

2 Interactieve websites met PHP en MySQL

1 INLEIDING

1.1 WAT IS PHP?

PHP is een server-side scripting taal.

PHP stond oorspronkelijk voor "**P**ersonal **H**ome **P**age" tools: ontworpen door Rasmus Lerdorf in 1994 en vrij ter beschikking gesteld op Internet.

Sindsdien is het door vele programmeurs omarmd en ontwikkeld waardoor het een krachtige en betrouwbare ontwikkelomgeving is geworden die nog steeds vrij te gebruiken is.

In PHP zijn veel elementen van andere programmeertalen geleend, zoals **C** en **Perl**. Maar het is ook zo ontworpen dat het eenvoudig in het gebruik is.

De belangrijkste informatiebron voor PHP vind je op Internet op het volgende adres:

www.php.net

Websites die gebruik maken van PHP hebben vaak dit logo: 

1.2 PHP EN ZIJN CONCURRENTEN

PHP is een van de "grote drie" server-side scripting talen. De andere zijn:

ASP: Active Server Pages, ontwikkeld door Microsoft en alleen maar beschikbaar op het Windows platform.

JSP: Java Server Pages, ontwikkeld door Sun en alleen maar te gebruiken in combinatie met een Java Web Server.

De belangrijkste voordelen van PHP boven de andere twee zijn:

- Vrij: PHP is gratis te gebruiken
- Platformonafhankelijk: is niet beperkt tot een enkel besturingssysteem of webserver
- Krachtig: PHP heeft uitstekende faciliteiten om te kunnen samenwerken met databases, grafische bestanden, XML en PDF.

1.3 PHP VERSUS JAVASCRIPT

De fundamentele verschillen tussen PHP en JavaScript zijn:

JavaScript

- Is een cliënt-side scripting taal; d.w.z. dat het script in de browser van je eigen computer wordt uitgevoerd
- Statements worden door de webbrowser gerund
- Scripts worden normaliter gerund na een event
- De pagina kan gewijzigd worden na het tonen van de pagina.



PHP

- Is een server-side scripting taal; d.w.z. dat het script op een andere computer (server) wordt uitgevoerd
- Statements worden gerund door de webserver
- De scripts worden uitgevoerd als de server de pagina laadt
- De pagina kan niet worden gewijzigd nadat deze naar de browser is verstuurd.

Daarnaast is PHP ook in combinatie met JavaScript te gebruiken.

1.4 WAARVOOR WORDT PHP GEBRUIKT?**Benaderen van databases**

- Een PHP script kan overweg met een groot aantal relationele databases door gebruik te maken van SQL. PHP kan de resultaten van SQL-queries in HTML-formaat weergeven.
- PHP bezit een aantal functies waarmee je standaard SQL bewerkingen kunt uitvoeren op verschillende database management systemen (DBMS).
- Verbinding met de DBMS server m.b.v. gebruikersnaam en wachtwoord.
- Gebruik van o.a. `CREATE`, `SELECT`, `INSERT` en `UPDATE` SQL statements.
- Het ontvangen van resultaten en het kunnen benaderen van de waardes van rijen en kolommen.
- Deze bewerkingen worden ondersteund door DBMS pakketten waaronder dBase, Informix, MySQL, mSQL, Oracle en Sybase.

Bewerken van webformulier gegevens

- Een PHP script kan gegevens ontvangen uit een webformulier; de gegevens uit dat formulier kunnen worden bewerkt, opgeslagen in een bestand of een database of verzonden via e-mail naar iemand.
- De gegevens uit een webformulier kunnen ook worden gevalideerd aan server kant.

Bestandsoperaties

- Een PHP script kan bestanden op de server benaderen (ongeacht besturingssysteem).
- Een PHP script kan ook de inhoud van mappen en bestanden lezen en schrijven.

Netwerkoperaties

- PHP scripts kunnen netwerk operaties uitvoeren zoals het versturen van e-mails.
- PHP scripts kunnen contact maken met andere netwerk servers.

Het draaien van programma's

- Een PHP script kan programma's op de computer laten draaien.
- De instelling hiervoor is standaard maar om veiligheidsredenen kun je dit eventueel ook uitschakelen.

Belangrijk punt

- PHP code wordt nooit naar de browser verstuurd. Je kunt daarom nooit de code in beeld krijgen.
- Als een webserver een PHP script laadt dan wordt door de server:
 - de ingebouwde PHP code uitgevoerd
 - tekst door het PHP script gegenereerd, ingebed in HTML
 - de HTML pagina naar de browser verstuurd.
- De browser ziet dus alleen maar de uiteindelijke HTML pagina.

1.5 OVER DEZE MODULE

Deze module is een introductie op PHP en MySQL. Het is de bedoeling dat je tijdens de (praktijk)lessen de voorbeelden uitprobeert en de opdrachten maakt om zodoende een goede basis te krijgen in het werken met PHP en MySQL. Er wordt van uitgegaan dat je over enige kennis van programmeren in Java of een andere hogere programmeertaal beschikt en databases hebt leren bevragen met SQL. Verder is kennis van HTML ook onontbeerlijk. In het verwerkingsboek is in bijlage 2 van het overeenkomstige hoofdstuk een overzicht te vinden van de gangbare HTML-tags die je kunt gebruiken in je code. Maak je geen zorgen over het feit dat in het begin sommige ideeën erg ondoorzichtig lijken, probeer gewoon te begrijpen wat de voorbeelden feitelijk doen zonder je al te zeer druk te maken over de details in de scriptcode. PHP zal je op den duur enkele krachtige instrumenten in handen geven waarmee je interactieve websites kunt maken. Je treft in dit boek een aantal voorbeelden aan. Het beste is om deze over te typen. Daardoor kun je meer praktische vaardigheid ontwikkelen. Je zult waarschijnlijk denken dat het eenvoudiger is om die voorbeelden te downloaden en in de editor te laden. Dat is natuurlijk wel zo, maar daardoor wordt het moeilijker de essenties van PHP te doorgronden en te waarderen. Het is net als alle andere talen: veel doen, door herhaling leer je meer. Het is beter dan alleen maar te luisteren naar een leraar of medeleerling die zegt hoe het moet.

2 PHP DE BASIS

2.1 EEN EENVOUDIG VOORBEELD

Een eenvoudig voorbeeld van een PHP script:

```
<html>
<head>
<title>Een eenvoudig voorbeeld van een PHP script</title>
</head>
<body>
<p>
    <?php
        print("Deze pagina is gegenereerd door PHP!" );
    ?>
</p>
</body>
</html>
```

De PHP code is verwerkt tussen de standaard HTML code. In dit voorbeeld is de PHP code geplaatst tussen de `<?php` en `?>` tags. Al het andere is normaal HTML. PHP scripts dien je op te slaan met de extensie: `.php`

Het is erg belangrijk dat je het bestand opslaat met de `.php` extensie. Als dat namelijk mislukt kan de server de PHP-code in het document niet parsen.

Als kladblok automatisch de extensie `.txt` er achter plaatst, kun je het beste het bestand eerst als `.txt` opslaan en vervolgens in de verkenner het bestand een nieuwe naam geven en de `.txt` vervangen door `.php`.

Er is een aantal dingen dat je beslist moet doen alvorens je de PHP bestanden succesvol kunt zien. Volg deze regels zorgvuldig op anders zal het niet lukken.



Je moet je pagina's online bekijken of offline als je een webserver op je machine hebt staan met ondersteuning van PHP. Alleen dan kan de code geparsed worden. Onder **parsen** van PHP code verstaan we het omzetten van de code naar HTML.

Start de browser. Typ op de adresbalk de juiste URL. Druk op Enter. Je krijgt dan een pagina te zien die er ongeveer zo uitziet:



Als je kijkt naar de broncode van deze getoonde webpagina dan ziet die er als volgt uit:



De PHP code is verdwenen en vervangen door tekst die door het script is gegenereerd. Dat is gebeurd tijdens het runnen van het script op de webserver. Onthoud goed: je mag nooit PHP code in de broncode van een webpagina aantreffen. Als dat wel het geval is heb je ergens een fout gemaakt in het script of is het bestand als `.txt` opgeslagen.

Om het bekijken van bestanden gemakkelijker te maken kun je in plaats van iedere keer de volledige URL in te typen ook een map in de `htdocs` map van je XAMPP aanmaken waarin je je scripts plaatst die je wilt testen. Voor deze module is op de lokale server een map `PHP-Enigma` aangemaakt. Let op! Je moet er wel op letten dat de meeste internetserveren hoofdlettergevoelig zijn. Als je nu in de browser `localhost/PHP-Enigma` intypt, krijg je iets te zien dat er ongeveer zo uitziet (zorg dat er geen `index.html` of `index.php` in de map `PHP-Enigma` staat):



Index of /PHP-Enigma

Name	Last modified	Size	Description
Parent Directory		-	
hallo.php	26-Nov-2007 12:16	198	

Klik je vervolgens op de link Parent Directory dan kom je terecht in de DocumentRoot van de webserver, in dit geval de map `htdocs`. Die map kun je benaderen door in de browser op de adresbalk `localhost` of `127.0.0.1` in te typen en te activeren.

opdracht 1

2.2 HET PRINT OF ECHO STATEMENT

Een PHP script wordt opgebouwd uit een aantal opeenvolgende statements die worden gescheiden door puntkomma's.

Het belangrijkste statement is `print`. Dit statement genereert tekst dat aan de webpagina wordt toegevoegd.

Een `print` statement heeft de volgende basisvorm:

```
print("dit is een stuk tekst");
```

Dit voorbeeld voegt de tekst `dit is een stuk tekst` toe aan de webpagina.

In opdracht 2 zagen we ook al een voorbeeld van het gebruik van het `print` statement.

OPMERKING: in plaats van `print` kan ook `echo` als statement gebruikt worden. Maar in deze module zullen wij steeds het `print` statement gebruiken.

opdrachten 2 - 4

2.3 PHP CODE FORMAT

Er bestaan 3 manieren om PHP in HTML code in te bedden:

1. De eenvoudigste manier is die om de code tussen de volgende tags te plaatsen:

```
<? code ?> →  
  
<? print("Dit is PHP"); ?>
```

Deze methode kan problemen opleveren als PHP gebruikt wordt in combinatie met XML omdat XML ook gebruik maakt van deze tags. Daarom zullen we deze manier in dit werkboek niet gebruiken.

2. Als PHP samen met XML gebruikt wordt is het raadzaam dat je deze stijl gebruikt. Dit is de meest gebruikte methode. We zullen die verder in dit werkboek blijven gebruiken. Het is absoluut aan te raden dit format voor je scripts te gebruiken.

```
<?php  
    print("Dit is PHP");  
?>
```

3. Als alternatief, maar wel een beetje uit de tijd, is de manier om de HTML script tags te gebruiken en daarom beter ook maar niet te gebruiken:

```
<script language = "php">  
    print("Dit is PHP");  
</script>
```



2.4 COMMENTAAR PLAATSEN IN PHP-SCRIPTS

In PHP code kun je op meerdere manieren commentaar verwerken:

1. `//` een regel commentaar voorafgegaan door een dubbele slash
2. `#` een regel commentaar voorafgegaan door een hekje
3. `/*` meerregelig commentaar wordt omsloten door slash-asterisk
..... asterisk-slash
`*/`

2.5 HET GEBRUIK VAN FORMULIEREN

De HTML code die door PHP wordt gegenereerd kan invoergegevens gebruiken om dynamische webpagina's te maken, gebaseerd op de waarden van de invoergegevens. Die gegevens kunnen komen uit een webformulier of uit andere bronnen.

Om de gebruiker informatie te laten invoeren en te verwerken moet je in php/html 2 stappen uitvoeren:

1. Een html pagina aanbieden waarop een form staat. In dit form kan de gebruiker dingen invullen en aanklikken. Componenten die in een form kunnen voorkomen zijn onder andere:
 - Tekstvakken
 - Knoppen
 - Radiobuttons
 - Comboboxen.
2. Een php pagina maken waar de verwerking van de gegevens plaatsvindt.

Laten we eens een eenvoudig voorbeeld bekijken. We halen een naam uit een webformulier op en laten die zien als onderdeel van een welkomstbericht.

Hier is de HTML code die je nodig hebt voor dat formulier.

```
<html>
<head>
  <title>Eenvoudig PHP voorbeeld 3</title>
</head>

<body>
  <form method = "POST" action = "hallojij.php">
    <p>Typ hieronder je naam in.</p>
    <p> Klik op de knop: <strong>Verstuur</strong></p>
    <p><input type = "text" size="25" name = "naam"></p>
    <p><input type = "submit" value = "Verstuur"></p>
  </form>
</body>
</html>
```

Let erop dat het formulier als een `.html` bestand wordt opgeslagen. Je kunt het echter ook als een `.php` bestand opslaan ondanks dat er geen enkele PHP code instaat. De reden daarvoor zou kunnen zijn dat je voor consistentie wilt kiezen: alle bestanden dezelfde extensie.

Het formulier bevat een variabele name. Deze variabele bevat straks de tekst die de gebruiker in het tekstvak intypt.

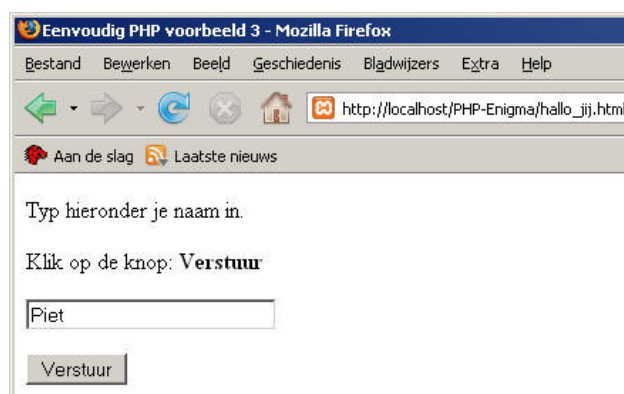
De action in het formulier bepaalt dat de server het bestand `hallojij.php` moet uitvoeren als er op de knop wordt geklikt. De code voor dat bestand ziet er als volgt uit:

```
<html>
<head>
    <title>
        Hallo <?php $naam = $_POST["naam"]; print( $naam ); ?>
    </title>
</head>

<body>
    <h1>Hallo <?php print( $naam ); ?> </h1>
</body>
</html>
```

In het script verwijst de variabele `$naam` naar de variabele in het invoerveld van het invulformulier dat door PHP door middel van de HTTP POST – methode wordt overgedragen.

Als je het formulier in je browser laadt, kan het er zo uitzien:



Als je bijvoorbeeld de naam Piet hebt ingetypt en vervolgens op de knop drukt krijg je het volgende in de browser te zien:



De bijbehorende HTML-broncode van deze webpagina ziet er dan als volgt uit:

```
Bron van: http://localhost/PHP-Enigma/hallojij.php - Mozilla Firefox
Bestand  Bewerken  Beeld  Help

<html>
<head>
    <title>
        Hallo Piet
    </title>
</head>

<body>
    <h1>Hallo Piet </h1>
</body>
</html>
```

Een iets ingewikkelder formulier kan er zo uitzien:

De bijbehorende code ziet er als volgt uit:

```
<html>
<head>
  <title>Een ingewikkelder formulier</title>
</head>
<body>
  <form method="get" action="form2afhandeling.php">

    <p> Hoe heet je?
      <input type="text" size="20" name="voornaam"></p>
    <p> Informatica is leuk:
    <input type="radio" name="stelling" value="eens" checked>Eens
    <input type="radio" name="stelling" value="oneens">Oneens
    </p>
    <p> Welk profiel heb je?
    <select name="profiel">
      <option>NT</option>
      <option>NG</option>
      <option>EM</option>
      <option>CM</option>
    </select>
    </p>
    <p> <input type="submit" value="Verstuur"></p>
  </form>
</body>
</html>
```

Zoals je ziet staat het formulier weer tussen de `<form>` en `</form>` tags. De pagina waar we naartoe gaan is `form2afhandeling.php`.

Ons formulier heeft 4 form onderdelen:

1. Een tekstveld

```
<input type="text" size="20" name="voornaam">
```

Deze kennen we al uit het vorige voorbeeld.

2. Radiobuttons

```
<input type="radio" name="stelling" value="eens" checked>Eens
```

```
<input type="radio" name="stelling" value="oneens">Oneens
```

Je ziet dat je voor elke radiobutton een aparte tag moet maken.

`type="radio"` geeft aan dat het een radiobutton is.

De `name="stelling"` is weer de naam waarmee de waarde opgehaald kan worden bij de afhandeling.

LET OP: radiobuttons die bij elkaar horen moeten dezelfde naam hebben!

`value="eens"` geeft de waarde aan die doorgestuurd wordt als deze radiobutton aan staat.

Bij de eerste optie staat er nog het woordje `checked` achter. Dat houdt in dat deze standaard geselecteerd is.

3. Een combobox

```
<select name="profiel">
    <option>NT</option>
    <option>NG</option>
    <option>EM</option>
    <option>CM</option>
</select>
```

De combobox wordt gegeven door een `<select>` blok.

De `name="profiel"` wordt gebruikt om straks de waarde op te halen.

Alle opties van de combobox worden gemaakt door aparte `<option>` blokken binnen het `<select>` blok.

4. Een knop

```
<input type="submit" value="Verstuur">
```

Deze kennen we al uit het vorige voorbeeld.

2.6 WERKEN MET VARIABELEN IN PHP

Variabelen beginnen in PHP altijd met het dollarteken (`$`), zoals we al hebben gezien in het script op bladzijde 10 waar we de variabele `$naam` hebben gebruikt. Het dollarteken moet worden gevolgd door een letter of een underscore. De rest van de variabelennaam kan bestaan uit letters, cijfers en underscores.

Voorbeelden van PHP variabelen kunnen zijn:

```
$i
$auteur
$query2
$student_ID
```

PHP variabelen zijn hoofdlettergevoelig. `$student_ID` is een andere variabele als `$student_id`.

Als variabelen uit een HTML formulier in een PHP script worden overgebracht wordt door PHP automatisch voor elke formuliervariabele een PHP variabele (Register_global) gemaakt:

- met dezelfde naam als de formuliervariabele
- en met een dollarteken ervoor.

Bijvoorbeeld: als er een formuliervariabele bestaat met de naam `wachtwoord` dan wordt de PHP variabele `$wachtwoord`. Dat kon je ook al zien in het script uit paragraaf 2.5: formuliervariabele naam wordt PHP variabele `$naam`.

In PHP worden aan variabelen waarden toegekend. Daarvoor wordt het is-teken (`=`) gebruikt en de toekenning wordt afgesloten, net zoals alle statements, met een puntkomma (`;`).



```
$i = 0;
$bewerker = "Jan Klaassen";
$query2 = "SELECT * FROM landen";
$student_ID = 720123;
```

Je kunt de waarden van PHP variabelen aan de tekst van een webpagina toevoegen door op verschillende manieren gebruik te maken van het `print` (of `echo`) statement:

```
print("De bewerker van dit werkboek is"); print($bewerker);
print("De bewerker van dit werkboek is $bewerker");
print("De bewerker van dit werkboek is " . $bewerker);
```

Al deze vormen hebben als resultaat dat op de webpagina de volgende tekst te zien is:

De bewerker van dit werkboek is Jan Klaassen

In het derde geval wordt gebruik gemaakt van de zogenaamde punt-operator. Hierover volgt later meer.

We hebben gezien dat je met het `print` statement tekst aan een webpagina kunt toevoegen. Als tekst wordt omgeven door dubbele aanhalingstekens

worden variabelennamen vervangen door hun waarden.

Letters voorafgegaan door een backslash (\) worden beschouwd als bijzondere tekens.

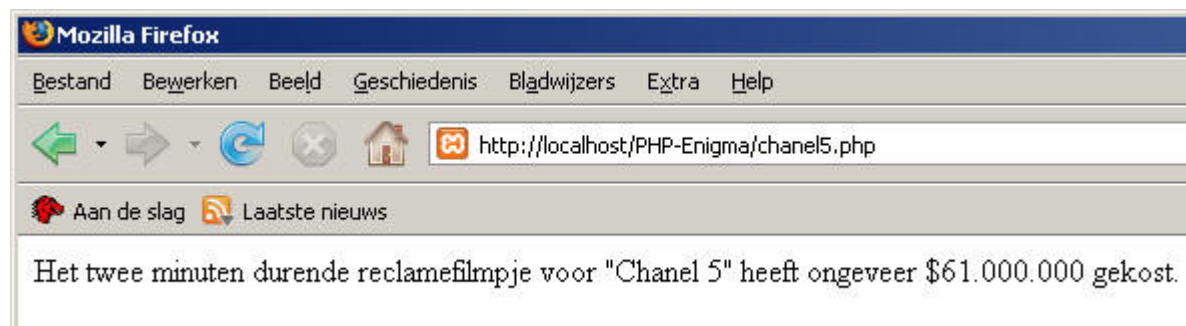
We noemen dit **escapen** van tekens.

```
\n - nieuwe regel
\$ - het dollarteken
\" - het dubbel aanhalingsteken
```

Een voorbeeld waarbij met bijzondere tekens wordt gewerkt:

```
print("Het twee minuten durende \nreclamefilmpje voor \"Chanel 5\"
\nheeft ongeveer \$61.000.000 gekost.");
```

Het resultaat in de browser ziet er ongeveer als volgt uit:



Als je naar de broncode van de HTML pagina gaat kijken ziet die er als volgt uit:



Je vraagt je misschien af wat het nut is om af en toe in de PHP code gebruik te maken van de code `\n` (nieuwe regel). Dat is vooral van belang bij complexe scripts. Het maakt het lezen van de HTML broncode eenvoudiger maar ook het opsporen van eventuele fouten.

2.7 DATATYPES

PHP ondersteunt de volgende belangrijke datatypes:

1. Integers of gehele getallen
2. Floating point getallen of komma getallen
3. Booleans (true/false of waar/onwaar)
4. Strings (letters en andere tekens eventueel gecombineerd met cijfers)
5. Arrays (rijen of reeksen)
6. Objecten

Een arrayvoorbeeld ziet er als volgt uit:

```
$namen = array("Marja", "Hans", "Ronald", "René", "Louis", "Piet");
```

Voor meer informatie over arrays kun je kijken in paragraaf 4 van dit hoofdstuk. (PHP ondersteunt ook objecten, maar die vallen buiten het bestek van deze module.)

Je kunt controleren welk datatype een variabele heeft. Daarvoor beschikt PHP over diverse functies. Om te controleren of de variabele `$getal` van het type `int` is ga je als volgt te werk (2 manieren):

```
1) is_int($getal);
2) is_numeric($getal);
```

De uitkomst zal `true` (1) zijn als het klopt en `false` (0) als het niet zo is.

2.8 REKENKUNDIGE OPERATOREN

Operatie	Rekenkundige bewerking
$X + Y$	X wordt bij Y opgeteld als beide numeriek (getallen) zijn
$X - Y$	Y wordt van X afgetrokken
$X * Y$	X wordt vermenigvuldigd met Y
X / Y	X wordt gedeeld door Y
$X \% Y$	Modulus van X en Y (het getal dat overblijft als je X door Y deelt)
$X ^ Y$	X tot de macht Y
$X . Y$	X wordt achter Y geplakt. Het maakt niet uit of het getallen zijn of Strings

Voorbeeld script:

```
<?php
$dag = 1;
$maand = "januari";
$gebeurtenis = "nieuwjaarsdag";
print($dag . " " . $maand . " is " . $gebeurtenis);
?>
```

Resultaat in browser:

1 januari is nieuwjaarsdag

Ook hier is gebruik gemaakt van de puntoperator. Die dient ervoor om gegevens aan elkaar te plakken (concatenatie). In dit geval variabelen (woorden) en spaties.

2.9 VERGELIJKINGSOPERATOREN

Vergelijkingsoperatoren vergelijken de waarden van twee variabelen en geven een Boolean waarde (waar(true) of onwaar(false)) terug.

Vergelijking	Waar als
<code>X == Y</code>	X en Y qua waarde aan elkaar gelijk zijn
<code>X != Y</code>	X en Y ongelijk aan elkaar zijn
<code>X === Y</code>	X identiek is aan Y, zowel qua datatype als waarde
<code>X > Y</code>	X groter is dan Y
<code>X < Y</code>	X kleiner is dan Y
<code>X >= Y</code>	X groter is dan of gelijk is aan Y
<code>X <= Y</code>	X kleiner is dan of gelijk is aan Y
<code>X && Y</code>	zowel X als Y waar zijn
<code>X Y</code>	óf X óf Y waar is
<code>!X</code>	X is onwaar

Voorbeeld script:

```
<?php
$getal_1 = 15;
$getal_2 = 25;
if($getal_1 > $getal_2){
    print("$getal_1 is groter dan $getal_2");
}
else{
    print("$getal_1 is kleiner dan $getal_2");
}
?>
```

2.10 HET GEBRUIK VAN CONTROLESTRUCTUREN IN PHP

2.10.1 HET IF-STATEMENT

Het voorbeeld uit paragraaf 2.9 bevat een enkelvoudig if-statement met een else component.

Na de expressie (voorwaarde) volgt ofwel een enkelvoudig statement of een opeenvolging van statements tussen accolades geplaatst. Zorg er wel voor dat elk statement wordt afgesloten met een puntkomma.

Enkelvoudig statement (if of if-else)	Meervoudige statements (if of if-else)
<pre>If(expressie) Statement;</pre>	<pre>If(expressie) { Statement; Statement; }</pre>
<pre>If(expressie) Statement; else Statement;</pre>	<pre>If(expressie) { Statement; Statement; } else { Statement; Statement; }</pre>

Plaats echter geen puntkomma na het sluihaakje van de voorwaarde en ook niet na de `else`. De regels die hier gelden zijn ook van toepassing op de `for`-loop, de `while`-loop en de `switch`.

2.10.2 DE FOR-LOOP

Voorbeeld van een `for`-loop:

Enkelvoudig statement (for)	Meervoudige statements (for)
<code>for(initialisatie; test; ophoging/verlaging) Statement;</code>	<code>for(initialisatie; test; ophoging/verlaging) { Statement; Statement; etc. }</code>

```
$namen = array("Marja", "Hans", "Ronald", "René", "Louis", "Piet");
```

Wil je nu alle namen uit de array in één keer onder elkaar in een genummerde lijst in een webpagina plaatsen, kun je de volgende gebruiken:

```
<html>
<head>
<title>Weergeven van namen met een for-loop</title>
</head>
<body>

<?php

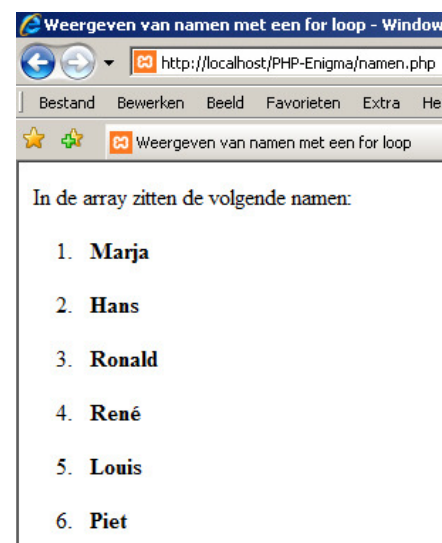
/* Deze loop zal alle namen van de array in een genummerde lijst onder
elkaar op het scherm zetten. */

$namen = array("Marja", "Hans", "Ronald", "René", "Louis", "Piet");
print("In de array zitten de volgende namen:<br>");
print("<ol>");
for($i = 0; $i < count($namen); $i++)
    print ( "<li><h4>$namen[$i]</h4></li>" );
print("</ol>");

?>

</body>
</html>
```

Als je het script uitvoert, krijg je het volgende te zien:



2.10.3 HET FOREACH-STATEMENT

Hetzelfde voorbeeld van de array maar nu met het `foreach` statement.

```
foreach($namen as $key => $value) {
    $key++; // ophogen van de teller om hem op 1 te zetten
    print "$key. <b>$value</b><br>";
}
```

Een array bestaat uit een aantal elementen met een index (`$key`) en een bijbehorende waarde (`$value`). Om nu elke index met bijbehorende waarde te laten zien gebruik je bovenstaande notatie.

2.10.4 DE WHILE-LOOP

Voorbeeld van de array met namen maar nu middels een `while`-loop

```
<html>
<head>
<title>Weergeven van namen met een while-loop</title>
</head>
<body>

<?php

/* Deze loop zal alle namen van de array in een genummerde lijst
onder elkaar op het scherm zetten. */

$namen = array("Marja", "Hans", "Ronald", "René", "Louis", "Piet");
print("In de array zitten de volgende namen:<br>");
print("<ol>");
$i = 0; // initialisatie om eerste element uit array weer te geven
while($i < count($namen) {
    print ( "<li><h4>$namen[$i]</h4></li>" );
    $i++;
}
print("</ol>");

?>

</body>
</html>
```

2.10.5 DE SWITCH

Als er een keuze gemaakt kan worden uit meerdere opties is het handig om een `switch` te gebruiken. Stel je wilt je website verschillende achtergrondkleuren laten krijgen, dan kun je dat o.a. door een `switch` doen. De code voor die `switch` ziet er dan bijvoorbeeld als volgt uit:

```
<?php

$achtergrondkleur = $_POST['achtergrondkleur'];

    switch($achtergrondkleur){
    case "grijs":
        $achtergrondkleur = "#aaaaaa";
        break;
    case "groen":
        $achtergrondkleur = "#aaffaa";
        break;
    case "blauw":
        $achtergrondkleur = "#abcdef";
        break;
    case "zalm":
        $achtergrondkleur = "#fedcba";
        break;
    default:
        $achtergrondkleur = "white";
        break;
    }

?>
```

De switch syntax werkt als volgt:

```
switch(input) {
    case voorwaarde 1:
        code die moet worden uitgevoerd als input = voorwaarde 1;
        break;
    case voorwaarde 2:
        code die moet worden uitgevoerd als input = voorwaarde 2;
        break;
    case etc.

    default:
        code die moet worden uitgevoerd als input ≠ alle cases;
}
```

De input is meestal een variabele. Als aan een bepaalde voorwaarde is voldaan, wordt de code die daarachter staat uitgevoerd en de switch via de break verlaten. Als aan geen der gestelde voorwaarden wordt voldaan, wordt de code uitgevoerd die achter default staat.

opdrachten 5 - 14

2.11 HET WERKEN MET STRINGS IN PHP

Vaak is het handig om te testen of een String variabele die uit een HTML formulier afkomstig is geen waarde heeft, dus een lege String is. Daarvoor gebruiken we de operator ! (uitroepteken).

Bijvoorbeeld:

```
if (!$wachtwoord)

    print("<strong>Je moet een wachtwoord invullen</strong>");
```



Een tweede manier om te kijken of een variabele een lege string is, is het gebruik van de functie `empty()`.

```
if(empty($wachtwoord))  
    print("<strong>Je moet een wachtwoord invullen</strong>");
```

Dit werkt in feite met elk type PHP variabele, om na te gaan of er aan die variabele een waarde is toegekend.

Gebruik de volgende operatoren om twee Strings met elkaar te vergelijken:

Gebruik `==` om te testen of twee Strings gelijk zijn aan elkaar.

Gebruik `<` om te testen of String 1 lexicografisch vóór String 2 komt.

Gebruik `>` om te testen of String 1 lexicografisch na String 2 komt.

Lexicografisch betekent alfabetisch en/of numeriek. Je kunt ook gebruik maken van de operatoren `<=` en `>=` die respectievelijk staan voor "vóór of gelijk aan" en "na of gelijk aan".

De volgende vergelijkingen zijn waar:

```
"abc" == "abc"  
"abc" <  "bcd"  
"bcd" >  "abc"
```

We zullen de `==` operator gebruiken in een PHP script dat een korte beschrijving geeft van de profielen die in de bovenbouw Havo en Vwo voorkomen.

Allereerst maken we een webpagina met een keuzemenu waar je de waarde van de variabele profiel kunt kiezen.

Houd er rekening mee dat het volgende voorbeeld een eenvoudige stringvergelijking uitvoert die werkt op basis van elke vergelijking die gevonden wordt. Daarom is het essentieel dat je ervoor moet zorgen dat de waarden die je in je HTML formulier stopt exact overeenkomen met de waarden waarin je in je PHP script naar op zoek bent.

Code voor het formulier:

```
<html>  
<head>  
<title>Informatie over profielen</title>  
</head>  
<body>  
<h4> De situatie in schooljaar 2007-2008</h4>  
  
<p> Kies een profiel en klik op de knop  
<strong>Informatie</strong>:</p>  
<form method="POST" action="profielinfo.php">  
    <select name="profiel">  
        <option selected> Natuur en Techniek </option>  
        <option> Natuur en Gezondheid </option>  
        <option> Economie en Maatschappij </option>  
        <option> Cultuur en Maatschappij </option>  
    </select>  
<input type="submit" value="Informatie">  
</form>  
</body>  
</html>
```


In het formulier wordt verwezen naar het bestand `profielinfo.php`. Hieronder is daarvoor de code weergegeven.

```
<html>
<head>
<title>
    Informatie over:
    <?php $profiel = $_POST["profiel"]; print( $profiel ); ?>
</title>
</head>
<body>
<?php
if ( $profiel == "Natuur en Techniek" ) {
?>
<h1>Natuur en Techniek</h1>
<p>Dit is het profiel met als profielvakken:<br> wiskunde B,
natuurkunde en scheikunde </p>
<?php
}
else if ( $profiel == "Natuur en Gezondheid" ) {
?>
<h1>Natuur en Gezondheid</h1>
<p>Dit is het profiel met als profielvakken:<br> wiskunde AB (of B),
biologie en scheikunde </p>
<?php
}
else if ( $profiel == "Economie en Maatschappij" ) {
?>
<h1>Economie en Maatschappij</h1>
<p>Dit is het profiel met wiskunde als profielvakken:<br> AB (of B)
en economie </p>
<?php
}
else if ( $profiel == "Cultuur en Maatschappij" ) {
?>
<h1>Cultuur en Maatschappij</h1>
<p>Dit is het profiel met als profielvakken:<br> wiskunde A en
geschiedenis.</p>
<?php
}
?>
</body>
</html>
```

In de browser ziet dat er zo uit:



opdrachten 15 - 18

2.12 HET WERKEN MET MULTIPASS SCRIPTS

Het is mogelijk dat we het voorbeeld van de profielen zodanig aanpassen dat we geen twee aparte webdocumenten nodig hebben. In plaats daarvan kunnen we die twee pagina's combineren tot één script. Dat betekent dat we het webformulier als onderdeel integreren in het PHP script.

In het aangepaste script is een extra voorwaardelijk statement ingebouwd om te bepalen wat er moet gebeuren:

1. Als het script wordt uitgevoerd terwijl er géén keuze is gemaakt (in feite geen waarde aan de variabele is toegekend). Dan moet alleen het formulier worden getoond, eventueel met de aanvulling dat er een selectie moet worden gemaakt.
2. Als het script wordt uitgevoerd als er wel een waarde aan de variabele is toegekend. Dan wordt de informatie van het profiel getoond zoals in het vorige voorbeeld is uitgewerkt.

Dit is een voorbeeld van een multipass script - een script dat zichzelf aanroept om te worden uitgevoerd. Het complete script ziet er als volgt uit:

```
<html>
<head>
<title>Informatie over profielen</title>
</head>
<body>

<?php
    $profiel = $_POST["profiel"]; // Haal de gegevens op van selectie
    if ( !$profiel ) {
?>
<h4> De situatie in schooljaar 2007-2008</h4>
<p> Kies een profiel en klik op de knop
<strong>Informatie</strong>:</p>
<form method="POST" action="profielinfo2.php">
<select name="profiel">
<option selected> Natuur en Techniek </option>
<option> Natuur en gezondheid </option>
<option> Economie en Maatschappij </option>
<option> Cultuur en Maatschappij </option>
</select>
<input type="submit" value="Informatie">
</form>
<?php
}
    else if ( $profiel == "Natuur en Techniek" ) {
?>
<h1>Natuur en Techniek</h1>
<p>Dit is het profiel met als profielvakken:<br> wiskunde B,
natuurkunde en scheikunde </p>
<?php
}
    else if ( $profiel == "Natuur en Gezondheid" ) {
?>
<h1>Natuur en Gezondheid</h1>
```

```
<p>Dit is het profiel met als profielvakken:<br> wiskunde AB (of B),  
biologie en scheikunde </p>
```

```
<?php  
>  
    else if ( $profiel == "Economie en Maatschappij" ) {  
>  
<h1>Economie en Maatschappij</h1>  
<p>Dit is het profiel met wiskunde als profielvakken:<br> AB (of B)  
en economie </p>  
<?php  
>  
    else if ( $profiel == "Cultuur en Maatschappij" ) {  
>  
<h1>Cultuur en Maatschappij</h1>  
<p>Dit is het profiel met als profielvakken:<br> wiskunde A en  
geschiedenis.</p>  
<?php  
>  
</body>  
</html>
```

Hoe werkt het?

Als je het script voor de eerste keer opent is er nog geen optie geselecteerd uit het formulier wat inhoudt dat de variabele `$profiel` nog geen waarde heeft.

Dat betekent dat in dat geval het formuliergedeelte wordt afgebeeld op het scherm om je in staat te stellen een keuze te maken.

Als je op de knop Informatie klikt, roept het script zich zelf aan. In dit geval heeft de variabele `$profiel` een waarde. Die waarde moet je voor de zekerheid ophalen met `$_POST["profiel"];`.

Na een keuze en de bevestiging wordt de informatie over het profiel getoond en niet het formulier.

Opmerkingen:

Bij de action in de form-tag op de vorige bladzijde kun je behalve de eigen naam van het document ook de volgende code invullen:

```
<form method="POST" action="<?php print $_SERVER["PHP_SELF"]; ?>">
```

Dit is ook een mogelijkheid om hetzelfde bestand nog een keer op te roepen na een submit.

Naast de POST-methode bestaat ook de mogelijkheid om gebruik te maken van de GET-methode om gegevens te verzenden en/of op te vragen. Om de gegevens te verkrijgen moet je dan in de PHP-code gebruik maken van de instructie `$_GET["inputnaam"]` waarbij "inputnaam" de naam is van het invulvenster in een formulier. Je moet dan uiteraard in het formulier wel bij method "GET" invullen. Bijvoorbeeld:

```
<form method = "GET" action="profielinfo.php">  
<select name="profiel">
```



Ophalen van gegevens →

```
<?php
$profiel = $_GET["profiel"];
?>
```

Op webservern met PHP 4.2.0 of hoger, en dat is tegenwoordig standaard, zijn de waarden niet direct meer benaderbaar via de GET-methode.

De GET-methode is een stuk onveiligere dan de POST-methode omdat de gegevens zichtbaar worden in de adresbalk van de browser. Bijvoorbeeld:

```
http://localhost/PHP_Enigma/hallojij.php?naam=Jan&wachtwoord=jan123
```

Hier wordt behalve de naam ook het wachtwoord afgebeeld. Dat kan natuurlijk niet de bedoeling zijn. Bij de POST-methode heb je dat niet.

opdrachten 19 - 23

3 FUNCTIES IN PHP

PHP heeft een grote hoeveelheid standaard functies die je uitgebreid gedocumenteerd kunt raadplegen in de functielijst op www.php.net. Maar naast die standaard functies biedt PHP jou ook de mogelijkheid zelf functies te maken. Dit kan handig zijn als je op webpagina's regelmatig dezelfde code wilt gebruiken.

3.1 FUNCTIES IN HET ALGEMEEN

Een functie in PHP bestaat in het algemeen uit 3 verschillende onderdelen:

- De functienaam voorafgegaan door het woord `function` en een spatie.
- Haakjes `()` achter de functienaam waarbinnen door een komma gescheiden argumenten kunnen voorkomen.
- De functie body omsloten door accolades `{ }`.

Format van een functie:

```
function functienaam($arg1, $arg2, ..., $argn) {
    code sectie
}
```

Stel je maakt een website en je wilt op elke pagina een copyright afgebeeld hebben. We maken daarvoor een aparte functie aan.

```
function show_copyright() {
    print("Copyright &copy; 2008 ENIGMA. Alle rechten voorbehouden.");
}
```

Als je in een PHP script deze functie aanroept wordt de copyright regel afgebeeld op de plek waar de functie wordt aangeroepen. Het enige wat je moet onderhouden is het jaartal in de functie.

Maar ook dat is met behulp van een ingebouwde PHP functie aan te passen. Stel dat het jaar 2009 eraan komt en je wilt vanaf nu dat je er geen omkijken meer naar hebt kun je gebruik maken van de functie `date()`.

De nieuwe functie `show_copyright()` gaat er dan als volgt uitzien:

```
function show_copyright() {  
    print("Copyright &copy; 2007 - date("Y") ENIGMA.  
        Alle rechten voorbehouden.");  
}
```

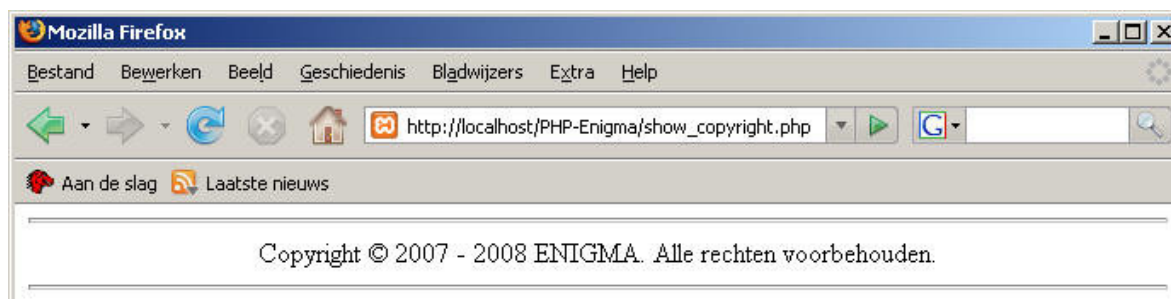
`date("Y")` geeft het jaartal weer in 4 cijfers, bijvoorbeeld 2008 tijdens het schrijven van deze module.

Het PHP script om deze functie te gebruiken ziet er als volgt uit:

```
<?php  
  
function show_copyright() {  
    print("Copyright &copy; 2007 - " . date("Y") . " ENIGMA . Alle  
        rechten voorbehouden.");  
}  
  
show_copyright();  
  
?>
```

Als functies binnen functies gebruikt worden spreken we van het nesten van functies.

De copyright regel wordt dan als volgt op het scherm weergegeven:



Door de code `show_copyright();` in elke webpagina op te nemen waar je het copyright wilt laten zien kun je dat verwezenlijken.

Het kan zijn dat je als webmaster van verschillende websites eenzelfde copyright regel wilt gebruiken. Alle websites hebben verschillende namen. Nu kun je een variabele `$site_naam` introduceren. Met behulp van deze variabele kun je nu elke website voorzien van een unieke copyright regel. Je geeft nu de functie `show_copyright()` het argument `$site_naam` mee.

Stel in ons geval betreft het de website `www.enigma-online.nl`. Dan komt het script met de functie er als volgt uit te zien:

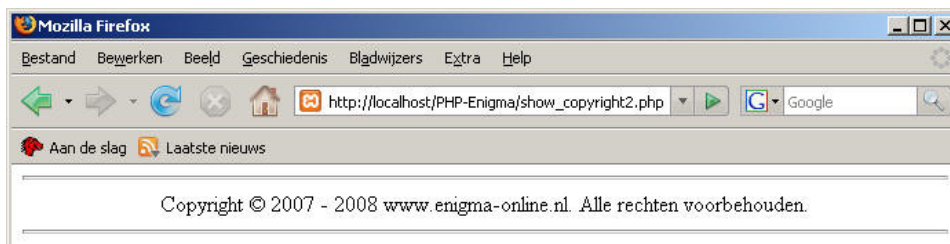


```
<?php

function show_copyright($site_naam) {
    print("Copyright &copy; 2007 - " . date("Y") . " $site_naam.
        Alle rechten voorbehouden.");
}
$site_naam = "www.enigma-online.nl";
show_copyright($site_naam);

?>
```

Uitvoering van dit script levert het volgende resultaat:



3.2 FUNCTIES MET TERUGKEERWAARDE

Vaak is het handig om functies te maken die een waarde als output geven. Dit wordt mogelijk gemaakt om bij het aanroepen van de functie een variabele een waarde toe te kennen. Het maakt niet uit welk datatype die variabele heeft; zelfs lijsten en arrays kunnen worden weergegeven.

Het volgende voorbeeld zal een en ander verduidelijken. Stel we hebben een bedrijf en we moeten de verkoopprijs van producten gaan bepalen. Daar hoort in veel gevallen het tarief van 19% BTW bij. Het zou handig zijn als we in een functie in een keer het eindbedrag voor de klant zouden kunnen bepalen.

Hoe gaan we te werk?

- Er moet een variabele `$prijs` en een variabele `$BTW` zijn.
- Er moet een functie `bereken_eindprijs()` gedeclareerd worden waarin de parameters `$prijs` en `$BTW` in worden opgenomen.
- Binnen de functie moet de verkoopprijs worden berekend en teruggegeven met `return()`.
- De functie moet binnen het script worden aangeroepen.
- In een boodschap moet de variabele `$verkoopprijs` worden meegedeeld.

In een script ziet dat er als volgt uit:

```
<?php

$prijs = 24.95;
$btw = .19;

function bereken_eindprijs($btw,$prijs) {
    $verkoopbelasting = $btw;
    return $prijs + ($prijs * $verkoopbelasting);
}
// Let op hoe de functie bereken_eindprijs() de waarde teruggeeft
```

```
$eindprijs = bereken_eindprijs($btw,$prijs);  
// Rond het bedrag af op twee decimalen  
$eindprijs = round($eindprijs,2);  
  
print "Verkoopprijs zonder BTW = &euro; $prijs<br>\n";  
print "Btw-tarief = 19%<br>\n";  
print "De eindprijs = &euro; $eindprijs";  
  
?>
```

Het is ook mogelijk om functies met returnwaarden rechtstreeks in controlestructuren te gebruiken. Het volgende voorbeeld zal dat hopelijk duidelijk maken.

Stel we willen online gaan shoppen. Maar een bepaalde webshop wordt gewerkt met een kredietlimiet. Elke keer als je iets in je winkelwagentje stopt wordt het bedrag dat je moet betalen bij de andere gekochte producten opgeteld en bovendien wordt dat totaalbedrag vergeleken met de kredietlimiet die je hebt. Het zou mooi zijn als bij de overschrijding van de limiet je gewaarschuwd wordt of als je nog kredietlimiet over hebt dat aangegeven wordt hoeveel dat is.

Dat kun je als volgt realiseren:

```
<?php  
  
$aankoopbedrag = 234.50;  
$kredietlimiet = 500.00;  
  
function check_kredietlimiet($aankoopbedrag, $kredietlimiet) {  
  
    if($aankoopbedrag > $kredietlimiet) {  
        return false;  
    }  
    else {  
  
        return true;  
    }  
  
}  
  
if (check_kredietlimiet($aankoopbedrag, $kredietlimiet)){  
    $bestedingsbedrag = number_format(($kredietlimiet -  
    $aankoopbedrag),2);  
    // Laat de klant rustig doorwinkelen  
    print "Winkel gerust nog even door! <br>";  
    print "U bent nog &euro; " . $bestedingsbedrag . " verwijderd  
        van uw kredietlimiet."  
}  
else {  
    $overschrijding = number_format(($aankoopbedrag -  
    $kredietlimiet),2);  
    print "Uw aankoopbedrag overstijgt de kredietlimiet met &euro;  
        ". $overschrijding . "!";  
}  
  
?>
```



Hoe werkt de functie `check_kredietlimiet()` ?

De functie heeft 2 parameters: `$aankoopbedrag` en `$kredietlimiet` . Als het aankoopbedrag groter is dan de kredietlimiet geeft de functie de returnwaarde of terugkeerwaarde 0 (`false`). Zoniet dan is de returnwaarde 1 (`true`) . Als de functie bij de check een 0 geeft, betekent dat dat de vergelijking `$aankoopbedrag > $kredietlimiet` niet klopt en dus het aankoopbedrag de kredietlimiet niet overstijgt, waardoor dus een 1 als returnwaarde moet worden weergegeven.

In Java hebben we deze constructies ook al eens gezien. We noemen dit dan geen functies maar methoden. Je hebt daar zogenaamde void-methoden zonder returnwaarde en methoden met terugkeerwaarden. (Zie deel 1, hoofdstuk 4)

3.3 HET ONDERBRENGEN VAN FUNCTIES IN EEN APART BESTAND

Er zijn nog meer typen functies te ontwerpen, zoals functies met meerdere returnwaarden, recursieve functies en variabele functies. Kijk daarvoor in de uitvoerige functielijst van PHP in de PHP-documentatie. Als je meerdere functies gebruikt die vaker terugkomen is het handig een zogenaamde functiebibliotheek aan te leggen. Je kunt alle gebruikte functies in één PHP bestand zetten en dat bijvoorbeeld opslaan als `functions.php` . Het enige wat je dan nog te doen hebt is dat bestand inladen in een PHP pagina d.m.v. de PHP functie `include()` .

De code om de eindprijs te berekenen zou er dan als volgt kunnen uitzien:

```
<?php
include("functions.php");

$prijs = 24.95;
$btw = .19;

// Let op hoe de functie bereken_eindprijs() de waarde teruggeeft
$eindprijs = bereken_eindprijs($BTW,$prijs);
// Rond het bedrag af op twee decimalen
$eindprijs = round($eindprijs,2);

print "Verkoopprijs zonder BTW = &euro; $prijs<br>\n";
print "Btw-tarief = 19%<br>\n";
print "De eindprijs = &euro; $eindprijs";
?>
```

opdrachten 24 - 26

4 ARRAYS

In bijna alle programmeer- en scripttalen is het mogelijk om met arrays te werken. Zo ook in PHP. We gaan nu kijken naar wat arrays zijn en hoe je ze kunt gebruiken.

4.1 WAT ZIJN ARRAYS?

Een array is een variabele die meerdere waardes tegelijk kan bevatten. Je kunt de individuele waardes uit een array benaderen door een indexgetal te gebruiken. Bekijk het onderstaande stukje PHP code:

```
<?php

$mijnArray = array("hallo", "ik", "ben", "een", "array");

print $mijnArray[0];

?>
```

In de eerste regel code wordt het array gevuld. In dit geval hebben we het array `$mijnArray` genoemd (het is dus gewoon een variabele). We hebben het array gevuld met 5 waardes, namelijk de Strings: "hallo", "ik", "ben", "een" en "array".

In de tweede regel staat een `print` statement, dus we drukken iets af op het scherm. Wat drukken we af? Het eerste element uit het array. Het resultaat van deze code zal dus zijn dat er "hallo" op het scherm wordt afgedrukt.

Let op, PHP begint te tellen bij 0! Onze array heeft dus 5 waarden, maar je moet deze waarden benaderen met `$mijnArray[0]` t/m `$mijnArray[4]`.

opdracht 27

Je kunt de waardes van een array veranderen. Dit gaat ongeveer op dezelfde manier als het opvragen van de waarde. Bekijk de volgende code:

```
<?php

$mijnArray = array("hallo", "ik", "ben", "een", "array");

print $mijnArray[0];

$mijnArray[0] = "dag";

print $mijnArray[0];

?>
```

Het resultaat van de code zal zijn dat eerst het woord hallo wordt afgedrukt, gevolgd door het woord dag.

opdracht 28



4.2 NUTTIGE ARRAY-OPERATIES

Er zijn een aantal handige operaties die je op arrays kunt gebruiken. Bekijk de code:

```
<?php

$mijnArray = array(1,2,3,4,5);
print count($mijnArray);
print "<br>";
$mijnArray[] = 6;
print count($mijnArray);

?>
```

Met de operatie `count()` kun je de lengte van een array opvragen. Dat betekent dat de uitkomst het getal is van het totaal aantal elementen in de array.

De regel `$mijnArray[] = 6` voegt het getal 6 toe als element achter aan het array.

In de code wordt dus de lengte van het array afgedrukt. Vervolgens wordt er een element aan het array toegevoegd. Tenslotte wordt weer de lengte van het array afgedrukt. De uitkomst zal er als volgt uit zien:

```
5
6
```

Er zijn ook functies om de elementen van een array om te draaien en om de elementen te sorteren:

```
<?php
$mijnArray = array(2,1,5);
print($mijnArray[0] " , " $mijnArray[1] " , " $mijnArray[2];
print "<br>";

$mijnArray2 = array_reverse($mijnArray);
print $mijnArray2[0] " , " $mijnArray2[1] " , " $mijnArray2[2];
print "<br>";

sort($mijnArray);
print $mijnArray[0] " , " $mijnArray[1] " , " $mijnArray[2];
print "<br>";
?>
```

In het eerste blok wordt een array gevuld met 3 elementen en afgedrukt. In het tweede blok wordt dat array omgedraaid en afgedrukt. In het derde blok wordt het array gesorteerd en afgedrukt.

De uitkomst ziet er als volgt uit:

```
2 , 1 , 5
5 , 1 , 2
1 , 2 , 5
```

opdrachten 29 - 30

Het handige aan arrays is dat je makkelijk door de elementen heen kunt lopen met een loopje. Bekijk de volgende code:

4.3 ARRAYS DOORLOPEN MET DE FOR-LOOP

```
<?php

$mijnArray = array("pindakaas", "is", "lekker");

for ($i=0; $i<3; $i++)
{
    print $mijnArray[$i];
    print "<br>";
}

?>
```

De uitkomst van deze code zal zijn:

```
pindakaas
is
lekker
```

We kunnen dit ook op een nog slimme manier doen:

```
<?php
$mijnArray = array("pindakaas", "is", "lekker");

for ($i=0; $i<count($mijnArray); $i++)
{
    print $mijnArray[$i];
    print "<br>";
}

?>
```

opdrachten 31 - 36

4.4 2D ARRAYS

We hebben al gezien dat de elementen van een array verschillende soorten waarden kunnen hebben, bijvoorbeeld getallen of Strings. Maar wat dacht je er van als de elementen van een array ook arrays zijn? Dan krijg je dus een array van arrays, ofwel een tweedimensionaal array. Bekijk de volgende code:

```
<?php

$rij1 = array(1,2,3);
$rij2 = array(4,5,6);
$rij3 = array(7,8,9);

$multi = array($rij1, $rij2, $rij3);

print ($multi[0][2]);

?>
```



In de code worden eerst 3 arrays van 3 elementen aangemaakt. Vervolgens wordt er een nieuw array gemaakt met als elementen de bovenstaande arrays. Daarna wordt het derde element van de eerste array afgedrukt naar het scherm. Let weer op dat we beginnen te tellen bij 0! De uitvoer zal zijn:

3

Ook 2D arrays kunnen met een loop worden doorlopen. Omdat je twee dimensies hebt, heb je ook een dubbele loop nodig. Dit heeft deze vorm:

```
for ($i=0; $i<10; $i++)
{
    for ($j=0; $j<10; $j++)
    {
        //code voor elk element
    }
}
```

opdrachten 37 - 39

4.5 ARRAYFUNCTIES GEBRUIKEN OM GEGEVENS WEER TE GEVEN UIT DATABASES

Als je met PHP en MySQL gegevens uit een MySQL database haalt, kom je ook in aanraking met arrays. Als je een query uitvoert en de functie `mysql_fetch_array()` gebruikt, dan krijg je een rij uit de database als array terug. De elementen van dit array zijn de waardes van de verschillende velden van die regel uit de database. Kijk in paragraaf 5.5 van dit hoofdstuk voor een voorbeeld.

5 PHP EN DATABASES

5.1 HET BENADEREN VAN EEN DATABASE MET PHP EN MYSQL

In deze module concentreren we ons vooral op het gegevens uit een database te halen en die in een webpagina te tonen en gegevens in een database op te slaan. PHP is in staat om (My)SQL bewerkingen door te voeren op een relationeel database managementsysteem (DBMS), maar we zullen ons alleen maar toeleggen op de `SELECT`, `INSERT` en `UPDATE` bewerkingen.

Het aanmaken van een database met tabellen met behulp van phpMyAdmin kun je vinden in bijlage 3

Het halen van gegevens uit een database met behulp van PHP verloopt in 4 stappen:

1. Maak contact met de database server via Database Management Services.
2. Kies op de server de juiste database.
3. Voer de query uit en geef de resultaten weer in een webpagina.
4. Verbreek de verbinding met de databaseserver.

PHP heeft voor elk DBMS een set database functies waarmee het kan werken.

In ons geval kijken we alleen maar naar het MySQL server.

opdrachten 40 - 41

5.2 DE FUNCTIE MYSQL_CONNECT()

Format:

```
$verbinding = mysql_connect($hostnaam,$gebruikersnaam,$wachtwoord);
```

Werking: maak een verbinding met het MySQL DBMS dat op de computer draait met de naam `hostnaam` waarbij je de vereiste gebruikersnaam en wachtwoord ingeeft.

De functie zorgt ervoor dat een foutmelding in beeld gebracht wordt als er om een of andere reden geen verbinding met het DBMS tot stand gebracht kan worden. Dat kan in de trant van: verkeerd wachtwoord, verkeerde gebruikersnaam, DBMS is niet op die computer aangetroffen.

We gaan ervan uit dat XAMPP of lokaal op de machines is geïnstalleerd of via het netwerk bereikbaar is. Daarin zit phpMyAdmin als DBMS. Je kunt hiermee werken door op de adresbalk van je browser in te typen:

```
localhost/phpmyadmin.
```

Voor de voorbeelden in deze module gebruiken we de volgende gegevens:

Host:	localhost
Gebruikersnaam:	root
Wachtwoord:	" "

De verbinding met het DBMS ziet er dan als volgt uit:

```
$verbinding = mysql_connect("localhost","root","");
```

Dit is in feite een erg onveilige situatie. Je moet eigenlijk nooit als gebruiker `root` een database benaderen. Bovendien is er hier geen wachtwoord ingesteld waardoor het voor hackers wel erg eenvoudig wordt om je database te benaderen. We hebben deze situatie gekozen omdat het een eenvoudige manier is om mee te werken. Zorg er dus in alle gevallen voor dat je bij het maken van websites die gebruik maken van een database wel goed beveiligd zijn.

opdracht 42

5.3 DE FUNCTIE MYSQL_SELECT_DB()

Format:

```
$db = mysql_select_db("database");
```

Werking: selecteert een specifieke database die gehost is door het DBMS.

In onze voorbeelden zullen we gebruik maken van de database `enigma_php`. Dat betekent dat we de database kunnen selecteren door de code:

```
$db = mysql_select_db("enigma_php");
```

Onze database `enigma_php` bevat de tabel `landen` met verschillende gegevens



land_id	land	hoofdstad	werelddeel	inwoners
1	Frankrijk	Parijs	Europa	59000000
2	Ierland	Dublin	Europa	3660000
3	Griekenland	Athene	Europa	10700000
4	Verenigde Staten	Washington	Amerika	281000000
5	Japan	Tokio	Azië	130000000
6	Zuid Afrika	Pretoria	Afrika	43800000
7	Marokko	Rabat	Afrika	31200000
8	China	Beijing	Azië	1200000000

5.4 DE FUNCTIE MYSQL_QUERY()

Format:

```
$resultaat = mysql_query($query);
```

Werking: zendt de SQL query naar de database.

De functie stuurt een PHP resultaat object terug dat alle rijen van een tabel uit een database bevat die de uitkomst vormen van de opgegeven query.

We kunnen nu een volgende SELECT query maken:

```
$query = "SELECT * FROM landen ORDER BY inwoners";
$resultaat = mysql_query($query);
```

5.5 DE FUNCTIE MYSQL_FETCH_ARRAY()

Format:

```
$row = mysql_fetch_array($resultaat);
```

haalt de volgende rij uit het objectresultaat. Geeft als terugkeerwaarde `false` als er geen nieuwe rijen meer zijn. Normaal wordt deze functie gebruikt in een `while` loop:

```
while ($row = mysql_fetch_array(resultaat) ) {
    doe iets met die rij;
}
```

Elke kolom van een rij wordt benaderd met een array notatie:

De eerste kolom is `row["land_id"]`

De tweede kolom is `row["land"]`

Etc.

Om nu de eerste kolom van alle rijen van de uitkomst van een query in beeld te brengen gebruik je de volgende code:

```
while ($row = mysql_fetch_array($resultaat)) {
    print($row["land_id"]. "<br>\n");
}
```

Nu zullen we al het voorafgaande eens in een compleet voorbeeld laten zien: selecteer alles uit de tabel landen en toon dat als een tabel op het scherm gesorteerd op werelddeel.

De code hiervoor is:

```
<html>
<head>
    <title>Lijst van alle landen uit de tabel</title>
</head>
<body>
<h2>Lijst van alle landen per werelddeel</h2>
<table border="1" cellpadding="2">
    <tr>
        <th>land_id</strong></th>
        <th>land</strong></th>
        <th>hoofdstad</strong></th>
        <th>werelddeel</strong></th>
        <th>inwoners</strong></th>
    </tr>
<?php
    mysql_connect("localhost","root","");
    mysql_select_db("enigma_php");
    $query = "SELECT * FROM landen ORDER werelddeel";
    $resultaat = mysql_query($query);
    while ($row = mysql_fetch_array($resultaat)) {
?>
        <tr>
            <td><?php print($row["land_id"]); ?></td>
            <td><?php print($row["land"]); ?></td>
            <td><?php print($row["hoofdstad"]); ?></td>
            <td><?php print($row["werelddeel"]); ?></td>
            <td><?php print($row["inwoners"]); ?></td>
        </tr>
<?php
    }
?>
</table>
</body>
</html>
```

Als de code goed is ingevoerd krijg je iets te zien dat hier op lijkt:

Lijst van alle landen per werelddeel

land_id	land	hoofdstad	werelddeel	inwoners
6	Zuid Afrika	Pretoria	Afrika	43800000
7	Marokko	Rabat	Afrika	31200000
5	Japan	Tokio	Azië	130000000
8	China	Beijing	Azië	1200000000
1	Frankrijk	Parijs	Europa	59000000
2	Ierland	Dublin	Europa	4200000
3	Griekenland	Athene	Europa	10700000
4	Verenigde Staten	Washington	Noord-Amerika	281000000



We kunnen de select query naar believen aanpassen om de gebruiker te laten bepalen wat voor informatie hij wil zien.

Normaal wordt dat gedaan door een HTTP variabele naar het script te sturen die dan als onderdeel van de query kan worden gebruikt.

Laten we het vorige script eens aanpassen:

Zorg ervoor dat er een HTTP variabele `$werelddeel` als input komt (bijvoorbeeld uit een webformulier)

Gebruik deze variabele om alleen maar de landen te selecteren uit dat specifieke werelddeel

```
SELECT * FROM landen WHERE werelddeel = \"$werelddeel\";
```

Als de waarde uit het invoerveld Europa is, wordt de query

```
SELECT * FROM landen WHERE werelddeel = \"Europa\";
```

De eenvoudigste manier om dit script te runnen is door gebruik te maken van de HTTP GET-methode om de variabele `$werelddeel` aan te bieden, waardoor het mogelijk wordt om de variabele aan de URL te koppelen.

Bijvoorbeeld:

```
http://localhost/PHP-Enigma/landenperwerelddeel.php?werelddeel=Europa
```

Als alternatief kun je met de opgedane kennis een webformulier ontwerpen dat de variabele `$werelddeel` doorgeeft.

Het is echter sneller om de vereiste gegevens aan de URL toe te voegen, bijvoorbeeld:

```
?land=Frankrijk
```

Maar wees voorzichtig! Als je niet de goede code aan de URL toevoegt of een formulier dat de gegevens naar het PHP script moet sturen onjuist opzet, zal het script wel runnen maar niet in staat zijn enig resultaat als output te laten zien.

Dit komt omdat de select query specifieke informatie vraagt over het werelddeel waarvan je de landen wilt laten zien.

Een algemene manier om in PHP met databases te werken is de zogenaamde tweetraps procedure:

Allereerst geef je de gebruiker een index waarmee je een samenvatting van alle items in de database te zien krijgt.

Daarna geef je de gebruiker de mogelijkheid om op elk item te klikken om alle gegevens per item weer te geven.

Dit patroon zie je vaker op het web. Bij www.bol.com kun je bijvoorbeeld voor boeken kiezen van een schrijver. Je krijgt dan alle titels in beeld en elke titel heeft een link waarachter een uitgebreide informatie van die titel te vinden is.

We zullen twee scripts gaan gebruiken: het index script en het detail script

De code van het script `landenindex.php`:


```
<html>

<head>
  <title>Index van alle landen</title>
</head>

<body><h2>Index van alle landen</h2>
<table border="1" cellpadding="2">
  <tr>
    <th>id</th>
    <th>naam</th>
  </tr>

  <?php
    mysql_connect("localhost","root","");
    mysql_select_db("enigma_php");
    $query = "SELECT land_id, land FROM landen ORDER BY land_id";
    $resultaat = mysql_query($query);
    while($row = mysql_fetch_array($resultaat)) {
  ?>
    <tr>
      <td><?php print($row["land_id"]); ?></td>
      <td><?php print
        ("<a href=\"landendetails.php?land=row[land]\">$row[land]</a>");
        ?></td>
    </tr>
  <?php
  }
  ?>
</table>

</body>

</html>
```

Het resultaat ziet er dan als volgt uit:

Index van alle landen	
land_id	land
1	Frankrijk
2	Ierland
3	Griekenland
4	Verenigde Staten
5	Japan
6	Zuid Afrika
7	Marokko
8	China



In het bestand `landenindex.php` wordt verwezen naar de code van `landendetails.php`. Dat bestand ziet er als volgt uit:

```
<?php
//Gebruik de functie addslashes() om misbruik (SQL-Injection) te voorkomen.
if($_GET["land"]){
    $land = addslashes($_GET["land"]);
}
else{
    $land = addslashes($_POST["land"]);
}
?>

<html>
<head>
<title>Details van <?php print($land); ?></title>
</head>
<body>
<h2 style="color:navy"> Details van <?php print($land); ?></h2>
<?php
    mysql_connect("localhost","root","");
    mysql_select_db("enigma_php");
    $query = "SELECT * FROM landen WHERE land = \"\$land\"";
    $resultaat = mysql_query($query);
    while($row = mysql_fetch_array($resultaat)) {
?>
<p>Landnummer:    <b><?php print($row["land_id"]); ?></b></p>
<p>Hoofdstad:     <b><?php print($row["hoofdstad"]); ?></b></p>
<p>Werelddeel:    <b><?php print($row["werelddeel"]); ?></b></p>
<p>Inwoners:      <b><?php print($row["inwoners"]); ?></b></p>
<?php
    }
?>
</body>
</html>
```

Klik vervolgens op de link Japan. Dan krijg je het volgende in beeld:



Let op hoe de URL op de adresbalk er uit ziet! Merk bovendien op dat aan het begin van het script de waarde van het gekozen land wordt opgehaald via de GET-methode of de POST-methode. Het is handig om dat op deze wijze te doen. Mocht de GET-methode niet werken heb je altijd nog de POST-methode achter de hand.

opdrachten 43 - 44

5.6 HET UITVOEREN VAN INSERT- EN UPDATE-QUERYS

Om gegevens in een database in te voeren kun je natuurlijk als webmaster via phpMyAdmin iedere keer braaf gegevens invoeren. Handiger zou het zijn als dit automatisch kan gebeuren. Stel je wilt een forum op Internet beheren en je zou van alle gebruikers zelf de gegevens moeten inkloppen. Dat is haast onbegonnen werk. Nu is er een mogelijkheid om met PHP en MySQL deze gegevens automatisch te verwerken. Je maakt daarvoor een geschikt webformulier waarin men de benodigde gegevens moet invullen, checkt eerst of de juiste gegevens zijn ingevuld en vervolgens sla je via een PHP script de gegevens middels een `INSERT` query op in de daarvoor bestemde tabel(len) in de database.

Wij werken hier het voorbeeld uit van onze landentabel. Hoe krijgen we via een invulformulier meerdere landen in de tabel opgeslagen. We zullen dus een formulier moeten maken met verschillende inputs en een submit knop. Bovendien moet dat formulier via de action verwijzen naar een PHP bestand. Laten we dat formulier nu eens totaal in PHP maken. Je moet dan wel goed opletten dat je op de juiste plekken dubbele quotes gaat escaperen.

De code van dat formulier ziet er dan bijvoorbeeld zo uit:

```
<?php
print "<html>";
print "<head>";
print "<title>Toevoegen van landen aan de database</title>";
print "</head>";
print "<body>";
print "<h2>Land toevoegen aan database</h2>";
print "<form action=\"landtoevoegen.php\" method=\"post\">";
print "<table cellpadding=\"2\">";
    print "<tr>";
        print "<td align=\"right\">Land:</td>";
        print "<td><input type=\"text\" name=\"land\"";
            print "<size=\"25\"></td>";
    print "</tr>";
    print "<tr>";
        print "<td align=\"right\">Hoofdstad:</td>";
        print "<td><input type=\"text\" name=\"hoofdstad\"";
            print "<size=\"25\"></td>";
    print "</tr>";
    print "<tr>";
        print "<td align=\"right\">Werelddeel:</td>";
        print "<td><input type=\"text\" name=\"werelddeel\"";
            print "<size=\"25\"></td>";
    print "</tr>";
    print "<tr>";
        print "<td align=\"right\">Aantal Inwoners:</td>";
        print "<td><input type=\"text\" name=\"inwoners\"";
            print "<size=\"25\"></td>";
    print "</tr>";
    print "<tr>";
        print "<td><input type=\"submit\" value=\"Voeg land";
            print "<toe\"></td>";
        print "<td><input type=\"reset\" value=\"Wis";
            print "<gegevens\"></td>";
    print "</tr>";
print "</table>";
```

```
print "</form>";
print "</body>";
print "</html>";
?>
```

De opmaak ziet er dan als volgt uit:



Land toevoegen aan database

Land:	<input type="text" value="Oostenrijk"/>
Hoofdstad:	<input type="text" value="Wenen"/>
Werelddeel:	<input type="text" value="Europa"/>
Aantal Inwoners:	<input type="text" value="8200000"/>
<input type="button" value="Voeg land toe"/> <input type="button" value="Wis gegevens"/>	

Je ziet dat de onderdelen d.m.v. een tabel keurig geordend zijn en dat de tekst in de eerste kolom rechts is uitgelijnd. We willen de gegevens van Oostenrijk aan de database toevoegen. In het formulier wordt bij de action verwezen naar het bestand `landtoevoegen.php`.

We gaan nu een apart include-bestand maken waarin we de gegevens zetten die de connectie met de databaseserver tot stand brengen. Dat is zeker uit veiligheidsoverwegingen raadzaam om te doen en bovendien ook praktisch. Want stel dat je van webhost verandert en in elk script de connectiegegevens moet gaan wijzigen, dat is een heleboel werk. Nu hoeft je alleen maar dat ene bestand te veranderen. Dat script ziet er als volgt uit:

```
<?php

$host = "localhost";
$gebruiker = "root";
$wachtwoord = "";
$databasename = "enigma_php";

$verbinding = mysql_connect("$host", "$gebruiker", "$wachtwoord")
or die("Verbinding met de server mislukt vanwege:" . mysql_error());
$db = mysql_select_db("$databasename", $verbinding) or die("Database
error: " . mysql_error());

?>
```

We noemen dit bestand `verbinding.inc.php` en slaan dat op in de map `includes` op de webserver.

Het bestand `landtoevoegen.php` komt er dan als volgt uit te zien:

```
<?php

// Connectie met de database maken
include("includes/verbinding.inc.php");
```

```
// Ophalen van gegevens uit het formulier
$land = addslashes($_POST["land"]);
$hoofdstad = addslashes($_POST["hoofdstad"]);
$werelddeel = addslashes($_POST["werelddeel"]);
$inwoners = addslashes($_POST["inwoners"]);

// Invullen van de gegevens in de tabel landen
$sql = "INSERT INTO landen(id,land,hoofdstad,werelddeel,inwoners)
VALUES('','$land','$hoofdstad','$werelddeel','$inwoners')";
$resultaat = mysql_query($sql);

// Verbreken van de verbinding met de database
$verbreken = mysql_close($verbinding);

// Bevestigen dat de gegevens zijn opgeslagen
print "De gegevens van $land zijn opgeslagen in de database.";
?>
```

Format INSERT INTO query:

```
$sql = "INSERT INTO tabelnaam(kol1 , kol2 , etc.) VALUES('var1',
'var2' , etc.)";
$resultaat = mysql_query($sql)
```

Let op dat bij het gebruik van de variabelen in de `VALUES` single quotes worden gebruikt i.p.v. dubbele quotes. Die hoeft je ook niet te escapen.

De eerste regel is de query die moet worden uitgevoerd. De tweede regel met de functie `mysql_query()` zorgt voor het daadwerkelijk toevoegen van de gegevens in de database.

Als we nu gaan testen of de gegevens inderdaad in de database staan kunnen we dat doe door een `SELECT` query uit te voeren.

Het resultaat ziet er dan als volgt uit:

 Aan de slag
 Laatste nieuws

Lijst van Europese landen

id	naam	hoofdstad	werelddeel	inwoners
1	Frankrijk	Parijs	Europa	59000000
2	Ierland	Dublin	Europa	4200000
3	Griekenland	Athene	Europa	10700000
9	Oostenrijk	Wenen	Europa	8200000

opdrachten 45 - 46

Stel dat er gegevens uit de database moeten worden aangepast. Daarvoor maken we gebruik van de `UPDATE` query. We willen de gegevens voor Ierland aanpassen. Het inwonertal moet worden opgehoogd naar 4,2 miljoen. Dat gaat als volgt:

```
<?php

// Connectie met de database maken
include("includes/verbinding.inc.php");
```



```
// Vervangen van gegevens in de tabel landen
$sql = "UPDATE landen SET inwoners = 4200000 WHERE land =
'Ierland'";
$resultaat = mysql_query($sql);

// Verbreken van de verbinding met de database
mysql_close($verbinding);

// Bevestigen dat de gegevens zijn opgeslagen
print "De gegevens zijn aangepast in de database.";
?>
```

Format UPDATE query:

```
$sql = "UPDATE tabel SET kol1 = waarde WHERE kol2 = waarde";
$resultaat = mysql_query($sql)
```

Je kunt van meerdere kolommen tegelijk de waarden vervangen. Je moet dan wel de kolommen met de bijbehorende waarde scheiden door een komma. Merk op dat je voor cijfers geen quotes hoeft te gebruiken en bij Strings wel, die je overigens weer moet escaper.

5.7 DATABASES EN AFBEELDINGEN

Het gebruik van afbeeldingen in een website komt natuurlijk veel voor. Hoe ga je daar mee om als van jouw website de meeste gegevens in een database staan? Je kunt natuurlijk de afbeeldingen in de database zelf stoppen maar dat is niet aan te raden en wel om de redenen dat de database erg snel heel groot wordt en daardoor ook overbelast kan raken.

Stel je hebt de plaatjes van de vlaggen als jpg-bestanden opgeslagen in een map afbeeldingen en het PHP bestand waarmee je naar die plaatjes gaat vragen staat een directory hoger. Dan moet je in de kolom vlag van de tabel landen (met phpMyAdmin als laatste kolom toegevoegd) voor het bestand `frankrijk.jpg` de link `afbeeldingen/frankrijk.jpg` invoegen.

In phpMyAdmin ziet het resultaat er dan zo uit:

land_id	land	hoofdstad	werelddeel	inwoners	vlag
1	Frankrijk	Parijs	Europa	59000000	afbeeldingen/frankrijk.jpg

Als je nu een query uitvoert met opnieuw als resultaat de landen van Europa maar nu met de bijbehorende vlag zal het script er dan deels als volgt uitzien:

```
<?php

//code voorafgaand aan de query





$sql = "SELECT * FROM landen WHERE werelddeel = 'Europa'";
$resultaat = mysql_query($sql);
```

```
// code achter de query
// weergeven van de afbeelding in de zesde kolom met de <img> tag
<td></td>

?>
```

Het resultaat ziet er dan als volgt uit:

Lijst van Europese landen met vlag

land_id	land	hoofdstad	werelddeel	inwoners	vlag
1	Frankrijk	Parijs	Europa	59000000	
2	Ierland	Dublin	Europa	4200000	
3	Griekenland	Athene	Europa	10700000	
9	Oostenrijk	Wenen	Europa	8200000	

opdracht 47

5.8 HET WERKEN MET MEERDERE TABELLEN

Als je complexe websites bouwt, ben je al gauw toe aan het gebruik van meerdere tabellen in een database. Het is van belang dat deze tabellen goede relaties onderling bezitten. Je moet tenslotte gegevens uit meerdere tabellen kunnen koppelen. Dan is het noodzakelijk gebruik te maken van subqueries of joins.

We gaan dit uitwerken voor onze landentabel. We maken een nieuwe tabel vlaggen aan.

De tabel vlaggen bevat 3 velden, namelijk `vlag_id`, `land_id` en `vlag_link`.

Primaire sleutel is `vlag_id`. Het datatype is `int (3)` en als extra kiezen we voor `auto-increment`. Ook `land_id` is van het type `int (3)`. De kolom `vlag_link` is van het type `varchar (55)`.

We moeten ervoor zorgen dat het `land_id` in de tabel vlaggen overeenkomt met het `land_id` van de tabel landen. Op die manier kunnen we de koppeling tussen de twee tabellen waarborgen.

Stel we willen nu een resultaat tabel hebben waarin alleen de naam van het land en de vlag voorkomen. Het script ziet er als volgt uit:

```
<html>
<head>
    <title>Europese landen met vlag</title>
</head>
<body>
<h2>Europese landen met vlag</h2>
<table border="1" cellpadding="2">
    <tr>
        <th>land</strong></th>
        <th>vlag</th>
    </tr>
<?php
```

```

include("includes/verbinding.inc.php");
$sql = "SELECT landen.land , vlaggen.vlag_link FROM landen , vlaggen
WHERE landen.land_id = vlaggen.land_id AND landen.werelddeel =
\"Europa\" or die(\"kan query niet uitvoeren\");
$resultaat = mysql_query($sql);
while ($row = mysql_fetch_row($resultaat)) {
    ?>
    <tr>
        <td><?php print($row['0']); ?></td>
        <td><img src=\"<?php print($row['1']); ?>\"></td>
    </tr>
<?php
}
?>
</table>
</body>
</html>

```

Let op dat je bij de weergave van het resultaat uitgaat van de nieuw te vormen resultaatstabel. Deze bestaat uit twee kolommen (land en vlag) en derhalve moet je hier de indexen 0 en 1 gebruiken en niet de kolomindexen uit de oorspronkelijke tabellen waaruit de gegevens zijn gehaald.

Het resultaat ziet er dan als volgt uit:

Europese landen met vlag

land	vlag
Frankrijk	
Ierland	
Griekenland	
Oostenrijk	

opdrachten 48 - 49

6 VALIDATIE VAN FORMULIERGEGEVENS MET PHP

PHP String functies zijn erg nuttig om aan de server kant formuliergegevens te valideren.

Daarbij is de multipass script techniek erg behulpzaam:

Het maken van een formulier, de validatiecode en de actie in één script

Als het script voor de eerste keer wordt uitgevoerd verschijnt een leeg formulier.

Als je op de knop drukt, wordt het script opnieuw aangeroepen. Als de data niet correct zijn, wordt het formulier weer getoond (met de gegevens die wel juist zijn, zodat de gebruiker die niet nog een keer hoeft in te vullen) samen met een foutmelding erbij.

Als de ingevulde gegevens correct zijn wordt de eigenlijke scriptactie uitgevoerd.

Typ onderstaande code over in je editor en sla het bestand op in de map werkboek als formuliervalidatie.php.

```

<html>
<head>
<title>Voorbeeld van een formulier validatie</title>
</head>

```



```

<body>
<?php
$naam = addslashes($_POST["naam"]);
$email = addslashes($_POST["email"]);
$opnieuw = addslashes($_POST["opnieuw"]);
if ($naam && $email) {
?>
<p>Welkom op onze website <b><?php print($naam); ?></b></p>
<?php
} else {
?>
<form method="POST" action="formuliervalidatie.php">
<input type="hidden" name="opnieuw" value="yes">
<p><strong>Naam:&nbsp;</strong> <input type="text" name="naam"
value="<?php print( $naam ); ?>" size="40"></p>
<?php
if ( !$naam && $opnieuw )
print( "<b style=\"color:red\">** je moet nog je naam invullen!</b>" );
?>
<p><strong>E-mail:</strong> <input type="text" name="email"
value="<?php print( $email ); ?>" size="30"></p>
<?php
if ( !$email && $opnieuw )
print( "<b style=\"color:red\">** Je moet nog je e-mailadres invullen!</b>"
);
?>
<p><input type="submit" value="Log in">
<input type="reset" value="Wis"></p>
</form>
<?php
}
?>
</body>
</html>

```

Als je het script runt, krijg je een formulier te zien dat er ongeveer zo uit ziet:

Probeer het script uit door niet alle gegevens in te vullen en klik vervolgens op Log in:

Vul **niets** in

Vul alleen je **naam** in

Vul alleen je e-mailadres in

Je krijgt dan een scherm te zien dat er ongeveer zo uit ziet:

Hoe werkt het?

Bij de eerste keer aanroepen van het script hebben de variabelen `$naam`, `$email` en `$opnieuw` geen waarde.

1. De eerste voorwaarde (bij `if`) is alleen maar waar als in beide invoervelden de gegevens zijn ingevuld. Let er wel op dat in beide velden ook onzin kan worden ingevuld om het script toch te laten werken.

```
if ($naam && $email) { ..... }
```

Dit resulteert in het bericht: Welkom op onze website waarbij op de plaats van de puntjes de op het formulier ingevulde naam wordt weergegeven.

De verborgen variabele `opnieuw` krijgt de waarde `yes` toegekend als de Login knop op zijn minst één keer is ingedrukt.

```
<input type="hidden" name="opnieuw" value="yes">
```

2. Als de Login knop is ingedrukt en de naam heeft geen waarde dan is de volgende voorwaarde waar:

```
if (!$naam && $opnieuw) { ..... }
```

Er verschijnt in rood onder het naamvenster een foutmelding dat de naam nog moet worden ingevuld

3. Als de Login knop is ingedrukt en het e-mailadres is nog niet ingevuld dan is de volgende voorwaarde waar:

```
if (!$email && $opnieuw) { ..... }
```

Er verschijnt in rood onder het naamvenster een foutmelding dat het e-mailadres nog moet worden ingevuld.

4. Als bij het aanroepen van het script meteen daarna de Login knop wordt ingedrukt zonder dat er ook maar iets is ingevuld worden beide foutmeldingen weergegeven.

opdracht 50

7 E-MAILEN MET PHP

PHP is in staat om e-mail te versturen met behulp van de functie `mail()`:

Format:

```
mail (aan, onderwerp, gegevens, [headers])
```

het betreft een boolean functie. Het geeft de waarde waar terug als het bericht succesvol naar de mail server is verstuurd.

attribuut	betekenis
<code>aan</code>	is het e-mailadres voor wie het bericht bedoeld is
<code>onderwerp</code>	Is het onderwerp van het e-mailbericht
<code>gegevens</code>	Is de eigenlijke inhoud van het e-mailbericht
<code>[headers]</code>	Zijn optioneel extra toevoegingen voor het e-mailbericht (bijv. de "van"-header)

We zouden nu het bestand `formuliervalidatie.php` zodanig kunnen veranderen dat de variabelen `$naam` en `$email` uit het formulier naar jouw eigen e-mailadres worden verstuurd. Daarvoor moet je in het script de volgende code invoegen:

```
mail ("mijnemail@provider.nl",  
      "gegevens van webformulier",  
      "gebruikersnaam = $naam\n e-mailadres = $email",  
      "From: webdomein.nl");
```

Zorg ervoor dat op de plaats van `mijnemail@provider.nl` je eigen e-mailadres is ingevuld. Bij `webdomein.nl` wordt het domein ingevuld waar de website is gehost.

Als het werkt zal niet alleen een begroeting op de website plaatsvinden maar ook de inloggegevens naar jou als webmaster worden verstuurd.

OPMERKING:

Deze functie werkt standaard niet als je XAMPP lokaal hebt geïnstalleerd omdat een mailserver standaard niet wordt geïnstalleerd.

opgave 51

8 SESSIES IN PHP

8.1 WAT IS EEN SESSIE?

Sessies maken het mogelijk om variabelen te gebruiken binnen een hele website. Op het moment dat een bezoeker een website bezoekt wordt een sessie id aangemaakt. Een dergelijk id is een uniek nummer. Het unieke nummer komt te vervallen op het moment dat de bezoeker uitlogt, de browser zal afsluiten of een op de server van te voren ingestelde tijd geen actie meer heeft ondernomen op de website. Je kunt met sessions werken via cookies of via de `session_id` in de URL.

8.2 WERKING VAN SESSIES

Een sessie wordt gestart door gebruik te maken van de functie `session_start()`. Dan wordt er voor iedere bezoeker een uniek `sessie_ID` aangemaakt dat gedurende het verblijf op de website wordt meegenomen. Doordat je nu zo'n uniek ID hebt is het mogelijk gegevens op te slaan in de vorm van sessievariabelen. Hiervoor wordt de PHP-variabele `$_SESSION[]` gebruikt. Je kunt sessievariabelen ook wissen door de functie `unset()`. Tot slot is het mogelijk om de sessie te vernietigen waarbij alle sessievariabelen, inclusief het `sessie_ID` worden verwijderd. Daarvoor gebruik je de functie `session_destroy()`.

8.3 HET NUT VAN SESSIES

VEILIGHEID

Pagina's kunnen alleen maar ingezien worden indien er ingelogd is en er een bekende gebruikersnaam als sessionvariabele op de server en cliënt aanwezig is.



BEHOUDEN VAN (INGEVULDE) INFORMATIE

Handig als er meerdere pagina's op een webserver moeten worden bekeken. Eerder ingevulde gegevens raken dan niet verloren maar zijn opgeslagen in sessionvariabelen. Erg handig, o.a. bij online Webshops:

- eerst verschillende pagina's bekijken,
- artikelen in winkelmandje stoppen,
- tussentijdse berekeningen uitvoeren,
- gegevens in formulieren invullen,
- eventuele wijzigingen doorvoeren,
- en tot slot afrekenen.

8.4 FUNCTIES DIE EEN ROL SPELEN BIJ SESSIES

In de tabel op de volgende pagina staan diverse functies die van belang zijn bij het werken met sessies.

functie	betekenis
<code>session_start()</code>	Opent nieuwe sessie of stelt gegevens bestaande sessie beschikbaar
<code>session_id()</code>	Geeft huidige session ID terug of je kunt zelf een ID meegeven als String voor een sessie
<code>session_register()</code>	Registreert een variabele als sessionvariabele. Let op! Geen \$-teken gebruiken. Resultaat is een array
<code>session_is_registered()</code>	Controleert of een sessionvariabele reeds bestaat. Boolean (True of False) als resultaat
<code>session_unregister()</code>	Verwijdert een variabele uit de bestaande session-array. De sessie blijft gewoon bestaan
<code>session_unset()</code>	Alle geregistreerde variabelen komen vrij, de sessie blijft bestaan, alleen de waarden van de variabelen zijn weg
<code>session_destroy()</code>	Verwijdert alle data voor een sessie. De sessie is daarmee ten einde

8.5 EEN EENVOUDIG VOORBEELD

```
<?php
    //dit is het bestand sessie_1.php
    session_start();
    $gebruikersnaam = "Piet" ;
    //sessievariabele registreren
    $_SESSION("gebruikersnaam");
?>
<html>
<head>
<title>Sessievariabele tijdelijk opslaan</title>
</head>
<body>
<h3>Uw gebruikersnaam voor deze sessie is:
<?php print ($_SESSION["gebruikersnaam"] . "<br>");
    $session_id = session_id();
    print ("Het session_id = $session_id");
?>
</h3>
</body>
</html>
```

Het resultaat ziet er zo uit:



Er kunnen foutmeldingen optreden. Deze kun je voorkomen door:

- In `php.ini` `register_globals = Off` vervangen door `On`. Let er dan wel op dat er veiligheidsproblemen kunnen optreden zoals SQL-injection!
- De browser een keer te verversen met F5.

opgaven 52 - 53

9 PROJECTOPDRACHTEN

Bij dit hoofdstuk vind je vier voorbeelden van projectopdrachten: een webbased muzieksysteem, een webwinkel, een systeem om sportdagen te organiseren en een urenregistratie systeem. Overleg met de docent welke opdracht je moet maken.

Projectopdrachten 54 -57