

# PHP VOOR BEGINNERS

1. Voorstelling
2. Inleiding
3. Basisprincipes van PHP
4. Werken met variabelen
5. Controlestructuren
6. MySQL
7. Doe-het-zelf



# *INLEIDING: Wat is PHP?*

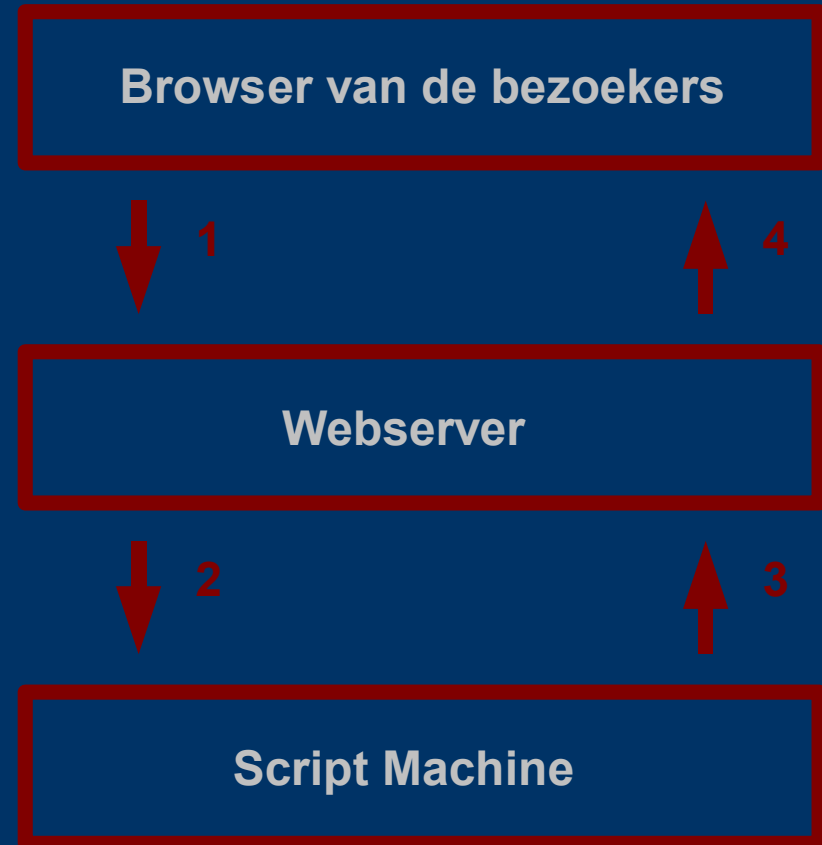
- Een programmeertaal voor dynamische pagina's
  - Voornamelijk gebruikt voor webpagina's te maken
  - PHP = Hypertext Preprocessor
  - Is een server-side scripting taal  
(vgl. html is client-side)
- 
-

# INLEIDING: Server-Side

## Client-Side



## Server-Side



# *INLEIDING: Waarom PHP?*

- **Waarom PHP?**
  - PHP is volledig gratis
  - PHP is redelijk eenvoudig en snel te leren
  - PHP is zeer populair
  - Meer documentatie beschikbaar
  - PHP is redelijk snel
  - ...
- 
-

# *BASISPRINCIPES: php-code*

- Begin php-code: `<?php`

Einde php-code: `?>`

- Commentaarblokken

`// hier uw commentaar`

`/* hier uw commentaar */`

Vergelijking met html:

`<!-- Hier uw commentaar -->`

(verschil voor de bezoeker: zichtbaarheid)

- Output met `echo`



# BASISPRINCIPES: Hello World

Het eerste php-bestand: *HelloWorld.php*

- `<?php`
- `// Openen van de php-code mbv <?php`
- 
- `// Hello World op het scherm zetten`
- 
- `echo "Hello World";`
- 
- `// Een regel wordt afgesloten met ;`
- 
- `/* Afsluiten van de php-code mbv ?> */`
- `?>`

# BASISPRINCIPES: Variabelen

- Wat zijn variabelen?  
*plaatsen in het geheugen waar je tijdens de uitvoering van je programma gegevens kan bewaren*
- Hoe worden ze aangeduid?  
*In php wordt de variabelen voorafgegaan door \$.*
- Hoe samenvoegen?  
*Variabelen worden samengevoegd mbv een punt.*
- Voorbeeld

```
<?php
$var = 2;           // In $var zit nu het getal twee
echo $var;         // zal het getal twee op het scherm zetten
?>
```

# BASISPRINCIPES: html en php

- Het gebruik van html samen met php

## Voorbeeld 1

```
<?php
    $var = "vet";
    echo "<b>";
    echo $var;           // Of korter op één regel echo "<b>".$var."</b>";
    echo "</b>";
?>
```

- Voorbeeld 2

```
<b>
<?php
    $var = "vet";
    echo $var;
?>
</b>
```

# ***BASISPRINCIPES: Oefening***

- Maak een php-bestand waarbij:
    - ✓ Sla een nummer op in een desbetreffende variabele
    - ✓ Je eigen naam op het scherm komt
    - ✓ Print de nummer uit op het scherm.
    - ✓ Maak een variabele aan met een woord
    - ✓ Maak een tweede variabele aan met een ander woord
    - ✓ Print deze twee variabele achter elkaar af.
- 
-

# VARIABELEN: Soorten

- Enkelvoudige types:

- String

```
$var = "een woord";
```

- Integer

```
$var = 34;
```

- Floating Points

```
$var = 2.5;
```

- Boolean

```
$var = TRUE; // of anders FALSE
```

- Meervoudige types:

- Array

```
$var = array("woord1", "woord2", ...);
```

```
$var[2] = "woord3";
```

```
echo $var[1]; // geeft woord2 weer op het scherm
```

- Objects (bij object-georiënteerd programmeren)
- 
-

# VARIABELEN: String

- Tussen quotes: “ of ’  
Wat mij een string met een quotes: \” of \’
- Samenvoegen met .
- Voorbeeld:

```
<?php
    $a = "Variabele 1";
    $b = 'Variabele 2\n'; // \n = nieuwe lijn toevoegen
    $c = "Variabele 3 met een \" quote";
    $d = "Variabele 4 met een ' accent\n";
    $e = $a.$b.$c.$d; // alle string achter elkaar
    echo $e;
?>
```

Geeft als output:

Variabele 1Variabele 2

Variabele 3 met een " quoteVariabele 4 met een ' accent

---

---

# VARIABELEN: Integers

- Opslaan van een geheel getal
- Negatief getal aanduiden met een -
- Wiskundige operaties
  - + , - , \* , / , %
- Voorbeeld

```
<?php
$a = 2;
$b = 5;
$c = $a + $b; // c = 7
$d = $a - $b; // d = -3
$e = $a * $b; // e = 10
$f = $a / $b; // f = 0.4 is een floating point, geen integer!!!
$g = $b % $a; // g = 2, een gehele deling
$h = $a.$b; // h = 25
?>
```

# *VARIABELEN: Floating Points*

- Helemaal analoog als de integers, enkel met kommagetallen
- Voorbeeld

```
<?php  
    $a = 2.5;  
?>
```

# VARIABELEN: Boolean

- Kan maar twee waarden hebben:  
TRUE of FALSE

- Relatieve operatoren  
&& , || , !

- Voorbeeld

```
<?php
    $a = TRUE;
    $b = FALSE;
    $c = $a || $b;    // c = TRUE, de OR-operator
    $d = $a && $b;    // d = FALSE, de AND-operator
    $e = !$a;        // e = FALSE, de negatie
?>
```

- Kunt ook AND gebruiken ipv &&
  - Kunt ook OR gebruiken ipv ||
- 
-

# VARIABELEN: Array's

- Meervoudige variabelen van (verschillende) eenvoudige types
- Eerste element heeft index 0.
- Voorbeeld

```
<?php
    $a = array("woord" , 1 , TRUE);
    echo $a[0]; // print woord uit, het eerste element
    echo $a[1]; // print het getal 1 uit, het tweede element
    echo $a[2]; // print TRUE uit, het derde element
?>
```

- Kan ook zonder index, maar met bv een string

```
<?php
    $b = array("eerste" => "woord", 2 => "test");
    echo $b["eerste"];
    echo $b[2];
?>
```

---

---

# VARIABELEN: Doorgeven

- Intern
    - > Binnen één bestand
    - > Tussen bestanden mbv  
`include()` of `require()`
  - Extern
    - > `$_GET`
    - > `$_POST`
    - > `$_SERVER`
    - > `$_SESSION`
    - > `$_COOKIE`
- 
-

# VARIABELEN: Intern doorgeven

- Binnen een bestand  
  zie vorige voorbeelden
- Tussen bestanden

Voorbeeld:

- bestand.php

```
<?php
    $a = "test";
?>
```

- globaal.php

```
<?php
    include('bestand.php');
    echo $a; // geeft test
?>
```

# VARIABELEN: Extern doorgeven

- GET-methode

Voorbeeld bestand.php

```
<?php  
    echo $_GET['var'];  
?>
```

Bij het bezoeken van *bestand.php?var=naam*  
geeft *naam* terug

- POST-methode  
 vooral gebruikt bij het invullen van formulieren  
 Voorbeeld 

# VARIABELEN: POST-methode

bestand\_verzenden.php

```
<?php
    echo $_POST['naam'];
?>
```

bestand.php

```
<form action="bestand_verzenden.php" method="post">
    <input type="text" name="naam" />
    <input type="submit" name="submit" value="verzenden" />
</form>
```

Bij het verzenden van het formulier wordt *bestand\_verzenden.php* aangeroepen. De inhoud van het formulier bevindt zich in de variabelen `$_POST` (een array!!!) (Kan ook met `get`, bij het aanpassen van het *method* bij `<form>`)

# VARIABELEN: SERVER-methode

- De variabele `$_SERVER` bevat informatie over
  - > De bezoeker (ip, browser, os, ...)
  - > De webpagina (url, header, ...)
  - > ...
- 
- Opgelet: Kan manueel door bezoeker gewijzigd worden (beveiliging!!!)
- Voorbeeld
- ```
<?php
echo $_SERVER['REMOTE_ADDR']; // ip-adres
echo $_SERVER['HTTP_USER_AGENT']; // browser
echo $_SERVER['PHP_SELF']; //naam van php-bestand
// Nog meer: zie php.net, google, ...

?>
```

# VARIABELEN: Session-methode

- Wat is een Sessie?

Een sessie onthoudt gegevens van een bezoeker. Elke bezoeker krijgt een eigen sessie toegewezen.

De sessie blijft beschikbaar zolang de browser waarin de pagina wordt bekeken open staat.

Wordt de browser afgesloten, is de verwijzing naar de sessie verloren.

- Hoe een sessie starten?

Plaats helemaal bovenaan de pagina `session_start()`;

- Hoe ziet een sessie-variabele eruit?

```
$_SESSION['naam'] = "inhoud";
```



# VARIABELEN: Cookie-methode

- Wat is een cookie?

Een cookie is een bestand op jouw computer, afkomstig van een website.

Een cookie bevat gegevens over die website.

- Hoe maak je een cookie aan?

```
Setcookie('naam', 'waarde', tijd, '/');
```

- Hoe vraag je de inhoud van een cookie op?

```
$_COOKIE['naam'];
```

---

---

# CONTROLE-INSTRUCTIES

- Controle-instructies
    - > Een blok code wel of niet uitvoeren
    - > if-elsif-else, switch
  - Lus-instructies
    - > Een blok code ev meermaals uitvoeren
    - > while, for, ...
  - 
  - Beide instructies mbv controle via boolean.
- 
-

# INSTRUCTIES: Boolean

- Boolean kan TRUE of FALSE zijn
- Gebruik maken van vergelijkingoperatoren

> == , != , <= , >= , < , >

```
<?php
$a = 2;
$b = "string";
/*      ($a == 2) geeft TRUE
        ($a < 2) geeft FALSE
        ($b != "Strin") geeft TRUE      */
?>
```

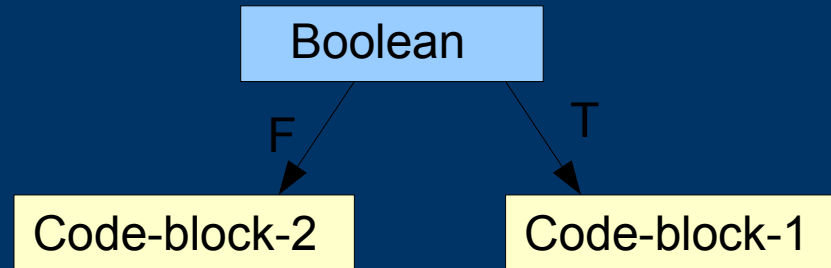
- Extra controle van type met ===, !==

```
<?php
$c = "2";
/*      ($a === 2) geeft TRUE
        ($c === 2) geeft FALSE
        ($c == 2)  geeft TRUE      */
?>
```

# CONTROLE-INSTRUCTIES: *if*

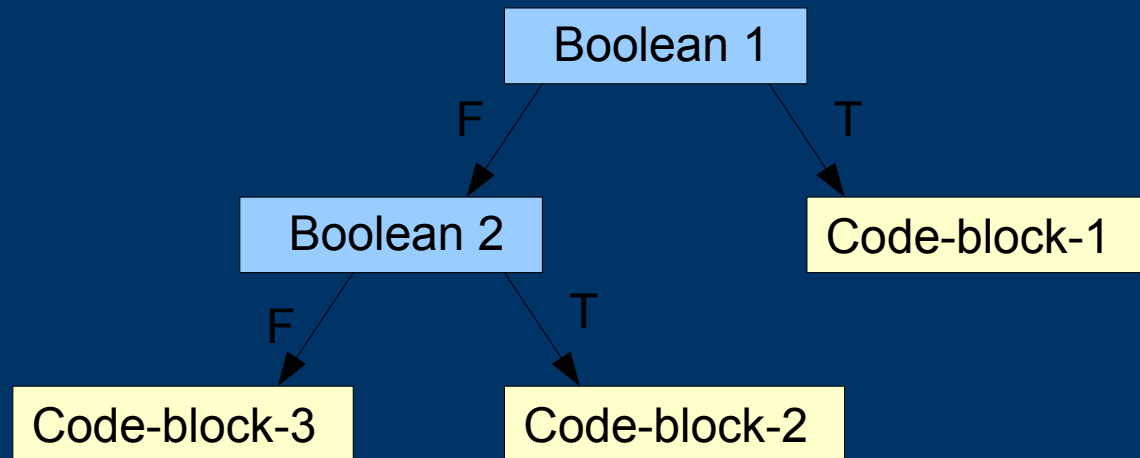
- If – else constructie

```
if (boolean) {  
    code-block-1;  
} else {  
    code-block-2;  
}
```



- If – elsif – else constructie

```
if (boolean1) {  
    code-block-1;  
} elsif (boolean2) {  
    code-block-2;  
} else {  
    code-block-3;  
}
```



# CONTROLE-INSTRUCTIES: *switch*

## Switch – constructie

```
switch($var) {  
    case waarde-1: code-block-1; break;  
    case waarde-2: code-block-2; break;  
    ...  
    default: code-block-default; break;  
}
```

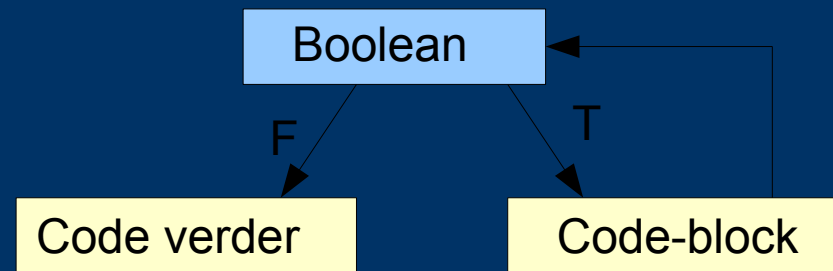
## Is hetzelfde als

```
if ($var == waarde-1) {  
    code-block-1;  
} elseif ($var == waarde-2) {  
    code-block-2;  
} else {  
    code-block-default;  
}
```

# LUS-INSTRUCTIES: *while*

- Mogelijk tot een code-block meermaals (of niet) uit te voeren

```
while(boolean) {  
    code-block;  
}
```

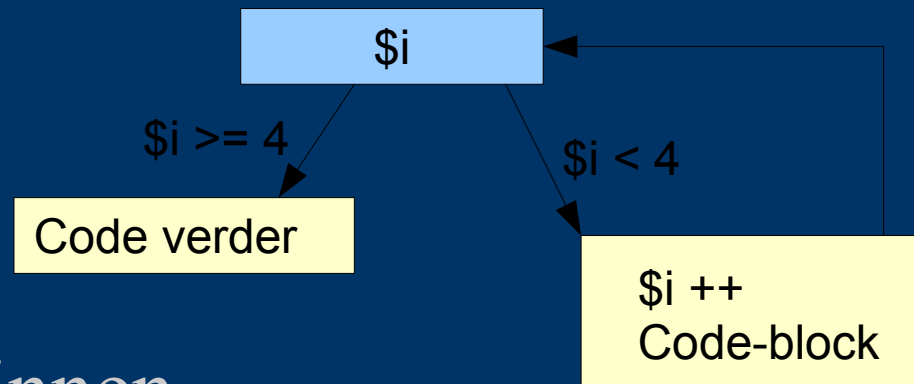


- Boolean, indien TRUE, moet in het code-block gewijzigd worden, anders krijg je een oneindige lus.

# LUS-INSTRUCTIES: for

- Mogelijk tot een code-block meermaals (of niet) uit te voeren

```
for($i = 0; $i < 4; $i++) {  
    code-block;  
}
```



- Moet niet vanaf 0 beginnen
- Kan ook  $\leq$  ,  $>$  ,  $\geq$  zijn
- Kunt ook \$i verminderen, mbv  $\$i--;$

# *MySQL: Wat?*

- Database: opslaan van gegevens
  - Vergelijkbaar met  
    Microsoft Access, OpenOffice.Org Base, ...
  - Informatie opslaan, bewerken, verwijderen, ...  
    met behulp van query's
  - De meest gebruikte database met php
  - Gemakkelijk in gebruik
- 
-

# *MySQL: PHPMysqlAdmin*



- Handig bij communiceren van database
- Gebruikt voor het aanmaken van databases
- Redelijk snel
- ...



# MySQL: Verbinding maken

## Verbinding maken met de server

```
mysql_connect("server", "gebruikersnaam", "wachtwoord");
```

*Opm.: Server is meestal localhost*

## Verbinding maken met de database

```
mysql_select_db("naam");
```

## Voorbeeld:

```
mysql_connect("localhost", "scoutnet", "password") or  
die("Kan niet verbinden: ".mysql_error());  
mysql_select_db("test_database") or  
die("Kan de database niet vinden");
```

# *MySQL: Een tabel aanmaken*

Een live voorbeeld met PHPMyAdmin

Voorbeeld:

Een tabel voor nieuwsberichten



# MySQL: Gegevens toevoegen

Gebruikt met de INSERT-query

```
INSERT INTO tabel (veld1, veld2, ...)  
VALUES (waarde1, waarde2, ...)
```

## Voorbeeld

```
<?php  
$titel = "Eerste bericht";  
$bericht = "Hallo iedereen,\n het allereerst nieuwsbericht";  
mysql_query("INSERT INTO nieuws (titel, bericht)  
VALUES ('.$titel.', '$bericht.');"");  
?>
```

# MySQL: Gegevens toevoegen

Wat bij slashes?

```
<?php
    $test = "zever ' gezever";
    mysql_query("INSERT INTO nieuws (titel) VALUES ('".$test."");");
    /* geeft de volgende query
       INSERT INTO nieuws (titel) VALUES('zever ' gezever');
       en dus ook een error, doordat de slashes "verkeerd" staan
    */
?>
```

Hoe op te lossen? => Gebruik maken van `addslashes()`

```
<?php
    $test = "zever ' gezever";
    mysql_query("INSERT INTO nieuws (titel)
                 VALUES ('".addslashes($test)."");");
    /* geeft de volgende query
       INSERT INTO nieuws (titel) VALUES('zever \' gezever');
    */
?>
```

# MySQL: Gegevens ophalen

Gebruikt met de SELECT-query

*SELECT veld1, veld2, ... FROM tabel*

*Opm.: SELECT \* FROM tabel selecteert alle velden*

## Voorbeeld

```
<?php
    $select_query = mysql_query("SELECT titel, bericht FROM nieuws");
    while($item = mysql_fetch_array($select_query)) {
        echo $item['titel']." : ".$item['bericht'];
    }
?>
```

Gebruik `stripslashes()` wanneer er `addslashes()` werd gebruikt

Wijziging van voorbeeld

```
echo stripslashes($item['titel'])." : ".stripslashes($item['bericht']);
```

---

---

# MySQL: Gegevens ophalen

- WHERE

*SELECT \* FROM tabel WHERE veld = 'waarde'*

Selecteert enkel de velden waarbij *veld = waarde*

- ORDER BY

*SELECT \* FROM tabel ORDER BY veld ASC*

Sorteert de resultaten volgens veld

ASC = ascending order

DESC = descending order

- LIMIT

*SELECT \* FROM tabel LIMIT 0, 10*

Geeft enkel de eerste 10 resultaten, beginnend bij de eerste (0).



# MySQL: Gegevens wijzigen

Gebruikt met de UPDATE-query

```
UPDATE tabel SET  
    veld1 = waarde1, veld2 = waarde2, ...  
[WHERE veld = waarde]  
[LIMIT 1]
```

## Voorbeeld

```
<?php  
$titel = "Eerste nieuwsbericht";  
$bericht = "Nieuw bericht";  
mysql_query("UPDATE nieuws SET  
    bericht = '".addslashes($bericht)."  
    WHERE titel = '$titel.'");  
?>
```

# MySQL: Gegevens verwijderen

Gebruikt met de DELETE-query

```
DELETE FROM tabel  
[WHERE veld = waarde]  
[LIMIT 1]
```

## Voorbeeld

```
<?php  
$titel = "Eerste nieuwsbericht";  
mysql_query("DELETE FROM nieuws  
            WHERE titel = '$titel.'");  
?>
```

# *DOE-HET-ZELF*

- Maak een gastenboek
- Berichten weergeven
- Berichten posten
- ... (ev leuke extra'tjes)



# *VRAGEN ACHTERAF?*

- PHP Documentation: <http://www.php.net>
- MySQL Documentation: <http://dev.mysql.com>
- Scoutnet Forum: <http://forum.scoutnet.be>
- Google
- ...



EINDE

