

# 2 PHP Simple Datatypes

## Variabelen

php variabelen beginnen altijd met \$ en zijn case sensitive. Verder mag je getallen en underscores gebruiken. In principe gebruiken we de Camelcase standaard.

Camelcase: elke variabele begint met een kleine letter en elk nieuw woord begint met een hoofdletter.

Voorbeeld: \$besteLeerling \$hoogsteCijfer of \$hoogsteCijferKlas

Je hoeft in PHP variabelen niet te declareren (zoals in JavaScript met var). Je kunt gewoon een variabele gebruiken. Variabelen zijn van het type string, integer of real. Het type wordt automatisch bepaald door PHP.

```
<?php
$stad="Parijs"; // $stad is een string
$aantal= 1;    // $aantal is een integer
$aantal="12";  // $aantal is een string
$prijs=1.95;   // $prijs is een real
$antwoord=False; // $antwoord is boolean (true of false)
?>
```

Over booleans wordt in de volgende les meer verteld.

## String Concatenate

In PHP plak je strings aan elkaar met een . (punt). Dit heet in het Engels string concatenate.

```
<?php
$naam="John";
echo "Welkom ".$naam;
?>
```

# Operator

Net als in JavaScript kent PHP min of meer dezelfde operatoren.

Operator	Beschrijving	Voor variabele type(n)	voorbeeld
+	optellen	int, real	\$a=\$a+\$b;
-	afrekken	int, real	\$a=\$a-4;
/	delen	int, real	\$x=12/3;
*	vermenigvuldigen	int, real	\$y=12*3;
.	concatenate	string	\$a="a"."b";
%	modulo	int, real	\$x=21%10;
**	machtsverheffen	int, real	\$y=2**3;
++	1 erbij optellen	int, real	\$a++;
--	1 er van afrekken	int, real	\$a--

## Typecasting

In PHP wordt het type variabele automatisch bepaald, maar als programmeur kun je een variabele een ander type geven. Dat noem je typecasting.

```
<?php
$myVar=12;
echo "Het nummer is ". (string)$myVar;

$nummer="125";
$nummmer=(int)$nummer+5;
echo "Het nummer is ".(string)$nummer;
?>
```

Typecasting gebeurt meestal vanzelf. PHP probeert te 'raden' wat je bedoeld. In sommige gevallen kan het echter nodig zijn om een variabele te casten.

Omdat je soms niet weet welk type een variabele is kun je met `var_dump()` het type het type opvragen.

```
<?php
$teller="0";
var_dump($teller);
```

```
echo "<br>";

$teller=$teller+1;
var_dump($teller);
?>
```

## Maar wanneer heb je typecasting nou echt nodig?

Omdat PHP zelf types omzet zul je typecasting niet vaak nodig hebben, maar onderstaand voorbeeld kun je het wel gebruiken:

```
$nummer=10;
echo 'De waarde is '. $nummer . ' en dat is goed';
```

Het type `$nummer` is `int`. Als je gewoon nummer afdruckt dan wordt het type van `$nummer` door PHP automatisch verandert naar een string. En dan plak je de drie strings aan elkaar. Niets aan de hand. Maar laten we een klein wijziging aanbrengen:

```
$nummer=10;
echo 'De waarde is '. $nummer+1 . ' en dat is goed';
```

Nu geef je PHP opdracht om 1 bij `$nummer` op te tellen. Dat betekent dat PHP `$nummer` zal (blijven) zien als een `int`. En een `int` kan je niet als een string aan elkaar plakken. Je kunt dit op verschillende manieren oplossen en type casting is er een van.

```
$nummer=10;
echo 'De waarde is '. (string)($nummer+1) . ' en dat is goed';
```

Opmerking: nu mag je zelfs `(string)` weer weglaten en alleen `$nummer+1` tussen haakjes zetten. Je voert dan hetzelfde uit maar het is minder duidelijk wat er dan precies gebeurt.

## En dan nog wat...

Er zijn nog een paar zaken die we niet heel erg duidelijk genoemd hebben. Maar dat komt omdat deze zaken hetzelfde zijn als dat we bij JavaScript hebben geleerd.

Zoals je hebt kunnen zien eindigt elke PHP-commando met een `;` net als in Javascript.

Je hebt in de voorbeelden ook gezien dat als je commentaar wilt toevoegen je `//` kan gebruiken.

Als je meer regels in commentaar wilt zetten dan kan je die tussen `/*` en `*/` zetten.

In de volgende lessen zullen we nog meer overeenkomsten met JavaScript tegenkomen.

We hebben ook het commando *echo* gezien dit is in JavaScript *document.write* maar in Unix Bash is het wel weer hetzelfde.

## Opgaven

1. Maak een PHP-programmaatje. Maak twee variabelen en geeft deze de waarden 12 en 10. tel deze variabelen bij elkaar op en druk het resultaat af.
2. Maak een PHP-programmaatje. Maak twee variabelen en geeft deze de waarden 12 en 10. vermenigvuldig deze variabelen en druk het resultaat af.
3. Maak een string en in PHP en zet daarin je naam. Druk daarna de boodschap af: "Welkom ...." op de plaats van de ... moet dan je naam worden afgedrukt.
4. Haal alle fouten uit de volgende code en schrijf de gecorrigeerde code op je antwoordblad.

```
<?php

// deze code bevat veel fouten
    haal alle fouten eruit!
//

$mijnNaam="Carl";
$jouwNaam=Camel;
echo $mijnNaam + " zit bij " + jouwNaam + " in de klas!"

?>
```

--

---

Revision #19

Created 19 January 2020 17:11:58 by Admin

Updated 1 February 2020 11:52:15 by Admin