

PHP herhaalt: for en while

Huub de Beer

Eindhoven, 4 juni 2011

Iteratie, repetitie en loops

PHP herhaalt

- ▶ Een voor een de elementen van een array doorlopen? Gebruik **foreach-statement**
- ▶ Tellen van een beginwaarde tot een eindwaarde en bij elke tel een stuk code uitvoeren? **for-statement**
- ▶ Een stuk code herhalen zolang een conditie geldt? **while-statement**

machtsverheffen

- ▶ Gebruik de PHP functie `pow(grondtal, exponent)`
- ▶ Maar je kunt ook zelf een machtfunctie definiëren met behulp van het `for`-statement.

machtsverheffen: g^e

- ▶ het grondtal g tot de macht (exponent) e
- ▶ in dit voorbeeld alleen voor $e \geq 0$
- ▶ g^e is:

$$\underbrace{g \times g \times g \times \cdots \times g}_{e \text{ keer}}$$

- ▶ $g^0 = 1$, voor alle g .

machtsverheffen

- ▶ Gebruik de PHP functie `pow(grondtal, exponent)`
- ▶ Maar je kunt ook zelf een machtfunctie definiëren met behulp van het `for`-statement.

machtsverheffen: g^e

- ▶ het grondtal g tot de macht (exponent) e
- ▶ in dit voorbeeld alleen voor $e \geq 0$
- ▶ g^e is:

$$\underbrace{g \times g \times g \times \cdots \times g}_{e \text{ keer}}$$

- ▶ $g^0 = 1$, voor alle g .

De machtsfunctie in PHP

```
1 function macht( $g, $e ) {  
2     // Bereken $g tot de macht $e en retourneer die waarde  
3  
4     $macht = 1; //  $g^0 = 1$  voor alle  $g$ , dus goede beginwaarde  
5  
6     for( $i = 1; $i <= $e; $i++ ) {  
7         // tellertje  $i$   
8         // van 1 tot en met  $e$ , elke tel bijgehouden in  $i$ , doe het volgende:  
9         // vermenigvuldig de macht tot nu toe met het grondtal  
10        $macht = $macht * $g;  
11    }  
12  
13    //  $g$  is  $e$  keer met zichzelf vermenigvuldigd  
14    return $macht;  
15 };
```

simulatie (I): 2^3

```
function macht( $g, $e ) {  
2   $macht = 1;  
3  
4   for( $i = 1; $i <= $e; $i++ ) {  
5       $macht = $macht * $g;  
6   }  
7  
8   return $macht;  
9 };
```

waarden van de variabelen:

\$i	\$macht	\$e	\$g
-	-	3	2

simulatie (II): 2^3

```
1 function macht( $g, $e ) {  
    $macht = 1;  
3  
4     for( $i = 1; $i <= $e; $i++ ) {  
5         $macht = $macht * $g;  
6     }  
7  
8     return $macht;  
9 };
```

waarden van de variabelen:

\$i	\$macht	\$e	\$g
-	-	3	2
-	1	3	2

simulatie (III): 2^3

```
1 function macht( $g, $e ) {  
2   $macht = 1;  
3  
4   for( $i = 1; $i <= $e; $i++ ) {  
5     $macht = $macht * $g;  
6   }  
7  
8   return $macht;  
9 };
```

waarden van de variabelen:

i	$macht$	e	g
-	-	3	2
-	1	3	2
1	1	3	2

is $i \leq e$? Ja, $1 \leq 3$

simulatie (IV): 2^3

```
1 function macht( $g, $e ) {  
2   $macht = 1;  
3  
4   for( $i = 1; $i <= $e; $i++ ) {  
5     $macht = $macht * $g;  
6   }  
7  
8   return $macht;  
9 };
```

waarden van de variabelen:

$\$i$	$\$macht$	$\$e$	$\$g$
-	-	3	2
-	1	3	2
1	1	3	2
1	1×2	3	2

simulatie (V): 2^3

```
1 function macht( $g, $e ) {  
2   $macht = 1;  
3  
4   for( $i = 1; $i <= $e; $i++ ) {  
5     $macht = $macht * $g;  
6   }  
7  
8   return $macht;  
9 };
```

waarden van de variabelen:

\$i	\$macht	\$e	\$g
-	-	3	2
-	1	3	2
1	1	3	2
1	1×2	3	2
1	1×2	3	2

simulatie (VI): 2^3

```
1 function macht( $g, $e ) {  
2   $macht = 1;  
3  
4   for( $i = 1; $i <= $e; $i++ ) {  
5     $macht = $macht * $g;  
6   }  
7  
8   return $macht;  
9 };
```

waarden van de variabelen:

\$i	\$macht	\$e	\$g
-	-	3	2
-	1	3	2
1	1	3	2
1	1×2	3	2
1	1×2	3	2
2	1×2	3	2

is $i \leq e$? Ja, $2 \leq 3$

simulatie (VII): 2^3

```
1 function macht( $g, $e ) {  
2   $macht = 1;  
3  
4   for( $i = 1; $i <= $e; $i++ ) {  
5     $macht = $macht * $g;  
6   }  
7  
8   return $macht;  
9 };
```

waarden van de variabelen:

\$i	\$macht	\$e	\$g
-	-	3	2
-	1	3	2
1	1	3	2
1	1×2	3	2
1	1×2	3	2
2	1×2	3	2
2	$1 \times 2 \times 2$	3	2

simulatie (VIII): 2^3

```
1 function macht( $g, $e ) {  
2   $macht = 1;  
3  
4   for( $i = 1; $i <= $e; $i++ ) {  
5     $macht = $macht * $g;  
6   }  
7  
8   return $macht;  
9 };
```

waarden van de variabelen:

\$i	\$macht	\$e	\$g
-	-	3	2
-	1	3	2
1	1	3	2
1	1×2	3	2
1	1×2	3	2
2	1×2	3	2
2	$1 \times 2 \times 2$	3	2
2	$1 \times 2 \times 2$	3	2

simulatie (IX): 2^3

```
1 function macht( $g, $e ) {  
2   $macht = 1;  
3  
4   for( $i = 1; $i <= $e; $i++ ) {  
5     $macht = $macht * $g;  
6   }  
7  
8   return $macht;  
9 };
```

waarden van de variabelen:

$\$i$	$\$macht$	$\$e$	$\$g$
-	-	3	2
-	1	3	2
1	1	3	2
1	1×2	3	2
1	1×2	3	2
2	1×2	3	2
2	$1 \times 2 \times 2$	3	2
2	$1 \times 2 \times 2$	3	2
3	$1 \times 2 \times 2$	3	2

is $\$i \leq \e ? Ja, $3 \leq 3$

simulatie (X): 2^3

```
1 function macht( $g, $e ) {  
2   $macht = 1;  
3  
4   for( $i = 1; $i <= $e; $i++ ) {  
5     $macht = $macht * $g;  
6   }  
7  
8   return $macht;  
9 };
```

waarden van de variabelen:

\$i	\$macht	\$e	\$g
-	-	3	2
-	1	3	2
1	1	3	2
1	1×2	3	2
1	1×2	3	2
2	1×2	3	2
2	$1 \times 2 \times 2$	3	2
2	$1 \times 2 \times 2$	3	2
3	$1 \times 2 \times 2$	3	2
3	$1 \times 2 \times 2 \times 2$	3	2

simulatie (XI): 2^3

```
1 function macht( $g, $e ) {  
2   $macht = 1;  
3  
4   for( $i = 1; $i <= $e; $i++ ) {  
5     $macht = $macht * $g;  
6   }  
7  
8   return $macht;  
9 };
```

waarden van de variabelen:

\$i	\$macht	\$e	\$g
-	-	3	2
-	1	3	2
1	1	3	2
1	1×2	3	2
1	1×2	3	2
2	1×2	3	2
2	$1 \times 2 \times 2$	3	2
2	$1 \times 2 \times 2$	3	2
3	$1 \times 2 \times 2$	3	2
3	$1 \times 2 \times 2 \times 2$	3	2
3	$1 \times 2 \times 2 \times 2$	3	2

simulatie (XII): 2^3

```
1 function macht( $g, $e ) {  
2   $macht = 1;  
3  
4   for( $i = 1; $i <= $e; $i++ ) {  
5     $macht = $macht * $g;  
6   }  
7  
8   return $macht;  
9 };
```

waarden van de variabelen:

\$i	\$macht	\$e	\$g
-	-	3	2
-	1	3	2
1	1	3	2
1	1×2	3	2
1	1×2	3	2
2	1×2	3	2
2	$1 \times 2 \times 2$	3	2
2	$1 \times 2 \times 2$	3	2
3	$1 \times 2 \times 2$	3	2
3	$1 \times 2 \times 2 \times 2$	3	2
3	$1 \times 2 \times 2 \times 2$	3	2
4	$1 \times 2 \times 2 \times 2$	3	2

is $i \leq e$? **Neen**, $4 > 3$

simulatie (XIII): 2^3

```
1 function macht( $g, $e ) {  
2   $macht = 1;  
3  
4   for( $i = 1; $i <= $e; $i++ ) {  
5     $macht = $macht * $g;  
6   }  
7  
8   return $macht;  
9 };
```

waarden van de variabelen:

\$i	\$macht	\$e	\$g
-	-	3	2
-	1	3	2
1	1	3	2
1	1×2	3	2
1	1×2	3	2
2	1×2	3	2
2	$1 \times 2 \times 2$	3	2
2	$1 \times 2 \times 2$	3	2
3	$1 \times 2 \times 2$	3	2
3	$1 \times 2 \times 2 \times 2$	3	2
3	$1 \times 2 \times 2 \times 2$	3	2
4	$1 \times 2 \times 2 \times 2$	3	2

**Einde functie: retourneer de
berekende macht**

Het for-statement

```
1 for ($i = beginwaarde; $i <= eindwaarde; $i++) {  
2   // Code die uitgevoerd wordt voor elke tel van  
3   // beginwaarde tot eindwaarde  
4  
5 };
```

- ▶ meestal ziet for-statement er zo uit
- ▶ een tellertje met de naam *i* of *j*
- ▶ tel van een beginwaarde tot een eindwaarde met stapjes van 1
- ▶ kan ook andersom: van eindwaarde tot beginwaarde terugtellen
- ▶ of grotere stapjes nemen

Machtsverheffen met het while-statement

```
1 function macht( $g, $e ) {  
2     $macht = 1;  
3     $i = 1;  
4  
5     while ( $i <= $e ) {  
6         $macht = $macht * $g;  
7         $i++;  
8     };  
9  
10    return $macht;  
11 };
```

- ▶ Een conditie zoals in if-statement: zolang de conditie geldt, voert PHP de code in het while-statement uit.
- ▶ Let op: zorg ervoor dat de code in het while-statement de code ooit onwaar maakt, anders geraakt PHP in een *oneindige lus*.
- ▶ In dit geval tel je zelf: het for-statement is hier geschikter

Welk iteratiestatement?

- ▶ foreach: een voor een alle elementen in een array inspecteren
- ▶ for: een stuk code een vast aantal stappen herhalen
- ▶ while: andere gevallen:
 - ▶ enkel een aantal elementen in een array inspecteren
 - ▶ door een array heen springen
 - ▶ berekenen van een wortel (benaderingsalgoritme)
 - ▶
- ▶ veel problemen kunnen met alle drie de iteratiestatements goed opgelost worden.