

PHP Aantekeningen

Aantekeningen module PHP informatica bovenbouw.

door Frans Peeters

PHP

© 2010 Frans Peeters

Deze reader mag gebruikt worden zonder verdere toestemming.

Geprint: december 2010

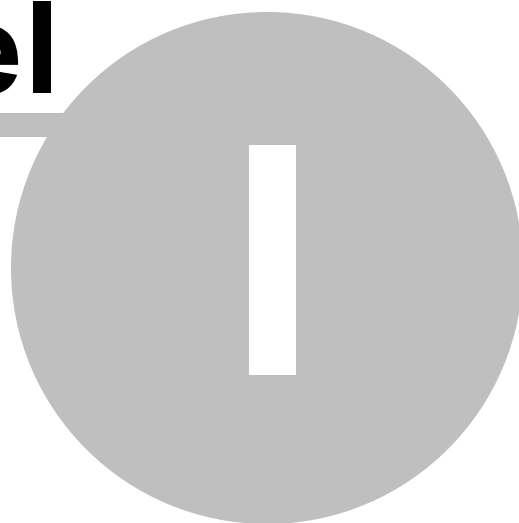
*Uitbreidingen op deze reader zijn welkom. Stuur ze naar
info@informaticavo.nl*

Inhoud

	0
Part I Introductie	6
Part II Control structures	8
1 if	8
2 else	8
3 elseif	8
4 switch.....	8
Part III Enkele of dubbele aanhalingstekens	12
Part IV Afronden	14
Part V Commentaar	16
Part VI Include	18
Part VII Radiobuttons en checkboxes	22
Part VIII Een eenvoudig formulier	24
Part IX Herhalingen	28
1 for	28
2 while.....	29
Part X Tabellen	32
Part XI Mail	36
Part XII Van twee naar één pagina	40
Part XIII Validatie	42
Part XIV Invoer verbeteren	46
Part XV Rekenmachine	48
Part XVI Escape de quotes	52
Part XVII Random	54

Part XVIII Samengestelde operatoren**56****Index****0**

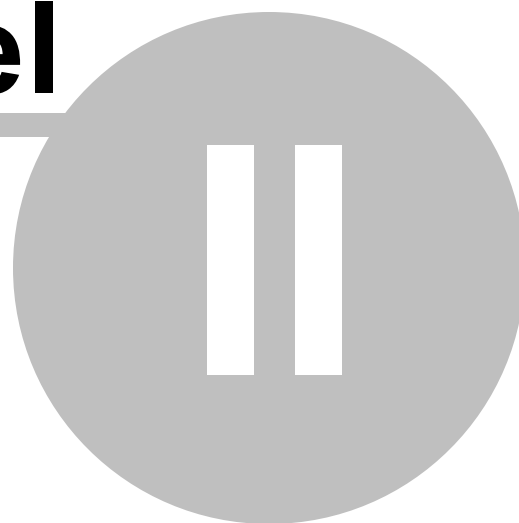
Deel



1 **Introductie**

Deze lessen PHP zijn bestemd voor een lessenserie PHP voor het vak informatica in de bovenbouw. De lijst van onderwerpen die aan de orde komen is willekeurig en kan zeker uitgebreid worden. Als er aanvullingen zijn op deze handleiding, dan kunnen die gestuurd worden naar info@informaticavo.nl. Als basis voor de lessen PHP wordt veelal het boekje van Peter Kassenaar gebruikt dat uitgegeven is bij Instruct. Zie www.instruct.nl voor verder informatie.

Deel



2 Control structures

Met control structures kun je kijken of een statement aan een bepaalde voorwaarde voldoet.

2.1 if

Met if kun je controleren of iets waar of onwaar is.

```
<?php
if ($a > $b)
echo "a is groter dan b";
?>
```

2.2 else

Je onderzoekt met if of een statement aan een voorwaarde voldoet. Als aan de voorwaarde voldaan wordt, wordt uitgevoerd wat erbij if staat. Als niet aan de voorwaarde voldaan wordt, wordt uitgevoerd wat er bij else staat.

```
<?php
if ($a > $b) {
echo "a is groter dan b";
} else {
echo "a is NIET groter dan b";
}
?>
```

2.3 elseif

elseif is een combinatie van if en else. Net als else is het een uitbreiding op if in geval de uitdrukking niet waar oplevert.

```
<?php
if ($a > $b) {
echo "a is groter b";
} elseif ($a == $b) {
echo "a is gelijk aan b";
} else {
echo "a is kleiner dan b";
}
?>
```

Je begint met een if, daarna mag er een onbeperkt aantal keer elseif komen en je besluit met else.

2.4 switch

Met if en elseif statements kun je allerlei voorwaarden testen maar soms werkt een switch sneller. Met switch wordt de waarde van een variabele getest en zodra er een overeenkomst is, wordt het programma afgebroken met break en

hoeft er niet verder gezocht te worden. Als er niets wordt gevonden, is er altijd nog een default werking (als je die tenminste hebt ingesteld). Als voorbeeld nemen we de bloembollen opdracht van 4.86 uit het boek van Instruct. Er zijn verschillende besteldrempels waar verschillende prijzen bij horen. Prijzen en drempels worden vastgelegd in variabelen.

```
<?php

$prijs_1=1.50;
$prijs_2=1.35;
$prijs_3=1.15;
$prijs_4=0.90;

$drempel_1=10;
$drempel_2=100;
$drempel_3=250;

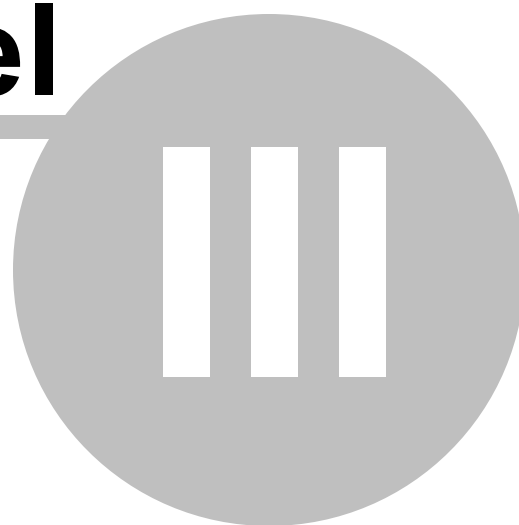
$bestelling=167;

switch($bestelling)
{
    case($bestelling>$drempel_3):
        $totaalprijs=$bestelling*$prijs_4;
        break;
    case($bestelling>$drempel_2):
        $totaalprijs=$bestelling*$prijs_3;
        break;
    case($bestelling>$drempel_1):
        $totaalprijs=$bestelling*$prijs_2;
        break;
    case($bestelling<$drempel_1):
        $totaalprijs=$bestelling*$prijs_1;
}

echo "<h1>Bestel nu uw bloembollen!</h1>";
echo "<img src=bloembollen.jpg align=right>";
echo "<table border=1>";
echo "<tr><th>Aantal bloembollen</th><th>Prijzen per stuk</th></tr>";
echo "<tr><td>minder dan " . $drempel_1 . "</td><td>" . $prijs_1 . " euro</td></tr>";
echo "<tr><td>" . $drempel_1 . " tot " . $drempel_2 . "</td><td>" . $prijs_2 . " euro";
echo "<tr><td>" . ($drempel_2+1) . " tot " . $drempel_3 . "</td><td>" . $prijs_3 . " euro";
echo "<tr><td>meer dan " . $drempel_3 . "</td><td>" . $prijs_4 . " euro</td></tr>";
echo "</table>";
echo "<h2>U bestelde " . $bestelling . " bloembollen.</h2>";
echo "<h2>De totaalprijs is " . $totaalprijs . " euro.</h2>"

?>
```


Deel



3 Enkele of dubbele aanhalingstekens

Het maakt verschil of je een echo maakt met enkele aanhalingstekens of met dubbele aanhalingstekens. Met dubbele aanhalingstekens wordt de php omgezet naar html. Bij enkele aanhalingstekens niet. Enkele aanhalingstekens zullen dus net iets sneller werken dan dubbele aanhalingstekens. Met dubbele aanhalingstekens kun je echter een hele regel tussen de aanhalingstekens zetten zonder dat je daar verder werk mee hebt. We gaan er even vanuit dat we de variabelen \$dag en \$maand hebben gedeclareerd, bijvoorbeeld met:

```
$vandaag=getdate();  
$dag=$vandaag['mday'];  
$maand=$vandaag['month'];
```

Nu kun je op verschillende manieren een echo maken.

```
echo "Het is vandaag $dag $maand.";
```

Dit toont de datum van vandaag op het scherm. Tussen de aanhalingstekens wordt gekeken wat er php is en dat wordt omgezet naar html. Dit is geschreven op de datum van vandaag en daarom wordt die datum (in het Engels) getoond. De datum die wordt getoond is de systeemdatum van de server waarop de php omgezet wordt in html. Nu gaan we dezelfde regel met enkele aanhalingstekens plaatsen.

```
echo 'Het is vandaag $dag $maand.';
```

Dit toont op het scherm: Het is vandaag \$dag \$maand. Met de enkele aanhalingstekens wordt er geen php omgezet in html maar wordt de php code getoond op het scherm. We kunnen ook een combinatie maken van html en php.

```
echo "Het is vandaag " . $dag . $maand;
```

Dit toont op het scherm wederom de juiste datum in het Engels op het scherm. Je zet de tekst tussen dubbele aanhalingstekens en verbindt de variabelen met de tekst door middel van een puntje. Er moet echter nog een spatie tussen de dag en de maand en er moet nog een punt achter de zin.

```
echo "Het is vandaag " . $dag . " " . $maand . ".";
```

Maar het kan dus ook met enkele aanhalingstekens. Binnen de enkele aanhalingstekens wordt geen php gebruikt en die tekst kan zo getoond worden. De php wordt verbonden met de html door het puntje.

```
echo 'Het is vandaag ' . $dag . ' ' . $maand . '.';
```

Deel

IV

4 Afronden

Bij bedragen wil je graag dat er maar twee cijfers achter de komma komen als de uitkomst een gebroken getal is met meerdere cijfers achter de komma. Dat kun je bereiken door de uitkomst af te ronden met `round()`. Met `round(uitkomst,2)` krijg je dan keurig een getal met twee achter de komma. De pagina roept zichzelf aan met de action op de eerste regel. Met een `if` stel je in dat er "Voer een bedrag in." komt te staan als het veld bedrag leeg is. Bij radio buttons is de name hetzelfde maar de value verschillend.

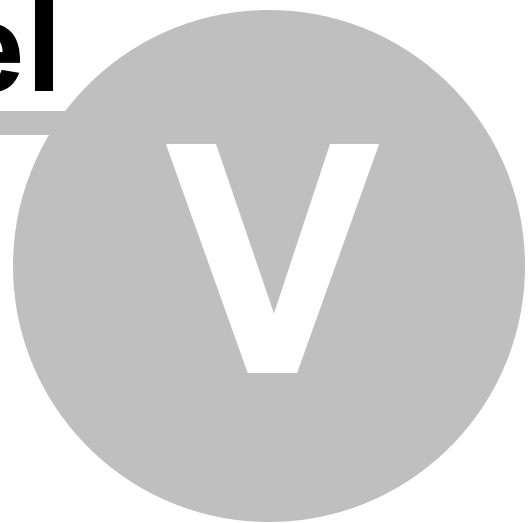
```
<form action="<?php echo $_SERVER["PHP_SELF"]?>" method="post">
Bedrag exclusief BTW: <input type="text" name="bedrag"><br>
<input type="radio" name="btw" value="0.06">laag tarief - 6%<br>
<input type="radio" name="btw" value="0.19" checked>hoog tarief - 19%<br>
<input type="submit" value="Bereken BTW">
</form>
<hr>

<?
$exbtw=$_POST['bedrag'];
$tarief=$_POST['btw'];
$btw=round(($tarief*$exbtw),2);
$inclusiefbtw=$exbtw+$btw;

if(empty($_POST['bedrag']))
{
    echo "Voer een bedrag in.";
}

else
{
    echo "Bedrag inclusief BTW: <b>$inclusiefbtw euro</b><br>";
    echo "De BTW bedraagt: <b>$btw euro</b>";
}
?>
```

Deel



5 Commentaar

Het is goed je eigen code van commentaar te voorzien zodat je later terug kunt zoeken hoe het programma in elkaar steekt of ter verduidelijking aan anderen die de code moeten lezen. Commentaar is ook een prima manier om programma's te debuggen als ze nog niet goed werken. Je voorziet een of meerdere regels van commentaar zodat ze niet uitgevoerd worden en kijkt of het wel werkt zonder die regels. Zodoende kun je vaak ontdekken in welke regel de fout zit.

Regels kun je van commentaar voorzien met twee slashes, dus `//`. Dat kan vóór de code of na de code. Als je meerdere regels van commentaar wil voorzien, dan kun je beginnen met `/*` en eindigen met `*/` voor het commentaar.

```
//Voorbeeld met while
$i=1;                // We stellen een getal vast op 1.
/* Vanaf hier doet de code niet meer mee.
while ($i<=7)
{
    echo "<font size=\"". $i . "\">Dit is lettergrootte " . $i . ".</font><br>\n";
    $i++;
}
Wat hierna komt doet weer mee in de code.*/
echo $i;
```


Deel

VI

6 Include

Include kan heel goed gebruikt worden om een configuratiebestand van je database in te voegen of om bijvoorbeeld een menuutje of een kop van een pagina over heel je website in te zetten. Vaak worden ook alle functies in één bestand gezet zodat daarna die functies in alle andere pagina's aangeroepen kunnen worden dankzij de include. Stel dat je bijvoorbeeld al je pagina's gegarandeerd w3c gevalideerd wil hebben. Dan ben je al een heel eind op weg als je zorgt dat de kop van de pagina inclusief menu met één include op al je pagina's hetzelfde is. Onderstaande code zet je in een bestand met een willekeurige naam. Het grote voordeel is dat je maar op één plaats hoeft te wijzigen als bijvoorbeeld het menu aangepast moet worden. Vaak wordt de extensie .inc gebruikt voor dergelijke bestanden. In ons geval slaan we onderstaande code op in een tekstbestand met als naam menu.inc.

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="nl" lang="nl">
  <head>
    <title>
      De titel van jouw website
    </title>
    <meta http-equiv="content-type" content="text/html;
      charset=utf-8" />
    <meta http-equiv="Content-Style-Type" content="text/css" />
    <link href="opmaak.css" rel="stylesheet" type="text/css" media="all" />
  </head>
  <body>
    <p align="center">
      [<a href="index.php">home</a>]
      [<a href="schaaklessen.php">schaaklessen</a>]
      [<a href="competitie.php">competitie</a>]
      [<a href="jtc.php">JTC</a>]
      [<a href="toernooien.php">toernooien</a>]
      [<a href="ratings.php">ratings</a>]
      [<a href="trainingen.php">trainingen</a></p>
```

Al onze php pagina's beginnen nu op de eerste regel met `<?php include("menu.inc") ?>`.

De kop van die pagina's is dan in ieder geval xhtml en op alle pagina's verschijnt het menu.

Of stel dat je meerdere functies hebt gemaakt die je graag op meerdere pagina's op je website wil gebruiken. Je zet dan die functies in een verzamelbestand (bijvoorbeeld functions.php) en met include kun je dan altijd beschikken over die functies. Als voorbeeld zetten we de functie `is_positief()` in het bestand functions.php.

```
function is_positief()  
{  
    if($getal>0)  
    {  
        return true;  
    }  
    else  
    {  
        return false;  
    }  
}
```

Op de pagina's waar we de functies willen gebruiken plaatsen we de include en daaronder kun je de functies aanroepen die gedefinieerd staan in functions.php.

```
include "functions.php";
```

Vervolgens kun je op die pagina's de functie gewoon aanroepen.

```
if(is_positief())  
{  
    echo "Het getal is positief."  
}  
else  
{  
    echo "Het getal is negatief."  
}
```


Deel

VII

7 Radiobuttons en checkboxes

Soms heb je bij formulieren de keuze een selectie te maken uit een aantal. Als je maar één keuze mag maken, maak je daarvoor een radiobutton (rond). Als er meerdere keuzes gemaakt mogen worden, dan doe je dat met checkboxes (vierkant).

We nemen een reactieformulier met een radiobutton (man/vrouw) en vier checkboxes (favoriete kleuren). Bij radiobuttons hebben alle buttons dezelfde name (naam) maar een verschillende value (waarde). Bij checkboxes hebben alle items een verschillende naam en waarde.

```
<form action="contact.php" method="post">

<p>Naam:<br>
<input type="text" name="naam"></p>

<p>Geslacht:<br>
<input type="radio" name="geslacht" value="man">man<br>
<input type="radio" name="geslacht" value="vrouw">vrouw</p>

<p>Favoriete kleur(en):<br>
<input type="checkbox" name="geel" value="geel">geel<br>
<input type="checkbox" name="groen" value="groen">groen<br>
<input type="checkbox" name="rood" value="rood">rood<br>
<input type="checkbox" name="paars" value="paars">paars</p>

<p>Geboren:<br>
<select name="geboren">
    <option value="1990" SELECTED>1990</option>
    <option value="1991">1991</option>
    <option value="1992">1992</option>
    <option value="1993">1993</option>
    <option value="1994">1994</option>
    <option value="1995">1995</option>
    <option value="onbekend">onbekend</option>
</select>

<p><input type="submit" value="Verstuur"></p>
</form>
```

Nu gaan we in contact.php zeggen dat alle waarden die ingevuld zijn getoond moeten worden:

```
<?php
while(list($variabele,$waarde)=each($_POST))
{
    echo "$variabele: $waarde<br>";
}
?>
```

Deel



8 Een eenvoudig formulier

Vaak worden er op internet formulieren gebruikt om ergens voor in te schrijven of om ergens op te reageren of zo. Meestal wordt er gewerkt met twee formulieren, namelijk een html bestand en een php bestand. In het html bestand wordt dan het php document opgevraagd. Met de meeste editors is het vrij gemakkelijk een formulier te maken, maar het kan ook gewoon met de kladblok.

Als voorbeeld zullen we een reactie formuliertje maken. Er moet een naam ingevuld worden en er kan een opmerking geplaatst worden.

Sla alvast een document op met als naam reactie.html. Een formulier maak je aan met onderstaande regel. De sluitcode voor het formulier noteren we ook alvast op de regel daaronder.

```
<form action="reactie.php" method="post">
</form>
```

Bij action vul je de naam in van het php document dat aangeroepen moet worden en je kiest voor post bij method.

Tussen de open code en de sluitcode van form moeten nu de velden en de knoppen komen. Maak nu een tekstveld voor de naam, een text area veld voor de opmerking en een submit button om het formulier te versturen. Je vult alleen de "name" in van een veld. Geef bijvoorbeeld de velden de namen "naam" en "opmerking". Bij het veld opmerking kun je een breedte (cols) en hoogte opgeven (rows). Bij de submit knop moet je een naam opgeven, bijvoorbeeld "go" en je kunt ook aangeven wat er op de knop moet komen te staan bij value. Boven de velden zet je de namen van de velden in de gewone tekst met een <p> ervoor zodat ze allemaal op een nieuwe regel beginnen. Je krijgt dan ongeveer dit:

```
<form action="reactie.php" method="post">
<p>Naam:
<p><input type="text" name="naam">
<p>Opmerking:
<p><textarea cols="40" rows="5" name="opmerking"></textarea>
<p><input type="submit" name="go" value="Verstuur">
</form>
```

Nu gaan we reactie.php maken . Open een leeg document en sla dat alvast op als reactie.php. We lezen de posts uit en tonen die met een echo op het scherm. Verderop wordt er ook een mailtje gestuurd naar een mailadres dat je hebt ingevuld bij mail. Met de stuurcode \t plaats je een tab in de mail en met \n ga je naar de volgende regel. Je krijgt dan de volgende code in reactie.php:

```
<?php
```



```
$naam= $_POST['naam'];  
$opmerking=$_POST['opmerking'];  
echo "Naam: ". $naam;  
echo "<p>Opmerking:<br>" . $opmerking;  
mail("pts@ostrealyceum.nl","Reactie contactfomulier","Naam:\t$naam\n\n Opmerking:\n$opmerking\n?>
```


Deel



IX

9 Herhalingen

Een van de sterkste kanten van een programmeertaal is dat herhalingen op eenvoudige wijze gerealiseerd kunnen worden. Dat kan bijvoorbeeld met `for` en met `while`.

9.1 `for`

Met `for` kun je php dingen laten herhalen. Zo kun je met enkele regels net zolang regels vullen tot er aan een bepaalde voorwaarde is voldaan. Stel bijvoorbeeld dat je de getallen 1 tot en met 10 wil laten zien.

```
<?
for($i=1; $i<=10; $i++)
{
    echo $i;
}
?>
```

We noemen de variabele `$i` maar dat kan uiteraard een willekeurige naam zijn. Het startgetal is 1 en het eindgetal is kleiner of gelijk aan 10. Elk getal wordt met een `echo` op het beeldscherm getoond en het getal wordt steeds met 1 verhoogd door `$i++` net zolang tot `$i` kleiner of gelijk is aan 10. De getallen worden nu allemaal achter elkaar getoond, zelfs zonder spaties. Het resultaat is dus 12345678910. Stel dat je de getallen 1 tot en met 10 onder elkaar wil hebben. Dan voeg je dus een `break` toe achter `$i`.

```
for($i=1; $i<=10; $i++)
{
    echo "$i <br>";
}
```

Je combineert nu php met html en dan moet je er op letten dat de html tussen aanhalingstekens moet, de php mag tussen aanhalingstekens maar dat hoeft niet. Als je echter `echo $i
` zonder aanhalingstekens gebruikt, krijg je een parse error. Een goede manier om html en php te scheiden is de php zonder aanhalingstekens te gebruiken en de html tussen aanhalingstekens te koppelen aan de php met een puntje. Spaties voor en na de punt zijn niet noodzakelijk maar verhogen de overzichtelijkheid.

```
for($i=1; $i<=10; $i++)
{
    echo $i . "<br>";
}
```

Stel dat ik nu de tafel van 7 wil laten zien op mijn scherm.

```
for($i=1; $i<=10; $i++)
{
```

```
    echo $i . " x 7 = " . $i*7 . "<br>";  
}
```

In de broncode van het html document staat alle code achter elkaar. Als we de broncode ook nog netjes op willen maken, kunnen we een stuurcode meegeven op het eind van de regel dat er in de code ook naar een volgende regel moet worden gesprongen. Dat kan met een backslash gevolgd door een n.

```
for($i=1; $i<=10; $i++)  
{  
    echo $i . " x 7 = " . $i*7 . "<br>\n";  
}
```

9.2 while

Met while zal het programma iets uitvoeren zolang een statement waar is.

```
$i = 1;  
while ($i <= 10)  
{  
    echo $i++;  
}
```

We geven de variabele de startwaarde 1 met `$i=1`. Zolang de variabele kleiner of gelijk is aan 10 wordt er 1 opgeteld bij `$i` met `$i++`.

Deel



10 Tabellen

Stel dat we de herhaling van de tafel met een variabel getal willen doen, dan maken we de noemer van de tafel variabel. Bij de tafel van 7 is dan \$noemer gelijk aan 7 en \$steller gelijk aan achtereenvolgens 1 tot en met 10. De uitkomst per regel is dan \$steller*\$noemer. Als we de tafel van een getal willen maken, dan krijgen we onderstaande code. Met <p> en </p> worden er netjes paragrafen geopend en gesloten.

```
$noemer=7;
echo '<p>De tafel van ' . $noemer . ' :</p><p>';

for ($steller=1; $steller<=10; $steller++)
{
    echo $steller . ' x ' . $noemer . ' = ' . $steller*$noemer . '<br>';
}
echo '</p>';
```

Het zou nog mooier zijn als we de tafel in een tabel zouden weergeven. Je moet dan wel weten hoe je een tabel moet maken in html. Met <table> en </table> open en sluit je een tabel. Met <tr> en </tr> maak je rijen. De bovenste rij kan gevuld zijn met <th> in plaats van <td> maar dat hoeft niet. De cellen in de rijen maak je met <td> en </td>. Hieronder de code voor een tabel met een koprij en twee rijen daaronder met elk twee cellen.

```
<table>
    <tr>
<th></th><th></th>
    </tr>
    <tr>
        <td></td><td></td>
    </tr>
    <tr>
        <td></td><td></td>
    </tr>
</table>
```

Als we nu de tafel van 7 netjes op willen maken met een tabel, dan gaat dat zoals hieronder. Met colspan pak je in de kop twee cellen samen. Met cellpadding stel je de afstand in van de tekst tot de rand van de cel. Onderstaande code is de uitwerking van 5.102 uit het boek van Instruct, behalve dat hier de tafel van 7 wordt gebruikt in plaats van die van 10.

```
<?php
$noemer=7;
echo '<table border=1 cellpadding=5><tr><th colspan=2>De tafel van ' . $noemer . '</th>';
for ($steller=1;$steller<=10;$steller++)
{
    echo '<tr><td>' . $steller . ' maal ' . $noemer . ' = </td><td>' . $steller*$noemer . '</td>';
}
```

```
echo '</table>';  
?>
```

Als je de border weg wil laten, stel je de border in op 0 in plaats van 1.

Deel

XI

11 Mail

Voorbeeld e-mail 1:

```
<?php
// In de variabele message komt de inhoud van het bericht.
// Let op de \n om naar de volgende regel te gaan.
// Die variabele kan bijvoorbeeld opgehaald zijn via een formulier.
$message = "Line 1\nLine 2\nLine 3";

// Als de regels langer worden dan 70 karakters, gebruik dan wordwrap.
$message = wordwrap($message, 70);

// Met onderstaande regel wordt de e-mail verstuurd. Vul het juiste e-mailadres in.
mail('informatica@kpnmail.nl', 'Mijn Onderwerp', $message);
?>
```

Voorbeeld e-mail 2:

Het is ook mogelijk variabelen te gebruiken voor de geadresseerde, het onderwerp, het bericht en de afzender. In dit voorbeeld wordt ook een antwoordadres opgegeven en de PHP versie getoond. De \r\n op het eind van de headers zijn nodig om naar de volgende regel te gaan. Met \r\n pas je een CRLF toe, een carriage return line feed.

```
<?php
$to      = 'informatica@kpnmail.nl';
$subject = 'Het Onderwerp';
$message = 'Hallo';
$headers = 'From: info@informaticavo.nl' . "\r\n" .
           'Reply-To: frans@depion.nl' . "\r\n" .
           'X-Mailer: PHP/' . phpversion();

mail($to, $subject, $message, $headers);
?>
```

Voorbeeld e-mail 3:

In voorbeeld drie zijn er verschillende ontvangers, wordt het bericht in HTML getoond, wordt er gebruik gemaakt van CC en BCC en wordt er een naam getoond bij de afzender in plaats van een e-mailadres.

```
<?php
// meerdere ontvangers
$to = 'frans@depion.nl' . ','; // let op de komma om de e-mailadressen te scheiden
$to .= 'bart@depion.nl';      // deze e-mailadressen kun je ook uit een post haalen

// onderwerp
```

```
$subject = 'Inschrijving Tweetallen Toernooi';

// message
$message = "
<html>
<head>
  <title>Inschrijving Tweetallen Toernooi</title>
</head>
<body>
  <p>Je bent ingeschreven voor het Tweetallen Toernooi met de volgende gegevens:</p>
  <table>
    <tr>
      <th>Bord</th><th>Speler</th><th>Club</th><th>E-mail</th>
    </tr>
    <tr>
      <td>1</td><td>$speler1</td><td>$club1</td><td>$email1</td>
    </tr>
    <tr>
      <td>2</td><td>$speler2</td><td>$club2</td><td>$email2</td>
    </tr>
  </table>
</body>
</html>
";

// Om HTML mail te versturen moet de Content-type header bepaald worden.
// Met de punt na $headers wordt $headers steeds uitgebreid.
$headers = 'MIME-Version: 1.0' . "\r\n";
$headers .= 'Content-type: text/html; charset=iso-8859-1' . "\r\n";

// Overige headers die je kunt gebruiken.
$headers .= 'To: Frans <frans@depion.nl>, Bart <bart@depion.nl>' . "\r\n";
$headers .= 'From: De Pion <tweetallen@depion.nl>' . "\r\n";
$headers .= 'Cc: voorzitter@depion.nl' . "\r\n";
$headers .= 'Bcc: secretaris@depion.nl' . "\r\n";

// Mail versturen
mail($to, $subject, $message, $headers);
?>
```


Deel

XII

12 Van twee naar één pagina

Je kunt er ook voor kiezen slechts één pagina in plaats van twee te maken. De pagina roept steeds zichzelf terug aan. Dat doe je met behulp van `$_SERVER["PHP_SELF"]`.

```
<?php
if(!empty($_POST))
{
    while(list($variabele, $waarde)=each($_POST))
    {
        echo "$variabele: $waarde<br>";
    }
    echo "<p><a href='\". $_SERVER[\"PHP_SELF\"]\" . \">Opnieuw</a></p>\";
}
else
{
    ?>

<form action='\"<?php echo $_SERVER[\"PHP_SELF\"]?>\" method='\"post\">

<p>Naam:<br>
<input type='\"text\" name='\"naam\"></p>

<p>Geslacht:<br>
<input type='\"radio\" name='\"geslacht\" value='\"man\">man<br>
<input type='\"radio\" name='\"geslacht\" value='\"vrouw\">vrouw</p>

<p>Favoriete kleur(en):<br>
<input type='\"checkbox\" name='\"geel\" value='\"geel\">geel<br>
<input type='\"checkbox\" name='\"groen\" value='\"groen\">groen<br>
<input type='\"checkbox\" name='\"rood\" value='\"rood\">rood<br>
<input type='\"checkbox\" name='\"paars\" value='\"paars\">paars</p>

<p>Geboren:<br>
<select name='\"geboren\">
    <option value='\"1990\" SELECTED>1990</option>
    <option value='\"1991\">1991</option>
    <option value='\"1992\">1992</option>
    <option value='\"1993\">1993</option>
    <option value='\"1994\">1994</option>
    <option value='\"1995\">1995</option>
    <option value='\"onbekend\">onbekend</option>
</select>

<p><input type='\"submit\" value='\"Verstuur\"></p>
</form>

<?php
}
?>
```


Deel

XIII

13 Validatie

We kunnen velden controleren op invoer van de gebruiker. Voorbeeldjes hiervan zijn:

- Is een bepaald veld wel ingevuld?
- Is een veld waar enkel een getal mag worden ingevuld ook echt gevuld met een getal?

We maken een validatie.html op de volgende manier:

```
<form action="validatie.php" method="post">
<p>Naam:</p>
<p><input type="text" name="naam"></p>
<p>Leeftijd:</p>
<p><input type="text" name="leeftijd"></p>
<p>Lengte:</p>
<p><input type="text" name="lengte"></p>
<p><input type="submit" name="go" value="verstuur"></p>
</form>
```

En de validatie.php gaat dan zoals hieronder. Met het uitroepteken maak je een uitdrukking ontkenkend. Dus elseif (!is_numeric("\$leeftijd")) betekent: als de variabele \$leeftijd niet numeriek is, dan ...

```
$naam=$_POST['naam'];
$leeftijd=$_POST['leeftijd'];
$lengte=$_POST['lengte'];

if($_POST)
{
    if (empty($naam))
    { echo "<p>Je hebt je naam niet ingevuld!</p><p>
Klik <a href=\"javascript:history.back(1)\">hier</a> om terug te gaan.</p>";
    }
    elseif (empty($leeftijd))
    { echo "<p>Je hebt je leeftijd niet ingevuld!</p><p>
Klik <a href=\"javascript:history.back(1)\">hier</a> om terug te gaan.</p>";
    }
    elseif (empty($lengte))
    { echo "<p>Je hebt je lengte niet ingevuld!</p><p>
Klik <a href=\"javascript:history.back(1)\">hier</a> om terug te gaan.</p>";
    }
    elseif (!is_numeric("$leeftijd"))
    { echo "<p>Je hebt geen juiste leeftijd ingevuld!</p><p>
Klik <a href=\"javascript:history.back(1)\">hier</a> om terug te gaan.</p>";
    }
    elseif (!is_numeric("$lengte"))
    { echo "<p>Je hebt geen juiste lengte ingevuld!</p><p>
Klik <a href=\"javascript:history.back(1)\">hier</a> om terug te gaan.</p>";
    }
}
```

```
else
{
echo "<p>U hebt de volgende gegevens ingevuld:</p>";
echo "<p>Naam: " . $naam . "</p>";
echo "<p>Leeftijd: " . $leeftijd . "</p>";
echo "<p>Lengte: " . $lengte . "</p>";
}
}
```


Deel



14 Invoer verbeteren

We kunnen ook de invoer van de gebruiker verbeteren. Voorbeelden hiervan zijn:

- Hoofdlettergebruik waar dat nodig is. Namen schrijf je met een hoofdletter.
- Allemaal kleine letters waar dat hoort. Tussenvoegsels staan in kleine letters.
- Alles in hoofdletters waar dat zo hoort, bijvoorbeeld bij de letters van de postcode.
- Ongewenste html code uit de invoer halen.

Maak een pagina `verbeter.html` met onderstaande code:

```
<form action="verbeter.php" method="post">
<table>
<tr><td>Voornaam:</td><td><input type="text" name="voornaam"></td></tr>
<tr><td>Tussenvoegsel:</td><td><input type="text" name="tussen"></td></tr>
<tr><td>Achternaam:</td><td><input type="text" name="achternaam"></td></tr>
<tr><td>Postcode:</td><td><input type="text" name="postcodecijfers" size="4">
<input type="text" name="postcodeletters" size="2"></td></tr>
<tr><td>Huisnummer:</td><td><input type="text" name="huisnummer"></td></tr>
</table>
<p>Opmerking:</p>
<p><textarea cols="40" rows="5" name="opmerking"></textarea></p>
<p><input type="submit" name="go" value="verstuur">
<input type="Reset" value="herstel"></p>
</form>
```

Maak de pagina `verbeter.php` waarmee de invoer wordt verbeterd.

```
<?php
$voornaam=ucfirst($_POST['voornaam']);
$tussen=strtolower($_POST['tussen']);
$achternaam=ucfirst($_POST['achternaam']);
$postcodecijfers=$_POST['postcodecijfers'];
$postcodeletters=strtoupper($_POST['postcodeletters']);
$opmerking=strip_tags($_POST['opmerking']);
echo "<p><b>Je hebt de volgende gegevens opgegeven:</b></p>";
echo "<p>" . $voornaam . " " . $tussen . " " . $achternaam;
echo "<p>" . $postcodecijfers . " " . $postcodeletters;
echo "<p>Opmerking:";
echo "<p>" . $opmerking;
?>
```

Deel

XV

15 Rekenmachine

Eerst halen we de drie variabelen op en dan gaan we kijken of het formulier ingevuld is of niet. Die drie variabelen zijn dus het eerste getal, het tweede getal en de operator (optellen, aftrekken, delen, vermenigvuldigen).

Met `if(!empty($_POST))` bedoel je: "Als het formulier niet leeg is, dan toon je de berekening."

Verderop staat `else`. En daar staat wat er moet gebeuren als het formulier wel leeg is. Dan krijg je namelijk een leeg formulier voor je neus. Kijk opnieuw hoe we een formulier zichzelf laten aanroepen met de action in de regel waar form begint.

Onze rekenmachine is nog niet foolproof. Zo kan er bijvoorbeeld nog tekst ingevuld worden en kun je nog proberen te delen door 0 (en dat mag niet).

We gebruiken `<h1>` om een groot lettertype te krijgen. Kijk waar die wordt geopend en waar die wordt gesloten met `</h1>`.

Dit is de opgave van bladzijde 7.152 van het boek van Instruct.

```
<?php
$getal1=$_POST['getal1'];
$getal2=$_POST['getal2'];
$operator=$_POST['operator'];

if(!empty($_POST))
{
    $som=$getal1+$getal2;
    $verschil=$getal1-$getal2;
    $deling=$getal1/$getal2;
    $vermenigvuldiging=$getal1*$getal2;

    echo "<h1>";
    if($operator=="plus")
        echo "$getal1 + $getal2 = $som";
    elseif($operator=="min")
        echo "$getal1 - $getal2 = $verschil";
    elseif($operator=="gedeeld")
        echo "$getal1 / $getal2 = $deling";
    elseif($operator=="keer")
        echo "$getal1 * $getal2 = $vermenigvuldiging";

    echo "</h1><p><a href=\". $_SERVER['PHP_SELF']\" . \">Opnieuw</a></p>";
}

else
{
    ?>
    <form action="<?php echo $_SERVER['PHP_SELF']?>" method="post">
    getal 1: <input type="text" name="getal1"><br>
    getal 2: <input type="text" name="getal2"><br>
    <input type="radio" name="operator" value="plus" checked checked>Optellen(+)
    <input type="radio" name="operator" value="min">Aftrekken(-)
```



```
<input type="radio" name="operator" value="gedeeld">Delen (/)
<input type="radio" name="operator" value="keer">Vermenigvuldigen (*)
<hr>
<input type="submit" name="rekenen" value="Bereken">
<input type="reset" name="reset" value="Leegmaken">
</form>
<?php
}
```


Deel



16 Escape de quotes

Je kunt in html een pagina een gele achtergrondkleur geven met `<body bgcolor=yellow>`. In betere html zou dat zijn: `<body bgcolor="yellow">`. Hier staat "yellow tussen aanhalingstekens. In php gebruik je echter de aanhalingstekens om strings af te bakenen. Na het tweede aanhalingsteken wordt de string afgesloten en dat wil je eigenlijk niet, want je wil echt een aanhalingsteken plaatsen. Dan kun je een backslash gebruiken om het teken te escaperen.

Dus met `echo "<body bgcolor=\"\$kleur\">";` in php toon ik `<body bgcolor="kleur">` in html. Dat escaperen geldt voor nog meer tekens, zoals het dollarteken en de ampersand (&). De uitleg hierover staat op pagina 3.58 van het boek van Instruct.

```
<?php
$kleur=$_POST['kleur'];
echo "<body bgcolor=\"\$kleur\">";
?>
```

```
<form action="<?php echo $_SERVER["PHP_SELF"]?>" method="post">
<input type="radio" name="kleur" value="red">rood
<input type="radio" name="kleur" value="green">groen
<input type="radio" name="kleur" value="blue">blauw
<input type="radio" name="kleur" value="silver">zilver
<input type="radio" name="kleur" value="white">wit
<input type="submit" name="instellen" value="Kleur instellen">
</form>
```

Deel

XVI

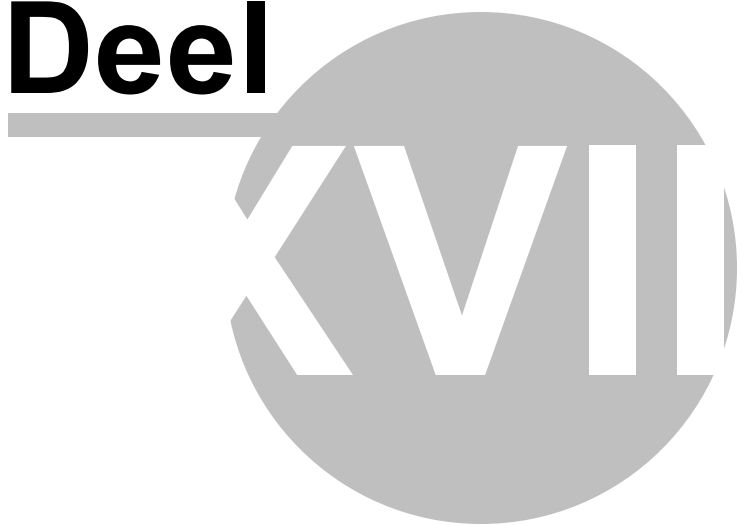
17 Random

Met `random()` kun je een willekeurig getal laten genereren. Met bijvoorbeeld `rand(5,15)` krijg je een willekeurig getal tussen 5 en 15, waarbij 5 en 15 inclusief zijn, dus die kun je ook krijgen. We kunnen het laagste en het hoogste getal met variabelen maken, zodat de gebruiker beide of één van de twee zelf op kan geven.

Stel bijvoorbeeld dat je een loterij hebt en dat de winnaar door de computer wordt bepaald nadat het aantal deelnemers door de gebruiker is ingevoerd met een formulier.

```
<?
$aantal=$_POST[aantal];
$winnaar=rand(1,$aantal);
echo $winnaar;
?>
```

Deel



18 Samengestelde operatoren

Met `.=` kun je een bestaande string uitbreiden. Een puntje gevolgd door een is gelijk teken dus.

```
echo $tekst = "Goede ";  
echo "<br>";  
echo $tekst .= "morgen ";  
echo "<br>";  
echo $tekst .= "allemaal";
```

Achtereenvolgens komt er te staan:

```
Goede  
Goede morgen  
Goede morgen allemaal
```

Hetzelfde kun je doen met rekenkundige operatoren zoals in onderstaand voorbeeld.

```
echo $getal =10;  
echo "<br>";  
echo $getal += 5;
```

Hier komt te staan:

```
10  
15
```

De opdracht op 5.102 kan op deze manier gemaakt worden met `for` of met `while`.

Met `while` gaat dat als volgt:

```
$getal=12;  
$totaal=0;  
echo '<p>Het getal is ' . $getal . "</p>";  
echo '<p>Het totaal van 1 tot en met ' . $getal . ' is<br>';  
echo '0 ';  
$i=1;  
while ($i<= $getal)  
{  
    echo ' + ' . $i;  
    $totaal += $i;           // $totaal wordt $totaal + $i  
    $i++;                  // Dat kan ook met $totaal=$totaal+$i  
}  
echo ' = ' . $totaal;
```

Met `for` kan het ook:

```
$getal=12;
```



```
$totaal=0;
echo '<p>Het getal is ' . $getal . "</p>";
echo '<p>Het totaal van 1 tot en met ' . $getal . ' is<br>';
echo '0 ';
for ($i=1;$i<=12;$i++)
{
    echo ' + ' . $i;
    $totaal += $i;
}
echo ' = ' . $totaal;
```