruiker **twee** getallen in te voeren. Dat mag/kan in één veld, maar dat mogen ook in twee aparte velden.

Vermenigvuldig (keer-som) deze twee getallen.

Toon het resultaat op het scherm.

Bijvoorbeeld, stel je hebt 3 en 5 ingevuld dan toon je:

```
3 x 5 = 15
```

Inleveren

- php-02-jouwnaam.php - met je PHP code (form en berekening).
- php-02-jouwnaam.png - met je schermafdruk van de hele browser waarin je het resultaat laat zien.

Opgave 3, loops

Hoe zat het ook alweer met al die loops. In deze opgave gaan we twee loops maken een *for-loop* en een *while-loop*.

Zoek eerst even hoe het ook weer zat met die loops; hoe werken die ook alweer?

Maak twee loops. Maak een *while* loop en een *for* loop en druk de getallen 1 tot en met 20 af.

De output moet er als volgt uit zien:

image-1656091001501.png

Zorge ervoor dat de output er ***precies*** zo uitziet!

De getallen zijn dus rood en **bold** en tussen de getallen staat een spatie-streepje-spatie en achter het laatste getal staat niets meer.

Rood en **bold**? Weet je niet meer hoe dat gaat. Zoek naar de juiste CSS code!

Inleveren

- php-03-jouwnaam.php - met je PHP code.
- php-03-jouwnaam.png - schermafdruk van jouw gehele browser met de juiste output.

Opgave 4, html refresh

Voor deze opgave moet je zelf één en ander opzoeken op internet. Dit hebben we nog niet eerder gehad, maar is wel iets dat soms erg handig kan zijn. We gaan een tijd afdrukken die automatisch loopt.

Zoek eerst uit hoe je de huidige tijd (in PHP) in uren en minuten op het scherm kan afdrukken.

Als je de tijd op het scherm laat zien dan veranderd deze alleen als je de pagina refreshed, probeer maar!

Er bestaat een zogenaamde HTML meta-tag die er voor zorgt dat je browser de pagina elke x seconden ververst. Zoek uit hoe dit werkt en laat de pagina die de tijd laat zien zichzelf elke 10 seconden verversen.

Controleer of jouw pagina zichzelf elke 10 seconde ververst.

Dit kan handig zijn als je een CRUD hebt gemaakt en je wilt dat de (Read) pagina zich elke paar minute ververst.

Opmerking: Om een tijd op het scherm te tonen kun je beter JavaScript gebruiken, maar we hebben het hier in PHP gedaan om de refresh te demonstreren.

Inleveren

- de php code in php-04-jouwnaam.php

Functies

Functies hebben soms één of meer parameters en kunnen een ding returnen.

Voorbeeld

```
<?php

function maalTwee($getal) {
  $uitkomst = $getal * 2;
  return $uitkomst;
}

echo maalTwee(4);
echo "<br>";
echo maalTwee(24);
echo "<br>";
```

Op regel 8 en 9 testen we de functie maalTwee() en we zien dat er 8 en 48 wordt afgedrukt.

In de volgende opgaven wordt telkens gevraagd om een functie te maken en om testen uit te voeren.

In dit voorbeeld hebben we dus **twee** testen uitgevoerd (input 4, output 8 en input 24 en output 48).

Input	Output
echo maalTwee(4);	8
echo maalTwee(4);	48

In de volgende opgaven gaan we dit zelf doen.

Opgave 5, Euro

Maak een functie die als input een getal krijgt en dat als een 'net' Euro getal terug geeft. In onderstaand schema staat wat de functie moet doen.

Test	Uitkomst
echo netEuro(12);	€ 12.00
echo netEuro(12.5);	€ 12.50
echo netEuro(12.6666);	€ 12.67

Bij elk bedrag wordt dus afgerond op twee decimalen (centen) en er wordt een euro getal afgedrukt voor het bedrag.

Maak de functie *netEuro* en test (tenmiste) met de waarden zoals die hierboven zijn gegeven.

Inleveren

- php-05-jouwnaam.php - met je PHP code.
- php-05-jouwnaam.png - schermafdruk van jouw gehele browser met de juiste output.

Opgave 6, onvoldoendes

Kijk naar het filmpje: https://www.youtube.com/watch?v=INHM0S_8GHA

En maak daarna een **functie** `onvoldoendes($PHPCijfers);` De functie krijgt als input een array met getallen en als output wordt een array gegeven waarin alleen de onvoldoendes staan.

Voorbeeld:

```
$PHPCijfers=[4,5,4,5,6,6,5,8,7,6,4,8];
```

Dus de output van de functie wordt:

```
4,5,4,5,5,4
```

Omdat de output een array is kun je die afdrukken met print_r(). Door een <pre> ervoor te gebruiken is de output beter leesbaar.

Input	Output
echo onvoldoendes([6,6,7]);	
echo onvoldoendes([6,3,6,7]);	3
echo onvoldoendes($PHPCijfers);	4,5,4,5,5,4

Maak de functie en test met tenminste de drie gegeven test cases.

Je code ziet er dus ongeveer zo uit:

```
function onvoldoendes($array) {
  ...
  ...
  ...
}

echo "<pre>";
print_r( onvoldoendes([6,6,7]) );
print_r( onvoldoendes([6,3,6,7]) );
print_r( onvoldoendes($PHPCijfers) );
```

> Let op de haakjes. Neem bijvoorbeeld regel 8.
> De buitenste () horen bij de print_r() functie. De () na de onvoldoendes horen bij onze functie onvoldoendes() en de [] haken geven aan dat we een array gebruiken.
>
> Probeer dit begrijpen want dit komt terug in frameworks (Yii en Laravel).

Op de plaats van de puntjes maak je jouw functie af.

Inleveren

- de php code in php-06-jouwnaam.php

Opgave 7, Goedemorgen

Maak een PHP programmaatje dat afhankelijk van de tijd een boodschap afdrukt.

- Het uur is kleiner dan 12: "Goedemorgen".
- Het uur is 12 uur of groter én kleiner dan 18: "Goedemiddag".
- Het uur is groter dan 18: "Goedenavond".

Er is een PHP een functie om de huidige tijd te bepalen. Je hoeft alleen maar te kijken naar het uur.Het maakt immers niet uit of het 11:00 of 11:59 is, het uur is kleiner dan 12 dus is de boodschap 'Goedemorgen'.

Zoek zelf hoe je in PHP de tijd en het uur kan bepalen.

Zet je code in een *functie* en noem deze functie *begroeting*).

De functie krijgt als input een naam mee.

De parameter (de naam) wordt achter de begroeting gezet en de complete begroeting wordt door de functie in een ***return*** terug gegeven.

Dus bijvoorbeeld, het is 4 uur 's middags en plaatst de volgende code:

Conditie	Input	Output
Het is 16:00 uur	echo begroeting("Vasco");	Goedemidag Vasco
Het is 16:00 uur	echo begroeting("Jesse");	Goedemidag Jesse
Het is 11:59 uur	echo begroeting("Nadir");	Goedemorgen Nadir

Maak een functie en test tenminste met de bovenstaande test cases.

Om de conditie te testen hoef je niet te wachten tot het (bijvoorbeeld) 16:00 uur is. Je kunt in de code ook even tijdelijk de variabele waarin je uur staat instellen.
Stel je hebt ergens PHP code die ervoor zorgt dat $uur de waarde krijgt van het huidige uur. Dan kun je daarna even tijdelijk om te testen `$uur=11` neerzetten.

Je code ziet er dus ongeveer zo uit:

```
function begroeting($naam) {
  ...
  ...
  ...
}

echo begroeting("Vasco");
```

```
echo onvoldoendes("Jesse");
```

Inleveren

- php-05-jouwnaam.php - met je PHP code.
- php-05-jouwnaam.png - schermafdruk van jouw gehele browser met de juiste output.

Opgave 8, Optellen

Je gaat een functie maken die alle getallen uit een array die groter zijn dan 10 optelt.

Neem bijvoorbeeld dit array.

```
$array = [11,11,5,2,12,6,7,8,1,10,9];
```

Er zijn drie getallen groter dan 10. Dat zijn 11, 11 en 12. Opgeteld is dat 34. De functie geeft dus 34 terug.

Noem de functie telGroteGetallen(array)

Input	Output
echo telGroteGetallen([10,20,20]);	40
echo telGroteGetallen([9, 10, 11]);	11
echo telGroteGetallen($array);	34

De $array in de laatste test heeft de waarden zoals in het voorbeeld is gegeven.

Maak een functie en test tenminste met de bovenstaande test cases.

Inleveren

- php-06-jouwnaam.php - met je PHP code.
- php-06-jouwnaam.png - schermafdruk van jouw gehele browser met de juiste output.

Opgave 9a, vier letters

Je gaat een functie maken die alle woorden telt die minder dan vier letters hebben.

Neem bijvoorbeeld deze string.

```
$string = "Voor de vormgeving is het handig om te weten hoe het eruit komt te zien voordat je daadwerkelijk tekst gaat plaatsen.";
```

De string heeft 9 woorden kleiner dan vier letters, tel maar na!

Hoe doe je dit in PHP?

1. Zet deze string om in een array. Tip: maak gebruik van de PHP-functie *explode*
2. Maak een variabele teller en zet die op 0 (je hebt immers nog 0 woorden gevonden die kleiner zijn dan vier letters).
3. Loop met een foreach loop door het array.
4. Maak een if in de loop en test of het woord minder dan vier karakters heeft.
5. Als de if waar is dan hoog je een teller op.
6. Aan het einde van de loop heb je in je teller het aantal woorden dat kleiner is dan vier. Return deze waarde

Noem nu een functie *kleineWoorden*(string) die de string als parameter mee krijgt en die als output het aantal woorden kleiner dan vier letter terug geeft.

Input	Outpur
echo kleineWoorden("dit is een voorbeeld");	3
echo kleineWoorden("a b c d");	4
echo kleineWoorden("abcd");	0
echo kleineWoorden($string);	9

De $string in de laatste test is dus de string zoals gegeven in het voorbeeld. Test je code met tenminste de 4 gegeven tests

Maak een functie en test tenminste met de bovenstaande test cases.

Inleveren

- php-09a-jouwnaam.php - met je PHP cod
- php-09a-jouwnaam.png - schermafdruk van jouw gehele browser met de juiste output.

Opgave 9b

Deze opgave is een vervolg-opgave van 9a.

Pas de functie kleineWoorden() aan zodat je aan de functie kan meegeven wat een klein woord is.

Stel je wilt alleen de woorden tellen die kleiner zijn dan 3 letters dan kun je de functie aanroepen met: kleineWoorden(3,$string) De functie telt dan alle woorden die kleiner zijn dat 3 karakters.

De $string in de testen is:

```
$string = "Voor de vormgeving is het handig om te weten hoe het eruit komt te zien voordat je daadwerkelijk tekst gaat plaatsen.";
```

Input	Output
echo kleineWoorden(4,"dit is een voorbeeld");	3
echo kleineWoorden(3,"dit is een voorbeeld");	1
echo kleineWoorden(1,"dit is een voorbeeld");	0
echo kleineWoorden(0,"dit is een voorbeeld");	0
echo kleineWoorden(2,"a b c d");	4
echo kleineWoorden(6,"abcde");	1
echo kleineWoorden(4,$string);	9
echo kleineWoorden(3,$string);	6
echo kleineWoorden(10,$string);	19

Maak een functie en test tenminste met de bovenstaande test cases.

Inleveren

- php-09b-jouwnaam.php - met je PHP cod
- php-09b-jouwnaam.png - schermafdruk van jouw gehele browser met de juiste output.

10 15Eindopgave/challenge

(15 punten)

Stel je hebt een grid view (Read) en je hebt een kolom met een opmerking. De opmerking kan lang zijn en voor het overzicht wil je alleen het eerste deel van de opmerking afdrukken. Je wilt geen woorden afbreken want dat ziet er lelijk uit.

De opgave is dus om een funtie te maken waarmee je een (lange) string afbreekt en alleen de eerst woorden laat zien zodat de totale lengte van de string niet langer is dan N karakters. N geef je mee aan de funtie.

OK, lastig? Hier komt een voorbeeld.

Stel je hebt deze opmerking.

```
opmerking = "Deze student is erg goed in programmeren in PHP";
```

In jouw overzicht heb je ongeveer ruimte voor 32 karakters. Je roept dan de functie aan

```
echo breekAf(32, $string);
```

Als je afbreekt op 32 karakters dan krijg je:

```
Deze student is erg goed in prog
```

Nu hebben we een 'half' woord (prog) dus de output wordt:

```
Deze student is erg goed in
```

Input	Output
echo breekAf(32, "Deze student is erg goed in programmeren in PHP");	Deze student is erg goed in
echo breekAf(18, "Deze student is erg goed in programmeren in PHP");	Deze student is
echo breekAf(19, "Deze student is erg goed in programmeren in PHP");	Deze student is erg
echo breekAf(20, "Deze student is erg goed in programmeren in PHP");	Deze student is erg

echo breekAf(21, "Deze student is erg goed in programmeren in PHP");	Deze student is erg
echo breekAf(24, "Deze student is erg goed in programmeren in PHP");	Deze student is erg goed
echo breekAf(3, "Deze student is erg goed in programmeren in PHP");	<lege string>

Maak een functie en test tenminste met de bovenstaande test cases.

Inleveren

- php-10-jouwnaam.php - met je PHP cod
- php-10-jouwnaam.png - schermafdruk van jouw gehele browser met de juiste output.

Revision #14
Created 2022-06-24 15:15:33 UTC by Max
Updated 2022-07-15 10:38:56 UTC by Max