



Web advanced

PHP

**DE HOGESCHOOL
MET HET NETWERK**

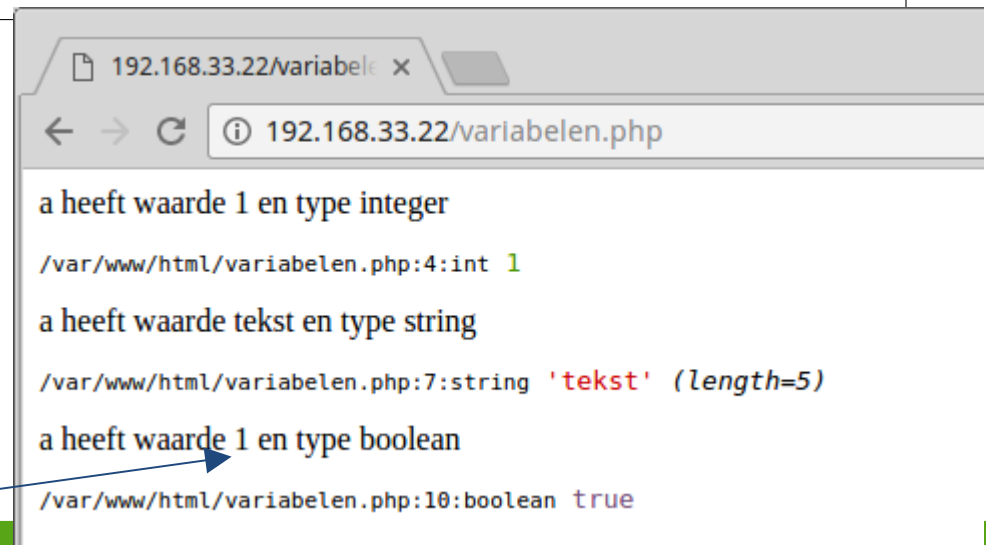
Hogeschool PXL – Dep. PXL-IT – Elfde-Liniestraat 26 – B-3500 Hasselt
www.pxl.be - www.pxl.be/facebook



PHP: variabelen

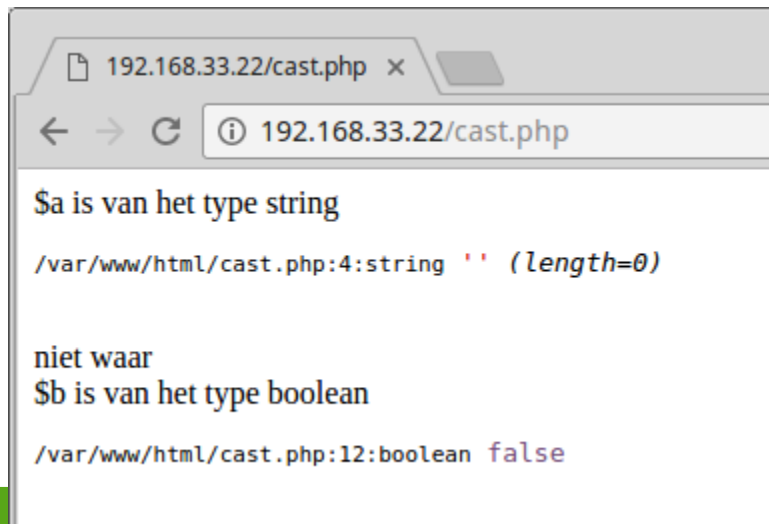
- aangeduid met \$
- datatype wordt bepaald door context

```
<?php
$a = 1;
print ('a heeft waarde ' . $a . ' en type ' . gettype($a) . '<br/>');
var_dump($a);
$a = "tekst";
print ('a heeft waarde ' . $a . " en type ' . gettype($a) . '<br/>');
var_dump($a);
$a = TRUE;
print ('a heeft waarde ' . $a . ' en type ' . gettype($a) . '<br/>');
var_dump($a);
```



PHP: cast

```
<?php
$a = "";
print ("\$a is van het type " . gettype($a));
var_dump($a);
$b = (boolean) $a;
if ( $b ) {
    print ("</br>waar <br/>");
} else {
    print ("<br/>niet waar <br/>");
}
print ( "\$b is van het type " . gettype($b));
var_dump ( $b );
```



PHP: cast

(boolean) 0 = false
(boolean) 0.0 = false
(boolean) "" = false
(boolean) "0" = false
(boolean) Null = false
(boolean) alle rest = true

(int) true	= 1	(float) true = 1.0
(int) false	= 0	(float) false = 0.0
(int) 12.33	= 12	(float) 12 = 12.0
(int) "12.22 aa"	= 12	(float) "12.22 aaaa" = 12.22
(int) "aa 12.222"	= 0	(float) "aaa 12.222" = 0.0
(int) Null	= 0	(float) Null = 0.0

(string) true = "1"
(string) false = ""
(string) 12 = "12"
(string) 12.222 = "12.222"
(string) Null = ""

PHP: operaties

$+$, $-$, $*$ LL of RL float (of een string die begint met een float):
beide worden als float beschouwd resultaat is float

ANDERS: LL en RL worden beschouwd als int en resultaat is int

$1 + \text{"a"} = 1$

$1 + \text{"12.22 a"} = 13.22$

$\text{"a"} + \text{"b"} = 0$

$/$ LL en RL worden altijd beschouwd als float en resultaat is float

$1 / 2 = 0.5$

$\text{"aaa"} / 2 = 0.0$

$\%$ modulo (rest na deling)

LL en RL worden altijd beschouwd als int en resultaat is int

$5 \% 2 = 1$

$\text{"a"} \% 2 = 0$

PHP: operaties

- . concatenatie: LL en RL worden beschouwd als string resultaat is string

"a" . "b" = "ab"

true . "a" = "1a"

PHP: operaties

==	gelijk in waarde
===	gelijk in waarde en van hetzelfde datatype
!=	verschillend in waarde
!==	verschillend in waarde of datatype
<	kleiner dan
>	groter dan
<=	kleiner dan of gelijk aan

```
<?php
var_dump(1=='1');
var_dump(1==='1');
```

```
jan@laptop-jan ~/Desktop/code/test $ php variabelen.php
/home/jan/Desktop/code/test/variabelen.php:2:
bool(true)
/home/jan/Desktop/code/test/variabelen.php:3:
bool(false)
```



PHP: control structures

```
<?php
$a=0;
if ( $a< 1 ) {
    print("$a is kleiner dan 1\n");
} else {
    print("$a is groter dan 1\n");
}

while( $a< 10 ) {
    print($a);
    $a++;
}

for ( $i = 0 ; $i < 10 ; $i++ ){
    print($i);
}
```


PHP: arrays

array bevat key, value pairs key int of string
value elk datatype

aangemaakt via array (arguments)

argument: key => value
of
value (eerst vrije int key wordt genomen)

```
<?php
$a = array ( 1, 2, 3, 4 );
$b = array ( 1, 2.1, true, "ja" );
$c = array ( 1 => 12, "Juist" => true );
$d = array ( 2 => 1, 10 );
```

	192.168.33.22/arrays.php
\$a = array(1,2,3,4);	<pre>/var/www/html/arrays.php:9: array (size=4) 0 => int 1 1 => int 2 2 => int 3 3 => int 4</pre>
\$b = array(1,2.1,true,"ja");	<pre>/var/www/html/arrays.php:12: array (size=4) 0 => int 1 1 => float 2.1 2 => boolean true 3 => string 'ja' (length=2)</pre>
\$c = array(1=> 12, "Juist" => true);	<pre>/var/www/html/arrays.php:15: array (size=2) 1 => int 12 'Juist' => boolean true</pre>
\$d = array (2 => 1, 10);	<pre>/var/www/html/arrays.php:18: array (size=2) 2 => int 1 10 => int 10</pre>

PHP: arrays

gebruik: array [key]

```
<?php
$a = array(1, 2, 3, 4);
echo "$a[0]\n";
$a[0] = "ja";
$b = array ( "een" => 1, "twee" => 3 );
echo $b["een"] . "\n";
echo "$b[een]\n";
$b["twee"] = 2;
```

waarden bijplaatsen

```
<?php
$a = array(1, 2, 3, 4);
$a[4] = 5;
$a[] = 6;    // eerste vrije int key
```

PHP: arrays

<code>count(\$a)</code>	geef het aantal waarden in de rij \$a
<code>array_keys(\$a)</code>	geef een rij met alle keys van \$a
<code>array_keys(\$a, val)</code>	geef een rij met de keys die overeenkomen met de waarde val (zoekfunctie)
<code>array_values(\$a)</code>	geef een rij met alle values van \$a
<code>sort(\$a)</code>	sorteer de rij \$a
<code>shuffle(\$a)</code>	schud de rij \$a
<code>min(\$a)</code>	minimum van de rij \$a
<code>max(\$a)</code>	maximum van de rij \$a

PHP: foreach

```
<?php
$a = array(1 => "ma", 2 => "di", 3 => "wo", 4 => "ma");
foreach ($a as $v) {
    echo "value: $v \n";
}
```

```
jan@laptop-jan ~/Desktop/code/test$ php foreach.php
value: ma
value: di
value: wo
value: ma
```

PHP: foreach

```
<?php
$a = array(1 => "ma", 2 => "di", 3 => "wo", 4 => "ma");
foreach ($a as $k=>$v) {
    echo "key: $k, value: $v \n";
}
```

```
key: 1, value: ma
key: 2, value: di
key: 3, value: wo
key: 4, value: ma
jan@laptop-jan ~/Desktop/code/test $ php foreach.php
```

PHP: functions

```
<?php
function som ( $getal1, $getal2 ) {
    return $getal1 + $getal2;
}

function printResultaat ( $resultaat ) {
    print ( $resultaat );
}

$resultaat = som ( 1 , 2 );
printResultaat ( $resultaat );
```

PHP: functions

Elke naam v. function mag 1 keer gebruikt worden.

Dus niet: definitie van som met 2 waarden, som met 3 waarden, ...

Oplossing 1 (default waarde)

```
<?php
function som ( $getal1, $getal2, $getal3 = 0 ) {
    return $getal1 + $getal2 + $getal3;
}

$resultaat = som ( 1 , 2 );
print ( $resultaat );
$resultaat = som ( 1 , 2 , 3);
print ( $resultaat );
```

PHP: functions

Oplossing 2 (func_get_args)

```
<?php
function som ( ) {
    $som=0;
    $argumenten = func_get_args();
    foreach ($argumenten as $argument) {
        $som += $argument;
    }
    return $som;
}

$resultaat = som ( 1 , 2 );
print ( $resultaat );
$resultaat = som ( 1 , 2 , 3, 7, 9);
print ( $resultaat );
```


PHP: functions

Oplossing 3 (splat operator ...)

```
<?php
function som ( ...$getallen ) {
    $som=0;
    foreach ( $getallen as $getal ) {
        $som += $getal;
    }
    return $som;
}

$resultaat = som ( 1 , 2 );
print ( $resultaat );
$resultaat = som ( 1 , 2 , 3, 7, 9);
print ( $resultaat );
```

PHP: functions

Type hinting: PHP 5.6 (enkel voor array en objects)

```
<?php
function printDatum ( DateTime $date ) {
    print ($date->format('Y-m-d H:i:s')) ;
}

function printRij ( array $rij ){
    foreach ($rij as $element){
        print($element."\n");
    }
}

$date=new DateTime();
printDatum ( $date );

$rij=[1,2,3];
printRij ( $rij );
```

PHP: functions

Type hinting: PHP 7 (ook voor int, float, string, boolean & returntype)

```
<?php
function som (int $getal1, int $getal2 ) : int {
    return $getal1 + $getal2;
}

$resultaat = som ( 1 , 2 );
print ( $resultaat );
```

PHP: voorgedef. functions

String functies

<code>strlen(\$s)</code>	lengte van de string \$s
<code>strpos (\$s, \$z)</code>	eerste positie van een zoekstring \$z in de string \$s false als \$z niet gevonden wordt
<code>substr(\$s, \$i)</code>	substring van de string \$s, beginnende vanaf positie \$i
<code>strtolower (\$s)</code>	geeft de string \$s omgezet naar kleine letters terug
<code>strtoupper (\$s)</code>	geef de string \$s omgezet naar hoofdletters
<code>trim(\$s)</code>	geef de string \$s zonder spaties voor en achteraan

Functies voor variabelen

<code>define("PI", 3.1415)</code>	definieer een constante
<code>unset (\$a)</code>	verwijder de variabele \$a
<code>isset(\$a)</code>	bestaat de variabele \$a?
<code>is_bool(\$a)</code>	is de variabele \$a een boolean?
<code>is_double(\$a)</code>	double?
<code>is_string(\$a)</code>	string?

`get_type($a)` geef het type van het variabele \$a

PHP: voorgedef. functions

require_once: plaats de inhoud van een bestand op deze locatie
niet gevonden --> foutmelding
_once --> het plaatsen v.h. bestand wordt maar 1 keer gedaan

Math.php

```
<?php
function som (int $getal1,int $getal2 ):int {
    return $getal1 + $getal2;
}
```

Som.php

```
<?php
require_once ( 'math.php' );
$resultaat = som ( 1, 2 );
print ( $resultaat ) ;
```