

# Arrays en Foreach

Huub de Beer

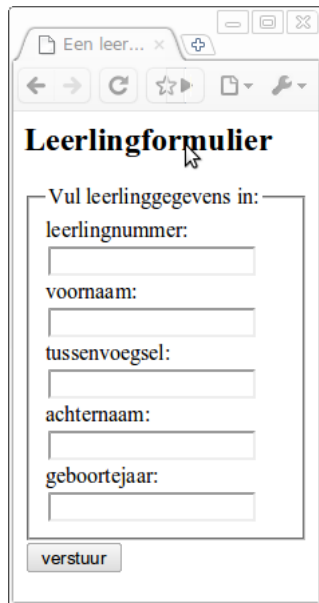
Eindhoven, 4 juni 2011

# Associatieve arrays (I)

- ▶ Je kent er al een aantal: `$_POST` en `$_GET`.
- ▶ PHP maakt er meer aan: `$_SERVER`, `$_SESSION`.
- ▶ Zie <http://www.php.net/manual/en/reserved.variables.php> voor een lijst met al deze associatieve arrays.

# Associatieve arrays (Ila): maak ze zelf in HTML

Voer informatie van een leerling in:



The image shows a browser window with a single tab titled "Een leer...". The address bar and navigation buttons are visible. The main content area displays the title "Leerlingformulier" in a large, bold font. Below the title, a mouse cursor is pointing at the text. The form is enclosed in a rectangular border and contains the following elements:

- A header text: "Vul leerlinggegevens in:"
- A label "leerlingnummer:" followed by a text input field.
- A label "voornaam:" followed by a text input field.
- A label "tussenvoegsel:" followed by a text input field.
- A label "achternaam:" followed by a text input field.
- A label "geboortejaar:" followed by a text input field.
- A "verstuur" button at the bottom left of the form area.

# Associatieve arrays (11b): maak ze zelf in HTML

Een stukje HTML code:

```
1 ...
2 <form action="..." method="POST">
3   <fieldset>
4   <legend>Vul leerlinggegevens in:</legend>
5   leerlingnummer: <input type="text" name="leerling[l1nr]"><br>
6   voornaam: <input type="text" name="leerling[voornaam]"><br>
7   tussenvoegsel: <input type="text" name="leerling[tussenvoegsel]"><br>
8   achternaam: <input type="text" name="leerling[achternaam]"><br>
9   geboortjaar: <input type="text" name="leerling[geboortjaar]">
10  </fieldset>
11  <input type="submit" value="verstuur">
12 </form>
13 ...
```

Let op

- ▶ naam van de control geschreven als een PHP array met unieke sleutel
- ▶ maar zonder de aanhalingstekens!

# Associatieve arrays (11b): maak ze zelf in HTML

Een stukje HTML code:

```
1 ...
2 <form action="..." method="POST">
3   <fieldset>
4   <legend>Vul leerlinggegevens in:</legend>
5   leerlingnummer: <input type="text" name="leerling[l1nr]"><br>
6   voornaam: <input type="text" name="leerling[voornaam]"><br>
7   tussenvoegsel: <input type="text" name="leerling[tussenvoegsel]"><br>
8   achternaam: <input type="text" name="leerling[achternaam]"><br>
9   geboortejaar: <input type="text" name="leerling[geboortejaar]">
10  </fieldset>
11  <input type="submit" value="verstuur">
12 </form>
13 ...
```

## Let op

- ▶ naam van de control geschreven als een PHP array met unieke sleutel
- ▶ maar zonder de aanhalingstekens!

# Associatieve arrays (III): hoe te gebruiken?

verwerk.php:

```
1 <?php
2 // Associatieve array inlezen
3 $leerling = $_POST['leerling'];
4 // Waarden gebruiken
5 $leeftijd = 2009 - $leerling['geb_jaar'];
6 // Waarden aanpassen
7 $leerling['naam'] = "Jan Jaap Jozef";
8 // Nieuwe waarden toevoegen
9 $leerling['klas'] = "3va";
10 // ...
11 ?>
```

- ▶ Eerst inlezen uit \$\_POST
- ▶ Sleutels overnemen uit HTML formulier, nu met aanhalingstekens!

# Associatieve arrays (III): hoe te gebruiken?

verwerk.php:

```
1 <?php
2 // Associatieve array inlezen
3 $leerling = $_POST['leerling'];
4 // Waarden gebruiken
5 $leeftijd = 2009 - $leerling['geb_jaar'];
6 // Waarden aanpassen
7 $leerling['naam'] = "Jan Jaap Jozef";
8 // Nieuwe waarden toevoegen
9 $leerling['klas'] = "3va";
10 // ...
11 ?>
```

- ▶ Eerst inlezen uit \$\_POST
- ▶ Sleutels overnemen uit HTML formulier, nu met aanhalingstekens!

# Associatieve arrays (IV): aanmaken in code

## De PHP functie array()

```
1 <?php
2 // We maken dezelfde array $leerling aan. Het verschil: we moeten
3 // nu *zelf* waarden geven en niet de gebruiker.
4 $leerling = array(
5     'naam' => 'Joop Jaapsens',
6     'lnr' => '012314',
7     'geb_jaar' => 1989 );
8 ?>
```

- ▶ standaardvorm: `array( key_1 => val_1, key_2 => val_2, ...)`
- ▶ Sleutels zijn strings (aanhalingstekens) of integers
- ▶ Waarden mogen elk willekeurig type hebben, zelfs arrays



# Associatieve arrays (IV): aanmaken in code

## De PHP functie array()

```
1 <?php
2 // We maken dezelfde array $leerling aan. Het verschil: we moeten
3 // nu *zelf* waarden geven en niet de gebruiker.
4 $leerling = array(
5     'naam' => 'Joop Jaapsens',
6     'lnr'  => '012314',
7     'geb_jaar' => 1989 );
8 ?>
```

- ▶ standaardvorm: **array**( key\_1 => val\_1, key\_2 => val\_2, ...)
- ▶ Sleutels zijn strings (aanhalingstekens) of integers
- ▶ Waarden mogen elk willekeurig type hebben, zelfs arrays

## Gewone arrays: zonder sleutels → indices

- ▶ Niet geïnteresseerd in sleutels: gewoon een lijst met waarden
- ▶ PHP maakt automatisch sleutels aan: de getallen 0, 1, 2, ...
- ▶ We noemen zo'n automatisch aangemaakte sleutel een *index*
- ▶ Indices beginnen bij 0! Dus niet bij 1.

### HTML

```
1 ...  
2 <input type="text" name="punten[]"><br>  
3 <input type="text" name="punten[]"><br>  
4 <input type="text" name="punten[]"><br>  
5 <input type="text" name="punten[]"><br>  
6 ...
```

### PHP

```
1 <?php  
2 $punten = array( 3.5, 6.5,  
3   7.6, 5.4, 9.5, 8.2 );  
4 ?>
```

## Gewone arrays: zonder sleutels → indices

- ▶ Niet geïnteresseerd in sleutels: gewoon een lijst met waarden
- ▶ PHP maakt automatisch sleutels aan: de getallen 0, 1, 2, ...
- ▶ We noemen zo'n automatisch aangemaakte sleutel een *index*
- ▶ Indices beginnen bij 0! Dus niet bij 1.

### HTML

```
1 ...  
2 <input type="text" name="punten[]"><br>  
3 <input type="text" name="punten[]"><br>  
4 <input type="text" name="punten[]"><br>  
5 <input type="text" name="punten[]"><br>  
6 ...
```

### PHP

```
1 <?php  
2 $punten = array( 3.5, 6.5,  
3   7.6, 5.4, 9.5, 8.2 );  
4 ?>
```

# Gewone arrays: zonder sleutels → indices

- ▶ Niet geïnteresseerd in sleutels: gewoon een lijst met waarden
- ▶ PHP maakt automatisch sleutels aan: de getallen 0, 1, 2, ...
- ▶ We noemen zo'n automatisch aangemaakte sleutel een *index*
- ▶ Indices beginnen bij 0! Dus niet bij 1.

## HTML

```
1 ...  
2 <input type="text" name="punten[]"><br>  
3 <input type="text" name="punten[]"><br>  
4 <input type="text" name="punten[]"><br>  
5 <input type="text" name="punten[]"><br>  
6 ...
```

## PHP

```
1 <?php  
2 $punten = array( 3.5, 6.5,  
3   7.6, 5.4, 9.5, 8.2 );  
4 ?>
```

# Werken met arrays (I)

## Bereken het gemiddelde punt

```
1 <?php
2 $punten = $_POST['punten'];
3
4 $som = $punten[0] + $punten[2] + $punten[3] + $punten[4] + ... ;
5 $gemiddelde = $som / count( $punten );
6
7 echo "Je gemiddelde is $gemiddelde";
8 ?>
```

- ▶ `count( $punten )` telt het aantal punten in het array \$punten.
- ▶ En als het arrays \$punten 50, 500 of nog meer waarden bevat?

# Werken met arrays (I)

## Bereken het gemiddelde punt

```
1 <?php
2   $punten = $_POST['punten'];
3
4   $som = $punten[0] + $punten[2] + $punten[3] + $punten[4] + ... ;
5   $gemiddelde = $som / count( $punten );
6
7   echo "Je gemiddelde is $gemiddelde";
8 ?>
```

- ▶ **count**( \$punten ) telt het aantal punten in het array \$punten.
- ▶ En als het arrays \$punten 50, 500 of nog meer waarden bevat?

# Werken met arrays (II), foreach (I)

## Laat PHP het werk doen

```
1 <?php
   $punten = $_POST['punten'];
3
4   $som = 0;
5   foreach( $punten as $punt ) {
6       $som += $punt
7   };
8   $gemiddelde = $som / count( $punten );
9
10  echo "Je gemiddelde is $gemiddelde";
11 ?>
```

### Voorbeeld

Alle punten worden in een keer ingelezen in de variabele punten

# Werken met arrays (II), foreach (I)

## Laat PHP het werk doen

```
1 <?php
   $punten = $_POST['punten'];
3
4   $som = 0;
5   foreach( $punten as $punt ) {
6       $som += $punt
7   };
8   $gemiddelde = $som / count( $punten );
9
10  echo "Je gemiddelde is $gemiddelde";
11 ?>
```

## Voorbeeld

Alle punten worden in een keer ingelezen in de variabele punten



# Werken met arrays (II), foreach (I)

## Laat PHP het werk doen

```
1 <?php
2 $punten = $_POST['punten'];
3
4 $som = 0;
5 foreach( $punten as $punt ) {
6     $som += $punt
7 };
8 $gemiddelde = $som / count( $punten );
9
10 echo "Je gemiddelde is $gemiddelde";
11 ?>
```

voorbeeld

loop	punt	som
		0

# Werken met arrays (II), foreach (I)

## Laat PHP het werk doen

```
1 <?php
2 $punten = $_POST['punten'];
3
4 $som = 0;
5 foreach( $punten as $punt ) {
6     $som += $punt
7 };
8 $gemiddelde = $som / count( $punten );
9
10 echo "Je gemiddelde is $gemiddelde";
11 ?>
```

### voorbeeld

loop	punt	som
		0

# Werken met arrays (II), foreach (I)

## Laat PHP het werk doen

```
1 <?php
2 $punten = $_POST['punten'];
3
4 $som = 0;
5 foreach( $punten as $punt ) {
6     $som += $punt
7 };
8 $gemiddelde = $som / count( $punten );
9
10 echo "Je gemiddelde is $gemiddelde";
11 ?>
```

voorbeeld

loop	punt	som
		0
1	6.7	

# Werken met arrays (II), foreach (I)

## Laat PHP het werk doen

```
1 <?php
2 $punten = $_POST['punten'];
3
4 $som = 0;
5 foreach( $punten as $punt ) {
6     $som += $punt
7 };
8 $gemiddelde = $som / count( $punten );
9
10 echo "Je gemiddelde is $gemiddelde";
11 ?>
```

### voorbeeld

loop	punt	som
1	6.7	0

# Werken met arrays (II), foreach (I)

## Laat PHP het werk doen

```
1 <?php
2 $punten = $_POST['punten'];
3
4 $som = 0;
5 foreach( $punten as $punt ) {
6     $som += $punt
7 };
8 $gemiddelde = $som / count( $punten );
9
10 echo "Je gemiddelde is $gemiddelde";
11 ?>
```

### voorbeeld

loop	punt	som
		0
1	6.7	6.7

# Werken met arrays (II), foreach (I)

## Laat PHP het werk doen

```
1 <?php
2 $punten = $_POST['punten'];
3
4 $som = 0;
5 foreach( $punten as $punt ) {
6     $som += $punt
7 };
8 $gemiddelde = $som / count( $punten );
9
10 echo "Je gemiddelde is $gemiddelde";
11 ?>
```

### voorbeeld

loop	punt	som
		0
1	6.7	6.7

# Werken met arrays (II), foreach (I)

## Laat PHP het werk doen

```
1 <?php
2 $punten = $_POST['punten'];
3
4 $som = 0;
5 foreach( $punten as $punt ) {
6     $som += $punt
7 };
8 $gemiddelde = $som / count( $punten );
9
10 echo "Je gemiddelde is $gemiddelde";
11 ?>
```

voorbeeld

loop	punt	som
		0
1	6.7	6.7

# Werken met arrays (II), foreach (I)

## Laat PHP het werk doen

```
1 <?php
2 $punten = $_POST['punten'];
3
4 $som = 0;
5 foreach( $punten as $punt ) {
6     $som += $punt
7 };
8 $gemiddelde = $som / count( $punten );
9
10 echo "Je gemiddelde is $gemiddelde";
11 ?>
```

### voorbeeld

loop	punt	som
		0
1	6.7	6.7



# Werken met arrays (II), foreach (I)

## Laat PHP het werk doen

```
1 <?php
2 $punten = $_POST['punten'];
3
4 $som = 0;
5 foreach( $punten as $punt ) {
6     $som += $punt
7 };
8 $gemiddelde = $som / count( $punten );
9
10 echo "Je gemiddelde is $gemiddelde";
11 ?>
```

voorbeeld

loop	punt	som
		0
1	6.7	6.7
2	7.2	

# Werken met arrays (II), foreach (I)

## Laat PHP het werk doen

```
1 <?php
2 $punten = $_POST['punten'];
3
4 $som = 0;
5 foreach( $punten as $punt ) {
6     $som += $punt
7 };
8 $gemiddelde = $som / count( $punten );
9
10 echo "Je gemiddelde is $gemiddelde";
11 ?>
```

### voorbeeld

loop	punt	som
		0
1	6.7	6.7
2	7.2	

# Werken met arrays (II), foreach (I)

## Laat PHP het werk doen

```
1 <?php
2 $punten = $_POST['punten'];
3
4 $som = 0;
5 foreach( $punten as $punt ) {
6     $som += $punt
7 };
8 $gemiddelde = $som / count( $punten );
9
10 echo "Je gemiddelde is $gemiddelde";
11 ?>
```

voorbeeld

loop	punt	som
		0
1	6.7	6.7
2	7.2	13.9

# Werken met arrays (II), foreach (I)

## Laat PHP het werk doen

```
1 <?php
2 $punten = $_POST['punten'];
3
4 $som = 0;
5 foreach( $punten as $punt ) {
6     $som += $punt
7 };
8 $gemiddelde = $som / count( $punten );
9
10 echo "Je gemiddelde is $gemiddelde";
11 ?>
```

### voorbeeld

loop	punt	som
		0
1	6.7	6.7
2	7.2	13.9

# Werken met arrays (II), foreach (I)

## Laat PHP het werk doen

```
1 <?php
2 $punten = $_POST['punten'];
3
4 $som = 0;
5 foreach( $punten as $punt ) {
6     $som += $punt
7 };
8 $gemiddelde = $som / count( $punten );
9
10 echo "Je gemiddelde is $gemiddelde";
11 ?>
```

voorbeeld

loop	punt	som
		0
1	6.7	6.7
2	7.2	13.9

# Werken met arrays (II), foreach (I)

## Laat PHP het werk doen

```
1 <?php
2 $punten = $_POST['punten'];
3
4 $som = 0;
5 foreach( $punten as $punt ) {
6     $som += $punt
7 };
8 $gemiddelde = $som / count( $punten );
9
10 echo "Je gemiddelde is $gemiddelde";
11 ?>
```

### voorbeeld

loop	punt	som
		0
1	6.7	6.7
2	7.2	13.9

# Werken met arrays (II), foreach (I)

## Laat PHP het werk doen

```
1 <?php
2 $punten = $_POST['punten'];
3
4 $som = 0;
5 foreach( $punten as $punt ) {
6     $som += $punt
7 };
8 $gemiddelde = $som / count( $punten );
9
10 echo "Je gemiddelde is $gemiddelde";
11 ?>
```

## voorbeeld

loop	punt	som
		0
1	6.7	6.7
2	7.2	13.9
3	5.4	19.3
4	8.3	27.6
5	5.6	33.2
6	6.3	39.5
7	7.2	46.7

Het foreach-statement wordt uitgevoerd voor elk item in het array

# Werken met arrays (II), foreach (I)

## Laat PHP het werk doen

```
1 <?php
2 $punten = $_POST['punten'];
3
4 $som = 0;
5 foreach( $punten as $punt ) {
6     $som += $punt
7 };
8 $gemiddelde = $som / count( $punten );
9
10 echo "Je gemiddelde is $gemiddelde";
11 ?>
```

## voorbeeld

loop	punt	som
		0
1	6.7	6.7
2	7.2	13.9
3	5.4	19.3
4	8.3	27.6
5	5.6	33.2
6	6.3	39.5
7	7.2	46.7

Het foreach-statement wordt uitgevoerd voor elk item in het array



# Werken met arrays (II), foreach (I)

## Laat PHP het werk doen

```
1 <?php
2   $punten = $_POST['punten'];
3
4   $som = 0;
5   foreach( $punten as $punt ) {
6       $som += $punt
7   };
8
9   $gemiddelde = $som / count( $punten );
10  echo "Je gemiddelde is $gemiddelde";
11  ?>
```

## voorbeeld

loop	punt	som
		0
1	6.7	6.7
2	7.2	13.9
3	5.4	19.3
4	8.3	27.6
5	5.6	33.2
6	6.3	39.5
7	7.2	46.7

Het gemiddelde is 46.7 gedeeld door het aantal elementen in het array punten (= 6.7)

# Werken met arrays (II), foreach (I)

## Laat PHP het werk doen

```
1 <?php
2 $punten = $_POST['punten'];
3
4 $som = 0;
5 foreach( $punten as $punt ) {
6     $som += $punt
7 };
8
9 $gemiddelde = $som / count( $punten );
10
11 echo "Je gemiddelde is $gemiddelde";
12 ?>
```

## voorbeeld

loop	punt	som
		0
1	6.7	6.7
2	7.2	13.9
3	5.4	19.3
4	8.3	27.6
5	5.6	33.2
6	6.3	39.5
7	7.2	46.7

Het gemiddelde is 46.7 gedeeld door het aantal elementen in het array punten (= 6.7)

# Werken met arrays (II), foreach (I)

## Laat PHP het werk doen

```
1 <?php
2 $punten = $_POST['punten'];
3
4 $som = 0;
5 foreach( $punten as $punt ) {
6     $som += $punt
7 };
8 $gemiddelde = $som / count( $punten );
9
10 echo "Je gemiddelde is $gemiddelde";
11 ?>
```

### voorbeeld

En het gemiddelde wordt geëchoot.

# Werken met arrays (II), foreach (I)

## Laat PHP het werk doen

```
1 <?php
2 $punten = $_POST['punten'];
3
4 $som = 0;
5 foreach( $punten as $punt ) {
6     $som += $punt
7 };
8 $gemiddelde = $som / count( $punten );
9
10 echo "Je gemiddelde is $gemiddelde";
11 ?>
```

## voorbeeld

En het gemiddelde wordt geëchoot.

# foreach (II)

## Het foreach-statement

```
1 foreach ( $array as $item ) {  
2   // Doe iets met $item, voor alle items in $array  
3   echo "$item <br>";  
4 };
```

- ▶ Het foreach-statement voert de code tussen { en } uit voor alle items in het array.
- ▶ Bij begin van elke "loop" krijgt \$item het volgende element uit het array als waarde toegekend. \$item noemen we de *loopvariabele*.
- ▶ De code uit het blok wordt uitgevoerd, \$item is daar bekend en heeft een waarde. Het werkt in het blok als elke andere variabele
- ▶ Als alle items langs zijn gekomen, is het foreach-statement afgelopen. \$item bestaat niet meer.

# foreach (II)

## Het foreach-statement

```
1 foreach ( $array as $item ) {  
2     // Doe iets met $item, voor alle items in $array  
3     echo "$item <br>";  
4 };
```

- ▶ Het foreach-statement voert de code tussen { en } uit voor alle items in het array.
- ▶ Bij begin van elke “loop” krijgt \$item het volgende element uit het array als waarde toegekend. \$item noemen we de *loopvariabele*.
- ▶ De code uit het blok wordt uitgevoerd, \$item is daar bekend en heeft een waarde. Het werkt in het blok als elke andere variabele
- ▶ Als alle items langs zijn gekomen, is het foreach-statement afgelopen. \$item bestaat niet meer.

# foreach (III): associatieve arrays

## Het foreach-statement

```
1 foreach( $assoc_array as $sleutel => $waarde ) {  
2     // Doe iets met $sleutel en $waarde voor elk  
3     // sleutel-waarde paartje in $assoc_array  
4     echo "$sleutel &rarr; $waarde <br>";  
5 };
```

- ▶ Zowel sleutel als waarde in het foreach
- ▶ Werkt verder precies hetzelfde als bij gewone arrays

# foreach (III): associatieve arrays

## Het foreach-statement

```
1 foreach( $assoc_array as $sleutel => $waarde ) {  
2     // Doe iets met $sleutel en $waarde voor elk  
3     // sleutel-waarde paartje in $assoc_array  
4     echo "$sleutel &rarr; $waarde <br>";  
5 };
```

- ▶ Zowel sleutel als waarde in het foreach
- ▶ Werkt verder precies hetzelfde als bij gewone arrays