

Variabelen aanmaken

Een 'variabele' is een container in een PHP script waar een waarde in bewaard kan worden.

De volgende benamingsconventie moet worden gehanteerd:

- Namen moeten met een dollar (\$) teken beginnen. **\$naam**
- Namen mogen bestaan uit letters, cijfers en underscore teken. **\$naam_1**
- Het eerste karakter na het dollar teken moet een letter zijn.

Variabelen namen in PHP zijn hoofdletter gevoelig. \$naam, \$NAAM en \$Naam zijn 3 verschillende variabelen.

Een PHP variabele kan elke waarden bevatten. Er hoeft niet aangegeven te worden van welk type deze moet zijn.

Een variabele maken in PHP gebeurt door de naam van de variabelen aan te geven. Je kunt dan de variabele een waarde toekennen. Deze instructie en alle andere instructies in PHP, moeten eindige met een punt-comma (;) zoals hier onder wordt aangegeven:

```
$leerling_leeftijd = 15;
```

Met behulp van de instructie **echo** wordt de waarde getoond in de browser.

```
echo $leerling_leeftijd;
```

Veel bruikbaarder is het wanneer de waarde van de variabele wordt getoond als onderdeel van een tekst string omsloten door dubbele aanhalingstekens, zoals:

```
echo "De leeftijd van Pedro is $leerling_leeftijd jaar.";
```

1

Maak het onderstaande HTML bestand in je teksteditor aan:

```
<!DOCTYPE HTML>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>
      Beginnen met PHP
    </title>
  </head>
  <body>
  </body>
</html>
```

- 2 Voeg de PHP tags toe binnen het body deel.

```
<?php
# hier komen PHP instructies
?>
```

- 3 Voeg nu de onderstaande regels toe tussen de PHP tags.

```
$leerling_leeftijd = 15;
echo $leerling_leeftijd;
echo "<br>";
echo "De leeftijd van Pedro is $leerling_leeftijd jaar.";
echo "<br>";
$leerling_leeftijd = 14;
echo "De leeftijd van 14 is $leerling_leeftijd jaar.";
```

OPMERKING: Mocht je teksten kopiëren zorg dan dat je in je tekst editor de aanhalingstekens aanpast. Word veranderd deze in slimme aanhalingstekens en dat werkt niet met PHP. Alleen rechte aanhalingstekens in PHP.

- 4 Sla het bestand op in de /htdocs map met de naam variabelen.php en open het script in je browser. Je ziet dan het onderstaande resultaat:

```
15
De leeftijd van Pedro is 15 jaar.
De leeftijd van 14 is 14 jaar.
```

Werken met strings

Een 'string' van tekst wordt op de zelfde manier als een numerieke waarde opgeslagen. Echter de toekenning moet omgeven worden door aanhalingstekens om het begin en het einde weer te geven. Geldige voorbeelden van een string toekenning zijn:

```
$lied_titel = "November Rain";
```

```
$lied_titel = 'November Rain';
```

Als je een tekst in een string wilt omringen door aanhalingstekens dan kun je dat doen door gebruik te maken van de backslash '\' om een

aanhalingsteken te tonen. Of je gebruikt de alternatieve manier zoals in de onderstaande voorbeelden:

```
$lied_titel = "Het lied \"November Rain\" door Guns n Roses";
```

```
$lied_titel = 'Het lied "November Rain" door Guns n Roses';
```

Waarden van een string kunnen worden getoond door gebruik te maken van de dubbele aanhalingstekens zoals hieronder wordt aangegeven:

```
echo "Velen zien $lied_titel als een populaire Rock balade.";
```

String waarden kunnen worden samengevoegd ('concatenated') in een enkele string door het gebruiken van een . 'punt'

```
$hallo = 'Hallo';
$doei = 'Doei';
$groet = $hallo . $doei; # 'HalloDoei'
```

Extra spaties en leestekens kunnen toegevoegd worden, zoals hieronder:

```
$groet = $hallo . ', '. $doei. '!'; # 'Hallo, Doe!
```

- 1 Tik het onderstaande script in je tekst editor in tussen de PHP tags. Gebruik daarbij het eerder gemaakte script.

```
<?php
# Instructies hier toevoegen
?>
```

```
$zin = 'De waarheid is zelden puur';
$schrijver = 'Oscar Wilde';
echo $zin;
echo "<p>Er wordt vaak gezegd dat <q>$zin<q></p>";
$zin = $zin. ' En nooit simpel';
echo "<p><q>$zin</q><cite>$schrijver</cite></p>";
```

- 2 Sla het document op in **/htdocs** als **string.php** en open het daarna in je browser. Je ziet dan het onderstaande resultaat.

De waarheid is zelden puur

Er wordt vaak gezegd dat "De waarheid is zelden puur"

"De waarheid is zelden puur En nooit simpel"*Oscar Wilde*

Arrays

Een array-variabele kan meerdere gegevens-items in sequentiële array-"elementen" opslaan die zijn genummerd ("geïndexeerd") beginnend bij nul.

Naar individuele array-elementwaarden kan worden verwezen met behulp van de variabele naam gevolgd door het indexnummer tussen vierkante haken. `$maanden[0]` verwijst bijvoorbeeld naar de waarde die is opgeslagen in het eerste arrayelement, `$maanden[1]` verwijst naar de waarde die is opgeslagen in het tweede arrayelement, enzovoort.

Variabele arrays kunnen worden gemaakt door meerdere instructies die een waarde toekennen, of door een enkele instructie die gebruikt maakt van de PHP functie `array()` om elementen te vullen.

```
$maanden[] = 'Januari';  
$maanden[] = 'Februari';  
$maanden[] = 'Maart';
```

```
$maanden = array('Januari', 'Februari', 'Maart');
```

Je kunt optioneel ook een sleutelnaam gebruiken om te refereren naar de verschillende elementen.

```
$maanden['jan'] = 'Januari';  
$maanden['feb'] = 'Februari';  
$maanden['mrt'] = 'Maart';
```

```
$maanden = array('jan' => 'Januari', 'feb' => 'Februari', 'mrt' =>  
'Maart');
```

Alle elementen van een array kunnen doorlopen worden met een `foreach` loop instructie waar elke element zijn waarde wordt toegekent aan een variabele:

```
foreach($maanden as $maand)  
{  
    # gebruik de huidige $maand waarde bij elke iteratie  
}
```

De sleutel kan ook aangeroepen worden tijdens elke iteratie van de lus.

```
Foreach($maanden as $sleutel => $maand)  
{  
    # gebruik de $sleutel en $maand waarden bij elke iteratie  
}
```

- 1 Maak een correct HTML5 document en tik het volgende script in tussen de PHP tags.

```
$dagen = array( 'Maandag', 'Dinsdag', 'Woensdag' );
foreach( $dagen as $waarde )
{
    echo "&bull; $waarde ";
}
$maanden = array( 'jan' => 'Januari',
                  'feb' => 'Februari', 'mrt' => 'Maart' );
echo '<dl>';
foreach( $maanden as $sleutel => $waarde )
{
    echo "<dt>$sleutel<dd>$waarde";
}
echo '</dl>';
```

- 2 Sla het document op in **/htdocs** met de naam **array.php** en open het script in je browser. Je hoort de onderstaande uitvoer te zien:

```
• Maandag • Dinsdag • Woensdag

jan
    Januari
feb
    Februari
mrt
    Maart
```

Arrays sorteren

PHP heeft twee handige functie voor conversies tussen arrays en strings.

De **implode()** functie converteert een array naar een string. De **explode()** functie verandert een string naar een array.

```
$csv_lijt = implode( ',', $array );
```

```
$array = explode( ',', $csv_lijt );
```

PHP heeft of functies voor het sorteren van een array.

Voor oplopend sorteren (a-z,1-9):

- **sort()** : Sorteert op waarde en haalt de originele sortering (key/sleutel) weg.

- **asort()** : Sorteert op waarde met behoudt van de originele sortering. (key/sleutel)
- **ksort()** : Sorteert op de sleutel (key)

Array elementen kunnen met een eenvoudige instructie worden gesorteerd:

```
$merk = array('Toyota', 'Hyundai', 'Isuzu');
Sort($merk);
```

Array elementen kunnen ook aflopend worden gesorteerd (z-a,9-1) met de volgende functies:

- **rsort()** : Sorteert op waarde en verwijderd de originele key/sleutel.
- **arsort()** : Sorteert op waarde en behoudt de originele key/sleutel.
- **krsort()** : Sorteert op de sleutel (key)

1 Maak een correct HTML5 document en tik het volgende script in tussen de PHP tags.

```
$autos = array('Toyota' => 'Tundra', 'Isuzu' => 'DMAX', 'Hyundai' =>
'Tucson');
echo '<dl><dt>Originele volgorde elementen: <dd>';
foreach($autos as $sleutel => $waarde)
{
    echo '&bull;','$sleutel.' ' '.$waarde;
}

asort($autos);
echo '<dt>Gesorteerd in waarde volgorde: <dd>';
foreach($autos as $sleutel => $waarde)
{
    echo '&bull;','$sleutel.' ' '.$waarde;
}

ksort($autos);
echo '<dt>Gesorteerd in sleutel volgorde: <dd>';
foreach($autos as $sleutel => $waarde)
{
    echo '&bull;','$sleutel.' ' '.$waarde;
}
echo '</dl>';
```

2 Sla het document op in **/htdocs** met de naam **sorteren.php** en open het script in je browser. Je hoort de onderstaande uitvoer te zien:

Originele volgorde elementen:

•Toyota Tundra•Isuzu DMAX•Hyundai Tucson

Gesorteerd in waarde volgorde:

•Isuzu DMAX•Hyundai Tucson•Toyota Tundra

Gesorteerd in sleutel volgorde:

•Hyundai Tucson•Isuzu DMAX•Toyota Tundra

Dimensies beschrijven

Het nummeren van array element is een 'array index'. PHP gebruikte het zero-base principe. Dit wil zeggen dat er altijd bij nul wordt begonnen.

Element waarde	A	B	C	D	E
Index nummers	[0]	[1]	[2]	[3]	[4]

Arrays kunnen ook meerdere indexen hebben – om meerdere dimensies te kunnen weer geven.

Eerste index	[0]	A	B	C	D	E
	[1]	F	G	H	I	J
Tweede index		[0]	[1]	[2]	[3]	[4]

In plaats van een index nummer kun je ook een sleutel naam gebruiken.

\$matrix['Letter'][0]

1 Maak een correct HTML5 document en tik het volgende script in tussen de PHP tags.

```
$letters = array( 'A' , 'B' , 'C' ) ;
$nummers = array( 1 , 2 , 3 ) ;
$matrix = array( 'Letter' => $letters , 'Number' => $nummers ) ;
echo "<p>Start : { $matrix['Letter'][0] } </p>" ;
foreach( $matrix as $array => $lijst )
{
    echo '<ul>' ;
    foreach( $lijst as $sleutel => $waarde )
    {
        echo "<li>$array [ $sleutel ] = $waarde " ;
    }
    echo '</ul>' ;
}
```

- 2 Sla het document op in **/htdocs** met de naam **matrix.php** en open het script in je browser. Je hoort de onderstaande uitvoer te zien:

```
Start : A

• Letter [ 0 ] = A
• Letter [ 1 ] = B
• Letter [ 2 ] = C

• Number [ 0 ] = 1
• Number [ 1 ] = 2
• Number [ 2 ] = 3
```

Rekenen

De tabel hier onder geeft de verschillende reken operatoren voor PHP weer.

Operator	Bewerking
+	Optellen
-	Aftrekken
*	Vermenigvuldigen
/	Delen
%	Modulus (restwaarde)
++	Vermeerdering
--	Vermindering

Alle standaard rekenregels zijn van toepassing. Let echter op de schrijfwijze. Duidelijke code voorkomt onnodig zoeken naar fouten.

a = b * c - d % e / f; # Dit is onduidelijk

a = (b * c) - ((d % e) / f); # Dit is duidelijk

Er zijn korte notaties voor expressies die handig kunnen zijn.

```
a += b;    # gelijk aan a = ( a + b );
a -= b;    # gelijk aan a = ( a - b );
a *= b;    # gelijk aan a = ( a * b );
a /= b;    # gelijk aan a = ( a / b );
```

- 1 Maak een correct HTML5 document en tik het volgende script in tussen de PHP tags.

```
$a = 4;
$b = 8;
$resultaat = $a + $b; echo "Optellen : $resultaat <br>";
$resultaat = $b - $a; echo "Aftrekken : $resultaat <br>";
$resultaat = $a * $b; echo "Vermenigvuldigen : $resultaat <br>";
$resultaat = $b / $a; echo "Delen : $resultaat <br>";
$resultaat = $a % $b; echo "Modulo : $resultaat <br>";
$a++; echo "Vermeerderen : $a <br>";
$b--; echo "Verminderen : $b <br>";
```

- 2 Sla het document op in **/htdocs** met de naam **arithmetic.php** en open het script in je browser. Je hoort de onderstaande uitvoer te zien:

```
Optellen : 12
Aftrekken : 4
Vermenigvuldigen : 32
Delen : 2
Modulo : 4
Vermeerderen : 5
Verminderen : 7
```

Waarden vergelijken

De onderstaande tabel geeft een overzicht van de operatoren die gebruikt worden om twee waarden te vergelijken in PHP.

Operator	Bewerking
==	Gelijk
!=	Ongelijk
>	Groter dan
<	Kleiner dan
>=	Groter dan of gelijk aan
<=	Kleiner dan of gelijk aan
?:	Waar of onwaar

De eerste 6 operatoren vergelijken twee operanden met elkaar en geeft een boolean (logische) waarde waar (true) of onwaar (false) terug.

```
6 > 3          # Dit is waar
7 <= 3         # Dit is onwaar
```

De ?: ternaire operator evalueert of een expressie waar of onwaar is en voer dan de bij behorende instructie uit.

`$variabele = ( expressie ) ? Als-waar-retour-dit:Als-onwaar-retour-dit;`

- 1 Maak een correct HTML5 document en tik het volgende script in tussen de PHP tags.

```
$nul = 0; $nil = 0; $een = 1; $hletter = 'A'; $kletter = 'a';
$resultaat = ($nul == $nil) ? 'TRUE' : 'FALSE';
echo "0 == 0 is $resultaat <br>";
$resultaat = ($nul == $een) ? 'TRUE' : 'FALSE';
echo "0 == 1 is $resultaat <br>";
$resultaat = ($hletter == $kletter) ? 'TRUE' : 'FALSE';
echo "A == a is $resultaat <br>";
$resultaat = ($hletter != $kletter) ? 'TRUE' : 'FALSE';
echo "A != a is $resultaat <br>";
$resultaat = ($een > $nil) ? 'TRUE' : 'FALSE';
echo "1 > 0 is $resultaat <br>";
$resultaat = ($nul >= $nil) ? 'TRUE' : 'FALSE';
echo "0 >= 0 is $resultaat <br>";
$resultaat = ($een <= $nil) ? 'TRUE' : 'FALSE';
echo "1 <= 0 is $resultaat <br>";
```

- 2 Sla het document op in **/htdocs** met de naam **comparison.php** en open het script in je browser. Je hoort de onderstaande uitvoer te zien:

```
0 == 0 is TRUE
0 == 1 is FALSE
A == a is FALSE
A != a is TRUE
1 > 0 is TRUE
0 >= 0 is TRUE
1 <= 0 is FALSE
```

Logica beoordelen

De onderstaande tabel geeft een overzicht van de logische operatoren die gebruikt worden in PHP.

Operator	Bewerking
&&	Logische AND
AND	Logische AND (alternatieve vorm van &&)
	Logische OR
OR	Logische OR (alternatieve vorm van)
!	Logische NOT

De logische operatoren worden gebruikt met operanden die een Boolean waarde hebben van TRUE of FALSE, of een expressie die kan converteren naar TRUE of FALSE.

- 1 Maak een correct HTML5 document en tik het volgende script in tussen de PHP tags.

```
$ja = TRUE; $nee = FALSE;
$resultaat = ($nee && $nee) ? 'TRUE' : 'FALSE';
echo "Nee AND Nee resulteerd in $resultaat <br>";
$resultaat = ($ja && $nee) ? 'TRUE' : 'FALSE';
echo "Ja AND Nee resulteerd in $resultaat <br>";
$resultaat = ($ja && $ja) ? 'TRUE' : 'FALSE';
echo "Ja AND Ja resulteerd in $resultaat <hr>";
$resultaat = ($nee || $nee) ? 'TRUE' : 'FALSE';
echo "Nee OR Nee resulteerd in $resultaat <br>";
$resultaat = ($ja || $nee) ? 'TRUE' : 'FALSE';
echo "Ja OR Nee resulteerd in $resultaat <br>";
$resultaat = ($ja || $ja) ? 'TRUE' : 'FALSE';
echo "Ja OR Ja resulteerd in $resultaat <hr>";
$resultaat = ( ! $ja ) ? 'TRUE' : 'FALSE';
echo "NOT Ja resulteerd in $resultaat <br>";
```

- 2 Sla het document op in **/htdocs** met de naam **logic.php** en open het script in je browser. Je hoort de onderstaande uitvoer te zien:

Nee AND Nee resulteerd in FALSE Ja AND Nee resulteerd in FALSE Ja AND Ja resulteerd in TRUE
Nee OR Nee resulteerd in FALSE Ja OR Nee resulteerd in TRUE Ja OR Ja resulteerd in TRUE
NOT Ja resulteerd in FALSE

Constanten definiëren

Constanten zijn container die een waarde bevat die niet veranderd. PHP constanten worden niet gemaakt op dezelfde manier als PHP variabelen. Er wordt een functie **define()** gebruikt om een constante te specificeren.

```
define('NAAM', waarde);
```

PHP kan het verschil tussen een constante naam en een normale tekst in hoofdletters niet onderscheiden.

```
define('GEBRUIKER', 'Juancho');  
echo "Bon dia GEBRUIKER";           # Toont: Bon dia GEBRUIKER
```

Er moet apart verwezen worden naar de waarde van de constante.

```
echo 'Bon dia '  
echo GEBRUIKER;
```

Of door concatenatie (samenvoeging):

```
echo 'Bon dia ' . GEBRUIKER;
```

- 1 Maak een correct HTML5 document en tik het volgende script in tussen de PHP tags.

```
define('GEBRUIKER','Juancho');  
echo '<p>Bon dia '.GEBRUIKER.'</P>';  
echo '<p>Je gebruikt PHP versie '.PHP_VERSION;  
echo ' op '.PHP_OS.'</p>';
```

- 2 Sla het document op in **/htdocs** met de naam **constant.php** en open het script in je browser. Je hoort de onderstaande uitvoer te zien:

Bon dia Juancho

Je gebruikt PHP versie 7.2.8 op WINNT