Inhoudstafel

[Inhoudstafel 1](#_Toc209878555)

[Inleiding 3](#_Toc209878556)

[Hoofdstuk 1: Basissyntax en Strings 4](#_Toc209878557)

[Commentaar in php 4](#_Toc209878558)

[Naamgeving van variabelen in PHP 4](#_Toc209878559)

[Datatypes in PHP 5](#_Toc209878560)

[Strings 5](#_Toc209878561)

[HTML toevoegen aan Strings 5](#_Toc209878562)

[Lengte van een String bepalen 6](#_Toc209878563)

[String doorzoeken voor een letter of een andere String 6](#_Toc209878564)

[Een String opsplitsen in delen 6](#_Toc209878565)

[Witruimte van een String verwijderen 7](#_Toc209878566)

[Oefeningen: 7](#_Toc209878567)

[Hoofdstuk 2: Selectie of keuzestructuren 8](#_Toc209878568)

[Enkelvoudige keuzestructuur – if/else 8](#_Toc209878569)

[Meervoudige keuze-structuur – if/elseif/else – switch 9](#_Toc209878570)

[Oefeningen: 10](#_Toc209878571)

[Hoofdstuk 3: Herhaalstructuren 11](#_Toc209878572)

[While()-lus (onbepaalde herhaling) 11](#_Toc209878573)

[Variant: do-while() 11](#_Toc209878574)

[For()-lus (bepaalde herhaling) 12](#_Toc209878575)

[Oneindige lus 12](#_Toc209878576)

[Foreach()-lus 13](#_Toc209878577)

[Lussen onderbreken en beëindigen 13](#_Toc209878578)

[Break 13](#_Toc209878579)

[Continue 13](#_Toc209878580)

[Exit 13](#_Toc209878581)

[Die() 13](#_Toc209878582)

[Oefeningen: 13](#_Toc209878583)

[Hoofdstuk 4: Formulieren 15](#_Toc209878584)

[Inleiding 15](#_Toc209878585)

[Get of Post 15](#_Toc209878586)

[Get 15](#_Toc209878587)

[Voordelen 15](#_Toc209878588)

[Nadelen 15](#_Toc209878589)

[POST 15](#_Toc209878590)

[Voordelen 15](#_Toc209878591)

[Nadelen 16](#_Toc209878592)

[Voorbeeld van Get-formulier: 16](#_Toc209878593)

[Voorbeeld van POST-formulier: 16](#_Toc209878594)

[Formuliervelden 17](#_Toc209878595)

[Oefeningen: 17](#_Toc209878596)

[Hoofdstuk 5: Sessies 19](#_Toc209878597)

[Inleiding 19](#_Toc209878598)

[Sessie starten 19](#_Toc209878599)

[Een sessievariabele gebruiken 19](#_Toc209878600)

[Bestaande sessievariabele oproepen 20](#_Toc209878601)

[Sessie verwijderen 20](#_Toc209878602)

[Oefeningen: 21](#_Toc209878603)

Inleiding

Php staat voor **PHP: hypertext Preprocessor** en is een **server-side scripttaal**. Dit wil dus zeggen dat de script op een webserver worden uitgevoerd en daarna de uitkomst wordt teruggezonden naar de browser.

PHP is een **open source taal** en is **gratis te downloaden en te gebruiken**. Dit vertaald zich dan ook direct in een grote en actieve community die van PHP een echt succesverhaal heeft gemaakt. Professionele developpers gebruiken iets meer ASP.NET maar als beginner is dit een uiterst geschikte taal om dynamische websites mee te crëeren.

Php kan goed overweg met de meeste **databanken** zoals **mysql**, **oracle**, etc. Aangezien Mysq eveneens een gratis databank is (voor persoonlijk gebruik) is dit de meest gebruikte combinatie.

Php kan met **verschillende besturingsystemen** overweg zoals Windows, linux, unix, etc. De webserver die meest gebruikt wordt is Apache maar Php kan ook draaien op een windows IIS.

De installatie kan wat moeilijk zijn voor leken aangezien je drie programma’s moet installeren en deze werken op alle platformen. Eerste stap is de installatie van de webserver (Apache), daarna kan je PHP installeren en daarna Mysql. We gebruiken een handig programmatje dat apache, php en myssql al voor ons installeert en toelaat om hiermee lokaal te werken. Dit programmaatje kan je downloaden op <http://www.usbwebserver.com>

Php bestanden kan je gewoon openen en schrijven in kladblok en kan je herkennen aan hun extensie, meestal .php.

Voor meer info kan je altijd terecht op de website: <http://www.php.net>.

Hoofdstuk 1: Basissyntax en Strings

Stukjes php-code laten we altijd beginnen met **<?php** en eindingen me **?>.** De php-code kan zich overal in het document bevinden. Sommige servers laten ook toe om met **<?** te beginnen.

Voorbeeld 1: ons eerste php-script

<html>

<body>

<?php

echo("Welkom bij KTA da Vinci");

?>

</body>

</html>

# Commentaar in php

Zoals altijd is het gebruik maken van commentaar belangrijk voor jezelf maar levensnoodzakelijk als je in een team van programmeurs gaat werken. Commentaar kan je inbrengen zoals in Java op 2 manieren.

//dit is de eerste manier

/\* dit is

een andere manier voor meerdere lijnen\*/

# Naamgeving van variabelen in PHP

Zoals steeds zijn er enkele regels om in het achterhoofd te houden wanneer we willen gebruik maken van variabelen.

* Elke variabele begint met een $
* Gebruik een zinvolle naam die weergeeft wat de inhoud is vb $naamKlant
* Een variabele moet beginnen met een letter of een liggend streepje \_
* Een variabele mag geen speciale tekens bevatten enkel a-z,A-Z en 0-9 en \_
* Een variabele mag geen spaties bevatten (anders denkt de interpreter dat je een andere functie/variabele begint). We kiezen ervoor om te beginnen met een kleine letter en daarna elk nieuw woord eraan vast te plakken en met een Hoofdletter te laten beginnen.

# Datatypes in PHP

Hoewel Php wel verschillende datatypes kent moet je niet explicitiet aangeven welk datatype je wil gebruiken. Php zal automatisch proberen te herkennen welke gegevens erin zitten en zijn gegevenstype aanpassen.

Voorbeeld 2:

<html>

<body>

<?php

//dit is ons eerste voorbeeld waarbij we variabelen gebruiken

$titel = "Welkom bij KTA da Vinci";

Echo $titel;

?>

</body>

</html>

# Strings

Aangezien PHP niet verplicht om een datatype aan te geven is het voldoende om “” te plaatsen rond een woord om aan te geven dat het een string is.

$tekst = “Welkom bij KTA”;

Verder kan je Strings ook nog aan elkaar plakken door gebruik te maken van het **.** symbool.

$tekst = “Welkom bij KTA”;

$tekst .=” da Vinci”; of $tekst = $tekst . “ da Vinci”;

## HTML toevoegen aan Strings

In php kan je altijd HTML-code laten uitvoeren vanuit Strings. Dit wordt meer gebruikt dan je misschien op het eerste zicht zou vermoeden. Stel je voor je wil gegevens uit een databank ophalen en tonen. Waarschijnlijk wil je dan ook wel voor iets van opmaak zorgen. Misschien wat CSS-codes of op zijn minst een tabelstructuur.

Voorbeeld 3:

<?php

echo “<font size=6 color=’pink’>Dag iedereen</font>”

?>

Dit zal ervoor zorgen dat “Dag iedereen” in het roos en vrij groot wordt weergegeven.

!Let op: Als je in HTML zelf “” zou moeten gebruiken moet je die in php vervangen door ‘ (enkele aanhalingstekens) anders wordt de String beëindigt en krijg je allerlei foutboodschappen.

## Lengte van een String bepalen

De functie strlen(string) bepaald de lengte van een opgegeven String. Dit kan handig zijn in verschillende scripts bvb om de een herhaallus zoveel maal te laten herhalen om een woord in stukjes te snijden.

Voordbeeld 4:

<?php

echo strlen("Welkom bij KTA da Vinci");

?>

Het scherm zou dan 23 tonen.

## String doorzoeken voor een letter of een andere String

De funcite strpos(string,string) doorzoekt de eerste String op zoek naar een “match’ met de 2de string. De 2de parameter kan ook gewoon 1 letter zijn.

!let op: het antwoord begint steeds te tellen vanaf 0 dus strpos(“hallo”,”a”) zou 1 opleveren.

Voorbeeld 5:

<?php

echo strpos("Welkom bij KTA da Vinci”,”KTA”);

?>

Het scherm zou dan 11 tonen omdat daar KTA begint.

## Een String opsplitsen in delen

De funcite substr(string, start,aantal) kan je gebruiken om een String in kleinere delen op te splitsen.

Voorbeeld 6:

<?php

echo substr("Welkom bij KTA da Vinci”,12,3);

?>

Het scherm zou dan KTA tonen.

## Witruimte van een String verwijderen

Als je veel gebruik maakt van formulieren met tekst vakken kan het wel eens zijn dat gebruikers teveel spaties plaatsen voor of na hun gegevens. Hiervoor heeft php enkele handige functies.

* Trim(String) verwijderd spaties voor en achteraan
* Ltrim(String) verwijderd spaties voor de String
* Rtrim(String) verwijderd spaties na de String

Voorbeeld 7:

<?php

echo trim(" Pizza!!! ”);

?>

Het scherm zou dan Pizza tonen.

# Oefeningen:

1. Maak een phppagina waarin je je voornaam en naam in variabelen stopt en ze beide toont op het scherm.
2. Idem als 1 maar zorg ervoor dat je naam pas op de volgende regel verschijnt.
3. Maak een php-pagina met daarin de variabele tekst waarin je “Jantje zag eens pruimen hangen” stopt. Hiervan toon je de lengte, de positie van de letter “g” en toon je de 8ste tot 10de letter.

Hoofdstuk 2: Selectie of keuzestructuren

Zoals in elke programmeer- of scriptaal zijn er hier ook weer enkele mogelijkheden om een keuzestructuur te implementeren. We zullen dit deel vrij kort houden aangezien jullie al kennis hebben van keuzestructuren.

# Enkelvoudige keuzestructuur – if/else

Het if-statement is een van de belangrijkste voorwaardelijke statements van PHP. Het test de boolean-waarde van een statement of van een variabele en afhankelijk van het resultaat worde een taak uitgevoerd.

Vb. **Als** warmer dan 30° dan gaan we naar de zee

In php:

If ($temperatuur > 30){

$strand = true;

Echo “Joepie, we gaan naar het strand”;

}

We kunnen dit nog verder uitbreiden naar een actie als niet aan de voorwaarde voldaan is.

Vb. **Als** warmer dan 30° dan gaan we naar de zee **anders** blijven we gezellig thuis.

In php:

If ($temperatuur > 30){

$strand = true;

Echo “Joepie, we gaan naar het strand”;

}else{

$strand = false;

Echo “Te koud, we blijven lekker gezellig thuis”;

}

}

Merk op: om dit script te testen moeten we wel eerst een waarde geven aan $temperatuur en ook onze php-pagina openen en sluiten.

# Meervoudige keuze-structuur – if/elseif/else – switch

Natuurlijk kan het zijn dat je meer dan 2 mogelijkheden hebt en dan kan je keuzestructuren nesten. Een mooiere oplossing voor als je een variabele wil testen op n-aantal mogelijkheden is de switch. We hebben dit ook al behandeld in VB maar daar wedr er gebruikt gemaakt van een select case.

Voorbeeld:

Je wil een nieuwe pc kopen van €1400. Als je genoeg spaargeld hebt dan wordt de tekst “De nieuwe pc komt eraan” getoond. Als je net tekort komt (<400) dan wordt er “Dat wordt nog even sparen.” Getoond. Als je nog minder hebt dan tonen we “Misschien toch eens zoeken naar een vakantiebaantje”.

In PHP:

If($spaargeld => 1400){

echo “De nieuwe pc komt eraan.”;

}elseif ($spaargeld =>1000){

echo “Dat wordt nog even sparen.”;

}else{

echo “Misschien toch eens zoeken naar een vakantiebaantje.”;

}

Als we nu maar een bepaald aantal mogelijkheden hebben bvb de dagen van de week of een productnaam dan kan je best opteren voor het switch-commando.

$productNaam = "Processor";

switch ($productNaam)

{

case "Videokaart":

echo "Videokaarten zijn leverbaar vanaf €50 tot €650";

break;

case "Monitoren":

echo "LCD monitoren kosten tussen €150 en €750";

break;

case "Processor":

echo "Intel processoren hebben we vanaf €89 tot €444";

break;

default:

echo "Sorry, maar dat product is ons niet bekent";

break;

}

Dit lijkt op het eerste moeilijker maar is bij grotere projecten veel overzichtelijker dan allemaal if-structuren om niet te spreken over het debuggen.

# Oefeningen:

Schrijf telkens erbij waarom je zou opteren voor if/elseof/else of switch

1. Maak een simulatie van een verkeerslicht en geef weer wat er moet gebeuren.

Vb. Het licht is oranje, je moet stoppen tenzij je dit niet meer veilig kan doen.

1. Sla de dag van de week op met:

$day = date("l");

Maak je output zo dat er komt: Vandaag is het zaterdag.

1. Maak een stukje code voor je facturatieprogramma. Als de btwcode 6,12 of 21 is dan toon je het btwbedrag.

Vb. $totaal = 1000;$btwcode=21;

Dan zou het programma moeten tonen :

“Het btwbedrag is €210;”

1. Je vraagt de leeftijd van je bezoeker. Als de leeftijd > is dan 15 dan zeg je “Welkom bij informaticabeheer”, anders “Binnen enkele jaren ben je welkom in Informaticabeheer”.

Hoofdstuk 3: Herhaalstructuren

Voor dit hoofdstuk hebben we het over vorowaardelijke (conditionele) statements gehad. De opdrachten werden uitgevoerd van zodra aan een bepaalde voorwaarde wordt voldaan.

# While()-lus (onbepaalde herhaling)

Nu willen we een bepaalde statement enkele malen uitvoeren aan de hand een expressie. We gebruiken hiervoor while() wat een lus-structuur is. De opdrachten worden uitgevoerd zolang aan een bepaalde voorwaarde wordt voldaan.

Syntax:

while(expressie){

statements;

}

Voorbeeld: de tafel van 10

<php

$teller = 1;

$getal = 10;

while($teller <= 10){

echo “$teller maal $getal = “.($teller \* $getal) . “<br>”;

$teller++;

}

?>

## Variant: do-while()

Bij deze variant wordt de test pas uitgevoerd op het einde van de lus ipv in het begin. Dit wil dus zeggen dat de lus altijd minstens 1 maal wordt doorlopen.

Syntax:

do{

statements;

}while(expresiie);

# For()-lus (bepaalde herhaling)

De for()-lus is eigenlijk ook weer een variant op de while()-lus met dit verschil dat de initialisatie van een teller of index moet gebeuren en de verhoging (verandering) ervan automatisch. Deze lus heeft altijd 3 parameters.

* de eerste parameter specificeert een variabele en kent hieraan een waarde toe
* de tweede parameter is een voorwaarde die op waar moet komen om de lus te doorbreken
* de derde parameter bevat de verandering van de eerste parameter elke keer de lus wordt doorlopen.

Syntax:

for($teller; voorwaarde; teller-bewerking){

statements;

}

voorbeeld: de tafel van 10

<?php

$getal=10;

for($teller=1;$teller <= 10;$teller++){

echo “$teller maal $getal = “ . ($teller \* $getal) . “<br>”;

}

?>

## Oneindige lus

Wees steeds op je hoede voor oneindige lussen. Deze komen vaker voor dan je zou wensen en zijn meestal een gevolg van het vergeten van de voorwaarde te veranderen (in while-lus bvb $teller++;) of door je < of > fout te zetten in je for()-lus.

# Foreach()-lus

Dit is een speciale lus ontworpen om op een makkelijke en efficiënte manier arrays te doorlopen.

voorbeeld:

<?php

$club=array(“Ajax”,”PSV”,”Feijenoord”);

foreach($club as $key => $value){

echo $key . “ bevat “ . $value . “<br>”;

}

?>

0 bevat Ajax  
1 bevat PSV  
2 bevat Feijenoord

# Lussen onderbreken en beëindigen

## Break

Dit commando staat u toe een lus te beëindigen.

## Continue

dit commando slaat deze lus 1x over maar zal dan verder gaan met de volgende. Dit kan bijvoorbeeld handig zijn om een lus enkel te laten uitvoeren bij even getallen maar niet bij oneven.

## Exit

Dit is een algemener commando dat u toelaat om uw gans script te stoppen.

### Die()

Dit commando laat u ook toe om het ganse script te stoppen maar heeft ook een String-parameter waarlangs u een boodschap(foutmelding) kan tonen aan de gebruiker.

De 2 laatste komen nog veelvuldig voor in de hoofdstukken over php en mysql.

# Oefeningen:

1. Een blaadje papier is 0.4mm dik. Hoeveel keer moet ik dit plooien om minimum 1m papier te krijgen.
2. Maak een array met kleuren (minimum 4) en laat deze waarden tonen op het scherm.
3. Schrijf een script dat de getallen 1 tot en met $getal bij elkaar optelt en grafisch weergeeft.

vb. 1+2+3+4=10

1. Toon de tafel van een ingegeven getal maar toon alles mooi in een tabelvorm (xhtml)

Hoofdstuk 4: Formulieren

# Inleiding

Tot nu toe hebben we nog geen interactie gehad met de gebruiker wat natuurlijk wel juist de bedoeling is. Wat zijn dynamische websites die altijd hetzelfde doen. Onze eerste vorm van interactiviteit gaan we bekomen door gebruik te maken van simpele HTML-formulieren.

Vorig jaar hebben we al kennis gemaakt met formulieren al hebben we deze maar summier behandeld. Ze kwamen voor in de cursus XHTML waar we de inhoud lieten doormailen en daarna nog eens in de cursus Javascript waar we de inhoud ervan leerden controleren. Nu gaan we dus een stapje verder, we gaan de ingevulde gegevens gebruiken om als invoer te dienen voor onze scripts. Zoals we reeds vorig jaar behandelen zien we hier nu ook weer het IVU-principe terug. Invoer van gegevens, verwerking van de gegevens naar nuttige informatie en ten slotte het tonen van deze informatie.

# Get of Post

Het formulier kent 2 methodes om de gegevens te verzenden namelijk GET en POST. De verschillen zijn niet zuiver estetisch maar vooral functioneel.

Get: als je deze methode gebruikt verstuurt het een arraymet alle gegevens naar de betreffende pagina.

Voordelen: Gebruik:

Gegevens zijn zichtbaar Bladwijzers voor Internet browsers  
Gegevens zijn herbruikbaar Links in mails voor registratie, e-card, etc

Nadelen:

Onveilig voor wachtwoorden  
Beperkt aantal gegevens die kunnen worden door gegeven (100 karakters)

POST: als je deze methode gebruikt verstuurt het een arraymet alle gegevens naar de betreffende pagina maar op een onzichtbare manier.

Voordelen: Gebruik:

Onzichtbaar in het doorgeven van gegevens Wachtwoorden  
Variabelen hebben een onbeperkte lengte Meeste formulieren die niet als link worden gebruikt

Nadelen:

Kan niet worden hergebruikt

# Voorbeeld van Get-formulier:

Aanroepformulier:

<form action="welkom-get.php" method="GET">

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

Voornaam: <input type="text" name="voornaam" />

<input type="submit" />

</form>

PHP-code van welkom-get.php:

Welkom <?php echo $\_GET["naam"]; ?>.” “.<?php echo $\_GET["voornaam"]; ?>

Adresbalk: <http://localhost:8080/welkom-get.php?naam=Tubbax&Voornaam=Koenraad>

Als je een you-tube filmpje wil doorsturen maak je ook gebruik van een GET-parameter, probeer maar eens <http://www.youtube.com/watch?v=ov07yLusCKs>

# Voorbeeld van POST-formulier:

<form action="welkom-post.php" method="POST">

Login: <input type="text" name="login" />

Wachtwoord: <input type="text" name="wachtwoord" />

<input type="submit" />

</form>

PHP-code van welkom-post.php

<? $login=$\_POST[”login”];  
 $wachtwood =$\_POST[“wachtwoord”] ;

echo “Welkom “. $login . “, je wachtwoord is “.$wachtwoord;

?>

Adresbalk: http://localhost:8080/welkom-post.php

# Formuliervelden

|  |  |  |
| --- | --- | --- |
| Soort Element | Beschrijving | Mogelijke attributen |
| text | Invoerveld met één regel | size,value, name |
| checkbox | Keuzevakje | value,checked,name |
| radio | Keuzerondje | value,checked,name |
| submit | Knop voor het verzenden van het formulier | value, name |
| reset | Knop om de orginele staat van het formulier te herstellen | value, name |
| password | verborgen invoerveld | size, value, name |
| hidden | onzichtbaar veld | value, name |
| button | knop | value, name |
| image | afbeelding als knop | src,name |
| file | bestand | name, accept |

# Oefeningen:

1. Toon de tafel van een ingegeven getal maar toon alles mooi in een tabelvorm (xhtml)
2. Maak een rekenmachine die 2 getallen die ingevoerd worden door de gebruiker optelt.
3. Maak een formulier dat de naam en de leeftijd vraagt (3 radio buttons). Als de gebruiker <18 dan gratis, tussen 18 en 65 volle prijs, > 65 10% korting.

Hoofdstuk 5: Sessies

# Inleiding

In het vorige hoofdstuk hebben we reeds behandeld hoe je kan reageren op invoer van de gebruik door gebruik te maken van formulieren. Zo kunnen we dus makkelijk een webapplicatie maken maar er is een groot probleem. De webserver heeft geen geheugen van webpagina’s en vergeet dus alle variabelen die niet in een formulier zitten. Stel u nu voor dat u elke keer op elke pagina een onzichtbaar formulier zou moeten toevoegen om variabelen over te zetten. Dit zou niet handig werken en ten tweede onveilig voor wachtwoorden als deze gewoon leesbaar zouden zijn in de bron-weergave.

Dit wordt opgelost in php door een speciale array bij te houden namelijk de array SESSION.

# Sessie starten

Voor we gebruik kunnen maken van deze speciale array moeten we natuurlijk wel eerst een sessie starten. Dit doen we met session\_start().

<?php session\_start(); ?>

<html>

<body>

</body>

</html>

# Een sessievariabele gebruiken

Nadat de sessie is geopend kunnen we sessievariabelen aanmaken bvb. $SESSION[“naam”] =waarde.

<?php

session\_start();

// sessievariabele opslaan

$\_SESSION['email']=jan@kta.be;

?>

# Bestaande sessievariabele oproepen

<?php

//sessievariabele uitlezen

echo "Emailadres :". $\_SESSION['views'];

?>

# Sessie verwijderen

Sessies komen vaak terug op sites waar je je kan aanmelden en natuurlijk wil je je dan ook kunnen afmelden. Door dat afmelden worden dan al je sessie-variabelen vernietigt. Dit gebeurt met session\_destroy(). Als de browser wordt afgesloten wordt ook de sessie beïndigt. Voor de geïntresserden een sessie is ook maar een bepaalde tijd geldig (staat in session.gc\_maxlifetime), laat je pagina van het leerplatform maar eens een uurtje open staan.

<?php  
session\_destroy();

?>

Alternatief kan je ook een enkele sessievariabele verwijderen en toch de sessie nog behouden. Dit doe je dan met unset() functie.

<?php  
unset($\_SESSION['email']);

?>

Controle of een bepaalde variabele bestaat kan je dan weer met de isset()-functie.

<?php

session\_start();

if(isset($\_SESSION['email'])){  
 echo “welkom”;

}else{  
 echo “gelieve je eerste aan te melden”;  
}?>

# Oefeningen:

1. Maak een formulier met naam en adres, verwerk het formulier en sla dit op in sessievariabelen. Maak een andere pagina en toon naam en adres.
2. Maak een formulier om je aan te melden. Maak een 2de pagina waar je enkel de tekst ‘Welkom” te zien krijgt als je inderdaad bent aangemeld. Maak het op deze pagina ook mogelijk om de sessie te beëindigen.

Hoofdstuk 6: Cookies

# Inleiding

Een cookie staat dikwijls in een negatief daglicht aangezien het gegevens bijhoudt op onze pc. Sommige mensen denken dan ook dat een cookie alles opslaat wat je doet op je pc. Dit is natuurlijk niet waar, een cookie is gewoon een tijdelijk tekstbestandje dat elke keer als je die pagina opvraagt ook het tekstbestandje meestuurt. Dit is dan ook iets handig om een identificatiebestand ofzo bij te houden zodat een gebruiker zich niet telkens moet aanmelden.

# Cookie aanmaken

Om een cookie aante maken moeten we enkel setcookie() uitvoeren. Dit is zoiets als session\_start() en moet ook voor alle andere html-codes verschijnen.

Syntax

setcookie(naam, waarde, vervaldatum, plaats, domein);

Voorbeeld:

We maken een cookie om de gekozen taal op te slaan. De naam van de cookie wordt taal en de waarde is “nederlands”. We willen dat de cookie 2uur blijft bestaan:

<?php

setcookie("taal", "Nederlands", time()+7200);

?>

<html>

<body>

</body>

# Hoe een cookie opvragen?

Juist zoals bij formulieren en sessies kan je dit ook weer oproepen vanuit een $\_COOKIE arrray.

Voorbeeld:

<?php  
echo $\_COOKIE["taal"];  
?>

Je kan dan ook weer met de isset() functie nakijken of er al een cookie bestaat en er anders eentje plaatsen.

<html>

<body>

<?php

if (isset($\_COOKIE["taal"])){

echo "Welkom in het " . $\_COOKIE["taal"] . "!<br />";

}else{

echo "Welcome! Please choose a language next time<br />";

?>

</body>  
</html>

# Cookies verwijderen

Aangezien cookies met een vervaldatum werken kan je het verwijderen door een vervaldatum in het verleden te kiezen.

voorbeeld:

<?php

setcookie("taal", "", time()-3600);

?>

# Oefeningen:

1. Maak een formulier en sla de gegevens ervan op met een cookie.
2. Maak een pagina die je cookie uitleest en toont.
3. Maak een pagina die je cookie verwijdert.

Overzicht PHP-Mysql funties

#### Connectie maken met de databank

Om een connectie te maken met de te gebruiken databank kan je best gebruik maken van de functie:

**mysql\_connect(“host”,”loginnaam”,”wachtwoord”)**

Bij ons dus meestal $db = mysql\_connect(‘localhost’,’root’,’usbw’);

#### Databank selecteren

Als je een connectie hebt met de databanken moet je natuurlijk nog aanduiden welke databank je wil gebruiken. Dit kan met de functie: **mysql\_select\_db(‘naam van de databank’,’connectievariabele’);**

Bij ons dus myslq\_select\_db(‘voorbeeld’,$db) or die(“er is een fout opgetreden”.mysql\_error());

Deze 2 regels stoppen we meestal in een apart bestand dat we dan toevoegen aan elk php-bestand dat we gebruiken met <? include(‘naambestand.php’); ?>

#### Query uitvoeren

Als we een sql-query willen uitvoeren gaan we meestal een aparte variabele aanmaken met de naam $query. Hierin kan je dan rustig je sql-statement in vormen vb. SELECT id,naam FROM producten WHERE id = $\_POST[‘id’]. Je kan natuurlijk ook insert into, delete, update statements gebruiken.

Om de query dan uit te voeren (tot nu toe hebben we gewoon een String met een sql-statement gemaakt) kunnen we volgende regel gebruiken.

**$resultaat=mysql\_query($query) or die("Er is een onverwachte fout opgetreden: " . mysql\_error());**

Als er een fout optreedt wordt deze weergegeven wat soms wel handig kan zijn.

#### Resultaten van de query 1 per 1 afhandelen

In $resultaat zitten momenteel alle records die aan de query voldeden. Aangezien we niet opvoorhand kunnen voorspellen hoeveel dit er zijn gebruiken we een onbepaalde herhaling om deze 1 per 1 af te handelen.

**while($rij= mysql\_fetch\_array($resultaat)) {**

//code wat er met de record moet gebeuren vb echo($rij[‘id’]);

**}**

Inleiding beveiliging

Dit hoofdstuk gaat over de mogelijke problemen met het zo maar uitvoeren van SQL-queries. Verder bekijken we ook wat SSL eigenlijk is en waarom het belangrijk is.

Doelstellingen:

* Kunnen verklaren wat SQL-injectie is.
* Een mogelijk probleem kunnen herkennen.
* Kunnen verklaren wat het SSL-protocol doet
* Hoe krijg je een SSL-certificaat

SQL-injectie

De term SQL-injectie (Engels: SQL injection) wordt gebruikt voor een type kwetsbaarheid van computerapplicaties, **meestal webapplicaties**. Applicaties die informatie in een database opslaan maken gebruik van SQL om met de database te communiceren. **SQL-injectie kan gebeuren als invoer van gebruikers op onvoldoende gecontroleerde wijze wordt verwerkt in een SQL-statement**. Om de precieze werking van SQL-injectie te begrijpen is het belangrijk om te weten hoe SQL werkt.

Voorbeeld:

Een eenvoudig voorbeeld van een SQL-injectie is een eenvoudige HTML-formulier waarin u een voornaam ingeeft en daarvan de achternaam toont:

<form name="test" action="t.php" method="post">

Voornaam: <input type="text" name="voornaam" />

<input type="submit" value="Verzenden" />

</form>

Gezien dit fragment van HTML, kan men afleiden dat de makkelijkste (en slechtste) manier voor het script "t.php" aan het werk zou zijn voor het bouwen en uitvoeren van een database query die er als volgt uitziet:

"SELECT \*from klanten WHERE voornaam = '$persoon'";

Onder deze omstandigheden, indien de variabele "$ persoon" direct is overgenomen van de gebruiker, kan het login-script gemakkelijk worden misleid om te geloven dat een geldige naam is verstrekt door het spelen met de syntaxis van het SQL statement. Stel de volgende tekenreeks als de naam:

'OR '1=1

en we gaven "Erna" als gebruikersnaam. Nadat de variabelen zijn geïnterpoleerd, de query hierboven zou er zo uitzien:

‘SELECT id from logins WHERE username = 'Erna'or'a'=a';

Deze query retourneert alle rijen, omdat de laatste zin:

... or''a=a''

altijd evalueren op true .

Als we dit nu toepassen krijgen we alle achternamen in plaats van enkel die van Erna. Met dezelfde logica kunnen mensen met slechte bedoelingen voor meer schade aanrichten.

Oplossingen

Een manier om dit tegen te gaan is:

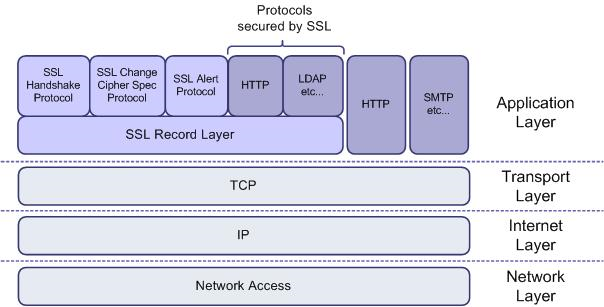
* escape de '$persoon' variabele met 'addslashes' of 'mysql\_real\_escape\_string'. Bijvoorbeeld:
* SELECT \*from klanten WHERE voornaam = '$persoon'='" . mysql\_real\_escape\_string($persoon) . "'");

Secure Sockets Layer

Deze beide protocollen leveren door middel van cryptografie zowel authenticatie als een beveiligde verbinding met het Internet. In alledaags gebruik wordt alleen de authenticiteit van de server gecontroleerd, terwijl de client geheel onbekend blijft. SSL versie 3.0 is ontwikkeld door Netscape en uitgebracht in 1996. Deze versie werd later de basis voor het **Transport Layer Security** (**TLS**)

Toepassing

Ondanks dat zowel SSL als TLS veiligheid kan bieden aan elk protocol dat gebruik maakt van TCP, wordt SSL het meest gebruikt voor HTTPS, bijvoorbeeld ter beveiliging van creditcard-gegevens.

Indien men het gebruikt om HTTP te beveiligen, wordt het "http://" gedeelte in een URL vervangen door "https://", waarbij de s staat voor "secure". Ook andere klassieke TCP/IP-applicatielaagprotocollen waarbij de informatie (zoals wachtwoorden) normaal onversleuteld over het netwerk gaan, kunnen met SSL/TLS worden beveiligd. 

TLS protocol gebruikt certificaten om de uitgewisselde gegevens te authenticeren en het verzekeren van de privacy. Elk certificaat bevat een publieke sleutel. De eigenaar van het certificaat bezit een privésleutel die geassocieerd is met de publieke sleutel in het certificaat. Er zijn verschillende bedrijven die certificaten uitreiken. Een van de bekendste is VeriSign.

OpenSSL

Dit is een OpenSource-oplossing die je ook een certificaten kan aanbieden/aanmaken met publieke sleutels.