Theorie PHP1

python reformatterCanvas.py 18661 c https://www.roc.ovh/books/2024/page/php1

Wat is een webserver

Wat ga je leren?

In deze les leer je het verschil tussen een front-end taal en een back-end taal. Je leert dat PHP een back-end taal is en dat je daarom een (web)server nodig hebt om PHP te kunnen gebruiken.

Lees de tekst goed want in de volgende opdracht krijg je vragen over deze tekst.

Een webserver is een computer die verbonden is met het internet en waar bestanden op staan op st
nodig zijn om een website te laten zien.

Elke website die jij bezoekt op het World Wide Web staat ergens op een webserver. Zo'n webserver
map (folder) met daarin webpagina's die bestaan uit HTML pagina's, CSS bestanden, Javascript, PHP
en derglijke.

Bestandssorrten	Functie
HTML	Basis opmaak van een webpagina
CSS	Detail opmaak van een webpagina
JavaScript	Interactie programmeren in de browser
PHP	Interactie programmeren op de server

Tot nu toe ben je gewend om de HTML-bestanden gewoon vanaf je eigen computer te starten door e
te klikken. De browser leest de bestanden en laat deze een mooie (of lelijke) pagina zien.

Je bent als gebruiker van webapplicaties gewend om bijvoorbeeld in te loggen, een bericht achter te
email te sturen. Deze applicaties zijn *dynamisch* . Dynamische applicaties hebben een **input** (inv
zetten deze om in een **output (uitvoer)** .

Bijvoorbeeld:

input : gebruikersnaam en wachtwoord
output : een boodschap dat het aanloggen is gelukt of een boodschap dat het aanloggen niet is ge

Om een dynamische webapplicatie te maken heb je meestal een webserver nodig die de pagina's **s**
De webserver krijgt van de browser, de cliënt een verzoek en de webserver behandeld het verzoek
antwoord terug. Net als een ober die een drankje **serveert** .

Stel je wilt een overzicht van alle studenten in een klas. Je vraagt via de browser dan om het overzic
bijvoorbeeld door op een menu te klikken. Het verzoek gaat dan via internet naar een **webserver**
het antwoord via internet weer **terug** naar jouw **browser** .

image-1655279215130.png
Image not found or type unknown

Omdat je nog geen toegang hebt tot een echte webserver gaan we er eentje simuleren. Simuleren i
doen alsof'. We simuleren de webserver met een gratis applicatie (wij doen dat met XAMPP) die je k
installeren op je eigen laptop. Dat heet dan een lokale server. Dat wordt ook wel **localhost** genoe
is dus door het gebruik van XAMPP een web server geworden.

Om een dynamische applicatie te maken gaan we in deze module gebruik maken van PHP.

De server heet ook wel back-end (achterkant). PHP is een back-end-taal. Dat betekent dat de taal d
back-end, de server draait.

Een bezoeker van de site (de eindgebruiker) kan de code die op de server staat niet zien of wijzigen

Bestandssorrten	Functie	Waar
HTML	Basis opmaak van een webpagina	Front-end / browser
CSS	Detail opmaak van een webpagina	Front-end / browser
Javascript	Interactie programmeren in de browser	Front-end / browser
PHP	Interactie programmeren op de server	Back-end / server

Database

Met PHP kun je ook een database benaderen waarin je gegevens kan opslaan voor een langere peri
bijvoorbeeld de gegevens van een gebruiker zoals voornaam, achternaam, email en wachtwoord.

Als je met PHP aan de slag gaat dan kun je pagina's dynamisch maken. Dit betekent dat de output a
van de input van de gebruiker.

XAMPP

Tijdens deze module maak je gebruik van een locale server. We gebruiken hiervoor XAMPP.

Je kunt XAMPP downloaden en installeren als je dat nog niet hebt gedaan.

Weet je (nog) niet hoe je XAMPP moet installeren op jouw Windows machine of heb je een "port con
hier

Links to an external site.

Inleveren

Volg de instructies in de les en lever je werk dat je in de les hebt gemaakt hier in.

XAMPP

Wat ga je leren?

Ter voorbereiding van PHP gaan we in deze les leren hoe we XAMPP kunnen installeren.

PHP is een back-end taal. Dat betekent dat deze taal op de server draait. Op de server wordt je code uitgevoerd. Omdat wij (nog) geen server op het internet hebben, installeren wij een soort server op onze laptop. Onze laptop is dan een *back-end* en ook *front-end* tegelijkertijd. Onze browser (*front-end*) 'praat' dus met de PHP server Xammp (*back-end*).

Later gaan we ook databases gebruiken en daarvoor installeren we een database server. Dit is een onderdeel van Xampp.

OK let's go!

We gaan XAMPP installeren! Volg alle stappen zoals die hier beschreven staan.

1. Download Xampp

1. Links to an external site.

2. Installeer Xampp met het installatieprogramma.
    - installeer deze op: C:\Xampp

- Tijdens de installatie kies je drie opties: Apache (dat is de PHP-server), MySQL (dat is de database server) en PHPMyAdmin, dat is de tool om de database mee te beheren.

1. Start Xampp en start daarna Apache en MySQL

   image_1653120218472.png
   Image not found or type unknown

2. Maak een nieuwe folder in c:\xampp\htdocs\phplevel1-jouw-naam
   - vervang *jouw-naam* door jouw naam
     Overal waar in het vervolg j*ouw-naam* staat, vervang je de tekst door jouw naam.

3. Start VCS en open de folder c:\xampp\htdocs\phplevel1-jouw-naam

4. Maak een nieuw bestand en noem deze php01-jouw-naam.php
   - Vergeet niet jouw-naam te vervangen!

5. Type de volgende code in:

```
<?php

echo "Hello world!";

?>
```

Voer de code uit.

Ga naar C:\xampp\htdocs en verwijder het bestand index.php

Ga nu naar de browser en type als url, *localhost* in.

Open nu de folder en het bestand dat je net hebt gemaakt.

Wat gebeurt er in deze code?

Op regel 1 <?php

wordt tegen de webserver verteld dat we code in php gaan maken. De server/XAMPP weet dan welke taal jij gata praten.

Op regel 3, echo

Het woordje *echo* betekent dat de server output moet gaan maken. Deze output wordt naar de web browser gestuurd.

Op regel 3 "Hello world!"

echo moet iets naar de browser sturen, dan moet je natuurlijk wel vertellen wat dat is. Het zinnetje na de echo is dat wat er naar de browser wordt gestuurd. Dit zinnetje staat tussen "" om aan te geven waar het zinnetje begint en waar het eindigt.

Op regel 3 ;

Na ELK commando in PHP staat een ; om aan te geven dat het commando afgelopen is en. De meest gemaakte fout in PHP is dat je een ; vergeet. Wedden dat jij dat dit jaar ook tig keer gaat vergeten?

Op regel 5 ?>

We zijn klaar met onze commando's in PHP. We vertellen de server/XAMPP dat we stoppen met "PHP praten".

Test de code en controleer of er in je browser de tekst Hello World! verschijnt.

Je hoeft niets in te leveren, maar als PHP/XAMPP niet werkt dan kan je de volgende opdracht niet maken.

Variabelen

Wat ga je leren?

In ***alle*** programmeertalen gebruik je variabelen. In deze les leer je wat een variabele is en hoe je deze kan gebruiken.

Computer geheugen

Een variabele is een plek in het geheugen van de computer waar je informatie in op kan slaan.

Het geheugen van een computer loopt bijvoorbeeld van adres 0 tot aan adres 32768.

Dan kan het zijn dat je op plaats 21 311 een plaats hebt waar je jouw naam hebt opgeslagen en op plaats 18711 heb je een plaats in het geheugen waar je jouw leeftijd hebt opgeslagen. Omdat deze getallen lastig zijn te onthouden kunnen we deze geheugenplaatsen een naam geven. Dat heet een variabele.

Variabele namen

Als je gaat programmeren ga je best veel gebruik maken van variabelen. We geven dus een *naam aan* een geheugenplekje, bijvoorbeeld plaats 21311 is waar jouw naam staat, noemen we *mijnnaam* en de plaats 18711 waar jouw leeftijd in staat noemen we *mijnleeftijd*.

Doosje of lade

Je zou een variabele ook kunnen zien als een doosje of een lade waar je iets kan instoppen.

image_1653122885951.png image_1653135643886.png

Net als bij een doos of een lade, kan je de doos/lade openen en kijken wat er in zit; er kan niets in zitten, maar er kunnen ook getallen of zinnetjes in zitten. En....je kunt de inhoud ook veranderen.

In de eerste 2 minuten van deze video (NL)

Links to an external site.

image not found or type unknown

wordt dit nog een keer uitgelegd.
In dit filmpje gaat het over de programmeertaal **Python**, maar dit geldt ook voor **PHP** (en bijna **alle** andere computertalen)

PHP Variabele

In PHP kun je een variabele makkelijk herkennen omdat die altijd met een $-teken begint, bijvoorbeeld `$mijnnaam`, `$mijnleeftijd` zijn twee variabelen.

Hoe ken je een waarde toe aan een variabele?

Stel jij heet *Mohammed* en je wilt de variable `$voornaam` de waarde *Mohammed* geven. Je gebruikt daar voor het = teken. Zie dit niet als '*is*' maar als '*wordt*' .

```
<?php
$mijnnaam="Mohammed";
?>
```

op regel 1 wordt de computer verteld dat je PHP code gaat gebruiken.

op regel 2 hier staat dat de variabele met de naam `$mijnnaam` de waarde Mohammed krijgt. Dus `$mijnnaam` wordt "Mohammed".

Mohammed is een zinnetje dus dat staat tussen "".

In dit geval is het een zinnetje van één woord, maar het zou ook meer woorden kunnen bevatten daarom staat een zinnetje, ook al is het maar één woordt altijd tussen "". De computer weet dan precies waar het zinnetje begint en waar het eindigt.

En niet vergeten de ; om aan te geven dat het commando klaar is!

> Elk PHP commando wordt in PHP (en ook in bijv. JavaScript en C#, C, C++ en Java) afgesloten met een ;

Even terug naar het doosje of de lade; we hebben dus een doosje gemaakt met het label *mijnnaam* en in het doosje hebben we de string-waarde "Mohammed" gezet.

Het = teken dat we hebben gebruikt kun je dus lezen als 'stop de volgende waarde in dit doosje'.

Variabele naamgeving

Zoals gezegd, een variabele naam in PHP begint altijd met een $-teken met daarachter een woord zonder spaties. De variabele bestaat dus altijd uit één woord. Omdat je soms duidelijk wil maken wat er in de variabele staat gebruik je goede namen die vaak wat langer zijn.

Bijvoorbeeld $*hoofdpersonagevandefilm*.

Om dit leesbaar te maken, maken we hiervan *hoofdPersonageVanDeFilm*.

Dus $*hoofdpersonagevandefilm* wordt $*hoofdPersonageVanDeFilm*

Zie je het verschil?

In code ziet het er dan zo uit.

```
$hoofdPersonageVanDeFilm = "Ryan Reynolds";
```

Zo bestaat de naam van de variabele nog steeds uit één woord, maar is het wel beter leesbaar.

Let op want de variabele $*mijnnaam* en $*mijnNaam* zijn twee verschillende variabelen. We zeggen dan dat variabelenamen hoofdlettergevoelig zijn (in het Engels; *case sensitive*).

Strings

Wat ga je leren?

Variabelen heb je in alle soorten en maten. De twee belangrijkste zijn getallen en strings.

In deze les leer je wat een string variabele is en hoe je er in PHP mee kan werken.

We hebben al kennis gemaakt met strings in de vorige les over variabelen, maar in deze les gaan we hier nog wat verder op in.

Je hebt verschillende soorten variabelen.

Om te beginnen zijn er twee hoofdsoorten: **strings en getallen**. In deze les gaan we kijken naar strings.

Strings

Strings (letterlijk vertaald een ketting) is een ketting van letters en/of getallen achter elkaar. Om het begin en een eind van een string aan te geven staat een string altijd tussen quotes. Je mag enkele of dubbele quotes gebruiken.

```
<?php

$string = "dit is een voorbeeld";
$nogEenString = 'dit is ook een string';

echo $string;
echo $nogEenString;

?>
```

Op regel 3 staat een string tussen dubbele quotes en op regel 4 staat een string tussen enkele quotes '.

Met strings kan je verschillende dingen doen. Zo kan je strings aan elkaar plakken. Dat heet *concatenation* *Links to an external site.* *(Engels)*.

In PHP plak je strings aan elkaar door tussen twee string een punt te zetten.

```
<?php
$voornaam="Alexandra";
$achternaam="Gaona";
```

```
echo $voornaam." ".$achternaam;
?>
```

Op regel 5 worden eigenlijk drie strings aan elkaar geplakt. Eerst de string die in de variabele $voornaam staat dan de string " " (deze staat niet in een variabele) en dan string die in de variabele $achternaam staat.

Als je de HTML code
 afdrukt dan wordt er een regel overgeslagen. "
" is een string.

```
echo "<br>"; // regel overslaan
```

Dus regel 1 drukt de string "
" af en de browser slaat dan een regel over.

De // aan het einde van de regel is overigens commentaar. Het doet niets en het heeft alleen als doel om je code duidelijker te maken (voor anderen of later voor jezelf).

Maak nu de opdrachten.

Getallen

Wat ga je leren?

Een veel gebruikt type variabele is integer. In deze les je wat een integer variabele is en hoe je daar mee kan werken in PHP.

Naast strings hebben we ook getallen. Deze zijn weer onderverdeeld in gehele getallen (integers) en gebroken getallen (float). Dat verschil is in PHP niet zo belangrijk omdat PHP zelf het juiste soort kiest.

Getallen zijn wel anders dan strings want je kunt er mee rekenen. Om aan te geven dat je een getal bedoeld en geen string, zet je een getal *niet* tussen quotes.

Dus $voorbeeld="9" is de string "9" en $voorbeeld=9 is het getal 9.

```
<?php

$getal1=12;
$getal2=13;
echo $getal1 + $getal2;

?>
```

Net zoals je getallen kan optellen, kun je ook andere bewerkingen uitvoeren.

Bewerking	teken	voorbeeld
optellen	+	echo $a + $b;
aftrekken	-	echo $a - $b;
vermenigvuldigen (keer)	*	echo $a * $b;
delen	/	echo $a / $b;

Ga door naar de opdrachten.

Condition if-then-else

Wat ga je leren?

Code wordt regel voor regel van boven naar beneden uitgevoerd,

Je hebt echter ook code die alleen maar wordt uitgevoerd als er aan een voorwaarde (of conditie) is voldaan.
Hiervoor heb je conditional statements in PHP, *if* (in het Nederlands: *als*).

In deze les leer je hoe je met een if in PHP kan werken en hoe je dus kan zorgen dat code alleen wordt uitgevoerd als er aan die ene voorwaarde wordt voldaan.

IF

Stel je maakt een spel en je moet bepalen of de speler het juiste antwoord heeft gegeven:

- Juiste antwoord? Score wordt verhoogd met 1.
- Onjuiste antwoord? Score blijft hetzelfde.

Er komt dus alleen iets bij de score als er aan een voorwaarde is voldaan.

Dat ziet er in PHP zo uit:

```
$score=0;


$vraag = "Hoeveel is 10+2?";
$antwoord = 12;
```

```
if ( $antwoord == 12 ) {
□$score = $score + 1;
}


echo "De score is: $score";
```

Op regel 6 wordt het antwoord gecontroleerd. Is het antwoord juist dan wordt alle code tussen { en } uitgevoerd. In dit geval is dat alleen regel 7.

Op regel 4 zie je = en op regel 6 ==

Dat komt omdat ze beide iets anders betekenen.

teken	voorbeeld	betekent
=	$antwoord = 12	De waarde van $antwoord **wordt** 12
==	$antwoord == 12	Is de waarde van $antwoord **gelijk aan** 12?

We kunnen ons afvragen of waarden gelijk zijn, maar er zijn nog een heel stel andere voorwaarden. De belangrijkste zijn:

teken	voorbeeld	betekent
==	$antwoord == 12	Is de waarde van $antwoord **gelijk aan** 12?
!=	$antwoord != 12	Is de waarde van $antwoord **ongelijk** aan 12?
>	$antwoord > 12	Is de waarde van $antwoord **groter** dan 12?
<	$antwoord < 12	Is de waarde van $antwoord **kleiner** dan 12?
>=	$antwoord >= 12	Is de waarde van $antwoord **groter of gelijk** aan 12?
<=	$antwoord <= 12	Is de waarde van $antwoord **kleiner of gelijk** aan 12?

ELSE

Stel voor dat als de score fout is dat er dan een punt van de score wordt afgetrokken, dan zou je dat zo kunnen coderen

```
$score=0;


$vraag = "Hoeveel is 10+2?";
```

```
$antwoord = 12;

if ( $antwoord == 12 ) {
$score = $score + 1;
} else {
  $score = $score -1;
}

echo "De score is: $score";
```

Maak de volgende opdrachten en gebruik indien nodig de volgende bronnen.

Bronnen

Jaap vd Veen - statements

Links to an external site.

W3 Schools - operators Links to an external site.

W3 Schools - if else

HTML, PHP en include

Wat ga je leren?

In deze les leer je hoe je HTML en PHP samen kunt gebruiken.

Je leert ook het wat het *include* en *require* in PHP doet en hoe je dat kunt gebruiken.

Include

Om je code overzichtelijk te houden, stop je niet al je code in één groot bestand, maar je deelt het op in meerdere kleinere bestanden. Je kunt dan snel de juiste code vinden.

Zo kun je php code hebben die een footer afdrukt. Stel je wilt je volgende footer:

```
<?php
echo "<hr>";
echo "(c) Copyright<br>";
echo "De inhoud en structuur van onze webpagina's zijn auteursrechtelijk beschermd.";
?>
```

Deze footer wil je in je pagina afdrukken dan kun je deze pagina 'includen'. Dat gaat als volgt:

```
 <!DOCTYPE html>
<html>
<body>
<h1>My First Heading</h1>
<p>My first paragraph.</p>
</body>

<?php
include 'footer.php';
?>

</html>
```

Zie je wat er op regel 9 gebeurt? Je voegt als het ware de inhoud van de footer.php toe op regel 9. Hieronder zie je dan wat de include eigenlijk doet.

```
 <!DOCTYPE html>
<html>
<body>
<h1>My First Heading</h1>
<p>My first paragraph.</p>
</body>

<?php
echo "<hr>";
echo "(c) copyright<br>";
echo "De inhoud en structuur van onze webpagina's zijn auteursrechtelijk beschermd.";
?>

</html>
```

De include verwijst altijd naar een bestand.

Let erop dat je alleen php code kan uitvoeren als de naam van het bestand op .php eindigt. Bovenstaand voorbeeld werkt dus alleen als de naam eindigt in .php.

Je kunt de naam van het bestand ook als string mee geven, tussen " zoals in het voorbeeld, maar je kunt natuurlijk ook een variabele meegeven:

```
include $footer;
```

De code van het bestand waarvan de naam in de variabele $footer staat wordt nu ingevoegd.

Nog een voorbeeld

Voorbeeld: **home.php**

```
<?php
   include 'database.php';
?>
<html>
<head>
</head>
<body>
   <div class="menu">
      <?php include 'menu.php'; ?>
   </div>
</body>
</html>
```

Je ziet in bovenstaand voorbeeld op regel 9 dat een menu met behulp van php wordt ingevoegd in de `<div class="menu"> .... </div>`

Je ziet dat php geopend wordt `<?php` en er dan normale php-code volgt. Zodra de php code klaar is dan sluiten we de php code weer met `?>`.

Logical operators en korte broeken

Wat ga je leren?

In deze les gaan we leren hoe we condities kunnen combineren.
We leren hoe we code alleen kunnen laten uitvoeren als er aan **twee** voorwaarden wordt voldaan.
En we leren hoe we code kunnen uitvoeren als er aan **één van de twee** voorwaarden wordt voldaan.

We hebben het gehad over *if-then-else*.

Met een *if-then-else* kun je *voorwaardelijk* code uitvoeren. Alleen als een bepaalde vergelijking waar is dan wordt bepaalde code uitgevoerd.

Stel je hebt een variabele $temperatuur waarin de temperatuur staat en stel je hebt een andere variabele waarin de dag van de week staat. Dus bijvoorbeeld:

```
$temperatuur = 21;
$dagVanDeWeek = 'ma';
```

Dus in dit voorbeeld is het maandag en de temperatuur is 21 graden. Stel dat ik code moet schrijven die mij advies geeft over het dragen van een korte broek. Ik wil alleen een korte broek aan doen als het zaterdag is én als de temperatuur minimaal 21

is. Dat zijn dus twee voorwaarden die *waar* moeten zijn.

Hoe doe je dat in code?

```
$temperatuur = 21;
$dagVanDeWeek = 'ma';

if ( $temperatuur >= 21 && $dagVanDeWeek == "za" ) {
  echo "Advies is om vandaag een korte broek aan te trekken.";
} else {
  echo "Advies is om vandaag geen korte broek aan te trekken.";
}
```

Zie je het **&&** - teken?

Dat betekent dat **beide** voorwaarden waar moeten zijn. Dus de temperatuur is groter of gelijk aan 21 graden **én** de dag van de week is 'za'.

Stel dat we nu op zaterdag of zondag altijd een korte broek willen dragen, hoe zou de code er dan uit zien?

```
$temperatuur = 21;
$dagVanDeWeek = 'ma';
```

```
if ( $dagVanDeWeek == "za" || $dagVanDeWeek == "zo" ) {
  echo "Advies is om vandaag een korte broek aan te trekken.";
} else {
  echo "Advies is om vandaag geen korte broek aan te trekken.";
}
```

Zie je het **||** - teken?

Dat betekent dat aan **één van de** voorwaarden moet worden voldaan. Dus als de dag van de week zaterdag **óf** zondag is dan volgt het advies voor het dragen van een korte broek.

In de voorbeelden wordt && gebruikt voor *and* en || voor *or*. In de meeste programmeertalen kun je ook gewoon het woord **and** en **or** gebruiken. Dat werkt in PHP ook.

Dus de volgende twee regels doen hetzelfde.

```
if ( $temperatuur >= 21 && $dagVanDeWeek == "za" ) {....

if ( $temperatuur >= 21 and $dagVanDeWeek == "za" ) {....
```

Dus samengevat

		Betekenis	Voorbeeld
&&	and	AND, aan **beide** voorwaarden moet zijn voldaan	if ($temperatuur >= 21 && $dagVanDeWeek == "za") {
||	or	OR, aan **één van beide** voorwaarden zijn voldaan	if ($dagVanDeWeek == "za" || $dagVanDeWeek == "zo") {

Haakjes in condities

Stel we willen alleen een korte broek aan als de temperatuur minimaal 21 graden is én de dag zaterdag is of de dag zondag is.

Dan krijgen we de volgende code:

```
if ( $temperatuur >= 21 && $dagVanDeWeek == "za" || $dagVanDeWeek == "zo" ) {
```

Maar stel dat het anders is. Op zondag willen we altijd een korte broek dragen en op zaterdag alleen als de temperatuur minimaal 21 graden is. Dus in code:

```
if ( $dagVanDeWeek == "zo" || $temperatuur >= 21 && $dagVanDeWeek == "za" ) {
```

Maar deze code is bijna hetzelfde als de vorige code, alleen de volgorde is anders. Om de code goed uit te voeren moeten we haakjes gebruiken:

Korte broek in het weekend als de temperatuur 21 graden of hoger is:

```
if ( $temperatuur >= 21 && ( $dagVanDeWeek == "za" || $dagVanDeWeek == "zo" )  ) {
```

Het geen tussen haakjes, de || wordt eerst uitgevoerd, daarna volgt de &&

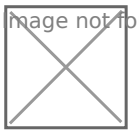
Altijd een korte broek op zondag én op zaterdag alleen een korte broek als de temperatuur 21 graden of hoger is.

```
if ( ($temperatuur >= 21 && $dagVanDeWeek == "za" ) || $dagVanDeWeek == "zo" ) {
```

Hier wordt eerst de && uitgevoerd en daarna volgt de ||.

KennisCheck

Image not found or type unknown



KC.jpg

Image not found or type unknown

Als je alles hebt gelezen en hebt uitgevoerd dan gaan jullie klassikaal een Kennischeck uitvoeren.

- Voer de juiste naam en klas in bij de kennis-check
- Zorg ervoor dat je 80% of hoger scoort.

Inleveren

1. Screenshot van je resultaat.

Revision #8
Created 8 June 2024 16:25:20 by Max
Updated 27 June 2024 19:47:11 by Max