



6 Bedrijfsondersteunende
Informaticawetenschappen

Basiscursus 2024-2025

PHP MySQL

Danny Devriendt

Inhoudsopgave

1	Inleiding	5
2	Kennismaking met PHP.....	6
2.1	Dynamische webpagina's.....	6
2.2	PHP.....	7
2.3	PHP-controlestructuren	11
2.4	Functies.....	15
2.5	Datum en tijd	17
2.6	Arrays	18
2.7	Tekens aan een string toevoegen met str_pad().....	20
3	Website opbouwen met include().....	22
4	Formulieren	24
4.1	Gegevens uit een formulier verwerken	24
4.2	Javascript voor het verwerken van je formulier	26
4.3	Ingevulde en gekozen gegevens in je formulier behouden	26
4.4	Invoer van diakritische en vreemde tekens	29
4.5	Oefeningen.....	29
5	MySQL databank	35
5.1	PHPMyAdmin	35
5.2	Database en tabellen aanmaken	35
5.3	Weergave van accentletters en vreemde tekens - de collatie	36
5.4	Relaties tussen tabellen	36
5.5	Connectie met de database	37
5.6	Gegevenstypes in MySQL.....	38
5.7	Enkele basis SQL-instructies.....	38
5.8	Gegevens raadplegen en weergeven.....	39
5.9	SQL joins: gegevens uit verschillende tabellen weergeven	40
5.10	Gegevens invoeren	42
5.11	Gegevens uitwisselen tussen MySQL databanken.....	43
5.12	Gegevens importeren uit Excel.....	44
5.13	Gegevens via PHP exporteren in Excel.....	45
5.14	Een mailbericht versturen vanuit PHP	47
5.15	Oefeningen.....	47
6	Session-variabelen en cookies	52
6.1	De sessie starten	52
6.2	Session-variabelen instellen	52
6.3	Een sessie stoppen.....	53
6.4	Waar worden sessievariabelen bewaard?.....	53
6.5	Praktisch voorbeeld – brood.....	53
6.6	Cookies.....	55

7	\$_GET en querystring	57
7.1	Praktische toepassing – brood.....	58
8	Fietsregistratie Brugge.....	60
8.1	Gegevens verwijderen	64
8.2	Gegevens wijzigen.....	66
8.3	Gegevens opzoeken.....	70
8.4	Detailinformatie opvragen.....	72
9	Missault roomijs.....	76
10	Bronnen	81

1 Inleiding

Je bent reeds vertrouwd met de opbouw van webpagina's via HTML en CSS. Meer en meer websites zijn echter dynamisch: ze bevatten interactieve elementen en/of de inhoud van de webpagina's wordt via een database gegenereerd en in de browser weergegeven.

*In deze cursus leer je dynamische websites opbouwen. De verleiding is groot om hierbij gebruik te maken van beschikbare kant-en-klaar-oplossingen waar je mits wat slepen en aanpassen tot een resultaat komt, maar dat doen we in deze cursus bewust niet. We starten vanaf nul en we kozen voor de populaire combinatie **PHP** (www.php.net) en **MySQL** (www.mysql.com), veel gebruikte open source software die je gratis kunt downloaden.*

*Voor het maken van onze PHP-pagina's maken we opnieuw gebruik van **Visual Studio Code**.*

We verwachten ten minste dat je volgende inhouden in een praktische context kan toepassen:

- *installatie, administratie en beheer van PHP/MySQL (server, database) via XAMPP;*
- *PHP variabelen, operatoren, gegevenstypes, controlestructuren, ingebouwde functies, arrays...*
- *formulieren en formulierobjecten voor het invoeren, weergeven, raadplegen en wijzigen van gegevens via PHP/MySQL.*
- *connectie met MySQL databank en opzetten/onderhouden van databanken*
- *kennis MySQL (invoegen, wijzigen, opvragen en verwijderen van gegevens uit een achterliggende database)*
- *invoer van gegevens controleren/valideren*
- *gebruik van session-variabelen*
- *integratie van PHP/MySQL-toepassing in een website*
- *praktische toepassingen: bv. inlogssysteem, registratie voor activiteiten, dynamische en/of gepersonaliseerde menu's/webpagina's, bestellen van producten, raadplegen van catalogus met opvraging van detailinformatie van een gekozen product...*

*En nog dit: **generatieve AI**. Je kan zeker chatGPT e.a. AI-tools ondersteunend gebruiken in je leerproces. Alles zonder meer door chatGPT laten oplossen en die oplossing overnemen, is evenwel geen goed idee. Je leerrendement is daarbij heel beperkt. Ook via Google kan je heel wat nuttige informatie en voorbeelden terugvinden. Dat is vaak handiger en vooral duurzamer dan generatieve AI inzetten. Generatieve AI verbruikt immers heel veel serverenergie en energie is nu eenmaal schaars. Kwestie van duurzaamheid dus.*

Alvast veel plezier en heel veel succes!

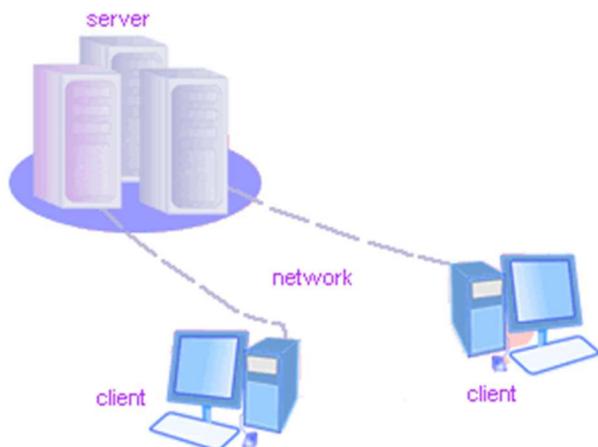
Danny Devriendt

2 Kennismaking met PHP

2.1 Dynamische webpagina's

2.1.1 Client-server

Het Internet is een typisch voorbeeld van een client-server model. Alle informatie is op webservers opgeslagen en via je webbrowser kopieer je die informatie naar je client-PC die met het Internet verbonden is, al dan niet draadloos (wireless).



Bij een statische webpagina is de webserver niet veel meer dan een altijd toegankelijke opslagplaats van webpagina's. Als je op het Internet surft, dan worden de opgeroepen webpagina's naar je client PC gekopieerd. Het is je webbrowser die ervoor zorgt dat je iets te zien krijgt: de HTML-codes worden omgezet in een tastbaar resultaat.

In een dynamische context is er op zijn minst één stap meer. Bij het aanroepen van een dynamische webpagina vanop je client PC, wordt de servercode in of bij je webpagina eerst op de server uitgevoerd. Het resultaat van die uitvoer –HTML-code – wordt naar je client-browser gestuurd die ervoor zorgt dat je de uitvoer op je scherm te zien krijgt.

2.1.2 XAMPP

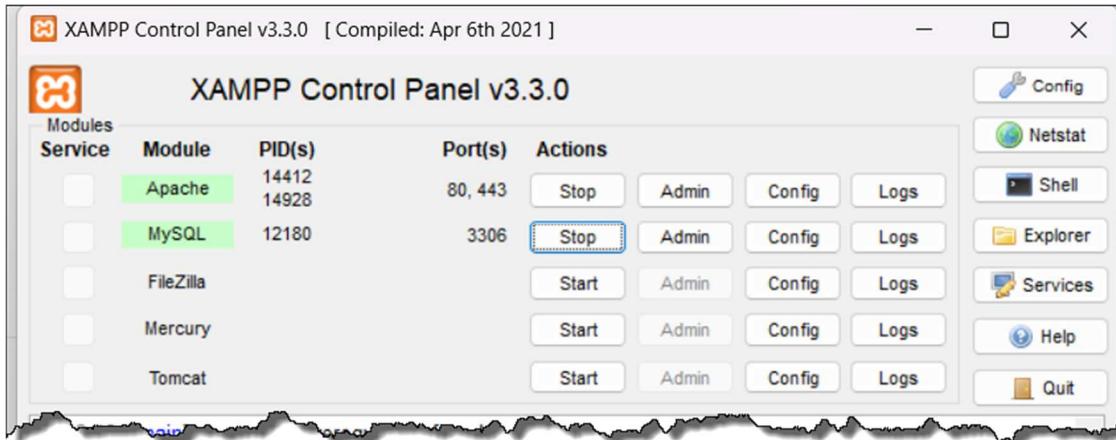
PHP is zo'n (programmeer)taal die je in je webpagina's kunt gebruiken om instructies mee te geven die op de webserver uitgevoerd worden.

Om te vermijden dat we onze pagina's telkens naar een externe webserver die PHP ondersteunt, moeten uploaden, installeren we zo'n webserver op onze eigen PC. We spelen dus server en client op één en hetzelfde toestel.

In plaats van PHP, MySQL en een webserver afzonderlijk te installeren, maak je best gebruik van een geïntegreerde oplossing als XAMPP. XAMPP bestaat zowel voor Linux, Mac als Windows en omvat onder meer volgende functionaliteit:

- Apache, een webserver
- MySQL, voor het gebruik van databases
- PHP, de servertaal die in deze cursus centraal staat
- Perl, een programmeertaal (die gebruiken we niet in deze cursus).

Na de installatie beschik je over een XAMPP Control Panel waarin je kunt controleren of Apache en MySQL draait. Standaard vind je op de C-schijf de map XAMPP met daarin de map HTDOCS, de hoofdmap waarin je webpagina's standaard terecht moeten komen. Vergeet niet eventueel Apache en MySQL te starten vooraleer je aan de slag gaat!



Xampp Control Panel

Het installeren van XAMPP is op zich eenvoudig. De installatiebestanden kan je vinden via www.apachefriends.org. Bekende alternatieven voor XAMPP zijn Easy PHP, WAMP...

Je kan ook voor XAMPP portable opteren en XAMPP USB Lite installeren zodat je op een USB-stick kan werken, maar zorg er zeker voor dat je ook lokaal op je laptop kunt werken.

Normaal gebruikt je webserver poort 80, wat soms tot conflicten leidt als je bv. programma's als Skype of TeamViewer hebt draaien. Sluit eventueel die programma's af of probeer Apache via een andere poort te draaien.

2.1.3 Servertest

Je kunt op je toestel nu zowel server als client spelen. Open je webbrowser en surf naar <http://localhost> of <http://127.0.0.1>.

Hiermee kom je in de map HTDOCS terecht en word je omgeleid naar de pagina INDEX.HTML die geopend wordt. Deze pagina vind je via: C:\XAMPP\HTDOCS\ASHBOARD → INDEX.HTML.



2.2 PHP

2.2.1 Het 3-lagen model

Het 3-lagen model in programmeren is jullie bekend en is hier ook van toepassing:

- De **presentatielaag**, de interface met de gebruiker (GUI) die data weergeeft en input van de gebruiker verzamelt. Hier spelen **HTML** en **CSS** de hoofdrol.

- De applicatielaag (business logic) die de data verwerkt en met de databanklaag in interactie gaat om gegevens op te slaan en op te halen. Hier treedt **PHP** op de voorgrond.
- De databanklaag, de laag waarin data gestockeerd worden. Dit is het domein van **MySQL**.

Hoewel er uiteraard interactie is, werkt elk van de lagen autonoom met eigen spelregels en een takenpakket. Het maakt het ontwikkelen en het onderhoud eenvoudiger.

2.2.2 De eerste test

Een PHP-pagina is een gewone tekstpagina die je zelfs met de Windows-kladblok kan maken. Een PHP-pagina moet PHP als extensie krijgen. Daarmee weet de server dat het een pagina betreft die op de server moet verwerkt worden.

PHP-code wordt gewoon tussen de HTML-code geschreven. Elk blokje PHP-code begint met de tag **<?php** en eindigt met **?>**.

Binnen de PHP-delimiters kan je returns en tabs gebruiken om je code te structureren. Die returns en tabs hebben geen effect op de uitvoering van de code zelf.

De webserver leest de PHP-pagina in, gaat op zoek naar de PHP-code en voert de instructies uit. Het resultaat is HTML-code die naar de client computer gestuurd wordt en door de client-browser gelezen wordt.

De klassieke test. Typ onderstaande instructie in de body van een lege webpagina test.php in.

```
<body>
<?php
echo 'Hello world!';
?>
</body>
```

```
<body>
<?= 'Hello world!';
?>
</body>
```

Echo is een ingebouwde functie waarmee je een tekenreeks uitvoert. Die komt tussen de HTML-code terecht. De argumenten kunnen tussen enkele of dubbele aanhalingstekens staan: het verschil wordt in 2.2.4 duidelijk.

Heb je enkel een echo-instructie in je PHP-codeblok, dan kan je de verkorte delimiters **<?=** en **?>** zonder de echo-functie in te typen. Zie hierboven rechts.

Sla de webpagina op in c:\xampp\htdocs en open de pagina in je browser:
<http://localhost/test.php>. (Je hoeft HTTP:// niet in te typen).

Het resultaat in je browser: Hello world!. Dit is de broncode van je webpagina:

```
<body>
Hello world!
</body>
```

Je merkt dus dat er van de PHP-code niets meer te zien is. Logisch, want die code is door de server verwerkt.

2.2.3 Enkele afspraken

Je kunt ook HTML-code mee opnemen in je PHP-instructies:

```
<?php
echo '<strong>Hello <br>world!</strong>';
?>
```

Het resultaat in je webbrowser:

```
Hello  
world!
```

Elke opdracht binnen een PHP-codeblok eindigt met een ; (puntkomma).

Zoals hoger al gezegd, hoef je geen nieuwe regels te nemen, al is dat in de praktijk wel handig en overzichtelijk. Ook deze code werkt prima:

```
<?php echo '<strong>Hello <br>world!</strong>'; ?>
```

Zoals in elke programmeertaal, kan je ook in PHP je code van commentaar voorzien. Als je commentaar beperkt is tot één regel, dan start je met //.

Bij commentaar die over verschillende regels loopt, gebruik je /* en */ om je commentaar af te bakenen. Alles wat tussen deze delimiters staat, wordt als commentaar aangezien.

Het is een goed idee je PHP-code te documenteren en van de nodige commentaar te voorzien. Dat maakt het onderhoud van de code eenvoudiger.

2.2.4 PHPinfo

Via de functie phpinfo() krijg je heel wat informatie over de gebruikte PHP-versie op je scherm te zien. Welke versie van PHP gebruik je?

```
<?php  
    phpinfo();  
?>
```

2.2.5 Variabelen

Zoals in elke programmeertaal, werken we ook in PHP vaak met variabelen. Er zijn enkele spelregels:

- een variabele begint altijd met een \$-teken
- na het dollarsteek volgt een letter of een _ (underscore), geen cijfer. Cijfers kunnen wel verder in de variabelenaam gebruikt worden.
- de naam van een variabele is hoofdlettergevoelig
- geen spaties of andere tekens in de variabelenaam gebruiken.

Het bereik van de variabele beperkt zich tot de pagina waarin de variabele geïnitialiseerd is.

In PHP worden variabelen normaal niet gedeclareerd. De variabele krijgt automatisch een type afhankelijk van de inhoud die je aan de variabele toewijst. Met de functie var_dump() kan je het datatype dat aan een variabele is toegekend opvragen.

In het voorbeeld hieronder zie je dat het concatenatieteken om tekenreeksen aan elkaar te plakken in PHP een punt (.) is.

Geef je aan een variabele een tekenreeks als waarde, dan plaats je de tekenreeks tussen enkele of dubbele aanhalingstekens:

- gebruik bij voorkeur enkele aanhalingstekens als het om letterlijke tekst of HTML-code gaat.
- gebruik dubbele aanhalingstekens als binnen de tekenreeks variabelen voorkomen waarvan je de actuele inhoud wil weergeven.

```
<?php  
$voornaam = 'Lutgarda';  
$familienaam = 'Petit';  
$totalnaam = $voornaam. ' '. $familienaam;  
echo "Mijn naam is: $totalnaam";  
?>
```

Mocht je in de laatste echo-instructie hierboven enkele aanhalingstekens gebruiken, dan krijg je letterlijk \$totalnaam te zien i.p.v. de inhoud van die variabele. Je kunt ook de letterlijke tekst tussen enkele aanhalingstekens plaatsen en de variabele eraan 'plakken' (concateneren):

```
echo 'Mijn naam is ' . $totalnaam;
```

Bij het toekennen van getallen aan een variabele gebruik je geen aanhalingstekens. PHP gebruikt het **punt** als scheidingsteken tussen eenheden en decimalen.

```
$aantalbesteld = 12;
```

2.2.6 Operatoren

De verkorte operatoren uit Javascript gelden ook in PHP:

Gewoon	Verkort	
\$var = \$var + 1;	\$var++;	verhoogt de variabele \$var met 1
\$var = \$var - 1;	\$var--;	vermindert de variabele \$var met 1
\$var = \$var + 5;	\$var+=5;	verhoogt de variabele \$var met 5
\$var = \$var - 5;	\$var-=5;	vermindert de variabele \$var met 5
\$var = \$var * 5;	\$var*=5;	vermenigvuldigt de variabele \$var met 5
\$var = \$var / 5;	\$var/=5;	deelt de variabele \$var door 5
\$var = \$var." Euro"	\$var.=" Euro"	Voegt de tekenreeks "euro" aan \$var toe
pow(grondtal,exponent)		machtenverheffing: pow(2,3) levert 8 op; pow(3,2) geeft 9.
%		modulus (de rest na een deling: 10%3 levert 1 op)
=		toewijzingsoperator
==		vergelijkingsoperator die de uitdrukkingen links en rechts vergelijkt. Resultaat: TRUE indien gelijk, zoniet FALSE.
!= of <>		niet gelijk aan. Verder ook <, >, <=, >=.
&& of and		logische en-operator
of or		logische of operator
!		logische not operator
.		concatenatieteken bij het samenvoegen van strings
"		begrenzing van strings met interpolatie van variabelen
'		begrenzing van strings zonder interpolatie van variabelen

Let zeker op het verschil tussen de **toewijzingsoperator =** en de **vergelijkingsoperator ==**.

2.2.7 Letterlijke waarden binnen een string

Bepaalde tekens hebben een specifieke betekenis, bv. de aanhalingstekens voor het afbakenen van een string, het \$-teken voor het aangeven van een variabele. Wat als je die tekens letterlijk nodig hebt binnen een string? Dan laat je ze voorafgaan door het **escape** teken \.

Is je tekenreeks afgebakend door enkele aanhalingstekens, dan kan je in de tekenreeks volgende speciale waarden gebruiken:

Teken	Resultaat
\'	levert letterlijk een enkele aanhaling op
\\	levert letterlijk een \ -teken (backslash) op
Alle andere tekens worden letterlijk weergegeven	

Binnen een tekenreeks tussen dubbele aanhalingstekens zijn o.a. volgende speciale waarden mogelijk:

Teken	Resultaat
\"	letterlijk een dubbel aanhalingsteken
\\$	letterlijk een \$-teken
\n	nieuwe regel in HTML broncode zodat de HTML-code leesbaar blijft

2.2.8 Gegevenstypes in PHP

Gegevenstype	
String	gegeven is een tekenreeks (wordt door aanhalingstekens afgebakend)
Integer	gegeven is een geheel getal, positief of negatief
Double	gegeven hoeft geen geheel getal te zijn. Scheidingstekens tussen eenheden en decimalen is een punt. Tot ca. 7 decimalen nauwkeurig. Ook float of real genoemd.
Boolean	gegeven kan enkel de waarde true of false hebben (ja/neen)
Array	een matrix van variabelen met dezelfde naam die via hun positie in de matrix opgeroepen worden (zie verder)
Object	een object is een instantie van een klasse
NULL	voor een variabele zonder waarde, ook geen nul.

2.2.9 Include

Code die vaker terugkomt, hoeven we niet elke keer opnieuw in te typen. Het is zeer gebruikelijk om die code in een afzonderlijke PHP-pagina op te slaan – zonder enige HTML-code buiten de PHP-tags – en die dan op de gewenste plaats in andere pagina's te gebruiken door die daar in te sluiten. Hiervoor gebruik je het include commando. Een voorbeeld:

```
include ("inc_controle.php")
```

In het bestand PHP.INI kan je via de parameter include_path een standaard pad opgeven voor ingesloten bestanden. Ingesloten bestanden worden automatisch in de opgegeven map gevonden. Vaak bevatten include-bestanden cruciale informatie, bijvoorbeeld de connectiegegevens naar een databank. Zie ook hoofdstuk 3.

2.3 PHP-controlestructuren

2.3.1 Keuzestructuren

```
if (voorwaarde)
{
    uit te voeren code
}
```

De te evalueren expressie staat altijd tussen ronde haakjes en wordt zeker niet beëindigd met een ;. De uit te voeren instructies volgen immers verder.

Als het resultaat van de expressie TRUE is, dan wordt het codeblok dat erop volgt uitgevoerd. Bij een FALSE resultaat, wordt dit codeblok niet uitgevoerd.

Het codeblok wordt tussen accolades geplaatst. De accolades kan je weglaten als er slechts één opdracht moet uitgevoerd worden, maar maak er een gewoonte van en plaats ze altijd.

```
<?php
$getal1 = 14;
$getal2 = 12;
```

```

if ($getal1>$getal2)
{
    echo $getal1.'<br>';
    echo $getal2.'<br>';
    echo 'getal1 is groter dan getal 2';
}
?>

```

Om het geheel overzichtelijk te maken, worden accolades vaak op afzonderlijke regels geplaatst en springt het codeblok in. Zeer handig en aanbevolen, maar niet verplicht.

In het voorbeeld hierboven wordt het codeblok enkel uitgevoerd als \$getal1 een grotere waarde bevat dan \$getal2.

Bemerk hierboven dat naast de getallen telkens letterlijk de HTML-code
 meegestuurd wordt. Deze code wordt door de browser van de client geïnterpreteerd en zorgt ervoor dat de gegevens op het scherm telkens op een nieuwe regel terechtkomen. Kan je de eerste twee echo-statements ook zonder concatenatie schrijven?

Je kunt je structuur uitbreiden en ook een actie laten uitvoeren bij *false*.

```

if (voorwaarde)
{
    uit te voeren code bij TRUE
}
else
{
    uit te voeren code bij FALSE
}

```

Of nog uitgebreider:

```

if (voorwaarde)
{
    uit te voeren code bij TRUE
}
elseif (voorwaarde)
{
    uit te voeren code
}
else
{
    uit te voeren code
}

```

Bestudeer het voorbeeld hieronder en test uit door de waarden van \$getal1 en \$getal2 telkens te wijzigen in je code.

```

<?php
$getal1 = 14;
$getal2 = 12;
if ($getal1>$getal2)
{
    echo $getal1.'<br>';
    echo $getal2.'<br>';
    echo 'getal1 is groter dan getal 2';
}
elseif ($getal1<$getal2)
{
    echo $getal1.'<br>';
}

```

```

echo $getal2.'<br>';
echo 'getal1 is kleiner dan getal 2';
}
else
{
    echo 'beide getallen zijn aan elkaar gelijk!';
} ?>

```

Het **ELSEIF** statement kan je zo vaak na elkaar gebruiken als nodig, met telkens een andere voorwaarde. De **ELSE** constructie wordt enkel uitgevoerd als geen van de bovenstaande voorwaarden true is.

Een alternatief is hier het gebruik van de **switch** constructie.

```

switch (expressie)
{
case waarde:
    uit te voeren code
case waarde:
    uit te voeren code
case waarde:
    uit te voeren code
...
default:
    uit te voeren code
}

```

Een voorbeeld:

```

<?php
$getal1 = 2;
switch ($getal1) {
case 1:
    echo 'Het getal is één.<br>';
    break;
case 2:
    echo 'Het getal is twee.<br>';
    break;
case 3:
    echo 'Het getal is drie.<br>';
    break;
default:
    print 'Het getal verschilt van 1, 2 of 3.';
}
?>

```

Als de waarde van de expressie overeenstemt met één van de case-blokken, dan wordt de code vanaf die case-blok uitgevoerd tot de eerstvolgende **break**-instructie. Zonder break-instructie wordt dus ook de code van alle nog volgende cases uitgevoerd, ook al verschilt de case-waarde van de switch-expressie.

De default-blok wordt enkel uitgevoerd als geen van de case-waarden overeenkomt met de switch-expressie.

2.3.2 Herhalingen

De herhaling-controlestructuren hebben volgende syntaxis:

Syntaxis herhaling met beginvoorwaarde

```
while (voorwaarde)
{
    uit te voeren code
}
```

De instructies in het codeblok worden herhaald zolang de voorwaarde true is. De voorwaarde wordt geëvalueerd vóór het codeblok uitgevoerd wordt.

```
echo 'Oneven getallen onder 10:<br>';
$getal = 1;
while ($getal<=10)
{
    echo $getal.'<br>';
    $getal=$getal+2;
}
```

Het resultaat:

Oneven getallen onder 10:

1

3

...

Let op het gebruik van regeleindes en vergeet natuurlijk niet je variabele binnen de lus te verhogen, zoniet kom je in een oneindige lus terecht!

Opdrachtje: doorloop alle getallen van 1 tot en met 100 en haal er de even getallen uit door gebruik te maken van de modulus operator. Geef de resultaten zoals hieronder weer:

2 – 4 – 6 – 8 – 10 – 12 – 14 – 16 ...

Syntaxis herhaling met eindvoorwaarde

```
do
{
    uit te voeren code
}
while (voorwaarde)
```

Hier wordt eerst het codeblok uitgevoerd en pas daarna gecontroleerd of de voorwaarde true is. Het codeblok wordt dus ten minste één keer uitgevoerd.

In het voorbeeld hieronder bouwen we de tafel van 3 op:

```
<?php
echo '<strong>Tafel van 3:</strong><br>';
$getal = 1;
do {
    if ($getal<10) {echo "0";}
    echo $getal." x 3 = ".($getal*3)."<br>";
    $getal++;
}
while ($getal<=10) ?>
```

Dit is het resultaat:

Tafel van 3:

01 x 3 = 3

02 x 3 = 6

03 x 3 = 9...

Syntaxis begrensde herhaling

```
for (expressie1; voorwaarde; expressie2)
{
    codeblok dat uitgevoerd wordt
}
```

Het codeblok wordt uitgevoerd zolang aan de voorwaarde voldaan is. De controle gebeurt vóór het uitvoeren van het codeblok. Expressie1 wordt één keer bij het begin van de lus ingelezen. Deze expressie bevat meestal een startwaarde.

Na het uitvoeren van het codeblok wordt expressie2 uitgevoerd en wordt de voorwaarde geëvalueerd. Als de voorwaarde true is, wordt het codeblok opnieuw uitgevoerd.

```
<?php
for ($i=1;$i<11;$i++)
{
    $ster = $ster."*";
    echo $ster."<br>";
} ?>
```

Dit is het resultaat:

```
*
```



```
**
```



```
***...
```

Doorlopen van een array: **foreach** > zie 2.6.2.

2.4 Functies

Je kan een beroep doen op de vele ingebouwde functies in PHP. We lijsten een aantal veel gebruikte functies op in de cursus.

2.4.1 Stringfuncties

Hieronder vind je een overzicht van enkele functies die je op tekenreeksen kunt toepassen.

Resultaat	Functie	Omschrijving
string	addslashes	zet in een string ', ", \ en nullbyte om in een escape reeks. Dit hoeft niet als magic_quotes_gpc actief is in php.ini (standaard). ' wordt in \' omgezet; " in \"...
string	stripslashes	zet een escape-reeks (zie hierboven) om naar een gewone reeks.
string	ltrim	verwijdt witruimte (spaties) aan het begin van een string
string	rtrim	verwijdt witruimte (spaties) op het einde van een string
string	trim	verwijdt witruimte (spaties) aan begin én einde van een string
string	str_ireplace str_replace	vervangt binnen een string een zoekstring door een vervangstring, bv. str_replace("<", "<", \$tekst). str_replace is case-sensitive, str_ireplace niet.
string	strip_tags	verwijdt alle HTML-tags uit een string, behalve de opgegeven tags, bv. strip_tags(\$tekst,"<u>")
int	strlen	geeft de lengte van een string weer
int	strpos	geeft de startpositie van een substring in een string mee, bv. strpos ("Zweden","en") > 4 (eerste teken heeft positie 0!).
string	strtolower strtoupper	zet een string om in kleine letters zet een string om in hoofdletters

Resultaat	Functie	Omschrijving
string	substr	haalt een deel uit een string. Negatieve getallen tel je vanaf rechts. substr("tekenreeks", startwaarde, aantal tekens) substr("website",3,2) levert <i>si</i> op. Eerste teken heeft positie 0! substr("website",-2,2) levert <i>te</i> op. substr("website",-4,-1) levert <i>sit</i> op.
array	range	bouwt een array met waarden op \$reeks = range (1,5) > array bevat 1, 2, 3, 4 en 5; \$reeks = range (1,14,3) > array bevat 1, 4, 7, 10 en 13; \$reeks = range ("A", "Z") > array bevat alle hoofdletters van A tot Z.
int	substr_count	telt hoeveel keer een substring in een string voorkomt: substr_count("string", "substring") substr_count("banaan", "a") levert het resultaat 3 op.
array	str_split	splitst een string op in een array van afzonderlijke tekens.

2.4.2 Wiskundige functies

Enkele wiskundige functies

Resultaat	Functie	Omschrijving
float	ceil	afronden naar hoger geheel getal
float	floor	afronden naar lager geheel getal
float	round	afronden naar een opgegeven aantal decimalen. round(\$getal,2) rondt de waarde uit de variabele \$getal af op 2 cijfers na de komma. Als derde parameter kan je nog een afrondmode opgeven, bv.: round(6.5,0,PHP_ROUND_HALF_UP) geeft 7 round(6.5,0,PHP_ROUND_HALF_DOWN) geeft 6.
Formatted version	number_format	bepaalt hoe numerieke gegevens weergegeven worden. Bv.: number_format (\$getal,2,".") bepaalt dat de waarde 3214.12 in \$getal zal weergegeven worden als 3 214,12. Je geeft dus achtereenvolgens op welke waarde er moet weergegeven worden, hoeveel cijfers er na de komma weergegeven worden en wat de scheidingstekens voor decimalen en duizendtallen zijn. Je kan ook gewoon number_format(\$getal,2) kiezen om enkel het aantal cijfers na de komma te bepalen.
void	mt_srand	initialiseren van de generator voor willekeurige getallen
int	mt_rand	genereert een willekeurig geheel getal tussen de opgegeven grenzen, inclusief de grenzen. mt_rand(2,9) levert een willekeurig getal tussen 2 en 9 op.

2.4.3 Isset() en empty()

Met de functie **isset()** ga je na of één of meer opgegeven variabelen wel een waarde toegekend kregen. Die waarde kan ook 0 of een lege string zijn. Als alle opgegeven variabelen in dat geval zijn, is het resultaat van de functie true. Van zodra één variabele geen waarde heeft (null), is het resultaat false.

```
<?php
if (isset ($var1,$var2,$var3))
{echo "Alle variabelen hebben een waarde.";} ?>
```

De functie **empty()** gaat na of de variabele een lege waarde heeft (lege string, 0, null of false). In dat geval is het resultaat van de functie true. De functie geeft false als resultaat als de variabele niet leeg is en geen 0-waarde bevat.

```

<?php
$var = 0;
if (empty($var)) {
$tekst = "De variabele is leeg of bevat 0.";
}
else {
$tekst = "De variabele bevat een niet-lege waarde.";
}
echo $tekst
?>

```

2.5 Datum en tijd

2.5.1 De huidige datum: date()

Met de functie date() geef je de huidige datum weer. Je bent verplicht een parameter mee te geven die het formaat van de datum bepaalt. De uitgebreide lijst van mogelijke parameters kan je vinden op php.net. Je kunt rechtstreeks php.net/date intypen voor informatie over de date() functie. Een beknopt overzicht volstaat ruim voor onze toepassingen:

Teken	Beschrijving	Voorbeeld
d	dag, met voorloopnullen	01 - 31
j	dag, zonder voorloopnullen	1 – 31
l (kleine L)	dag, voluit (Engels!)	sunday - saturday
w	dag, in week	0 (zondag) tot 6 (zaterdag)
N	dag, in week (ISO 8601)	1 (maandag) tot 7 (zondag)
F	maand, voluit (Engels!)	January – December
m	maand, met voorloopnullen	01 – 12
n	maand, zonder voorloopnullen	1 – 12
Y	jaar, vier cijfers	2009
y	jaar, twee cijfers	09
H	uur, twee cijfers, 24 uur	00-23
i	minuten, met voorloopnullen	00-59
s	seconden, met voorloopnullen	00-59

Een voorbeeld:

```

<?php
echo 'vandaag is '.date("Y / m / d ");
?>

```

Het resultaat is bij voorbeeld: vandaag is 2025 / 01 /06.

Diezelfde functie date() kan je ook gebruiken om de huidige tijd weer te geven:

```

<?php
echo 'nu is het '.date("H:i:s").' u.';
?>

```

Het resultaat is: nu is het 10:48:30 u.

In het voorbeeld hieronder testen we via een selectie (`IF`) op de datum van vandaag en passen we via een class de achtergrond van de webpagina aan de datum aan.

```

<style type="text/css">
.groen {background-color: #00CC33;}
.blauw {background-color: #6666FF;}
</style>
</head>
<body class=
<?php
if (date("j") <= 15)
{echo '"blauw"';}
else
{echo'"groen"';}
?>>...

```

2.5.2 Rekenen met datums

Heel vaak speelt het aantal dagen tussen twee datums een belangrijke rol. Het berekenen van dat aantal dagen heeft wel wat voeten in de aarde. Er zijn verschillende functies. Wij houden het hier bij de functie **strtotime**.

Met de functie **strtotime** zet je een datum om in een zogenaamde timestamp, een geheel getal dat de waarde in seconden uitdrukt t.o.v. 1 januari 1970, 0 uur, 0 minuten, 0 seconden.

Op die manier kan je met datums rekenen door de timestamps van twee data van elkaar af te trekken. Het getal dat je uitkomt, is een aantal seconden.

Deel dit getal door 60 om minuten te krijgen, door 3600 om uren te krijgen en door 3600*24 om het verschil in dagen te bekomen. Gebruik eventueel een afrondfunctie.

Als je de datum via een tekstvak invoert of via een variabele in bv. de NBN-norm, heb je eventueel wat bijkomend werk. Gebruik de functie substr voor het opsplitsen van de datum.

In het eerste voorbeeld hieronder krijg je een berekening van het aantal seconden tussen twee tijdstippen, uitgedrukt tot op de seconde.

In het tweede voorbeeld berekenen we het aantal dagen tussen twee datums.

```

/*Voorbeeld 1*/
$start=strtotime("2025-01-04 23:59:02");
$einde=strtotime("2025-01-05 00:01:01");
$verschil = $einde - $start;
echo $verschil; /*verschil in aantal seconden, hier 119*/

/*Voorbeeld 2*/
$start2=strtotime("2025-01-28");
$einde2=strtotime("2025-02-02");
$verschil2 = $einde2 - $start2;
$aanstdagen = $verschil2/(24*60*60);
echo "Het aantal dagen tussen beide data is $aanstdagen.";
```

2.6 Arrays

2.6.1 Arrays gebruiken

In PHP maken we heel vaak gebruik van arrays, waarin verschillende waarden kunnen bijgehouden worden op basis van een key en een value. Je kunt arrays op verschillende manieren maken. In het voorbeeld hieronder is de key de positie in de array, te beginnen met 0, een **indexed array**. Zo levert de key 0 de waarde Robbe op, de key 1 Jan...

```
$studenten = array("Lobke", "Simon", "Emma", "Alexander");
```

Via de key spreek je elke waarde uit een array aan. Wat levert onderstaande instructie?

```
echo $studenten[1];
```

Niet Lobke, maar Simon, want Lobke heeft key of index 0 in de array.

Je kunt een waarde aan de array toevoegen door geen indexwaarde op te geven. Het nieuwe item krijgt dan automatisch de eerstvolgende vrije indexplaats.

```
$studenten[] = "Adam";
```

In plaats van met genummerde sleutels te werken, kan je ook je key benoemen. We spreken hier van een **associatieve array**.

```
$frans=array("huis"=>"maison", "auto"=>"voiture", "muis"=>"souris");
```

Hetzelfde verkrijg je door:

```
$frans["huis"]="maison";
$frans["auto"]="voiture";...
```

Via de **key** kan je nu de **value** (waarde) opvragen:

```
echo $frans["huis"]
```

Het resultaat van de echo-opdracht hierboven: maison.

Met de functie **count** of **sizeof** kan je het aantal waarden in een array tellen:

```
echo 'Er zitten '.count($studenten).' studenten in de klas.';
```

Te onthouden: arrays zullen bij formulieren en databanken nooit ver weg zijn.

Voorbeeld: de datum voluit weergeven in het Nederlands

In de code hieronder gebruiken we de array's \$maanden en \$weekdagen om een datum \$mijndatum voluit in het Nederlands weer te geven. Bemerk dat we van de dag en maand twee keer 1 aftrekken omdat de eerste waarde in de array 0 is.

```
$maanden = array("januari", "februari", "maart", "april", "mei", "juni",
"juli", "augustus", "september", "oktober", "november", "december");
$weekdagen = array("maandag", "dinsdag", "woensdag", "donderdag",
"vrijdag", "zaterdag", "zondag");
echo 'Tijdstip: '.$weekdagen[date("N", $mijndatum)-1]."
.date("j", $mijndatum). " ".$maanden[date("n", $mijndatum)-1]."
.date("Y", $mijndatum). '<br><br>';
```

Je kan evenwel weekdagen of maanden ook in het Nederlands weergeven zonder een array met namen te gebruiken.

```
setlocale(LC_ALL, 'nld_NLD');
strftime("%A %e %B %G", $mijndatum);
```

In de code hierboven staat:

- %A – weekdag, voluit
- %e – dag van de maand, 1 tot 31
- %B – naam van de maand, voluit
- %G – jaar in 4 cijfers

2.6.2 Een array doorlopen

Heel vaak worden gegevens in een array doorlopen. In onderstaand voorbeeld drukken we een lijstje af met alle waarden uit de array.

```
$studenten = array("Milo", "Celestin", "Senna", "Gilles", "Hugo", "Joshua",
"Arthur", "Julian", "Maxim", "Bouke", "Sem");
for ($i=0;$i<count($studenten);$i++)
{echo 'Student: '.$studenten[$i].'  
'; }
```

Je kan een array evenwel ook met een **foreach** lus doorlopen:

```
$studenten = array("Lobke", "Simon", "Emma", "Alexander", "Daan", "Aaron",
"Pahul", "Tuur", "Stan", "Luka");
foreach($studenten as $a)
{echo 'Student: '.$a.'  
'; }
```

Een voorbeeld bij een associatieve array:

```
$ww=array ("An Smet"=>"smetje", "Mieke Vos"=>"vossie", "Jo Sys"=>"jojo");
foreach($ww as $persoon=>$persoon_value)
{
echo "Het wachtwoord van ".$persoon." is
".$persoon_value."  
";
}
```

In het voorbeeld hierboven is \$persoon gebruikt als variabele voor de *key* en \$persoon_value als variabele voor de *value*.

Alle elementen van de array worden automatisch doorlopen met foreach. Na het laatste element stopt het doorlopen.

2.6.3 Een reeks variabelen opbouwen via een iteratie

Veronderstel even dat je de variabelen \$getal1, \$getal2... \$getal50 moet genereren en gebruiken in een iteratie door bv. een teller \$i te gebruiken.

Op het eerste gezicht kan je hier \$getal en \$i concateneren: \$getal.\$i. Dit zal niet werken, omdat hier gezocht wordt naar de waarde van \$getal die niet ter beschikking is.

De oplossing is het gebruik van **accoades**: \${'getal'}.{\$i}. De expressie binnen de accolades wordt nu eerst uitgewerkt: letterlijk de letters getal met daaraan geplakt de waarde van de teller \$i. We krijgen dus getal1, getal2... getal50 als resultaat en met het dollarteken ervóór wordt dat \$getal1, \$getal2... precies wat we willen.

2.6.4 Een willekeurig element uit een array ophalen: array_rand()

Met de **array_rand()** functie haal je een willekeurige sleutel uit een array, bij voorbeeld:

```
$studenten = array("Mattias", "Henri", "Jani", "Jules", "Mathijs", "Josse",
"Bas", "Tatiana");
$gekozen = $studenten[array_rand($studenten) ];
echo "<p>Gekozen student: $gekozen</p>";
```

Elke keer als je deze pagina laadt of refresht, wordt een willekeurige student uit de array gekozen en in de variabele \$gekozen gestockeerd.

2.7 Tekens aan een string toevoegen met str_pad()

Met **str_pad()** voeg je extra tekens aan een string toe tot een opgegeven aantal.

```
$input ="Ipanema";
echo str_pad($input,15,"*");
```

Het resultaat van de code hierboven is Ipanema*****. Er worden rechts *-tekens toegevoegd tot het totaal aantal tekens n de tekenreeks 15 is. Hier is dat 8 keer een * dus. Geef je geen teken op, dan worden spaties toegevoegd.

Met een optionele vierde parameter bepaal je waar de extra tekens toegevoegd worden:

```
$input = "Ipanema";
echo str_pad($input, 15, "*", STR_PAD_BOTH);
```

- STR_PAD_BOTH verdeelt de extra tekens links en rechts: ****Ipanema****. Bij oneven aantal, wordt rechts een teken meer geplaatst.
 - STR_PAD_LEFT voegt de extra tekens links toe: *****Ipanema.
 - STR_PAD_RIGHT voegt de extra tekens rechts toe: Ipanema*****.
- Dat is de standaardwaarde en die kan je ook weglaten.

Let op: de vierde parameter moet in hoofdletters ingegeven worden!

Een praktisch voorbeeld is het toevoegen van voorloopnullen bij getallen kleiner dan 10:

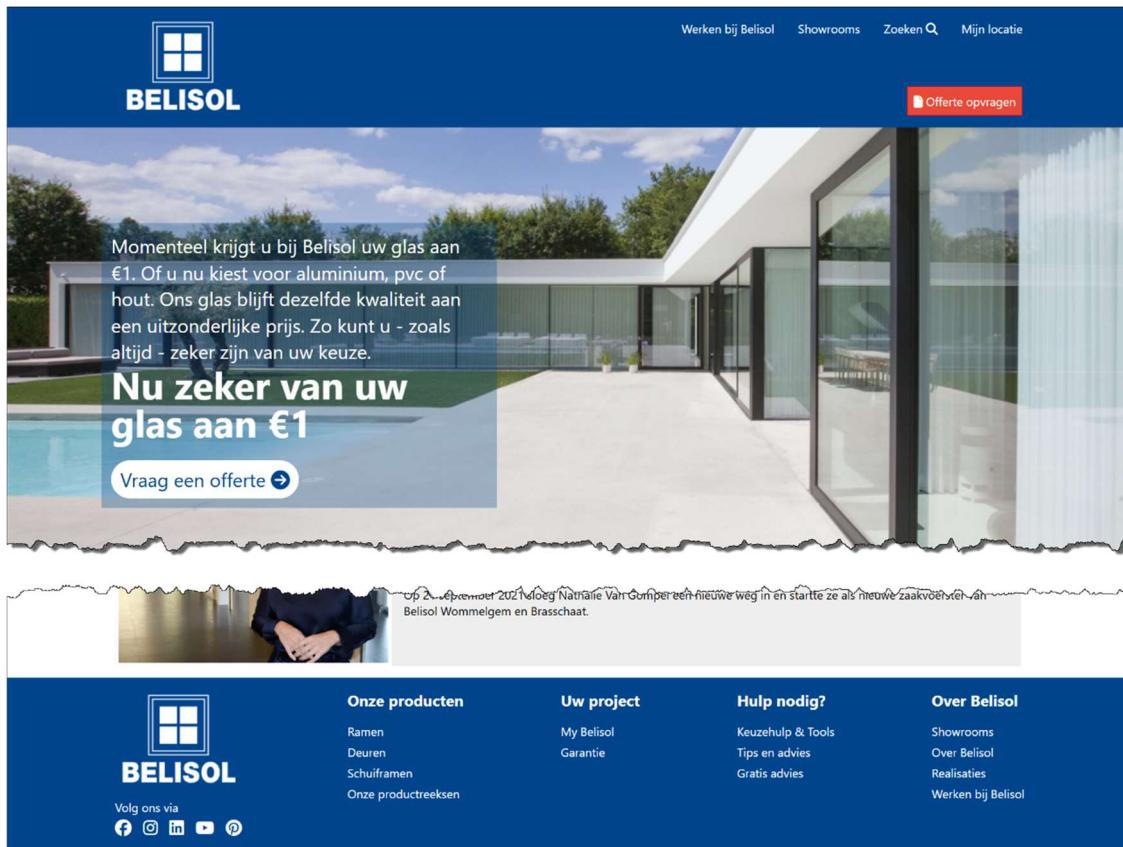
```
$getal = mt_rand(1,100);
echo str_pad($getal, 3, "0", STR_PAD_LEFT);
```

Via str_pad wordt bv. de willekeurige waarde 6 als 006 weergegeven, wat soms handig kan zijn.

3 Website opbouwen met include()

Om een website op te bouwen maken we handig gebruik maken van de **include()** functie die in 2.2.8 al aan bod kwam om informatie die op elke webpagina herhaald wordt in te sluiten.

Neem als voorbeeld de webpagina hieronder die bestaat uit een navigatiezone met header bovenaan en een footer onderaan die op elke pagina terugkomen. De inhoud van de zones ertussen verandert pagina per pagina.



In de verschillende containers breng je geen inhoud aan, maar haal je eigenlijk telkens de inhoud uit een andere, losse pagina op met de **include()** functie:

navigatie <?php include("inc_nav.php");?>
header <?php include("inc_header.php");?>
footer <?php include("inc_footer.php");?>

Wil je dan achteraf bv. de footer in alle webpagina's wijzigen, dan moet je enkel de pagina **INC_FOOTER.PHP** aanpassen en alle pagina's zijn aangepast.

Uiteraard geldt dit niet voor de content-zone: in die zone komt per webpagina telkens een andere inhoud.

De opbouw van de webpagina ziet er dan bijvoorbeeld als volgt uit:

```
<div id="wrapper">
    <header><?php include("inc_header.php");?></header>
    <nav><?php include("inc_nav.php");?></nav>
    <div id="content">
        ...
    </div>
    <footer><?php include("inc_footer.php");?></footer>
</div>
```

Via een extern CSS-bestand regel je dan de positie en opmaak van de verschillende elementen. Aanpassingen in dit bestand gelden dan ook automatisch voor alle webpagina's.

4 Formulieren

4.1 Gegevens uit een formulier verwerken

In de basiscursus HTML/CSS maakte je al met formulieren en de meeste courante formulierobjecten kennis. Je leerde ook met HTML aan client side validatie te doen. Nu versturen we de informatie uit een formulier naar de server.

Met de parameter *name* krijgt elk formulierobject een naam, bv. *txtVoornaam* bij het tekstvak waarin de voornaam ingetypt wordt.

Bij het versturen van het formulier met de method **POST** worden de waarden van alle formulierobjecten in de superglobale array **\$_POST** opgeslagen. Dat gebeurt automatisch. De naam van de formulierobjecten geldt als key.

Met **\$_POST['txtVoornaam']** haal je de ingevoerde waarde uit het tekstvak *txtvoornaam*. Gebruik altijd hoofdletters om de **\$_POST** array aan te spreken!

```
echo $_POST['txtVoornaam'];
```

Het resultaat van bovenstaande echo-opdracht is de inhoud van het tekstvak *txtVoornaam* uit het verstuurde formulier.

4.1.1 Controle op bestaan en inhoud

Met HTML5 kan je een aantal controles op de inhoud van je formulier doen vóór je het verstuurt, *client side*, op het toestel van diegene die je website bezoekt: controle of er wel iets ingevuld is in een bepaald tekstvak, of een waarde wel binnen een bepaald bereik ligt...

Ook bij de verwerking van het formulier op de server – *server side* dus – kan je controleren of een object wel bestaat en of het een inhoud heeft: met de **isset()** en **empty()** functies die ook al in 2.4.3 aan bod kwamen:

```
if (isset($_POST['txtVoornaam']))
```

controleert of er wel een object *txtVoornaam* bestaat.

```
if (!empty($_POST['txtVoornaam']))
```

controleert of het object niet leeg is. Het !-teken vóór *empty* is de not-operator zoals je weet.

Je kunt ook eenvoudig controleren of een formulier al verstuurd is:

```
if (isset($_POST['btnVersturen']))
```

Het codeblok dat op bovenstaande selectie volgt, wordt enkel uitgevoerd als er op de knop met de naam *btnVersturen* geklikt is.

Zo kan je bv. bij het aanroepen van een pagina een formulier weergeven dat je bij het verwerken van dezelfde pagina – na het klikken op de knop – niet meer weergeeft.

```
if (!isset($_POST['btnVersturen']))
{
?>
<form>
... hier komt het formulier dat enkel de eerste keer zichtbaar is.
</form>
```

```

<?php
}
else
{
Hier komt code die uitgevoerd wordt nadat de formulierknop is
gebruikt. Het formulier zelf krijg je niet meer te zien.
}

```

4.1.2 Strip_tags

In de tekstvakken kunnen gebruikers met minder fraaie bedoelingen allerlei HTML-code intypen, zelfs script aanroepen... Dat is natuurlijk niet de bedoeling.

Het is daarom niet overbodig om de doorgegeven waarde te controleren op lengte en om de functie **strip_tags** toe te passen. De functie **strip_tags** verwijdert alle HTML, XML en PHP-tags uit een string, behalve de eventueel als tweede argument opgegeven tag(s).

```
$voornaam = strip_tags($_POST['txtVoornaam'], "<strong><u>")
```

4.1.3 Redirect header

Onderstaande redirect-header kan je gebruiken om vanuit het PHP-script de navigatie naar een bepaalde pagina te sturen. Het is belangrijk dat er **vóór deze instructie geen uitvoer** is. Deze code komt best in een php-blok dat helemaal bovenaan zelfs vóór de <HTML>-tag staat. Er mag ook geen lege regel of spatie voorafgaan.

```
Header("Location: bestel.php");
```

```
Header("Location: http://www.vlam.be");
```

Een voorbeeld

In het eenvoudige voorbeeld hieronder krijg je bij een eerste aanroep het formuliertje te zien waar enkel een wachtwoord moet ingevuld worden.

Afhankelijk of het wachtwoord *admin* of *student* is ingevoerd, word je doorgestuurd naar respectievelijk de pagina ADMIN.PHP of STUDENT.PHP.

Geef je een ander wachtwoord in, dan blijf je het formulier te zien krijgen.

```

<?php
if ($_POST["txtWachtwoord"] == 'admin')
{
    header("Location:admin.php");
}
elseif ($_POST["txtWachtwoord"] == 'student')
{
    header("Location:student.php");
}
else
{
?>
...
<body>
<form name="frmWachtwoord" method="post" action="">
    <p>paswoord?
        <input type="password" name="txtWachtwoord"/>
    </p>
    <p>

```

```

        <input type="submit" name="cmdLogIn" value="Log in" />
</p>
</form>
<?php
}
?>
</body>
</html>

```

4.2 Javascript voor het verwerken van je formulier

Het verwerken van een formulier gebeurt meestal door het klikken op een knop in het formulier. Soms is het vervelend eerst nog op een knop te moeten klikken.

Je kunt het verwerken van je formulier ook laten afhangen van een andere gebeurtenis, met bv. een javascript. Een voorbeeld:

```

<form id="frmKeuzelijst" name="frmKeuzelijst" method="post"
action="verwerk.php">
    <select name="cboLijst" id="cboLijst"
onChange="document.frmKeuzelijst.submit();"

```

In dit voorbeeld zal bij het kiezen van een item in de keuzelijst – correcter gezegd: bij het wijzigen van je keuze – de event handler **onChange** ervoor zorgen dat het bijhorende javascript wordt uitgevoerd. De javascript submit() functie zorgt ervoor dat het formulier *gesubmit* wordt, doorgestuurd voor verwerking naar de webserver.

Bij keuzerondjes en selectievakjes gebruik je in Internet Explorer beter de **onClick** event handler. De onChange event handler werkt hier niet correct.

Wil je met de eerste keuze in je keuzelijst werken, dan moet je dus eerst een andere keuze maken en dan terug de eerste keuze maken om het *onChange* event te triggeren. Een workaround is dat je ervoor zorgt dat het eerste element in je keuzelijst geen echte keuze is, maar bv. de inhoud *Maak je keuze...* laat zien.

Bij een tekstvak, selectievakje... kan je ook nog de **onBlur** event handler gebruiken om een actie uit te voeren als het object de focus verliest, dus als je vanuit het tekstvak op een ander element klikt of tabuleert.

De **onFocus** event handler triggert een actie die uitgevoerd wordt als een object de focus krijgt, het tegengestelde van onBlur dus.

4.3 Ingevulde en gekozen gegevens in je formulier behouden

Bij het invullen of kiezen van gegevens in een formulier, komt het vaak voor dat je na het *submit*ten van het formulier – via submit-knop of via een javascript – de formuliergegevens nog in het formulier wilt behouden als het formulier nog zichtbaar blijft. Normaal wordt het formulier volledig opnieuw opgebouwd en is het dus terug in zijn oorspronkelijke toestand, meestal leeg.

4.3.1 Voorbeeld

In het voorbeeld hieronder is het de bedoeling om de ingevulde gegevens na het *Registreren* nog in het formulier zichtbaar te houden en een registratiebericht te laten verschijnen: *Registratie van XXX bij docent YYY om ZZZ uur.*

The form consists of two main sections. The top section contains input fields for 'Voornaam' (with placeholder 'Michiel') and 'Familienaam' (with placeholder 'Devos'), a dropdown menu for 'Kies je vaklector' (with 'Amelie Caveye' selected), and two buttons: 'Registreren' and 'Annuleren'. The bottom section contains a radio button group for 'Keuze uur' ('14.15' is selected) and another pair of buttons. A message box at the bottom displays the registration confirmation: 'Registratie van Michiel Devos bij lector Wim Petit om 14.15 uur.'

Hiervoor vangen we de waarden van het geposte formulier op via de `$_POST` array en geven we die gegevens standaard in het formulier weer via de `value`-eigenschap.

Bij een **tekstvak** geef je dan als value de waarde mee die vóór het versturen ingevuld was:

```
<input name="txtFamilienaam" type="text" id="txtFamilienaam"  
value='<?php echo $_POST['txtFamilienaam']; ?>' />
```

Bij een **keuzerondje** voeg je aan de input-tag de optie **checked** toe om een keuzerondje geactiveerd weer te geven. Je controleert daarvoor eerst of de value van elk specifiek keuzerondje overeenstemt met de waarde die gekozen werd en via de `$_POST` array wordt doorgegeven.

```
<input type="radio" name="uur" id="radio10" value="17.15" <?php if  
($_POST['uur']=='17.15') {echo "checked";}?> />
```

Bij een **keuzelijst** werk je analoog en voeg je **selected='selected'** toe aan de gekozen optie. De meeste browsers accepteren nu ook gewoon **selected** als attribuut.

In het voorbeeld dat volgt wordt de keuzelijst opgebouwd met waarden uit een array. Telkens wordt nagegaan of de waarde overeenstemt met de eerder gekozen waarde uit de lijst. Is dat het geval, dan wordt `selected='selected'` aan de option-tag toegevoegd.

```
$docenten = array("Amelie Caveye", "Ilse De Bolster", "Lut De  
Jaegher", "Danny Devriendt", "Fleur Liefooghe", "Wim Petit", "Lynn Van  
Elsen", "Jasper Wouters");  
echo "<form name='frmtest' method='post' action='><select name  
='cboDocenten' onChange='document.frmtest.submit()'>";  
foreach($docenten as $a)  
{echo "<option value='$a'";  
if ($_POST['cboDocenten']===$a) {echo " selected='selected'";}  
echo ">$a</option>";}  
echo "</select></form>";
```

In dit voorbeeld komen nog twee extraatjes aan bod:

- de keuze van het uur komt pas te voorschijn nadat een docent gekozen is.
- de registratie wordt pas uitgevoerd als voornaam en familienaam zijn ingevuld en als er ook een uur gekozen werd.

4.3.2 Keuze van het uur

Om de uurkeuze pas te laten verschijnen na het kiezen van een docent, moet het formulier gepost worden bij de keuze van de docent. Dat gebeurt via een javascript (zie hoger).

```
<select name="cboDocenten" onChange="document.frmRegistreer.submit()">
```

Daarbij is er één probleem: het onChange-event werkt maar als je een andere keuze maakt in de lijst. Als je toevallig de eerste keuze nodig hebt, zou je dus eerst een andere docent moeten kiezen en dan terug de eerste docent.

Een workaround is het toevoegen van een eerste keuze – buiten de lus die de keuzelijst genereert – waarbij je uitnodigt een keuze te maken.

Het tabellertje met de uren laat je pas verschijnen als een keuze gemaakt is in de lijst:



```
<?php if (isset($_POST['cboDocenten'])) { ?>
<table width="100%" border="0"><tr>
<td width="90">Keuze uur </td>
<td width="90"><input type="radio" name="uur" id="radio11" value="14.15" <?php if ($_POST['uur'] == '14.15') {echo "checked";}?>/>14.15</td>
<td width="90"><input type="radio" name="uur" id="radio10" value="17.15" <?php if ($_POST['uur'] == '17.15') {echo "checked";}?>/>17.15</td>
</tr><table>
<?php } ?>
```

4.3.3 Controle op het invullen

De registratie mag pas gebeuren als alle gegevens ingevuld zijn. Hoewel je die controle bij/na het versturen van het formulier kunt doen – *server side* – ben je met het HTML5 attribuut **REQUIRED** in je formulierobjecten veel beter af om er bv. voor te zorgen dat voornaam en familienaam zijn ingevuld vóór je het formulier stuurt (*client side*). Snel geregeld en je hoeft de server-side controlecode die hieronder volgt niet te schrijven.

Als geen van de tekstvakken leeg is en ook een uur gekozen werd, drukken we het registratiebericht af. Werk je met een databank, dan schrijf je hier de gegevens weg. In ons voorbeeld is dat niet het geval, en laten we een passend bericht verschijnen.

```
<?php
if (isset($_POST['btnRegistreren']))
{
    if (!empty($_POST['txtVoornaam'])
and !empty($_POST['txtFamilienaam'])
        and !empty($_POST['uur']))
    {
        $bericht = "Registratie van ".$_POST['txtVoornaam']."
".$_POST['txtFamilienaam']."' bij docent ".$_POST['cboDocenten']."
om ".$_POST['uur']. " uur.";
    }
}
```

```

else {
    if (empty($_POST['txtVoornaam'])) { $bericht .= "Je voornaam is
niet ingevuld!<br>"}
    if (empty($_POST['txtFamilienaam'])) { $bericht .= "Je familienaam
is niet ingevuld!<br>"}
    if (empty($_POST['uur'])) { $bericht .= "Je hebt geen uur
gekozen!<br>"}
}
echo $bericht;
} ?>

```

Test PHP

Registratie test PHP 23 mei 2025

Voornaam Familienaam
Kies je vaklector: Amelie Caveye

Keuze uur 14.15 17.15

Je voornaam is niet ingevuld!
Je familienaam is niet ingevuld!
Je hebt geen uur gekozen!

4.3.4 Het formulier leegmaken

Met de knop Annuleren herstel je het formulier in zijn oorspronkelijke toestand. Ook de knop Annuleren is een submit-knop (een reset-knop werkt niet).

Controleer bovenaan in de pagina of op de knop Annuleren geklikt is. Is dat zo, schakel dan de `$_POST` array uit met de `unset()` functie. Bij het klikken op Annuleren krijg je nu terug het 'lege' formulier omdat de opgevraagde inhoud uit de `$_POST` array niet meer bestaat.

```

if ($_POST['btnLeegmaken']=="Annuleren")
{unset($_POST);}

```

4.4 Invoer van diakritische en vreemde tekens

Accentletters, vreemde tekens worden niet altijd netjes weergegeven.

Controleer eerst en vooral in de code van je webpagina's welke tekenset (charset) gebruikt wordt. Mocht dat bv. nog charset iso-8859-1 zijn, vervang dan door onderstaande meta-tag die tekenset UTF-8 aanroept in het `<head>`-deel te plaatsen:

```
<meta charset="utf-8">
```

4.5 Definities

4.5.1 Letters/tekens in een woord of tekenreeks tellen - LETTERS.PHP

In een formulier moet je een letter of teken en een woord of tekenreeks kunnen invullen.

Druk op de knop Resultaat en volgende informatie moet verschijnen:

Letter of teken	e
in woord	onzelieveheersbeestje
Resultaat	

De letter e komt 8 keer voor in het woord onzelieveheersbeestje.

- De letter x komt y keer voor in het woord z. **OF**
- De letter x komt niet voor in het woord z.

Aandacht en extra's

- Bij het openen van de pagina mag enkel het formulier verschijnen.
- Bij het doorsturen van het resultaat moeten de ingevulde waarden in de tekstvakken zichtbaar blijven.
- Controleer of er één teken is ingevoerd in het eerste vak.
- Controleer of er een woord of tekenreeks is ingevoerd.
- Pas als de invoer klopt, verschijnt het resultaat. In de andere gevallen verschijnt een passende foutmelding.

4.5.2 Deelbaar door.. – DEELBAAR.PHP

Als je een geheel getal ingeeft, moet je als resultaat alle getallen krijgen waardoor het ingegeven getal deelbaar is. Zie voorbeeld hieronder met 132 als invoer.

<input type="text"/> is deelbaar door...	132 is deelbaar door... 1, 2, 3, 4, 6, 11, 12, 22, 33, 44, 66 en 132.
Toon resultaat	Toon resultaat

Voer een geheel getal in!

Controleer of de invoer correct is voordat je het resultaat weergeeft.

4.5.3 Geboortedatum voluit weergeven – GEBDATUM.PHP

- Voorzie een tekstvak voor het intypen van de naam.
- Voorzie 3 keuzelijstjes voor het kiezen van de geboortedatum
 - dag: van 1 tot 31
 - maand: de maanden voluit. Maak daarvoor een array met alle maanden.
 - jaar: vanaf 1920 tot 6 jaar voor het actuele jaar. In 2025 bv. is het hoogste jaar 2019.

Mijn naam	<input type="text" value="Jan Janssen"/>		
Mijn geboortedatum	Dag	Maand	Jaar
	4	februari	1930
Resultaat			

Jan Janssen is geboren op 4 februari 1930.
Notatie NBN: 1930-02-04.

- Als resultaat krijg je, zoals je hierboven ziet, de datum met de maand voluit, maar ook de datum volgens de NBN-norm: yyyy-mm-dd.

Extra

- Kan je het aantal dagen dynamisch aanpassen in functie van de gekozen maand. Kies je bv. maart, dan zijn er 31 dagen in de keuzelijst, in april zijn er dat maar 30.
- Februari is een speciaal geval: daar zijn er 28 of 29, afhankelijk of het jaar al dan niet een schrikkeljaar is.
Om te weten of een jaar een schrikkeljaar is, vind je hieronder een Excel-formule die controleert of het jaartal in de cel A1 een schrikkeljaar is.

=ALS(OF(EN(REST(A1;4)=0;REST(A1;100)<>0);REST(A1;400)=0);"schrikkeljaar";"neen")

Je kan natuurlijk ook even opzoeken welke regels bepalen of een jaar al dan niet een schrikkeljaar is.

4.5.4 De tafel van... - TAFELS.PHP

- Voorzie een tekstvak zoals hieronder geïllustreerd waarin een getal kan ingetypt worden.
- Controleer of de invoer een numerieke waarde is zowel via HTML5 als server-side (functie `is_NUMERIC`) en of die waarde tussen 1 en 99 ligt. Zoniet, geef dan de instructie: "Geef een waarde tussen 1 en 99" weer.
- Ligt de ingevoerde waarde tussen 1 en 99, geef dan de tafel van dat getal weer (1 tot 10), zoals hieronder links geïllustreerd.
- Uitbreiding: geef de tafel weer zoals hieronder rechts, met gebruik van DIV-tags. Je mag een embedded stylesheet gebruiken.

De tafel van **34** Resultaat

$1 \times 34 = 34$
 $2 \times 34 = 68$
 $3 \times 34 = 102$
 $4 \times 34 = 136$
 $5 \times 34 = 170$
 $6 \times 34 = 204$
 $7 \times 34 = 238$
 $8 \times 34 = 272$
 $9 \times 34 = 306$
 $10 \times 34 = 340$

De tafel van **150** Resultaat

Geef een waarde tussen 1 en 99.

De tafel van **23** Resultaat

$1 \times 23 = 23$ $6 \times 23 = 138$
 $2 \times 23 = 46$ $7 \times 23 = 161$
 $3 \times 23 = 69$ $8 \times 23 = 184$
 $4 \times 23 = 92$ $9 \times 23 = 207$
 $5 \times 23 = 115$ $10 \times 23 = 230$

4.5.5 Hoeveel kost mijn zending? – PORTOKOSTEN.PHP

Gewicht van de brief (in gram) Posttarief

- Als je een gewicht intypt (minimum 1 en maximum 2000 gram – controleren!), moet je onder het formulier het posttarief te zien krijgen voor een Belgische zending, op basis van de tarieven hieronder, type D+3 stuk (één zegel; levering binnen 3 dagen).
- Uitbreiding: voeg een keuzelijstje toe met de verschillende types zegels en zorg dat je het juiste tarief te zien krijgt volgens de tabel hieronder.

Geef het resultaat als *Het tarief voor deze zending is XXX EUR* onder het formulier weer.

Gewicht van de brief (in gram) - Type zending: D+1 stuk Posttarief

D+1 stuk
D+1 aantal
D+3 stuk
D+3 aantal

Gewicht tussen 1 en 2000 gram invoeren!

Gewicht (gram)	België		België		België	
	Eenheid	D+1 stuk	D+1 aantal	D+3 stuk	D+3 aantal	
0-50	1	€ 2,37	€ 2,33	€ 1,53	€ 1,49	
51-100	2	€ 4,74	€ 4,66	€ 3,06	€ 2,98	
101-350	3	€ 7,11	€ 6,99	€ 4,59	€ 4,47	
351-1000	5	€ 11,85	€ 11,65	€ 7,65	€ 7,45	
1000-2000	7	€ 16,59	€ 16,31	€ 10,71	€ 10,43	

4.5.6 Vermenigvuldigingstabel – TAFELS.PHP

Maak een pagina die de vermenigvuldigingstabel (tafels tot 10) weergeeft. De bovenste rij en de meest linkse kolom bevatten de cijfers van 1 tot 10 en worden vetjes weergeven.

Een aanwezige klasse *randen* zorgt ervoor dat de eigenlijke berekende waarden omrand worden en rechts uitgelijnd zijn.

Uitbreiding: geef de oneven tafels (rijen) in witte letters op een blauwe achtergrond weer (klasse *blauwerij*). Even tafels geef je in blauwe letters op witte achtergrond (klasse *witterij*) weer.

0	1	2	3	4	5	6	7	8	9	10
1	1	2	3	4	5	6	7	8	9	10
2	2	4	6	8	10	12	14	16	18	20
3	3	6	9	12	15	18	21	24	27	30
4	4	8	12	16	20	24	28	32	36	40
5	5	10	15	20	25	30	35	40	45	50
6	6	12	18	24	30	36	42	48	54	60
7	7	14	21	28	35	42	49	56	63	70
8	8	16	24	32	40	48	56	64	72	80
9	9	18	27	36	45	54	63	72	81	90
10	10	20	30	40	50	60	70	80	90	100

4.5.7 Fibonacci – FIBONACCI.PHP

Geef in deze pagina via een tabel de *reeks van Fibonacci* weer (de eerste 50 waarden). Na het uitschrijven van 10 getallen wordt telkens een nieuwe rij genomen.

In de reeks van Fibonacci is elke getal van de reeks steeds de som van de twee voorgaande getallen, beginnend met 0 en 1.

1	2	3	5	8	13	21	34	55	89
144	233	377	610	987	1597	2584	4181	6765	10946
17711	28657	46368	75025	121393	196418	317811	514229	832040	1346269
2178309	3524578	5702887	9227465	14930352	24157817	39088169	63245986	102334155	165580141
267914296	433494437	701408733	1134903170	1836311903	2971215073	4807526976	7778742049	12586269025	20365011074

4.5.8 BMI – BMI.PHP

In deze toepassing bereken je de Body Mass Index op basis van gewicht (in kg) en lengte (in meter). Als je beide gegevens invoert in het formulier, krijg je het resultaat te zien en een interpretatie van het resultaat op basis van de vereenvoudigde tabel hieronder die geen onderscheid maakt tussen geslacht, leeftijd...:

Kleiner dan 18.5	Ondergewicht	Vanaf 26.9 tot 29.9	Matig overgewicht
Vanaf 18.5 tot 24.9	Normaal gewicht	Vanaf 29.9 tot 40	Ernstig overgewicht
Vanaf 24.9 tot 26.9	Licht overgewicht	40 en meer	Ziekelijk overgewicht

Je berekent de BMI door het gewicht (kg) te delen door het kwadraat van de lengte (m).

Bereken je Body Mass Index (BMI)
Mijn gewicht: <input type="text"/> (in kg) -- decimaal teken = punt
Mijn lengte: <input type="text"/> (in m) -- decimaal teken = punt
<input type="button" value="Bereken je BMI"/>
Je BMI is 23.1.
Je hebt een normaal gewicht.

Bij de weergave van het resultaat, verdwijnt het formulier.

Extra 1: controleer of gewicht en lengte goed ingevuld zijn. Indien niet, geef een boodschap weer naast de knop. Gebruik de klasse *opvallend* voor de weergave.

Extra 2: zorg voor een correcte berekening als bij gewicht en/of lengte toch een komma gebruikt is als decimaal teken. **Tip:** de functie *str_replace* kan helpen.

Extra 3: zorg dat de BMI op 1 cijfer na de komma afgerond is.

Bereken je Body Mass Index (BMI)
Mijn gewicht: <input type="text"/> (in kg) -- decimaal teken = punt
Mijn lengte: <input type="text"/> (in m) -- decimaal teken = punt
<input type="button" value="Bereken je BMI"/>
Gewicht ontbreekt! Lengte is niet ingevuld!

4.5.9 Foto's tonen via keuzelijst – FOTOS.PHP

The left screenshot shows a dropdown menu titled 'Leerling' with a list of student names. 'Daan Hautekeete' is selected. Below the menu is a placeholder image of a person. The right screenshot shows the same interface after selecting 'Nico Van Hevel'. It displays a large blue silhouette of a person's head with a white question mark in the center, indicating that no photo was found for that name.

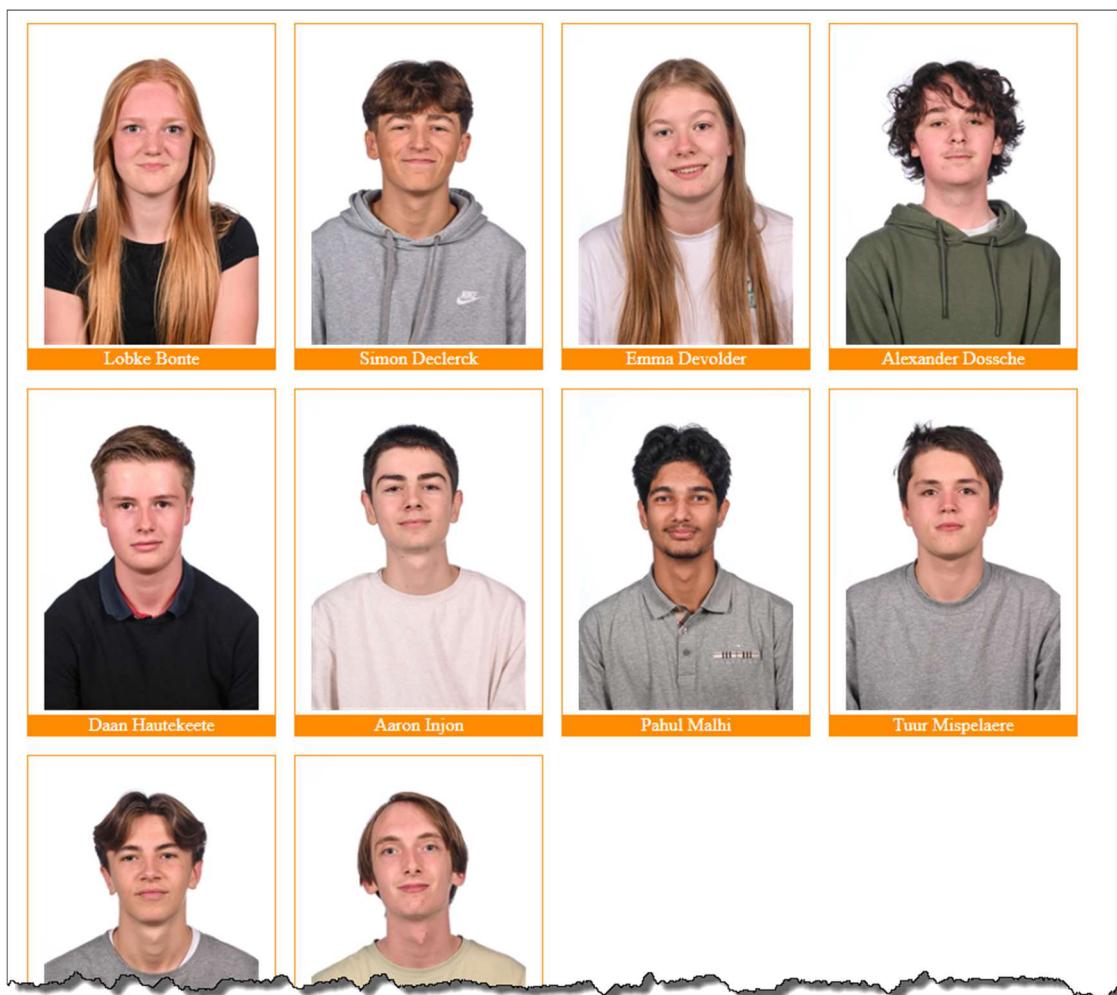
Zorg voor een keuzelijstje op basis van de leerlingenarray met leerlingengegevens. Als geen foto beschikbaar is, laat je de foto UNKNOWN.JPG zien.

Tip: met de functie `file_exists("fotosSJZ/foto.jpg")` kan je nagaan of een opgegeven extern bestand op de opgegeven plaats al dan niet beschikbaar is (true/false).

4.5.10 Foto's tonen in overzicht – FOTOLIJST.PHP

Maak een overzichtslijst van de foto's naar onderstaand model op basis van een array met volgende structuur:

```
$leerlingen=array ("Lobke Bonte"=>"BonteLobke", "Simon Declerck"=>"DeclerckSimon", "Emma Devolder"=>"DevolderEmma", "Alexander Dossche"=>"DosscheAlexander", "Daan Hautekeete"=> "HautekeeteDaan" ...
```



4.5.11 Euro's – EURO.PHP

In een eenvoudig formuliertje typ je een bedrag in. Het ingetypte bedrag wordt omgezet in aantal biljetten en munten dat nodig is om het bedrag uit te betalen, naar onderstaand model:

€ 1688,93	Biljetten en munten			
				
3 x 500 euro	1 x 100 euro	1 x 50 euro	1 x 20 euro	
1 x 10 euro	1 x 5 euro	1 x 2 euro	1 x 1 euro	2 x 50 cent
1 x 20 cent	1 x 10 cent			

Houd bij de uitwerking rekening met onderstaande informatie:

- de bedragen liggen tussen 0.01 en 5000 EUR, met maximum 2 cijfers na de komma.
Controleer de ingetypte waarde vóór het formulier effectief verstuurd wordt.
- In ons land wordt een komma als scheiding tussen eenheden/decimalen gebruikt, terwijl PHP een punt als scheidingsteken gebruikt. Vang dit op.
- Het ingetypte bedrag blijft in het tekstvak zichtbaar na het versturen van het formulier.
- Voor elk biljet/munt gebruik je de div *eenheid* met daarin een div *aantal* voor de weergave van het aantal biljetten/munten en een div *foto* voor de weergave van de illustratie van het biljet of de munt. De opmaak is beschikbaar in een embedded CSS-stylesheet.
- de bestandsnamen van de illustraties komen met de waarde van het biljet/muntstuk overeen, behalve bij de centen, bv. 500.jpg, 10.jpg, 2.jpg, 1.jpg. Bij de centen is het decimale punt niet in de bestandsnaam opgenomen, 0.50 EUR is dus 050.jpg en 0.02 EUR is 002.jpg.

Variant

Geef het effectief aantal biljetten/ muntstukken weer om dit bedrag uit te betalen. Bij 1688,93 EUR krijg je dus effectief 3 briefjes van 500 EUR en 2 stukken van 0.20 EUR te zien.

€ 1688,93	Biljetten en munten							
								
3 x 500 euro	1 x 100 euro	1 x 50 euro	1 x 20 euro					
2 x 500 euro	1 x 100 euro	1 x 50 euro	1 x 20 euro	1 x 10 euro	1 x 5 euro	1 x 2 euro	1 x 1 euro	2 x 50 cent
3 x 500 euro	1 x 100 euro	1 x 50 euro	2 x 20 euro	1 x 10 euro	1 x 5 euro	1 x 2 euro	1 x 1 euro	2 x 50 cent
3 x 500 euro	1 x 100 euro	1 x 50 euro	2 x 20 euro	1 x 10 euro	1 x 5 euro	1 x 2 euro	1 x 1 euro	2 x 50 cent

5 MySQL databank

In dynamische websites wordt heel vaak gebruik gemaakt van een achterliggende database waarin gegevens opgeslagen worden en waaruit gegevens weergegeven worden.

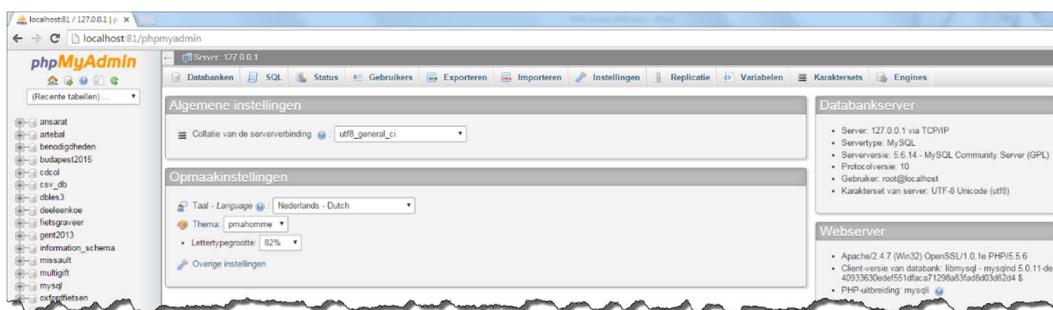
We maken gebruik van **MySQL** als relationele database. De installatie ervan is gebeurd bij het installeren van XAMPP waar de M voor MySQL staat.

Via de webbrowser leren we gegevens invoeren, wijzigen, raadplegen en indien nodig ook weer verwijderen. Maar eerst maken we kennis met **PHPMyAdmin**, een tool waarmee je je MySQL omgeving kunt beheren.

Door PHPMyAdmin te gebruiken, hoef je zelf geen SQL-code te schrijven.

5.1 PHPMyAdmin

Je start je PHPMyAdmin-interface via <http://localhost/PHPMyAdmin>. Links krijg je een overzicht van de aanwezige database(s).

The screenshot shows the main configuration page of PHPMyAdmin. On the left, there's a sidebar with a tree view of databases: 'arsarat', 'arbeal', 'benodigdhen', 'budapest2015', 'cdcol', 'csv_db', 'deales', 'deals', 'deals_voorhoe', 'fotograver', 'print2013', 'information_schema', 'missault', 'multigift', 'mysql', and 'ontdeksters'. The main panel has several sections: 'Algemene instellingen' (General settings) with 'Collate van de serververbinding' set to 'utf8_general_ci'; 'Opmaakininstellingen' (Formatting settings) with 'Taal - Language' set to 'Nederlands - Dutch' and 'Lettertypegrottte' at '82%'; 'Databanksserver' (Database server) listing 'Server: 127.0.0.1 via TCP/IP', 'Servertype: MySQL', 'Serverversie: 5.5.14 - MySQL Community Server (GPL)', 'Protocollivierse: 10', 'Gebruiker: root@localhost', and 'Karacterset van server: UTF-8 Unicode (utf8)'; and 'Webserver' (Web server) listing 'Apache/2.4.7 (Ubuntu) OpenSSL/1.0.2e PHP/5.5.6', 'Client versie van standaard: libmysql - myzend 5.5.11-dev 409336300aef551dfe47328a53faab0d3a623d44', and 'PHP-uitbreiding: mysqli'.

5.2 Database en tabellen aanmaken

Om een nieuwe database toe te voegen, geef je de naam op bij *Nieuwe database aanmaken* en klik je op de knop **Aanmaken**.

De SQL-instructie die hier uitgevoerd wordt is bv.:

```
CREATE DATABASE 'test1' ;
```

Je krijgt de uitgevoerde SQL-instructie ook te zien.

Na het maken van de database, moet je natuurlijk één of meer tabellen eraan toevoegen:

A screenshot of the 'Tabel aanmaken' (Create Table) form. It has a title bar 'Tabel aanmaken'. Below it, there are two input fields: 'Naam:' and 'Aantal kolommen:'. Both fields have a placeholder value of '1'.

Van zodra je de database links selecteert, kan je een nieuwe tabel toevoegen in het tabblad **Structuur**. Geef de naam van je tabel en het aantal voorziene velden (*aantal kolommen*). Uiteraard kan je dat aantal later nog aanpassen.

Bij het klikken op **Start** kan je nu de gegevens per veld ingeven: naam, type, lengte...

Databanken

A screenshot of the 'Databanken' (Databases) page. It shows a table with one row. The first column contains a small icon, the second column contains the database name 'test1', and the third column contains a link labeled 'Aanmaken'.

Neem nu dat je de velden ID, voornaam, familienaam en gemeente voorziet. Voor voornaam, familienaam en gemeente, kan je het algemene type varchar behouden. Hier moet je wel een maximale veldlengte ingeven.

Voor de ID kan je eventueel een INT kiezen (integer). Door voor de optie 'AI' (auto increment) te kiezen, wordt de ID telkens automatisch ingevuld. Vergeet niet op te slaan!

5.3 Weergave van accentletters en vreemde tekens - de collatie

Wil je accentletters correct weergeven, dan zijn er heel wat aandachtspunten. Hoger hadden we het reeds over de karakterset die in de webpagina gebruikt wordt om gegevens weer te geven: bij voorkeur UTF-8.

Bij het aanmaken van een tabel in PHPMyAdmin kan je zowel voor de databank als geheel als voor individuele velden in een tabel een *collatie* instellen. Zonder hier dieper op in te gaan onthoud je dat de collatie heel belangrijk is voor de juiste weergave van accentletters en voor de wijze waarop gesorteerd en geselecteerd wordt. De collatie **utf8_unicode_ci** is zeker bruikbaar in velden waarin accentletters voorkomen.

In de actie-zone van het veldoverzicht kan je later de velden nog wijzigen of verwijderen, maar ook een primaire sleutel instellen, bijvoorbeeld op het LoginID-veld.

Veld	Type	Collatie	Attributen	Null	Standaardwaarde	Extra	Actie
LoginID	int(3)			Nee	Geen		
Familienaam	varchar(30)	utf8_unicode_ci		Nee	Geen		
Voornaam	varchar(25)	utf8_unicode_ci		Nee	Geen		

Via het tabblad **Invoegen** kan je records aan je tabel toevoegen. Dat verloopt niet zo gebruikersvriendelijk, maar het is uiteraard de bedoeling gegevens via de webinterface (webformulier) toe te voegen en te beheren en niet rechtstreeks in de databank te werken.

Selecteer je een tabel, dan krijg je dus de tabbladen voor tabellen te zien, o.a.:

- **Verkennen:** inhoud van de tabel bekijken en bewerken.
- **Structuur:** structuur van de tabel bekijken en bewerken.
- **Invoegen:** nieuwe records invoegen...

5.4 Relaties tussen tabellen

Als je je database selecteert, beschik je over een tabblad **Designer/Ontwerper**. In dit venster leg je relaties tussen tabellen die je eerst met het pijltje linksboven openklapt. Gebruik

daarvoor de vierde knop *Create relation/Maak relatie*. Klik in je mastertabel op het relatieveld waarop een primaire sleutel moet ingesteld worden (*Referenced key*) en daarna op het afhankelijke veld in de andere tabel (*Foreign key*) en bevestig de relatie.



Kies je bij on delete/on update voor **cascade**, dan worden wijzigingen in de mastertabel automatisch aangepast in de child tabel. Verwijder je in zo'n geval bv. een klant uit de tabel *tblklanten*, dan worden alle orders van die klant in *tblorders* automatisch ook verwijderd.

Let op: om een relatie te kunnen leggen, moet het veld dat als Foreign key dienst doet, geïndexeerd zijn:

#	Column	Type	Collation	Attributes	Null	Default	Extra	Action
1	orderID	varchar(8)	latin1_swedish_ci		No	None		Change Drop More ▾
2	klantID	varchar(8)	latin1_swedish_ci		No	None		Change Drop More ▾
3	orderDatum	date			No	None		Change Drop More ▾

Check All / Uncheck All With selected: [Browse](#) [Change](#) [Drop](#) [Primary](#) [Unique](#) [Index](#)

[Print view](#) [Relation view](#) [Propose table structure](#) [Track table](#)

Add 1 column(s) At End of Table At Beginning of Table After orderID Go

Indexes:

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Drop	PRIMARY	BTREE	Yes	No	orderID	4	A		
Edit Drop	klantID	BTREE	No	No	klantID	4	A		

5.5 Connectie met de database

Om toegang tot je database te hebben, moet je over verschillende informatie beschikken:

- de naam van uw *host* waar de database ondergebracht is. Standaard: **localhost**.
- je **gebruikersnaam** (standaard is dat **root**)
- je **wachtwoord** (standaard is er geen wachtwoord).
- de naam van de **database** die je wil benaderen.

Voor de connectie met onze databank gebruik je tegenwoordig best de PHP **MySQLi** class; de *i* staat voor *improved*.

Wil je accentletters en andere vreemde tekens uit je databank netjes weergeven, voeg dan ook de regel `$db->query("SET character_set_results=utf8")` toe.

Met gebruik van enkele variabelen kan de connectie er als volgt uitzien, inclusief foutmelding bij niet-geslaagde connectie:

```
$dbhost="localhost";
$dbuser="root";
$dbpaswoord="";
$dbdatabase="missault";
$db = new mysqli($dbhost, $dbuser, $dbpaswoord, $dbdatabase) or die
("<p>connectie niet geslaagd!</p>");
$db->query("SET character_set_results=utf8");
```

Omdat je die database vanuit verschillende webpagina's moet kunnen benaderen, maak je van je connectie best een afzonderlijk bestand, bv. CONNECTION.PHP dat je dan in de verschillende webpagina's met een **include()** statement insluit. Zo kunnen aanpassingen in de verbinding snel en centraal gebeuren, als dat nodig is.

5.6 Gegevenstypes in MySQL

Gegevenstype	
tinyINT	geheel getal, tussen -127 en 127
smallINT	geheel getal, tussen -32767 en 32767
mediumINT	geheel getal, tussen – 8388607 en 8388607
INT, Integer	geheel getal, tussen -2147483647 en 2147483647
DOUBLE, FLOAT	decimaal getal
date	datum, opgeslagen in YYYY-MM-DD formaat
datetime	datum en tijd
char	veld met vaste breedte, tot 255 tekens
varchar	veld met variabele breedte, tot 255 tekens
tinytext	tot 255 tekens
text	tot 65535 tekens
mediumtext	tot 16 777 215 tekens
longtext	tot 4 GB

5.7 Enkele basis SQL-instructies

Wil je nu echt met gegevens uit de database werken, dan kan je niet om een aantal belangrijke SQL-instructies heen. We beperken ons tot de basisfuncties waar we al heel wat mee kunnen doen, ook wel als CRUD aangeduid: Create, Read, Update en Delete

Alles draait rond vier SQL-instructies voor de vier basisbewerkingen:

- SELECT: om gegevens uit de database te selecteren
- INSERT: om gegevens aan de database toe te voegen
- UPDATE: om gegevens in de database te wijzigen
- DELETE: om gegevens uit de database te verwijderen.

Hieronder vind je bij wijze van voorbeeld enkele eenvoudige SQL-commando's:

- Alle velden uit de tabel tblKlanten opvragen.

```
SELECT * FROM tblKlanten
```

- De velden familienaam, woonplaats en telefoon uit de tabel tblKlanten opvragen

```
SELECT familienaam, woonplaats, telefoon FROM tblKlanten
```

- De velden voornaam en fnaam worden aaneengeplakt (geconcateneerd, met spaties ertussen) als totnaam (vrij te kiezen naam) en samen met de woonplaats uit de tabel tblKlanten opgevraagd.

```
SELECT concat(voornaam, ' ', fnaam) as totnaam, woonplaats FROM  
tblKlanten
```

- Familienaam en woonplaats uit de tabel tblKlanten opvragen van de klanten uit Gent:

```
SELECT familienaam, woonplaats FROM tblKlanten WHERE woonplaats='Gent'
```

- Familienaam en woonplaats uit de tabel tblKlanten opvragen van de klanten uit Gent en Brugge:

```
SELECT familienaam, woonplaats FROM tblKlanten WHERE woonplaats='Gent'
OR woonplaats='Brugge'
```

- Elke klant waarbij de tekenreeks Walle in het veld FNAAM voorkomt opvragen:

```
SELECT * from tblKlanten WHERE fnaam Like '%Walle%'
```

Hier worden dus o.m. de klanten Wallein, Vandewalle, Van de Walle, Van Wallegem... geselecteerd.

- Een alfabetische naamlijst van alle Nederlandstalige klanten die in Gent wonen:

```
SELECT familienaam, woonplaats FROM tblKlanten WHERE woonplaats='Gent'
AND taal='NL' ORDER BY familienaam
```

- Met het sleutelwoord DISTINCT haal je enkel unieke waarden uit een veld op. In onderstaand voorbeeld selecteer je elke klas slechts één keer om bv. een klaslijst te maken:

```
SELECT DISTINCT klas from tblLeerlingen order by klas
```

- Tellen van aantal records in de tabel tblLeerlingen:

```
SELECT COUNT(*) from tblLeerlingen
```

- Een nieuwe klant Janssens uit Aalter met telefoonnummer 09 230 14 22 toevoegen:

```
INSERT INTO tblKlanten (Familienaam, Woonplaats, Telefoon) VALUES
('Janssens', 'Aalter', '09 230 14 22')
```

- De klant met klantID 436 – numeriek veld – verwijderen

```
DELETE FROM tblKlanten WHERE klantID=436
```

Let op: je merkt dat de inhoud van een numeriek veld niet tussen aanhalingsstekens verschijnt. Gebruik bij alfanumerieke velden altijd aanhalingsstekens.

- Het telefoonnummer van klant Janssens die klantID 518 heeft wijzigen in 09 230 22 14

```
UPDATE tblKlanten SET telefoon='09 230 22 14' WHERE klantID=518
```

Met de basisvoorbeelden hierboven komen we dus al een heel eind.

5.8 Gegevens raadplegen en weergeven

Eens je connectie hebt met je database, gebruik je de query-methode van je object om SQL-opdrachten uit te voeren:

```
$result = $db->query($sql)
```

Het resultaat van die opdracht, sla je in een variabele op, in het voorbeeld hierboven de variabele \$result.

Via een IF-statement kan je de instructie stoppen in geval van fouten. De **die()** functie zorgt voor een foutmelding en stopt de uitvoer.

Vandaar dat we het uitvoeren van een query best als volgt aanvullen:

```
if (!$result = $db->query($sql)) {
    die('<p>fout in query!</p>');
}
```

Als de query niet resulteert (! = not) in een recordset \$RESULT, dan is er iets fout glopen bij de uitvoer ervan en melden we dit.

Met de opdracht **fetch_assoc()** vraag je de gegevens uit je querybron rij per rij op. De gegevens van een record worden telkens in een associative array opgeslagen. In onderstaand voorbeeld vraag je de gegevens per rij in een array \$row op en geef je telkens voornaam en familienaam weer.

```
$sql="SELECT * FROM tblVertegenwoordigers";
if(!$result = $db->query($sql)) {
    die('<p>fout in query!</p>');
}
while($row= $result->fetch_assoc())
{
    echo $row["voornaam"]." ".$row["familienaam"]."<br />";
}
?>
```

Na de laatste rij stopt het doorlopen automatisch.

Het aantal rijen – records – kan je opvragen met de opdracht **num_rows**.

```
$aantalrijen = $result->num_rows;
```

Heb je je resultatenset niet langer nodig, maak die dan 'vrij'.

```
$result->free()
```

Ben je helemaal klaar met het gebruik van je databank, sluit dan netjes de connectie:

```
$db->close()
```

Werk je liever met de genummerde index, gebruik dan **fetch_row()** i.p.v. **fetch_assoc()**:

```
while($row= $result->fetch_row())
{
    echo $row[0]." ".$row[1]."<br />";
}
```

In het voorbeeld hierboven krijg je een lijstje met telkens het eerste en twee veld uit de array.

5.9 SQL joins: gegevens uit verschillende tabellen weergeven

Om gegevens uit twee of meer tabellen weer te geven, gebruiken we **joins** tussen de tabellen. We leggen de verschillende types joins uit aan de hand van een praktisch voorbeeld met een klantentabel (tblklanten; klantID is de primaire sleutel) en een ordertabel (tblorders, orderID is de primaire sleutel).

Bemerkt dat er klanten zijn die nog niets besteld hebben (KL1001, KL1004) en orders die komen van klanten die niet in de klantentabel zitten (ORD9002 en ORD9004). Dat laatste moeten we uiteraard vermijden!

tblorders		
orderID	klantID	orderDatum
ORD9001	KL1002	2014-02-13
ORD9002	KL1008	2014-03-20
ORD9003	KL1003	2014-04-15
ORD9004	KL1008	2014-04-28

tblklanten		
klantID	klantNaam	klantGemeente
KL1001	Janssens BVBA	Jabbeke
KL1002	Gheeraert NV	Gent
KL1003	Pauwels	Paal
KL1004	Thiel	Tienen

5.9.1 Inner join

De verbinding tussen beide tabellen gebeurt natuurlijk via de klantID.

```
$sql = "SELECT tblorders.orderID, tblorders.klantID,
tblorders.orderDatum, tblklanten.klantID, tblklanten.klantNaam,
tblklanten.klantGemeente
FROM tblorders
INNER JOIN tblklanten
ON tblorders.klantID=tblklanten.klantID
ORDER BY tblorders.orderID";
```

orderID	klantID	orderDatum	klantID	klantNaam	klantGemeente
ORD9001	KL1002	2014-02-13	KL1002	Gheeraert NV	Gent
ORD9003	KL1003	2014-04-15	KL1003	Pauwels	Paal

Bij de **inner join** krijg je enkel de rijen te zien waar de join-voorwaarde klopt. Concreet zijn dat alle orders die verwijzen naar een bestaande klant.

5.9.2 Left join

Met een **left join** – of **left outer join** – krijg je alle rijen uit de linkse tabel, de eerste tabel, te zien. Zijn er geen overeenkomende gegevens in de rechtse tabel, dan blijven die kolommen leeg, zoals dat het geval is bij ORD9002 en ORD9004.

```
$sql = "SELECT tblorders.orderID, tblorders.klantID,
tblorders.orderDatum, tblklanten.klantID, tblklanten.klantNaam,
tblklanten.klantGemeente
FROM tblorders
LEFT JOIN tblklanten
ON tblorders.klantID=tblklanten.klantID
ORDER BY tblorders.orderID";
```

orderID	klantID	orderDatum	klantID	klantNaam	klantGemeente
ORD9001	KL1002	2014-02-13	KL1002	Gheeraert NV	Gent
ORD9002	KL1008	2014-03-20			
ORD9003	KL1003	2014-04-15	KL1003	Pauwels	Paal
ORD9004	KL1008	2014-04-28			

5.9.3 Right join

Met een **right join** – of **right outer join** – krijg je alle rijen uit de rechtse tabel, de tweede tabel dus, te zien. Zijn er geen overeenkomende gegevens in de linkse tabel, dan blijven die kolommen leeg, zoals dat het geval is bij KL1001 en KL1004: die hebben nog niets besteld. In het voorbeeld hebben we de rijen op klantID gerangschikt.

```
$sql = "SELECT tblorders.orderID, tblorders.klantID,
tblorders.orderDatum, tblklanten.klantID, tblklanten.klantNaam,
tblklanten.klantGemeente
FROM tblorders
RIGHT JOIN tblklanten
ON tblorders.klantID=tblklanten.klantID
ORDER BY tblklanten.klantID";
```

orderID	klantID	orderDatum	klantID	klantNaam	klantGemeente
			KL1001	Janssens BVBA	Jabbeke
ORD9001	KL1002	2014-02-13	KL1002	Gheeraert NV	Gent
ORD9003	KL1003	2014-04-15	KL1003	Pauwels	Paal
			KL1004	Thiel	Tienen

5.9.4 Gegevens weergeven

De gegevens van de SQL-statements hierboven kunnen als volgt in tabelvorm weergegeven worden. Via CSS werden de tabelcellen omrand.

```
if (!$result = $db->query($sql)) {die ('<p>fout!</p>');}
echo "<table>";
//rij met veldnamen
$columns = $result->field_count;
echo "<tr>";
while ($fieldinfo=$result->fetch_field())
{
    echo "<td>". $fieldinfo->name."</td>";
}
echo "</tr>";

//gegevensrijen
while ($row = $result->fetch_row())
{
    echo "<tr>";
    for ($i=0;$i<$columns;$i++)
    {
        echo "<td>".$row[$i]."</td>";
    }
    echo "</tr>";
}
echo "</table>";
```

5.10 Gegevens invoeren

Met het kleine formuliertje hieronder proberen we gegevens aan de database *missault* toe te voegen in de tabel *tblVertegenwoordigers*.

Naam	<input type="text"/>
Voornaam	<input type="text"/>
Gemeente	<input type="text"/>
<input type="button" value="Toevoegen"/>	

De PHP-code hieronder mag pas uitgevoerd worden als je op de knop *Toevoegen* klikt. Hiervoor slaan we de waarde die door de knop *Toevoegen* doorgestuurd wordt op in een variabele (*\$actie*). We schrijven de gegevens pas in de database als de waarde 'Toevoegen' van de knop – eigenschap *value* - in *\$actie* is opgeslagen en testen daar dus op:

```
$actie= $_POST['btnToevoegen'];
if ($actie=="Toevoegen") {
    ...
}
```

Onmiddellijk daarna volgt een nieuwe controle: geen enkele van de drie tekstvakken mag leeg zijn. Let op: die controle kan je natuurlijk – en eigenlijk beter – client side doen via het HTML5 **required** attribuut in je tekstvakken!

```
if (!empty($_POST['txtVoornaam']) && !empty($_POST['txtFamilienaam']) && !empty($_POST['txtGemeente'])) {
```

Als dat wel het geval is, geven we een boodschap weer. De totale code ziet er als volgt uit:

```
$actie= $_POST['btnToevoegen'];
if ($actie=="Toevoegen")
{
if (!empty($_POST['txtVoornaam']) && !empty($_POST['txtFamilienaam']) && !empty($_POST['txtGemeente']))
{
    $voornaam = $_POST['txtVoornaam'];
    $familienaam = $_POST['txtFamilienaam'];
    $gemeente = $_POST['txtGemeente'];
    $sql="INSERT INTO tblVertegenwoordigers (voornaam, familienaam, gemeente) VALUES ('$voornaam', '$familienaam', '$gemeente')";
    $db->query($sql) or die('<p>fout!</p>');
    echo"$voornaam $familienaam uit $gemeente is toegevoegd.";
}
else
{
    echo "Voornaam, familienaam en gemeente moeten alle drie ingevuld zijn!";
}
}
```

In plaats van de **die()** functie kan je ook onderstaand statement gebruiken. Als de SQL-instructie niet correct uitgevoerd wordt, krijg je nu ook een indicatie van het soort fout (**\$db->error**) te lezen.

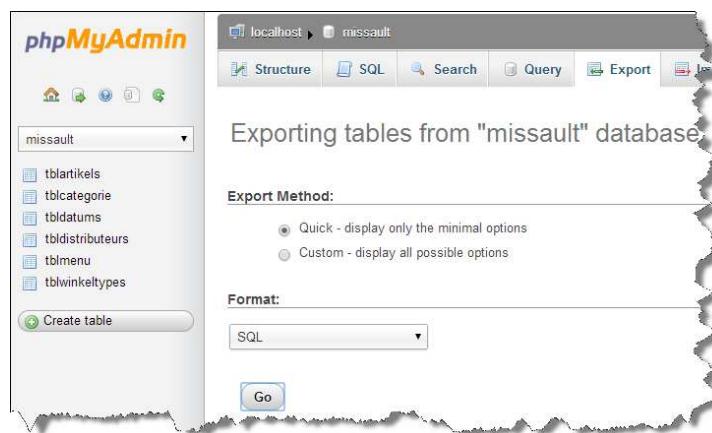
```
if ($db->query($sql) === TRUE) {
    echo "Nieuw record succesvol toegevoegd.";
} else {
    echo "Fout: " . $sql . "<br>" . $db->error;
}
```

5.11 Gegevens uitwisselen tussen MySQL databanken

Gegevens uitwisselen tussen twee MySQL databanken kan je doen door ze uit de ene databank te exporteren en in een andere databank te importeren. Dat kan zowel op het niveau van de volledige databank als op het niveau van één of meer geselecteerde tabellen gebeuren.

Ga daarbij als volgt te werk:

- open de brondatabank en kies het tabblad Export/Exporteren.



Via de Custom/Uitgebreid export method kan je eventueel één of meer tabellen selecteren.

- kies als format SQL (standaard). Bemerk dat je ook kunt exporteren naar Excel, Word...
- klik op Go/Starten om de gegevens te exporteren. Er wordt een sql-bestand geëxporteerd.

Om de gegevens te importeren open je de databank waarin geïmporteerd moet worden:

- kies het tabblad *Import*, blader naar het sql-bestand en importeer de gegevens.

Bemerk dat je ook CSV-bestanden kunt importeren (Format > CSV) zoals hieronder toegepast.

5.12 Gegevens importeren uit Excel

Vanuit Excel

Om gegevens vanuit Excel in een MySQL-databank te importeren, sla je de gegevens vanuit Excel vooraf als **CSV-bestand** op (Bestand > Opslaan als > CSV (gescheiden door lijstscheidingstekens) (*.csv)). Uiteraard zorg je voor een goed gestructureerd bestand dat uit aaneensluitende rijen en kolommen bestaat, de eerste kolom liefst met veldnamen.

Aandachtspunt: zorg er bij numerieke gegevens met decimalen voor dat er geen spaties tussen duizendtallen staan en gebruik een punt als scheiding eenheden/decimalen!

Zo'n CSV-bestand is een gewoon tekstbestand dat je zelfs met de kladblok kunt openen. Elk record is één regel. De veldinhoud wordt telkens gescheiden door een ;-teken.

	A	B	C	D	E	F
1	ID	Klas	Nr	Vnaam	Fnaam	
2	6BI101	6BI1	1	Danté	De Metsenaere	
3	6BI102	6BI1	2	Nicolas	Demeyer	
4	6BI103	6BI1	3	Aileen	De Roo	
5	6BI104	6BI1	4	Shannon	Lescrauwet	
6	6BI105	6BI1	5	Josephine	Leterme	

Vanuit phpMyAdmin

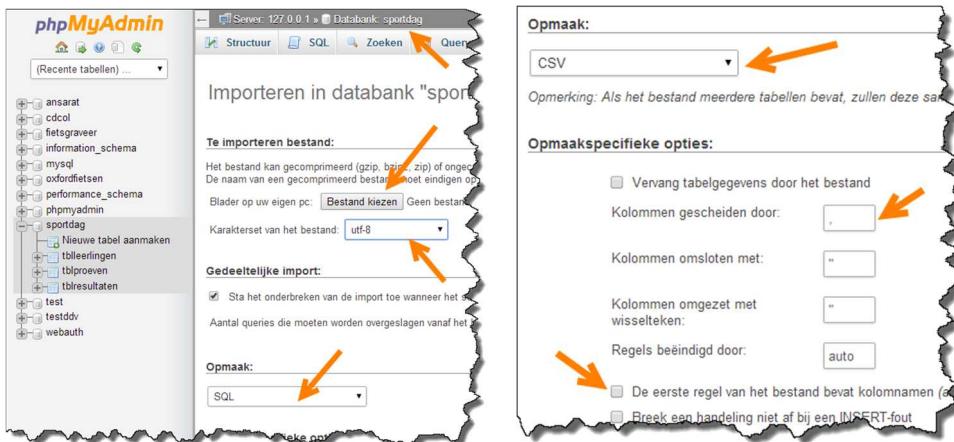
Je kunt de CSV-gegevens:

- als nieuwe tabel binnen een bestaande databank importeren. Open gewoon de databank.
- of binnen een bestaande tabel importeren. De gegevens worden aan de tabel toegevoegd, al dan niet met verwijdering van de reeds aanwezige gegevens. Uiteraard moeten de velden in de databank matchen met de kolommen in het Excel-bestand.

In beide gevallen kies je voor **Importeren** en kies je het betreffende CSV-bestand. Automatisch wordt bij Opmaak voor CSV gekozen.

Kies eventueel als karakterset voor bv. **iso-8859-15** of **windows-1256/1257** i.p.v. UTF-8 om te vermijden dat accentletters niet correct geïmporteerd worden.

Bij de opties CSV kan je het kolomscheidingsteken opgeven: een puntkomma dus. Vink ook *De eerste regel van het bestand bevat kolomnamen* aan als je in een nieuwe tabel importeert en de eerste rij effectief de veldnamen bevat.



Importeer je in een nieuwe tabel, dan krijgt die tabel de naam *table x*. Via Handelingen > Tabelopties > Tabel hernoemen naar, pas je de tabelnaam aan.

Afhankelijk van het soort informatie, zal je ook nog de structuur van de tabel (veldtypes) moeten aanpassen en instellen dat de ID unieke waarden bevat en/of de primaire sleutel is. Dit is nodig om de gegevens in je tabel binnen phpMyAdmin aan te passen of te verwijderen.

Accentletters

Je kunt ook je Excel-gegevens in **Notepad++** (gratis te downloaden) openen en daar via het menu Codering > Naar UTF8 (zonder BOM) converteren en daarna opslaan. Dit helpt zeker om mogelijke problemen met accentletters op te vangen. Controleer uiteraard grondig!

5.13 Gegevens via PHP exporteren in Excel

Je gegevens uit een databank vanuit je webpagina rechtstreeks in bv. Excel openen? Je vindt verschillende oplossingen op het internet, maar vaak is de export van accentletters een probleem. Met onderstaande code, die je gemakkelijk kunt kopiëren aanpassen, zijn er alvast geen problemen wat accentletters betreft.

We gebruiken een hulpbestand van Vitaliy N. Kravchenko (excel.inc.php) dat je in je website moet opnemen en in je webpagina insluiten met de include-instructie:

```
include("excel.inc.php");
```

Je vindt het bestand zeker via het internet. Het bestand bevat een reeks functie waarvan we gebruik maken.

Onderstaande code – die je kunt aanpassen natuurlijk – zorgt voor de export van een volledige MySQL-tabel naar Excel.

```

001 $sql= "select * from tblklanten";
002 if(!$result = $db->query($sql)){die('<p>fout!</p>');}
003 $columns = $result->field_count;
004 while ($fieldinfo = $result->fetch_field()) {
005     $arrayvelden[] = $fieldinfo->name;
006 }
007 //hieronder volgt de export naar Excel
008 $xlsRow = 1;
009 $xls = new Excel(true);
010 $row = $result->fetch_row()
011 for ($i=0;$i<$columns;$i++)

```

```

012     {
013         $xls->WriteLabel(0,$i,$arrayvelden[$i]);
014     }
015     while($row=$result->fetch_assoc())
016     {
017         for ($i=0;$i<$columns;$i++)
018         {
019             $xls->WriteLabel($xlsRow,$i,$row[$arrayvelden[$i]]);
020         }
021         $xlsRow++;
022     }
023     $xls->SendFile('klanten.xls');
024     exit();

```

Bij het voorbereiden van de export, gebruiken we onderstaande functies

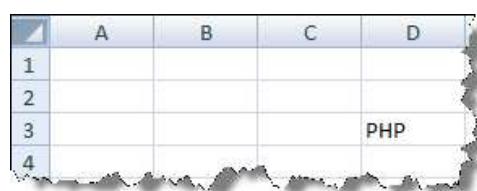
Functie	Omschrijving
field_count	geeft het aantal velden in een recordset weer.
fetch_field()	haalt een veld uit de recordset als object.

Een woordje uitleg bij de code hierboven

Regel	Toelichting
003	Met de eigenschap field_count tellen we het aantal velden in de recordset en slaan die waarde op in de variabele \$columns.
004-006	In deze lus worden de veldnamen uit de recordset één voor één aan de array \$arrayvelden toegevoegd.
011-014	In deze lus worden de veldnamen één voor één in de eerste rij van de spreadsheet geplaatst. De kern in de code is de functie WriteLabel waarmee je een bepaalde waarde in een cel van je spreadsheet kunt plaatsen.
015-022	Eén voor één worden de records uit de recordset doorlopen. In de lus op de regels 017-020 worden de velden van het actieve record één voor één opgehaald en weergegeven in de rij die door de variabele \$xlsRow bepaald wordt. Zo worden de gegevens in de Excel-tabel gevuld.

```
WriteLabel(2,3,"PHP");
```

Bovenstaande code plaatst de waarde PHP in de cel D3. De eerste parameter geeft de rij aan: rij 1 is 0, rij 2 is 1...



De tweede parameter geeft de kolom aan, te starten vanaf 0.

```

WriteLabel(0,0,"Adressenlijst");
WriteLabel(2,0,"Familienaam");
WriteLabel(2,1,"Voornaam");
WriteLabel(2,2,"Adres");
WriteLabel(2,3,"Postcode");
WriteLabel(2,4,"Gemeente");

```

De code hiernaast plaatst de kolomkoppen op rij 3 (rij 2 in de code, maar we starten in de code met 0).

Met rij 0 en kolom 0 adresseer je dus de cel A1 voor de titel 'Adressenlijst'.

	A	B	C	D	E
1	Adressenlijst				
2					
3	Familienaam	Voornaam	Adres	Postcode	Gemeente
4					
5					

5.14 Een mailbericht versturen vanuit PHP

Je kunt vrij eenvoudig vanuit PHP een mailbericht versturen, bv. een bevestiging na het invullen van een formulier. Hiervoor gebruik je de ingebouwde **mail()** functie. Het resultaat is true (mail met succes verstuurd) of false (mail niet verstuurd).

De syntaxis van de functie:

```
mail ($aan,$onderwerp,$bericht,$koppen)
```

Een voorbeeld:

```
$aan = "amelie.caveye@bamaflex.be,jan.janssens@bamaflex.be";
$onderwerp = "Bevestiging insturen oefeningen";
$bericht = "Dag Jan\n\n";
$bericht .= "We hebben je aanvraag goed ontvangen en maken er werk
van!\n\n";
$bericht .= "Beste groeten\n\nDéborah Huyghe";
$koppen= "From: Bamaflex besteldienst\r\nBcc:bestel@bamaflex.be";
$koppen .= 'Content-type: text/html; charset=UTF-8' . "\r\n";
mail ($aan,$onderwerp,$bericht,$koppen)
```

Bemerkt in het voorbeeld:

- dat \n gebruikt wordt voor het nemen van een nieuwe regel in je mailbericht. Een regel mag maximum 70 tekens bevatten.
- vergeet nooit een From: kop toe te voegen. Je kunt een standaard waarde in PHP.INI plaatsen. Andere mogelijke koppen: Cc, Bcc. Gebruik je verschillende koppen, dan scheid je die door \r\n (CRLF).
- het toevoegen/meesturen van '**Content-type: text/html; charset=UTF-8**' helpt om accentletters correct weer te geven in de verstuurde mail.
- uiteraard kan je in je mail variabele gegevens opnemen die bv. uit het juist ingevulde formulier gehaald worden.

5.15 Defineringen

Maak vooraf een databank **dbles3** en importeer de databank **dbles3.sql.zip**. Leg een connectie naar die databank op basis van een extern connectiebestand dat je telkens insluit.

5.15.1 Redder aan zee – RAZ_KEUZELIJST.PHP

Je biedt een keuzelijst aan met alfabetisch alle kandidaten voor het examen *Redder aan zee* uit de tabel raz_kandidaten.

Bij het kiezen krijg je als uitvoer de naam, ID en het bijhorende wachtwoord. De keuzelijst blijft zichtbaar en toont de weergegeven naam.



Variant: na het kiezen van een persoon is de keuzelijst niet langer zichtbaar en krijg je enkel de uitvoer te zien.

5.15.2 RAZ wachtwoord – RAZ_WACHTWOORD.PHP

De keuzelijst bevat alle namen uit de tabel raz_kandidaten, alfabetisch gerangschikt. In het tekstvak wordt een wachtwoord ingetypt en met de Log in knop controleer je of het wachtwoord klopt.

Als dat wachtwoord klopt, verschijnt onder de keuzelijst de melding *Wachtwoord OK*, zoniet verschijnt *Wachtwoord verkeerd!*

The image contains two screenshots of a login form. The top screenshot shows a dropdown menu with names and an empty text input field labeled 'Wachtwoord:' followed by a 'Log in' button. The bottom screenshot shows the same setup, but the 'Wachtwoord:' field now contains five blacked-out characters (dots) and the 'Log in' button remains.

Wachtwoord OK!

Varianten:

- Voeg als eerste element in de keuzelijst een keuze met de tekst "Kies een persoon>>>" toe. Pas als je een persoon gekozen hebt, verschijnt het tekstvak voor het ingeven van een wachtwoord en de Log in knop.

A screenshot of a dropdown menu with the text 'Kies een persoon>>>' selected.

De controle op het wachtwoord gebeurt enkel als een wachtwoord is ingevuld.

- Bouw een controle in op de lengte van het wachtwoord en controleer het wachtwoord enkel als het ten minste 2 tekens en maximum 20 tekens bevat.

Om gemakkelijk te kunnen controleren is het wachtwoord hier telkens de voornaam.

5.15.3 Verkooppunten opzoeken (VERKOOPPUNTEN.PHP)

Je start met een keuzelijst waarin alfabetisch op deelgemeente alle postcodes en deelgemeentes voorkomen (tabel tblPostcodes).

Bij het kiezen van een deelgemeente krijg je een overzicht van de verkooppunten in die deelgemeente (op basis van postcode, tabel tblverkooppunten).

9000 Gent			
GENT CENTRUM	Lange Kruisstraat 55	9000	GENT
GENT STAPELPLEIN	Stapelplein 75	9000	GENT
GENT STATIONSWIJK	Kortrijksesteenweg 194	9000	GENT

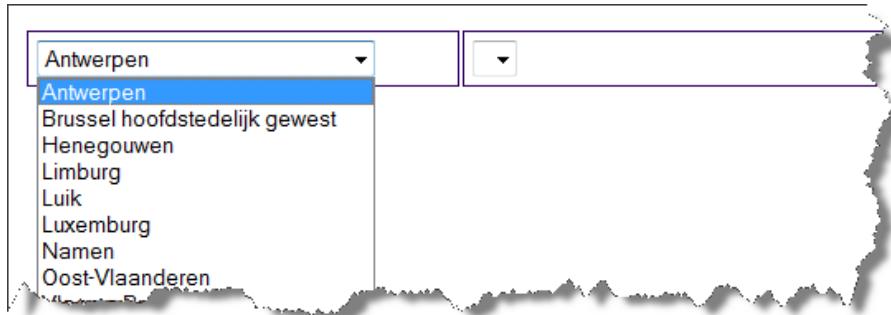
Kies je een deelgemeente zonder verkooppunten zijn, dan krijg je daar een melding van.

9860 Scheldewindeke
Geen verkooppunt gevonden. Kies een andere gemeente.

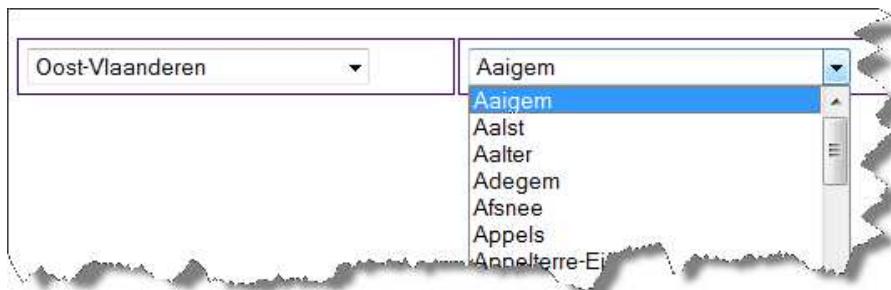
Je hoeft geen rekening ermee te houden dat soms verschillende deelgemeentes toch eenzelfde postcode hebben. Na het kiezen van een deelgemeente in de keuzelijst krijg je soms een andere deelgemeente te zien.

5.15.4 Postcodes en deelgemeentes (POSTCODES.PHP)

Bij de start krijg je een keuzelijst met alle provincies (inclusief Brussel hoofdstedelijk gewest).



Na het kiezen van een provincie wordt de keuzelijst ernaast – die je eventueel eerst kunt verbergen – gevuld met alle gemeentes uit die provincie.



Bij het kiezen van een gemeente in de tweede keuzelijst, krijg je ernaast een overzicht van alle deelgemeentes die hetzelfde postnummer hebben. Hieronder het resultaat bij *Balegem*.

Oost-Vlaanderen	Scheldewindeke	9860 - Balegem 9860 - Gijzenzele 9860 - Landskouter 9860 - Moortsele 9860 - Oosterzele 9860 - Scheldewindeke
-----------------	----------------	---

Na het kiezen van *Balegem* verspringt de gemeente op *Scheldewindeke*. Probeer te verklaren hoe dat komt. Een oplossing hoef je nog niet echt te zoeken...

5.15.5 De Chinese vrijwilliger (VRIJWILLIGER.PHP)

Voor steekproefcontrole ontwikkel je een kleine tool om een willekeurige student uit een groep studenten te kiezen.

Klik je op de knop, dan verschijnt in het vak eronder een willekeurig gekozen student.

Kies vrijwilliger>>>

De student van dit moment is... Simon Declerck.

De studenten haal je uit een array, bv:

```
$leerlingen=array("Lobke Bonte"=>"BonteLobke","Simon Declerck"=>  
"DeclerckSimon","Emma Devolder"=>"DevolderEmma","Alexander ...")
```

Variant:

- Zelfde toepassing, maar met namen uit een databank.

5.15.6 Resultaten Redder aan zee (RAZ_PUNTEN.PHP)

In de tabel RAZ_KANDIDATEN vind je alle kandidaten van het theoretisch examen redder aan zee. Op basis van die gegevens bouw je een keuzelijst.

Haal de examenresultaten van je keuze op uit de tabel RAZ_EXRESULT. Daarin vind je de ID van de kandidaat (EXKAND) en de punten (op 100) voor de 7 hoofdstukken (H1 tot H7).

Bij de start krijg je enkel de keuzelijst te zien:

Redder aan zee
A job so hot...
it's cool!

Kies een examenkandidaat: Tacia Ackx

Kies een naam uit de lijst en je krijgt de resultaten van de gekozen te zien. Scores die lager dan 50 % liggen, worden in witte letters op donkere achtergrond weergegeven.

Onder de resultaten krijg je het eendoordeel te zien: niet geslaagd of geslaagd. Bij geslaagde deelnemers wordt het totaalprocent vermeld (met 1 cijfer na de komma). Let ook op de verschillende weergaves bij geslaagd versus niet geslaagd.

Houd rekening met volgende evaluatieregels:

- Een kandidaat die op elk hoofdstuk 50 of meer punten behaalt slaagt.
- Wie op één of meer hoofdstukken minder dan 50 scoort, beschikt nog over een reservepotje, afhankelijk van het totaalprocent:
 - bij 90 % of meer, zijn er 9 tolerantiepunten beschikbaar;
 - bij 80 % of meer, zijn er 8 tolerantiepunten beschikbaar;
 - bij 70 % of meer, zijn er 7 tolerantiepunten beschikbaar;
 - bij 60 % of meer, zijn er 6 tolerantiepunten beschikbaar;
 - bij 50 % of meer, zijn er 5 tolerantiepunten beschikbaar;
 - een kandidaat met minder dan 50 % als totaal slaagt niet.

De tolerantiepunten kunnen ingezet worden om één of meer kleinere tekorten te compenseren. Bij meer tekortpunten dan tolerantiepunten slaagt de kandidaat niet.

Redder aan zee
A job so hot... it's cool!

Kies een examenkandidaat: ▾

	Hoofdstuk 1	Hoofdstuk 2	Hoofdstuk 3	Hoofdstuk 4	Hoofdstuk 5	Hoofdstuk 6	Hoofdstuk 7
Scores	54	69	41	39	68	60	52

niet geslaagd

Redder aan zee
A job so hot... it's cool!

Kies een examenkandidaat: ▾

	Hoofdstuk 1	Hoofdstuk 2	Hoofdstuk 3	Hoofdstuk 4	Hoofdstuk 5	Hoofdstuk 6	Hoofdstuk 7
Scores	71	79	54	68	73	54	53

geslaagd met 64.6 %

6 Session-variabelen en cookies

Bij het bezoek aan een website surf je van de ene pagina naar de andere. Gegevens die je in de ene pagina opvraagt en bijhoudt, zijn in de volgende pagina('s) normaal niet meer beschikbaar, behalve dan gegevens uit een gepost formulier die je in de bij het ACTION attribuut opgegeven pagina nog kunt opvragen via de `$_POST` array. In verdere pagina's zijn ook die gegevens niet meer beschikbaar.

Om gegevens over verschillende pagina's beschikbaar te houden, maak je gebruik van **SESSION**-variabelen. Een sessievariabele is gedurende de volledige sessie beschikbaar. De sessie start wanneer je met je webbrowser een pagina op de website opent en loopt zolang je binnen de website actief blijft en je browser niet sluit. Binnen een sessie kan je alle sessievariabelen in principe in alle pagina's gebruiken.

Sluit je de browser en open je die opnieuw, dan open je een nieuwe sessie. De session-variabelen uit de vorige sessie zijn niet langer beschikbaar.

Om veiligheidsredenen is de duur van een sessie beperkt in tijd (standaard ca. 20 minuten) maar je kunt die *timeout* als serverbeheerder aanpassen via de **session.cookie_lifetime** instelling in je **php.ini** bestand.

6.1 De sessie starten

Om een sessie te starten, neem je helemaal boven in elke webpagina – boven alle HTML-code – de instructie **session_start()** op.

Elke sessie krijgt een uniek nummer, de **session_id**, die je eventueel kunt weergeven met de `session_id()` functie.

```
<?php  
session_start();  
echo session_id();  
?>
```

Een voorbeeld van zo'n session-id: c7df8a20220a9b517fc80110793b35a9.

6.2 Session-variabelen instellen

Session-variabelen worden in de array **\$_SESSION** bijgehouden. Je kunt er zoveel session-variabelen in stockeren als je wil. Om bijvoorbeeld bij het verwerken van een formulier de ingetypte familienaam bij te houden over de webpagina's heen, kan je code er als volgt uitzien:

```
$_SESSION['Familienaam'] = $_POST['txtFamilienaam'];
```

Op de plaats waar je de inhoud uit de session-variabele wil gebruiken, haal je die gewoon als volgt op:

```
echo 'Je familienaam is: '.$_SESSION['Familienaam'];
```

6.3 Een sessie stoppen

Je kunt een session stoppen met de functie **session_destroy()**. De sessie wordt gestopt en alle sessievariabelen worden verwijderd. De sessievariabelen zijn pas weg als de pagina opnieuw geladen wordt.

Om een sessievariabele onmiddellijk te verwijderen, kan je de **unset()** functie gebruiken. Met **session_unset()** verwijder je alle sessievariabelen in één actie.

```
unset($_SESSION['Familienaam']);
```

6.4 Waar worden sessievariabelen bewaard?

Sessievariabelen worden op de server bewaard. Vermits je op je toestellen zowel client als server speelt, worden sessievariabelen op je toestel opgeslagen. Via de **session_save_path()** kan je opvragen in welke map sessievariabelen opgeslagen worden.

```
echo session_save_path();
```

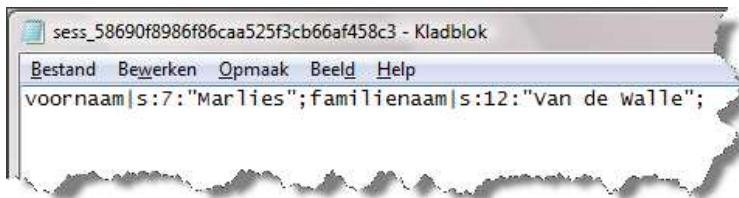
Die map vind je ook in je PHP.INI bestand (xampp\php\php.ini) terug in volgende instructieregel:

```
session.save_path = "C:\xampp\tmp"
```

In die map – standard xampp\tmp – vind je je sessiebestand terug en daarin vind je de opgeslagen gegevens.

Bekijk de PHP-code hieronder en het resultaat in het sessiebestand.

```
<?php  
$_SESSION['voornaam']='Marlies';  
$_SESSION['familienaam']='Van de Walle';  
?>
```



6.5 Praktisch voorbeeld – brood

In onderstaande website is de inhoud van de pagina's die je via het horizontale menu kunt kiezen enkel toegankelijk na correct inloggen.

Is dat niet het geval, dan krijg je een algemene illustratie te zien (BROODJES_ALL.JPG) met de verwijzing naar de loginmodule in de homepage (<<< Eerst aanmelden!).

Alhoewel we het inloggen kunnen koppelen aan een databank met gebruikers en wachtwoorden, houden we het in dit voorbeeld beperkt tot een algemene login *brood* met wachtwoord *pain*.

Probleemstelling: het probleem is dat we in de verschillende pagina's moeten nagaan en bijhouden of er ingelogd is, want enkel dan mag de inhoud van de pagina's getoond worden. Dat doen we via een sessievariabele.



Brood, lekker gezond

Brood is onontbeerlijk voor je gezondheid. Het is je voornaamste bron van koolhydraten en glucose. En die heb je broodnodig om je kortetermijngeheugen en je visuele en verbale capaciteiten een boost te geven. Van's morgens tot 's avonds.

Brood is hip Smakelijk, de klok rond Brood, lekker gezond Brood in alle soorten

	BAGEL	BAGUETTE	BRIOCHE	CHELSEA BUNS	
	CIABATTA		DUITS ROGGE BROOD		LAHMACUN
				NAAN	

<<< Eerst aanmelden!

Login

In elke pagina controleer je er in- of uitgelogd wordt. Is correct ingelogd (juiste naam en wachtwoord), dan stop je de waarde OK in de sessievariabele login:

`$_SESSION['login'] = "OK".`

Bij het klikken op de knop *Logout* wordt die sessievariabele leeg gemaakt.

```
<?php
if ($_POST['btnLogin']=="Login" and $_POST['txtWW']=='pain' and
$_POST['txtLogin']=='brood')
{
$_SESSION['login']= "OK";
}
if ($_POST['btnLogin']=="Logout")
{
$_SESSION['login']= "";
}
?>
```

Je kunt nu dezelfde sessievariabele gebruiken om te bepalen welk deel van de login je weergeeft: de tekstvakken met Login-knop of enkel een Logout-knop.

Je start natuurlijk met het login-gedeelte, maar na het correct inloggen geef je enkel een knop Logout weer. Klik je op die Logout-knop dan wordt de sessievariabele opnieuw leeg gemaakt... en nu krijg je weer het logingededeelte van het formulier te zien.

```
<form method="post">
<?php if ($_SESSION['login']=="OK")
{
?>
<p><input name="btnLogin" type="submit" value="Logout" /></p>

<?php
}
else
```

```

{
?>
<p>Login:<br><input name="txtLogin" type="text" size="15" required /></p>
<p>Wachtwoord:<br><input name="txtWW" type="password" size="15" required /></p>
<p><input name="btnLogin" type="submit" value="Login" /></p>
<?php
}
?>
</form>

```

Werk bij voorkeur ingesloten externe PHP-bestanden. Zo moet je de pagina's niet updaten na elke wijziging in je algemene PHP-code die voor alle pagina's geldt.

```

<?php
session_start();
include("cnnbrood.php"); //connectie met de databank
include ("algemeen.php"); //algemene PHP-code voor alle pagina's
?>

```

In de body van elke pagina controleer je de inhoud van de sessievariabele login. Als die verschilt van OK, dan krijg je een algemene foto + de boodschap 'eerst aanmelden' te zien.

Is de sessievariabele login op OK ingesteld, dan krijg je de eigenlijke inhoud van de webpagina te zien.

```

<?php
if ($_SESSION['login'] != "OK")
{
echo "<p><img src='images/broodjes_all.jpg' class='nofloat' width='100%'></p><br><p><strong>&lt;&lt;&lt; Eerst aanmelden!</strong></p>";
}
else
{
?>
<h1>brood is hip</h1>
<p>Brood is wereldwijd ... </p>
<?php
}
?>

```

Zolang je de browser niet sluit en/of de tijdslimiet niet overschrijdt, blijft de waarde van de sessievariabele over de verschillende pagina's gelden.

6.6 Cookies

Je wordt er dagelijks mee geconfronteerd op het internet: het al dan niet (gedeeltelijk) aanvaarden van *cookies*. Het zijn kleine stukjes data die via websites die je bezoekt soms stiekem op je computer worden opgeslagen. Ze houden o.a. voorkeuren en wat je op het internet doet bij en worden bij een volgend bezoek aan de website opgehaald om je gericht in een bepaalde richting te sturen.

Cookies worden op het toestel van de bezoeker – de client - opgeslagen, jouw toestel dus als je naar een website surft. Je browser bepaalt waar de cookies precies terecht komen. Via de browserinstellingen kan je je cookies beheren.

We maken uw winkelervaring persoonlijk

Zalando, Zalando Lounge, Zalon, Zircle en Outlets maken gebruik van cookies en andere technologieën om onze websites betrouwbaar en veilig te houden, om hun prestaties te meten, om een persoonlijke shopervaring en om persoonlijke advertenties aan te bieden. Hiervoor verzamelen wij informatie over gebruikers, hun gedrag en hun toestellen. Als u kiest voor "OK", gaat u ermee akkoord dat wij deze informatie delen met derde partijen, zoals onze marketingpartners. Dit betekent dat uw data misschien verwerkt kan worden in Amerika. Als u **toestemming weigert** gebruiken wij alleen de noodzakelijke cookies en ontvangt u helaas geen gepersonaliseerde inhoud. Selecteer "Voorkeuren instellen" voor meer informatie en om uw opties te beheren. U kunt uw voorkeuren te allen tijde wijzigen. Lees ons [privacybeleid](#) voor meer informatie.

Alleen essentiële cookies

Voorkeuren instellen

OK

Omwille van je privacy (GDPR) moet je dat opslaan van cookies op je toestel tegenwoordig al dan niet explicet toelaten. Je krijgt die vraag als je websites die cookies gebruiken bezoekt.

Hierboven en hiernaast een voorbeeld van de bevestiging die gevraagd wordt rond het gebruik van cookies bij Zalando.

Je kunt zelf nog iets meer in detail kiezen wat je al dan niet toelaat.

Je kunt met PHP relatief eenvoudig cookies maken en gebruiken.

```
<?php  
$cname = "interesse";  
$cvalue = "beamter";  
setcookie($cname, $cvalue, time() + (86400 * 60), "/");  
?>
```

Met de code hierboven stop je de interesse voor het product *beamter* in een cookie met de naam *interesse* die twee maand op het toestel van de bezoeker opgeslagen blijft.

86400 is de duur van één dag in seconden: 24 uur x 60 minuten x 60 seconden.

Let op: de setcookie() functie moet boven de <html>-tag verschijnen!

Met de / geven we aan dat de cookie in de volledige website beschikbaar is.

Cookies worden opgeslagen in de **\$_COOKIE** array.

Als de gebruiker volgende keer de website bezoekt, hoeft hij/zij niet verbaasd te zijn een aanbiedingen voor beamers te zien:

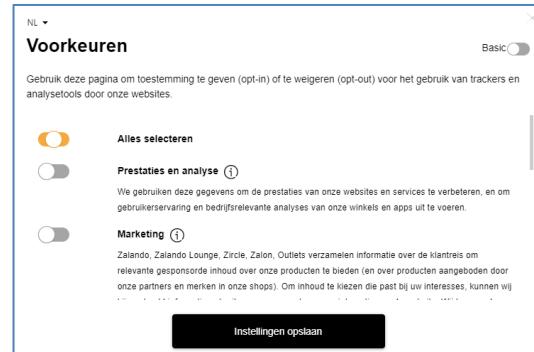
```
<?php  
if (isset($_COOKIE['interesse']))  
{  
$interesse = $_COOKIE['interesse'];  
Header("Location: $interesse.php");  
}  
?>
```

Let op: cookies kunnen niet in eenzelfde script geplaatst en uitgelezen worden!

Om een cookie te verwijderen, kan je een datum in het verleden toewijzen, bv.

```
<?php  
setcookie("interesse","",time()-60);  
?>
```

Hiermee stel je de vervaltijd op een minuut geleden in.



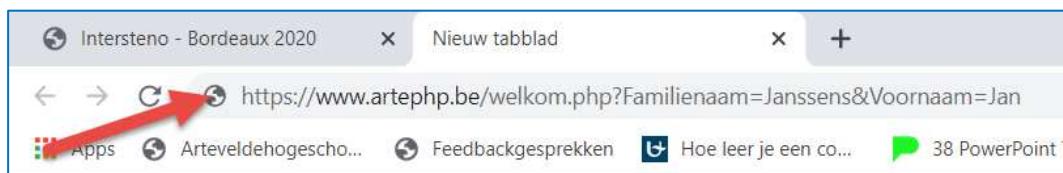
7 \$_GET en querystring

De \$_GET array verzamelt in eerste instantie informatie uit een formulier waar de methode GET is gebruikt (i.p.v. POST).

```
<form action="welkom.php" method="get">
Name: <input type="text" name="Familienaam" />
Age: <input type="text" name="Voornaam" />
<input type="submit" />
</form>
```

Bij het versturen van het formulier wordt de pagina welkom.php geopend via het action attribuut van de form-tag. De waarden uit de formuliervelden worden met de URL meegestuurd via een zogenaamde querystring. In het voorbeeld hieronder zie je duidelijk hoe dat gebeurt:

- een ? scheidt de pagina van de variabelen. De variabelen zijn de namen van de formulierobjecten waaraan de inhoud uit het formulier wordt toegekend.
- een & scheidt de verschillende variabelen als er meer dan één is.



Er zijn enkele belangrijke nadelen aan deze werkwijze:

- de informatie die doorgestuurd wordt is zichtbaar voor iedereen in de adresbalk van de browser
- de informatie is beperkt tot maximum 100 tekens.

Bij formulieren geven we dus de voorkeur aan de POST-methode, maar toch gaan we vaak van \$_GET gebruik maken, vooral om via **hyperlinks** informatie naar een bepaalde pagina door te sluizen.

Je kunt ook waarden doorgeven via een URL (link). Deze waarden zijn dan enkel bruikbaar in de pagina die via de URL opgeroepen wordt.

In het voorbeeld dat volgt wordt de code van de pagina TAAL.PHP getoond. De twee hyperlinks in die pagina roepen dezelfde pagina op, TAAL.PHP, maar geven hierbij een verschillende waarde voor de variabele 'taal' mee. De waarde van deze variabele wordt bewaard in de array **\$_GET**.

```
<?php
if (ISSET($_GET['taal']))
{
    if($_GET['taal']=='nl')
    {
        echo 'Je hebt Nederlands gekozen';
    }
    else
    {
        echo 'Je hebt Frans gekozen';
    }
}
```

```

        }
else
{
?>
<body>
<a href="taal.php?taal='nl'"> Nederlands</a>
<a href="taal.php?taal='fr'"> Frans</a>
</body>
</html>
<?php
}
?>

```

Bij de eerste aanroep van het formulier heeft `$_GET['taal']` nog geen waarde en dus worden de beide talen getoond, voorzien van een hyperlink. Bij het klikken op één van beide links krijgt `$_GET['taal']` een waarde die bepaalt welke boodschap je te zien krijgt, afhankelijk van de geklikte hyperlink.

7.1 Praktische toepassing – brood

Achter het menu *Brood in alle soorten* zit een foto-overzicht van alle broodsoorten, gegenererd uit de database. Klik je op een foto, dan krijg je detailinformatie van de gekozen soort.

The screenshot shows a website layout. At the top, there's a banner with the text "Brood, lekker gezond" and a photo of a person eating a sandwich. Below the banner is a navigation bar with four items: "Brood is hip", "Smakelijk, de klok rond", "Brood, lekker gezond" (which is highlighted in orange), and "Brood in alle soorten". The main content area features a large image of a sandwich with cream cheese and basil. To its right is a grid of smaller bread images with labels: BAGEL, BAGUETTE, BRIOCHE, CHELSEA BUNS, CIABATTA, DUTS ROGGE BROOD, LAHMACUN, and NAAN. A vertical scrollbar is visible on the right side of the content area.

De detailinformatie van een broodsoort wordt in de pagina `brooddetail.php` weergegeven. Achter elke foto zit dus een link naar die pagina, maar tegelijk wordt in de querystring ook de ID van de gekozen soort meegegeven.

In de pagina `BROODDETAIL.PHP` haal je dan aan de hand van de doorgegeven ID de juiste informatie uit de database.

Brood in alle soorten – BROOD.PHP

Naast de inlogcontrole, haal je hier de foto's uit de database. Achter elke foto ligt een hyperlink naar `BROODDETAIL.PHP` waarbij de ID van de broodsoort wordt doorgegeven.

```

<body>
<?php
if  ($_SESSION['login'] != "OK")  {

```

```

echo "<img src='images/broodjes_all.jpg'><br><br><strong><<< Eerst
aanmelden!</strong>";
}
else
{
$sql = "select * from tblbrood order by soort";
if(!$result = $db->query($sql)){
    die('<p>fout!</p>');
}
while ($row = $result->fetch_assoc())
{
echo "<a href='brooddetail.php?soort=".$row['ID']."'><img
src='images/".$row['foto']."'></a>";
}
}
?>

```

BROODDETAIL.PHP

Opnieuw de inlogcontrole. Op basis van de doorgegeven soort (`$_GET['soort']`) wordt de bijhorende tekst uit de databank gehaald en weergegeven. In `$_GET['soort']` verwijst *soort* naar de variabele in de querystring en niet naar het veld *soort* in de databank!

```

<body>
<?php
if ($_SESSION['login'] != "OK") {
echo "<img src='images/broodjes_all.jpg'><br><br><strong><<< Eerst
aanmelden!</strong>";
}
else
{
$sql = "select * from tblbrood where ID=".$_GET['soort'];
if(!$result = $db->query($sql)){
    die('<p>fout!</p>');
}
$row = $result->fetch_assoc();
echo $row['uitleg'];
} ?>

```

8 Fietsregistratie Brugge

Maak eerst een databank **fietsgraveer** aan in MySQL (via PHPMyAdmin) en importeer de gegevens in die databank. Je krijgt de structuur van de website ter beschikking.

De toepassing situeert zich in de stad Brugge, waar de preventiedienst gratis fietsen graveert om ze zo beter tegen diefstal te kunnen beschermen.

The screenshot shows the homepage of the Fietsregistratie Brugge website. At the top, there is a banner featuring a red logo with a stylized bicycle and the word 'FIETSSTAD' below it, set against a background of a historic building in Bruges. Below the banner, the menu includes 'Overzicht', 'Registreer', 'Lijsten', 'Wijzig', 'Verwijder', 'Detailinfo', 'Zoeken', and 'Home'. There are also fields for 'Voornaam', 'Familienaam', 'Wachtwoord', and a 'Login' button. The main content area has a red sidebar on the left with the 'FIETSSTAD' logo and text about the service. The main text area discusses the service of marking bike frames with a self-adhesive label. It includes a photo of a worker applying a label to a bike frame. The sidebar also provides contact information for the service.

Het menu wordt dynamisch uit de databank gegenereerd (**tblmenu**) en bevat bij de start drie keuzes. Later breiden we het menu uit.

	menuID	menutekst	menulink
	1	Overzicht	overzicht.php
	2	Registreer	registreer.php
	3	Lijsten	lijsten.php

(tblmenu)

- **Overzicht:** geeft een overzicht van de verschillende locaties waar het graveren kan gebeuren. Deze informatie komt uit de tabel **tblplaatsen**.

graveerID	gemeente	locatie	adres	datum	uur
1	Dudzele	gemeentehuis	Sint-Lenardsstraat 28	2025-05-07 00:00:00	15.00-18.00 u.
2	Koolkerke	hoeve De Grendel	Koolkerkse Steenweg 199	2025-05-08 00:00:00	15.00-18.00 u.
3	Sint-Michiels	gemeentehuis	Rijselstraat 98	2025-05-11 00:00:00	14.00-18.00 u.
4	Sint-Andries	gemeentehuis	Gistelse Steenweg 524	2025-05-14 00:00:00	14.00-18.00 u.
5	Sint-Jozef	buurthuis	Landjuwelenstraat 43	2025-05-21 00:00:00	15.00-18.00 u.
6	Assebroek	gemeentehuis	Generaal Lemanlaan 123	2025-05-22 00:00:00	14.00-18.00 u.
7	Zeebrugge	gemeenschapshuis	Marktplein 12	2025-05-18 00:00:00	15.00-18.00 u.
8	Sint-Pieters	fietsendienst	Sint-Pieterskaai 37	2025-06-04 00:00:00	15.00-18.00 u.
9	Sint-Kruis	gemeenschapshuis	Moerkerkse Steenweg 190	2025-06-18 00:00:00	14.00-18.00 u.
10	Lissewege	Oude pastorie	Oude Pastoriestraat 5	2025-06-25 00:00:00	15.00-18.00 u.

- **Registreer:** hiermee registreer je je komst op één van de voorziene fietsgraveersessies in de tabel **tblregistratie**.
- **Lijsten:** overzichtslijsten per gekozen locatie.

Overzicht

Je krijgt een overzichtslijst van alle locaties zoals hieronder geïllustreerd. Aandachtspunten:

Veld	Type
<input type="checkbox"/> registratielID	int(10)
<input type="checkbox"/> fnaam	varchar(60)
<input type="checkbox"/> voornaam	varchar(30)
<input type="checkbox"/> telefoon	varchar(20)
<input type="checkbox"/> email	varchar(50)
<input type="checkbox"/> plaats	int(10)

1 - tblRegistratie



Home Overzicht Registreren Lijsten Wijzig Verwijder Detailinfo Zoeken Mijn data Voornaam Familiernaam Wachtwoord Login

Overzicht fietsgraveeractie voorjaar 2025				
Gemeente	Locatie	Adres	Datum	Uur
Assebroek	gemeentehuis	Generaal Lemanlaan 123	2025-05-22	14.00-18.00 u.
Dudzele	gemeentehuis	Sint-Lenardsstraat 28	2025-05-07	15.00-18.00 u.
Koolkerke	hoeve De Grendel	Koolkerkse Steenweg 199	2025-05-08	15.00-18.00 u.
Lissewege	Oude pastorie	Oude Pastoriestraat 5	2025-06-25	15.00-18.00 u.
Sint-Andries	gemeentehuis	Gistelse Steenweg 524	2025-05-14	14.00-18.00 u.

Registreren

Hier registreren inwoners zich voor één van de aangeboden plaatsen en data. Het volstaat familienaam, voornaam, telefoon en eventueel e-mailadres op te geven en een plaats te

kiezen. Verzorg de opbouw en opmaak van je formulier. Gebruik het hierna geïllustreerde voorbeeld als inspiratiebron.

Na de registratie krijg je – in dezelfde pagina – een samenvatting van de geregistreerde gegevens (zie illustratie). Je krijgt er niet enkel een overzicht van de persoonlijke gegevens, maar ook detailinformatie over de gekozen locatie: adres en tijdstip.

Extraatje: dag en maand in de datum worden voluit weergegeven in het Nederlands!

The screenshot shows a registration form for bicycle engraving. At the top, there's a red header bar with links: Overzicht, Registreer, Lijsten, Wijzig, Verwijder, Detailinfo, Zoeken, Home, and a login section with fields for Voornaam, Familienaam, Wachtwoord, and Login. Below the header is a logo for 'FIETSSTAD' with a stylized bicycle icon. To the right of the logo is a large image of the Bruges Market Square (Grote Markt) with its characteristic Gothic town hall and belfry. The main form area has a red border and contains fields for personal information (Familienaam, Voornaam, Telefoon, E-mailadres), a dropdown menu for engraving location (Kies je graveerplaats with options like Dudzele, Koolkerke, Sint-Michiels, etc.), and a red button labeled 'Registreer je fietsgravure'.

Het invulformulier

This screenshot shows the confirmation page for the registration. It features the same red header and logo as the previous form. The main content area displays a summary of the registered data: Naam: Yelena Van Haverbeke, Telefoon: 0476 26 57 38, E-mailadres: yelena_vh@gmail.com, Graveerplaats: Koolkerke - hoeve De Grendel (Koolkerkse Steenweg 199), and Tijdstip: maandag 8 mei 2013 tussen 15:00-18:00 u.. A red button at the bottom of the summary area says 'Registreer je fietsgravure'.

De registratie wordt in dezelfde pagina bevestigd!

Lijsten

Tenslotte bieden we ook nog overzichtslijsten van de geregistreerde deelnemers per locatie aan.

Hiervoor gebruik je een keuzelijst die dynamisch en alfabetisch alle locaties bevat waar fietsen gegraveerd kunnen worden.

Als op de gekozen locatie nog niemand geregistreerd is, krijg je de boodschap '*Nog niemand geregistreerd.*'

Zijn er wel één of meer deelnemers geregistreerd, dan krijg je een alfabetische lijst met onderaan het aantal geregistreerde deelnemers. De lijst bevat naam, telefoon en e-mailadres.

Baseer je op de schermcaptenties op de volgende pagina voor de uitwerking.

The screenshot shows a website header with a red logo containing a stylized bicycle and the word 'FIETSSTAD'. Below the header is a large image of a Gothic-style building with multiple towers and flags. At the bottom of the page is a navigation bar with links like Overzicht, Registreren, Lijsten, Wijzig, Verwijder, Detailinfo, Zoeken, Home, and a login form.

Lijst van geregistreerde personen per locatie

Kies je locatie: **Maak je keuze!**

- Assebroek
- Dudzele
- Koolkerke

De keuzelijst met locaties

The screenshot shows the same website structure as the previous one. The main content area displays a message: 'Locatie: Koolkerke - hoeve De Grendel', 'Adres: Koolkerkse Steenweg 199', 'Datum en uur: donderdag 8 mei 2025, 15.00-18.00 u.', and 'Nog niemand geregistreerd.'

Een locatie waar nog niemand geregistreerd is.

Lijst van geregistreerde personen per locatie

Locatie: Lissewege - Oude pastorie
Adres: Oude Pastoriestraat 5
Datum en uur: woensdag 25 juni 2025, 15.00-18.00 u.

Naam	Telefoon	Email
Marie De Cuyper	0474 21 39 55	marie.decuyper@gmail.com
Chelsea De Graer	0479 86 26 76	chelsea.degraer@gmail.com

Totaal aantal geregistreerde personen: 2.

De detailinformatie en lijst van geregistreerden voor de gekozen locatie

8.1 Gegevens verwijderen

Met het verwijderen van gegevens ga je uiteraard zeer zorgvuldig om, want weg is weg. Vergeet vooral niet het juiste record te selecteren (WHERE), want zonder selectie verwijder je alle records uit de opgegeven tabel!

```
DELETE FROM tblKlanten WHERE klantID=436
```

In het voorbeeld hieronder gebruiken we één pagina voor drie verschillende soorten informatie:

- bij de start: een overzicht van de records waarvan je er eentje kunt verwijderen.
- een bevestigingsvraag nadat je een record geselecteerd hebt om te verwijderen.
- de bevestiging dat het record verwijderd is (of terug het overzicht als je niets verwijderd hebt).

Verwijder registratie

Victoria Aernoudt	>
Gauthier Andries	>
Pol Aneca	>
Leen Baert	>
Beau Bauto	>

De lijst waaruit je een record kan kiezen om te verwijderen.



De bevestigingsvraag om te verwijderen.



De bevestiging dat het record verwijderd is met een link terug naar de lijst.

Je krijgt de overzichtslijst te zien als geen record gekozen is of geen bevestiging gegeven is van verwijderen: geen knop ingedrukt of de Nee-knop ingedrukt.

```
if ((!isset($_GET['id'])) and !isset($_POST['btnKnop'])) or  
$_POST['btnKnop']=="Nee")
```

Het bevestigingsformulier verschijnt als een record gekozen is (ID doorgegeven via GET) maar nog niet op een knop (Ja of Nee) geklikt is.

```
if (isset($_GET['id']) and !isset($_POST['btnKnop']))
```

Pas als op de Ja-knop geklikt is, wordt het record echt verwijderd en een bericht weergegeven.

```
if ($_POST['btnKnop']=="Ja")
```

Let in de volledige code hieronder ook op het gebruik van enkele sessievariabelen die je uitschakelt als een record verwijderd is.

```
<?php  
session_start();  
include("cnnfietsgraveer.php");  
?  
...  
<body>  
<?php  
//het overzicht van records die verwijderd kunnen worden  
if ((!isset($_GET['id'])) and !isset($_POST['btnKnop'])) or  
$_POST['btnKnop']=="Nee")  
{  
$sql = "select * FROM tblregistratie order by fnaam,voornaam";  
if(!$result = $db->query($sql)){die('<p>fout!</p>');}  
echo "<table border='1'>";  
while ($row = $result->fetch_assoc())  
{  
echo "<tr><td>".$row['voornaam']." ".$row['fnaam']."</td><td><a  
href='verwijder.php?id=".$row['registratieID']."'> </a></td></tr>";  
}  
echo "</table>";
```

```

}

//het bevestigingsformulier om al dan niet te verwijderen
if (isset($_GET['id']) and !isset($_POST['btnKnop']))
{
$_SESSION['id']= $_GET['id'];
$sql1 = "select * FROM tblregistratie where
registratieID=".$_GET['id'];
if(!$result1 = $db->query($sql1)){
    die('<p>fout!</p>');
}
$row1 = $result1->fetch_assoc();
$_SESSION['fnaam']= $row1['fnaam'];
$_SESSION['voornaam']= $row1['voornaam'];
?>
<form id="form1" name="form1" method="post" action="">
    <table border="0" cellspacing="0" cellpadding="0">
        <tr>
            <td width="500">Wilt u de gegevens van <?php echo
$row1['voornaam']."' ".$row1['fnaam'] ?> definitief verwijderen? </td>
            <td width="50"><input name="btnKnop" type="submit" id="btnKnop"
value="Ja" /></td>
            <td width="50"><input name="btnKnop" type="submit" id="btnKnop"
value="Nee" /></td>
        </tr>
    </table>
</form>
<?php
}
//de bevestiging van het verwijderen
if ($_POST['btnKnop']=="Ja")
{
$sql2 = "delete FROM tblregistratie where
registratieID=".$_SESSION['id'];
if(!$result2 = $db->query($sql2)){
    die('<p>fout!</p>');
}
echo $_SESSION['voornaam']."' ".$_SESSION['fnaam']."' is definitief
verwijderd!<br><br>";
unset($_SESSION['id'],$_SESSION['voornaam'],$_SESSION['fnaam']);
echo "<a href='verwijder.php'>Terug naar lijst</a>";
}
?>
</body>
</html>

```

8.2 Gegevens wijzigen

Naast raadplegen, invoeren en verwijderen is het wijzigen van gegevens via PHP/MySQL de vierde en laatste vorm van SQL-interactie die we bespreken.

De basisinstructie voor het wijzigen van gegevens ziet er als volgt uit:

```
UPDATE tblPersonenel SET fnaam='Janssens', voornaam='Jan', aantalKind=3
WHERE personeelsID=436
```

We werken een concreet voorbeeld uit in onze website voor het laten graveren van fietsen.

Overzicht Registrer Lijsten Wijzig Verwijder Detailinfo Zoeken Home Voornaam Familienaam Wachtwoord Login

Gegevens van geregistreerde personen wijzigen

Kies een persoon: Bonnet Florence

Familienaam: Bonnet

Voornaam: Florence

Telefoon: 0479 50 92 47

Email: florence.bonnet@gmail.com

Plaats: Assebroek - gemeentehuis

Wijzigingen registreren

FIETSSTAD
B R U G G E
WERELDFOODSTAD
Fietsendienst
L. Coiseaukai 2
8000 Brugge

We onderscheiden volgende fases bij het wijzigen:

- voorziet een formulier waarin de gegevens van het te wijzigen record uit de databank weergegeven worden
- wijzig de gegevens in het formulier via je browser
- schrijf de gewijzigde gegevens in het juiste record weg in de databank.

Aan de website Brugge – fietsgraveren is een menukeuze *Wijzig* toegevoegd waarmee gegevens van geregistreerde gebruikers gewijzigd kunnen worden.

Mogelijke varianten die we hier niet beschouwen maar die prima oefenmateriaal kunnen zijn:

- de pagina *Wijzig* enkel toegankelijk maken na registratie via een wachtwoord zodat niet iedereen gegevens van iedereen kan wijzigen.
- iedereen enkel zijn eigen gegevens laten wijzigen na inloggen. In de pagina *Wijzig* krijgt hij/zij altijd en direct de eigen gegevens te zien zonder ze nog expliciet te moeten selecteren.

In onze uitwerking krijg je bij het openen van de pagina een keuzelijst waarin alle geregistreerde personen voorkomen en waaruit je iemand kunt selecteren.



Direct na de selectie worden de gegevens van de gekozen persoon netjes in de formuliervelden ingevuld.

De *plaats* vraagt speciale aandacht omdat die in een keuzelijst weergegeven wordt. In de databank zelf wordt enkel een nummer bewaard!

Na het aanpassen van de informatie, worden de wijzigingen opgeslagen. In feite slaan we in deze toepassing alle informatie opnieuw op, ook de niet gewijzigde gegevens.

IP-adres en tijdstip van wijzigen bijhouden.

Na het wijzigen laten we een bericht verschijnen waarin we het tijdstip van de wijziging en het IP-adres van waaruit gewijzigd werd weergeven.

Het is niet onverstandig om die gegevens – in plaats van ze weer te geven – in je databank bij te houden ter controle. Voeg dus twee velden aan de tabel *tblregistratie* toe. Voor het

bijhouden van het tijdstip kan je een veld van het type *varchar* gebruiken. De veldtypes *datetime* en *timestamp* zijn moeilijker in gebruik en bespreken we hier niet verder.



Om het IP-adres dat bij het versturen gebruikt is op te vragen, gebruik je de servervariable `$_SERVER['REMOTE_ADDR']`. Het is normaal dat je als ip-adres ::1 krijgt als je met je lokale webserver werkt.

Met `date('c')` geef je de huidige datum en tijd in het ISO 8601 formaat weer.

Het formulier voorziet twee keer een actie naar de databank toe:

- bij het kiezen van een persoon moeten de gegevens van die persoon opgehaald en weergegeven worden;
- bij het klikken op de knop *Wijzigingen registreren* worden de gegevens weggeschreven.

In de code die volgt merk je dus dat we kijken of de knop *btnWijzig* ingedrukt is, dan wel of er een persoon gekozen is in de keuzelijst *cboPersoon*.

```
<?php
include("cnnfietsgraveer.php");
if (isset($_POST['btnWijzig']))
{
    //wijzigingen registreren
$fnaam = $_POST['txtFamilienaam'];
$voornaam = $_POST['txtVoornaam'];
$telefoon = $_POST['txtTelefoon'];
$email = $_POST['txtEmail'];
$plaats = $_POST['cboPlaats'];
$id = $_POST['cboPersoon'];
$ipadres = $_SERVER['REMOTE_ADDR'];
$tijdreg = date('c');

$db->query("update tblregistratie SET
fnaam='$fnaam',voornaam='$voornaam',telefoon='$telefoon',email='$email',
plaats=$plaats,ipadres='$ipadres',tijdreg='$tijdreg' where
registratieID =$id") or die('probleem bij update');
$bericht = "$voornaam $fnaam gewijzigd op $tijdreg vanuit ip-adres
$ipadres";
}
if (isset($_POST['cboPersoon']))
{
$result = $db->query("select * from tblregistratie where registratieID
=". $_POST['cboPersoon']) or die(mysql_error());
$arpersoon = $result->fetch_assoc();
} ?>
```

De beide keuzelijsten worden op dezelfde manier opgebouwd. Bemerk hoe ook na het kiezen de gekozen waarde in de keuzelijst weergegeven blijft!

Bij het maken van een keuze in de keuzelijst met personen, is een submit naar de server voorzien. Kies je een (andere) plaats, dan moet er natuurlijk niets op de server gebeuren. Hetwegschrijven van de wijzigingen gebeurt immers pas als op de daarvoor voorziene knop geklikt wordt.

Keuzelijst personen

```
<select name="cboPersoon" onChange="document.frmPersoon.submit() ">
    <option value="0">Maak je keuze!</option>
<?php
$result = $db->query("select * from tblregistratie order by
fnaam,voornaam") or die(mysql_error());
while ($rij = $result->fetch_assoc())
{
    echo "<option value='".$rij['registratieID']."' ";
    if ($_POST['cboPersoon']===$rij['registratieID']) {echo
"selected='selected'";}
    echo ">".$rij['fnaam']." ".$rij['voornaam'] . "</option>\n";
} ?>
</select>
```

De plaats die in de keuzelijst weergegeven wordt, moet uiteraard overeenstemmen met de plaats die bij de betrokken persoon in de databank geregistreerd is.

Keuzelijst plaats

```
<select name="cboPlaats">
<?php
if (isset($_POST['cboPersoon']))
{
$sqlgemeente = $db->query("select * from tblplaatsen order by
gemeente") or die(mysql_error());
while ($rijgemeente = sqlgemeente->fetch_assoc())
{
    echo "<option value='".$rijgemeente['graveerID']."' ";
    if ($arpersoon['plaats']===$rijgemeente['graveerID']) {echo
"selected='selected'";}
    echo ">".$rijgemeente['gemeente'] . " - ".$rijgemeente['locatie'] .
"</option>\n";
}
?>
</select>
```

In de tekstvakken geef je telkens de opgeslagen inhoud uit het bijhorende veld in de databank weer:

```
<input name="txtFamilienaam" type="text" id="txtFamilienaam"
value="<?php echo $arpersoon['fnaam']?>" />
```

8.3 Gegevens opzoeken

Aan de hand van een voorbeeld leer je ook gegevens opzoeken. In het formulier hieronder kan je voornaam en/of familienaam invullen en op zoek gaan naar overeenstemmende gegevens.

The screenshot shows a search form titled "Gegevens van geregistreerde personen opzoeken". It features two input fields: "Familienaam" and "Voornaam", both currently empty. Below these fields is a dropdown menu labeled "Wie zoek je?" containing the value "bruyn". To the right of the dropdown is a red "Zoeken" button. The background of the page is a photograph of a historic building in Bruges.

Als resultaat krijg je een lijstje met alle personen die aan de opzoekcriteria beantwoorden. De tekstvakken worden bij het submitten leeggemaakt voor een volgende zoekopdracht.

Worden geen resultaten gevonden die aan de criteria beantwoorden, dan verschijnt de melding *Er zijn geen gegevens gevonden die aan je zoekcriteria beantwoorden*.

The screenshot shows the search results for the name "bruyn". The results table has columns: "Familienaam", "Voornaam", "Telefoon", "Email", and "Plaats". There are 8 rows of data:

Familienaam	Voornaam	Telefoon	Email	Plaats
Bruynooghe	Cindy	0478 78 64 96	cindy.bruynooghe@gmail.com	22
Bruynooghe	Dieter	0474 92 42 52	dieter.bruynooghe@gmail.com	4
Bruynooghe	Lien	0479 46 49 24	lien.bruynooghe@gmail.com	1
Bruynooghe	Ralph	0475 21 97 62	ralph.bruynooghe@gmail.com	13
Debruyne	Delfine	0474 26 10 74	delfine.debruyne@gmail.com	6
Debruyne	Emile	0473 70 61 39	emile.debruyne@gmail.com	7

De code op zich herbergt weinig verrassingen, al moet ze met verschillende situaties rekening houden, afhankelijk of beide, één of geen tekstvakken zijn ingevuld. Ze wordt bovendien pas uitgevoerd als op de knop Zoeken (btnZoeken) geklikt is.

```
<?php
if (isset($_POST['btnZoeken']))
{
$fnaam = $_POST['txtFamilienaam'];
$vnaam = $_POST['txtVoornaam'];

if (!empty($_POST['txtFamilienaam'])
and !empty($_POST['txtVoornaam']))
{
$sql = "select * from tblregistratie where fnaam='$fnaam' and
voornaam = '$vnaam' order by fnaam,voornaam";
}
elseif (!empty($_POST['txtVoornaam']))
{
$sql = "select * from tblregistratie where voornaam = '$vnaam'
order by fnaam,voornaam";
}
```

```

}
elseif (!empty($_POST['txtFamilienaam']))
{
    $sql = "select * from tblregistratie where fnaam ='$fnaam' order
by fnaam,voornaam";
}
else
{
    $sql = "select * from tblregistratie where fnaam='xyzxyz'";
}
if (!$result = $db->query($sql)){die('<p>fout!</p>');}
if ($result->num_rows==0)
{
    $bericht = "Er zijn geen gegevens gevonden die aan je zoekcriteria
beantwoorden.";
}
else
{
    $bericht = "Aantal gevonden personen: ".$result->num_rows;
}
}
?>

```

De tabel met resultaten wordt enkel weergegeven als er op de knop geklikt is en als er ten minste één persoon aan de criteria beantwoordt:

```

<?php
if (isset($_POST['btnZoeken']))
{
    if ($result->num_rows>0)
    {
        ?>
<table width="100%" border="0" cellspacing="0" cellpadding="0">
    <tr>
        <td><strong>Familienaam</strong></td>
        <td><strong>Voornaam</strong></td>
        <td><strong>Telefoon</strong></td>
        <td><strong>Email</strong></td>
        <td><strong>Plaats</strong></td>
    </tr>
    <?php
    }
    while ($arresult = $result->fetch_assoc())
    {
        echo "<tr>";
        echo "<td>".$arresult['fnaam']."</td>";
        echo "<td>".$arresult['voornaam']."</td>";
        echo "<td>".$arresult['telefoon']."</td>";
        echo "<td>".$arresult['email']."</td>";
        echo "<td>".$arresult['plaats']."</td>";
    }
    ?>
</table>
<?php } ?>

```

Zoeken naar delen van namen

Met bovenstaande code worden enkel personen gevonden waarvan voornaam en/of familienaam volledig overeenstemmen met hetgeen ingetypt is in de tekstvakken.

Met de **like** operator kan je ook op delen van namen zoeken: *where fnaam like '%ketel%'* zal je zowel de familienamen *Ketels* als *De Ketelaere* vinden: *ketel* komt in beide namen voor.

```
select * from tblregistratie where fnaam like '%$fnaam%'
```

Bij de like operator horen twee wildcards:

- %, vertegenwoordigt 0, 1 of meer tekens, bv. ...*like 'Vande%'* [alles wat met *Vande* begint].
- _ (underscore), vertegenwoordigt één teken, bv. ...*like 'DDV__'* [*DDV* + 3 tekens].

8.4 Detailinformatie opvragen

In een hoofdpagina krijg je een lijst of overzicht van een aantal gegevens of producten. Bij selectie volgt de detailinformatie, hetzij in dezelfde pagina, hetzij in een andere pagina van het gekozen product of gegeven. In het voorbeeld hieronder krijg je een overzicht van de locaties waar je terecht kunt om je fiets te graveren. Een klik op de detailknop moet meer informatie bezorgen die in dezelfde pagina naast het overzicht links verschijnt.

8.4.1 Detailinformatie in dezelfde pagina weergeven

The screenshot shows a website with a header in Dutch: Overzicht, Registrer, Lijsten, Wijzig, Verwijder, Detailinfo, Zoeken, Home, Voornaam, Familienaam, Wachtwoord, Login. Below the header is a logo for 'FIETSSTAD BRUGGE' featuring a stylized bicycle and the text 'WERELDFOODESTAD'. A main heading reads 'Overzicht en detailinformatie waar u terecht kunt.' Below it, a sub-instruction says 'U kunt op één van volgende locaties terecht voor het graveren van je fiets. Klik op detail> voor detailinformatie.' A table lists locations:

Gemeente	Locatie	detail>
Assebroek	gemeentehuis	detail>
Dudzele	gemeentehuis	detail>
Koolkerke	hoeve De Grendel	detail>
Lissewege	Oude pastorie	detail>
Sint-Andries	gemeentehuis	detail>
Sint-Jozef	buurthuis	detail>

The screenshot shows a detailed view for Dudzele. The top navigation bar is identical to the previous screenshot. The main heading is 'Overzicht en detailinformatie waar u terecht kunt.' Below it, the sub-instruction is the same. The table of locations is partially visible on the left. On the right, a detailed information box for Dudzele is shown:

Detailinfo Dudzele	
Gemeente	Dudzele
Locatie	gemeentehuis
Adres	Sint-Lenardsstraat 28
Datum	2025-05-07 00:00:00
Uur	10:00-18:00

Om te beginnen voorzie je natuurlijk een onzichtbare tabel die de ruimte in twee delen splitst: links de overzichtsweergave en rechts de detailweergave van één plaats.

Het overzicht links haal je gewoon uit je databank en geef je verzorgd weer. Achter *detail>* zit een hyperlink die een variabele *graveerid* meestuurt via de querystring. Vóór het ? is geen pagina opgegeven waardoor automatisch dezelfde pagina gebruikt wordt.

```
<a href=?graveerid=". $overzicht['graveerID'] .">
<em>detail&gt;</em></td>
```

In de detailweergave rechts haal je de juiste gegevens uiteraard op aan de hand van de door-gegeven *graveerid* die je via de `$_GET` array uit de querystring haalt.

Bij de start van de pagina zou een fout ontstaan in de SELECT-instructie omdat er nog geen *graveerid* kan opgevraagd worden uit de querystring – er is nog geen enkele keer op een `detail>` knop geklikt. Vandaar de onontbeerlijke controle op het niet-leeg zijn van de *graveerid*.

```
if (!empty($_GET['graveerid']))
{
$qrydetail = $db->query("select * from tblplaatsen where
graveerid=".$_GET['graveerid']) or die(mysql_error());
$ardetail = $qrydetail->fetch_assoc();
}
```

Voor de weergave van detailgegevens haal je die gegevens op uit de `$ardetail` array.

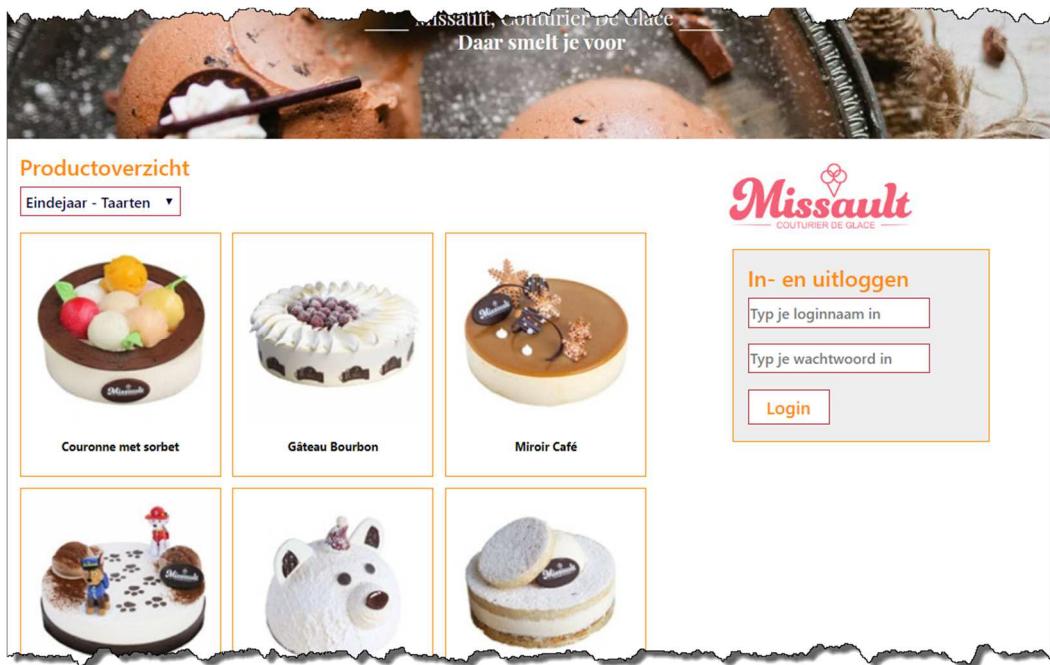
```
<tr>
<td class="onderrand">Adres</td>
<td class="onderrand"><?php echo $ardetail['adres']; ?></td>
</tr>
```

Om het geheel af te werken, geef je in het detailgedeelte enkel informatie weer nadat de bezoeker op één van de `detail>` links geklikt heeft. Ook hier controleren we opnieuw of de querystringvariabele *graveerid* al een waarde heeft. Is dat niet het geval, dan blijft het detailgedeelte compleet leeg.

```
<?php
if (!empty($_GET['graveerid'])) {
?>
⇒ HIER KOMT DE GEHELE DETAILINFORMATIE ⇌
<?php } ?>
```

8.4.2 Detailinformatie in een andere pagina weergeven

In dit voorbeeld geven we detailinformatie in een andere pagina weer. In plaats van een overzichtslijst krijg je deze keer van elk product in de gekozen categorie een foto te zien.



In de code die volgt wordt bij elke foto via de hyperlink het artikelID van het gekozen artikel naar de detailpagina doorgegeven.

The screenshot shows a product detail page for a 'Couronne met sorbet'. The product image is a round cake topped with colorful sorbet balls. The description reads: 'Couronne met sorbet' (Eindejaar - Taarten), 'Vanille- en woudvruchtenroomijs, afgewerkt met een ring van chocolade en bolletjes sorbet in diverse smaken.', '8 - 10 personen', and '€ 25,95'. Below the product information is a link 'Terug naar overzicht'. To the right of the product details is a login form with fields for 'Typ je logingebruikersnaam in' and 'Typ je wachtwoord in', and a 'Login' button. The Missault logo is at the top right. At the bottom of the page are links for 'Roomijs Missault (NV)', 'info@missault.be', and 'Openingsuren Cash & Carry'.

Om het geheel gebruikersvriendelijk te maken, is vanuit de detail-pagina een link voorzien die terug naar de fotopagina gaat (*Terug naar overzicht*).

```
if (isset($_POST['cboCategorie'])) {$gekozencat =  
$_POST['cboCategorie'];}  
elseif (isset($_GET['categorie'])) {$gekozencat = $_GET['categorie'];}  
...  
$sql="select * from tblproducten ";  
if (!empty($gekozencat) and $gekozencat<>"1") {$sql.="where  
categorie='".$gekozencat"';}  
$sql .= " order by categorie,productNL";  
$result = $db->query($sql) or die(mysql_error());  
while ($row = $result->fetch_assoc())  
{  
$id = $row[prodID];  
echo "<div id='product'>";  
echo "<a href='detail.php?id=$id'>";  
echo "<img src='images/$id.jpg' class='img-fluid'>";  
echo "<p class='midden'><strong>".$row['productNL'];  
echo "</strong></p>";  
echo "</a></div>";  
}
```

In de detailpagina halen we de detailgegevens op door te selecteren op het doorgegeven artikelID. Via **\$_GET** halen we de waarde uit de querystring op.

```
<?php  
if (isset($_GET['id']))  
{$sql = "select * from tblproducten where prodID='".$_GET['id']."'";  
$result = $db->query($sql) or die(mysql_error());  
$row = $result->fetch_assoc();  
?>
```

Het is nu een koud kunstje om de informatie uit het geselecteerde artikel netjes weer te geven.

In de link die terug naar het overzicht gaat, sturen we de categorie in de querystring mee. Op die manier kan je terug het overzicht opbouwen dat voordien geselecteerd was.

```
<a href='master.php?categorie=<?php echo $row['categorie']; ?>'>Terug  
naar overzicht</a>
```

Het weergeven van de vorige waarde in de keuzelijst kan dus uit twee bronnen gehaald worden:

- de gekozen optie in de keuzelijst bij het maken van een keuze die via `$_POST` doorgegeven wordt;
- de optie in de keuzelijst die vóór het opvragen van de detailinformatie actief was; deze optie wordt via `$_GET` uit de querystring gehaald.

```
if (isset($_POST['cboCategorie'])) {$gekozencat =  
$_POST['cboCategorie'];}  
elseif (isset($_GET['categorie'])) {$gekozencat = $_GET['categorie'];}  
  
$sql1="select distinct categorie from tblproducten order by  
categorie";  
$result1 = $db->query($sql1) or die(mysql_error());  
while ($row1 = $result1->fetch_assoc())  
{  
    $cat = $row1['categorie'];  
    echo "<option";  
    if ($cat==$gekozencat) {echo " selected='selected'";}  
    echo ">$cat</option>\n";  
}
```

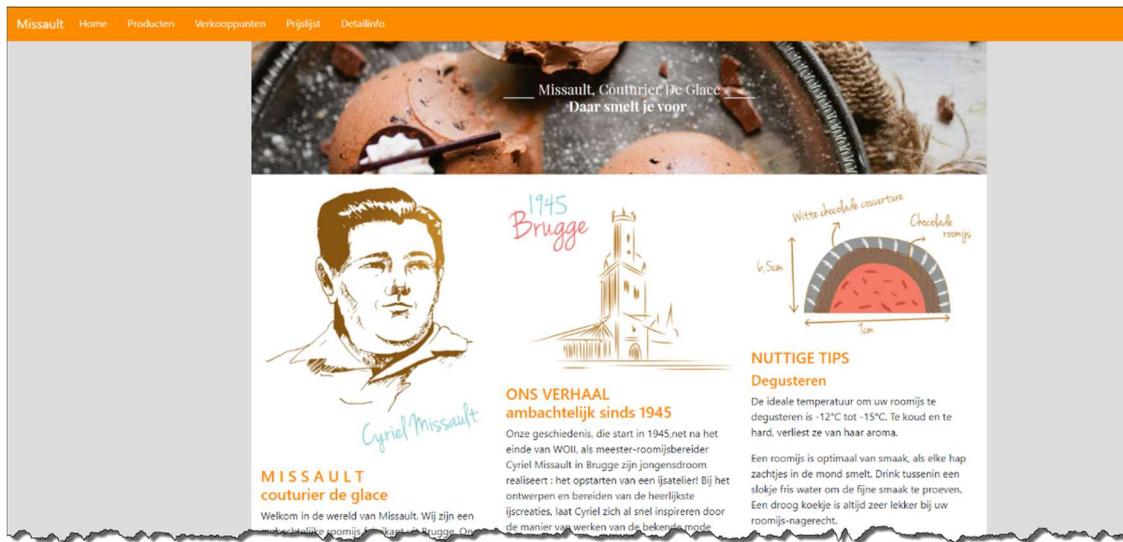
Hierboven halen we met het **select distinct** statement elke categorie slechts één keer uit de tabel met producten. Je kunt natuurlijk ook gebruik maken van een afzonderlijke tabel met alle categorieën als die in de databank beschikbaar is.

Ook voor de selectie van gegevens in het overzicht, komt het selectiecriterium – afhankelijk van de situatie – uit de `$_POST` array of uit de `$_GET` array. Bekijk in de code bovenaan vorige pagina hoe daarmee rekening wordt gehouden.

9 Missault roomijs

Missault is een landelijk gerenommeerd producent van kwaliteitsijs voor allerlei gelegenheden, o.m. de communiefeesten die er volop zijn. Via **missaultijs.zip** beschik je reeds over de database, de structuur van de homepage (index.php), enkele hulpbestanden en alle nodige illustraties.

Bij het openen van **INDEX.PHP** krijg je een website die op het bootstrap framework gebaseerd is.



Het menu

Het menu wordt uit de databank gegenereerd (**TBLMENU**). Bij de start krijg je altijd het Nederlandstalige menu.

Later kunnen nog taalknoppen toegevoegd worden om de menutaal te wijzigen (NL en FR).

Opgelet: als er geen distributeur aangemeld is, krijg je het menu-onderdeel *wijzig wachtwoord* niet te zien! Van zodra en zolang een distributeur aangemeld is, is de menukeuze *wijzig wachtwoord* wel zichtbaar.

Aanmelden distributeurs

Op basis van hun bedrijfsnaam en wachtwoord kunnen distributeurs zich aanmelden op de website. Via de login-knop controleer je eerst of de opgegeven naam bestaat en zo ja, of het wachtwoord klopt. Je geeft onder de login-knop feedback als een van beide gegevens niet klopt.

The image shows two side-by-side login forms for distributors. Both forms have a header with the Missault logo and the text "COUTURIER DE GLACE". Below the header is a section titled "In- en uitloggen" with a text input field labeled "Typ je loginnaam in". Underneath is another text input field labeled "Typ je wachtwoord in". At the bottom of each form is a red "Login" button. The left form is a standard login form. The right form is identical but includes the message "Distributeur Walschap is ingelogd." above the "Logout" button.

Zijn bedrijfsnaam en wachtwoord correct, dan krijg je een logout-knop te zien met daarboven de naam van de aangemelde distributeur. Die melding blijft zichtbaar zolang de distributeur aangemeld is. Afmelden doe je uiteraard met de knop *Logout* die prompt weer plaats maakt voor het login-gedeelte van het formulier.

Producten

De pagina met producten (**PRODUCT.PHP**) biedt in eerste instantie een overzicht van alle producten, per categorie en productnaam (NL) gerangschikt. Je krijgt de informatie te zien zoals hieronder geïllustreerd.

Met de keuzelijst bovenaan kan je ook een bepaalde categorie producten selecteren en weergeven. Later kan je altijd opnieuw alle producten weergeven door *Alle categorieën* in de keuzelijst te kiezen. De keuzelijst wordt dynamisch uit de databank opgebouwd.

Productoverzicht

- Kies een categorie ▾
 - Kies een categorie
 - Alle categorieën
 - Coulis
 - Dessert
 - Eindejaar - Buches
 - Eindejaar - Desserts
 - Eindejaar - Taarten
 - Halloween
 - Kindertaart
 - Lange taart
 - Taart

		
Dieren Dessert Beker gevuld met vanille roomijs. 1 - 0 personen € 3.75 Ook per 6 stuks verkrijgbaar.	Dôme Framboos Dessert Sorbet framboos, een stukje gebakken meringue, versierd met frambozen coulis en frambozen sponscake. 1 - 0 personen € 2.75 Ook per 6 stuks verkrijgbaar.	Double café Dessert Extra sterke koffie roomijs met een vleugje Armagnac. 1 - 0 personen € 2.10 Ook per 6 stuks verkrijgbaar. PROMO - 15 % korting!
		

Prijslijst

De prijslijst bevat een alfabetische lijst van de producten met daarnaast de categorie, volume, aantal personen, verkoopprijs per stuk en de eventuele verkoopprijs voor 6 of 12 stuks.

Bij het aantal Personen is soms een minimum én een maximum aantal opgegeven. Enkel als ook een maximum aantal is opgegeven, plaats je een liggend streepje tussen minimum en maximum aantal.

Enkel als er een prijs voor 6 of 12 stuks vastgesteld is, vermeld je die prijs. In alle andere gevallen krijg je niets te zien.

In de prijzen wordt een punt als decimaal scheidingsteken gebruikt. Zou je de prijzen ook met komma als decimaal scheidingsteken kunnen weergeven?

The screenshot shows a website for 'Missault, Couturier De Glace'. At the top, there's a banner with the text 'Daar smelt je voor' and an image of chocolate. Below it is a 'Prijslijst' (Price List) table:

Product	Categorie	Volume	Personen	Prijs	Doos6	Doos12
Chocolade coulis	Coulis	200 ml	25 - 50	€ 3.00		
Framboos coulis	Coulis	200 ml	25 - 50	€ 3.00		
Aardbei 130 ml	Dessert	130 ml	1	€ 1.35		€ 14.10
Beker Puzzel	Dessert	100 ml	1	€ 3.15	€ 27.40	
Beker suikervrij	Dessert	130 ml	1	€ 3.10	€ 21.60	
Beker Tekening	Dessert	100 ml	1	€ 3.05	€ 21.25	
Bekers praliné	Dessert	150 ml	1	€ 1.80		€ 18.80
Bekers vanille	Dessert	150 ml	1	€ 1.80		€ 18.80

To the right, there's a 'Missault' logo and a 'In- en uitloggen' (Log in/out) form with fields for login name and password, and a 'Login' button.

Verkooppunten

Je krijgt aanvankelijk een overzicht van alle types verkooppunten. Voorzie een keuzelijst waarmee je op een winkeltype kunt selecteren.

Bij selectie van een winkeltype past de lijst zich aan. Je geeft naam, adresgegevens, telefoon en winkeltype (afgekort) in de lijst weer.

The screenshot shows a 'Verkooppunten' (Sales Points) section with a dropdown menu 'Kies een winkeltype' (Select a shop type). The dropdown shows options like 'Alle winkeltypen', 'Groothandel', 'Kleinhandel - beperkt', etc. To the right is a table of sales point details:

Adres	Postcode	Gemeente	Telefoon	Type
Berlingmolenstraat 64	8850	Ardoorie	051 74 71 81	KHB
Parklaan 9	8730	Beernem	050 78 18 36	KHB
Avenue Jos Wauters 69	4500	Ben-Ahin	085 23 35 29	KHH
Hoogstraat 2	3580	Beringen	011 43 33 82	KHB
Brandheide 11	2880	Borrem	03 890 64 70	GH
Gistelse steenweg 260	8200	Brugge	050 38 77 34	KHV

To the right, there's a 'Missault' logo and a 'In- en uitloggen' (Log in/out) box stating 'Distributeur Walschap is ingelogd.' (Distributor Walschap is logged in) with a 'Logout' button.

Wijzig wachtwoord

Enkel als een distributeur aangemeld is, kan hij ook zijn wachtwoord wijzigen door het oude wachtwoord in te typen en daarna twee keer het nieuwe wachtwoord. Eerst wordt nagegaan of het oude wachtwoord overeenstemt met de aangemelde distributeur, daarna of het nieuwe wachtwoord twee keer hetzelfde is en ten minste 5 tekens lang is.

Indien niet alles klopt, krijg je de gepaste foutmelding naast de knop te zien. Als alles klopt wordt het wachtwoord effectief gewijzigd en krijg je daar bericht van.

Inloggen met een bedrijfsnaam zoals 't Koetje is zeker een extra uitdaging!



Missault – structuur van de databank

tblproducten

prodID	categorie	productNL	productFR	omschrijvingNL	omschrijvingFR	volume	minpax	maxpax	prijs	doos6	doos12	promo	magazijn
COU01	Coulis	Chocolade coulis	Coulis Chocolat	Chocolade coulis	Coulis Chocolat	200	25	50	3.00	0.00	0.00		D6
COU02	Coulis	Framboos coulis	Coulis Framboise	Framboos coulis	Coulis Framboise	200	25	50	3.00	0.00	0.00		D7
DES01	Dessert	Hartje	Petit Coeur	Roomijs vanille. coulis framboise. meringue. afgewerkt met go...	Crème glacée vanille. coulis framboise. meringue. ...	130	1	0	3.00	15.65	0.00		D11
DES02	Dessert	Hartje 3D	Petit Coeur 3D	Roomijs Vanille - coulis framboos afgewerkt met go...	Crème glacée Vanille - coulis framboise décoré de ...	100	1	0	3.00	15.65	0.00	X	D8
DES03	Dessert	Hart	Cœur	Roomijs vanille. mousse van passievrucht met stukjes...	Crème glacée vanille. mousse de fruits de la passi...	600	4	0	12.70	0.00	0.00		D9
DES04	Dessert	Dôme Framboos	Dôme Framboise	Sorbet framboos. een stukje gebakken meringue. ver...	Sorbet framboise. un petit morceau de meringue cu...	100	1	0	2.75	13.85	0.00		D2
DES05	Dessert	Rivoli	Rivoli	Vanille roomijs met passievruchten mousse	Crème glacée à la vanille avec mousse de	130	1	0	3.10	16.20	0.00		D10

tblcategorie

catID	catOmschrijving	catTaal	catArtikel
CAT02	Bûches	FR	ijsstronk
CAT04	Desserts individuels	FR	individueel
CAT08	Gâteaux	FR	taart
CAT06	Gâteaux pour enfants	FR	kindertaart
CAT01	Ijsstronken	NL	ijsstronk
CAT03	Individuele porties	NL	individueel
CAT05	Kindertaarten	NL	kindertaart
CAT07	Taart	NL	taart

tbldistributeurs

disID	disCode	disNaam	disAdres	disPostcode	disGemeente	disTel	disType
DIS1044	9700	'De Lekkerbek'	Nestor De Tièrestraat 60	9700	Oudenaarde	055 30 94 81	KHB
DIS1010	1860	't Hoeveke	Kapellenlaan 340	1860	Meise	02 270 18 28	KHB
DIS1005	2800	't Kadetje	Geerdegem Schone berg 17	2800	Mechelen	015 42 33 01	KHB
DIS1034	8730	't Kiekse	Parklaan 9	8730	Beernem	050 78 18 36	KHB
DIS1006	1910	't Koetje	Mechelsesteenweg 52	1910	Kampenhout	016 65 07 03	KHB
DIS1017	8940	't Werviks Chocoladehuis	Komenstraat 29	8940	Wervik	0472 77 64 70	KHB
DIS1053	8400	Arthos	Nieuwpoortsesteenweg 256	8400	Oostende	059 43 31 18	KHB

tblmenu

MenuItemID	MenuItem	MenuLink	Volgorde	MenuCode	MenuTaal
MEN01	Producten	product.php	1	ALL	NL
MEN02	Verkooppunten	verkooppunten.php	2	ALL	NL
MEN03	Nuttige tips	tips.php	3	ALL	NL
MEN04	Wijzig wachtwoord	wijzigww.php	5	DIS	NL
MEN05	Les produits	product.php	1	ALL	FR
MEN06	Points de vente	verkooppunten.php	2	ALL	FR
MEN07	Conseils utiles	conseils.php	3	ALL	FR

tblwinkeltypes

ID	distype	winkeltype
1	GH	Groothandel
2	KHV	Kleinhandel - volledig
3	KHB	Kleinhandel - beperkt
4	KHH	Thuisverkoop

10 Bronnen

w3schools, Learn PHP - <https://www.w3schools.com>

w3schools, Learn SQL - <https://www.w3schools.com>

diverse andere online bronnen

