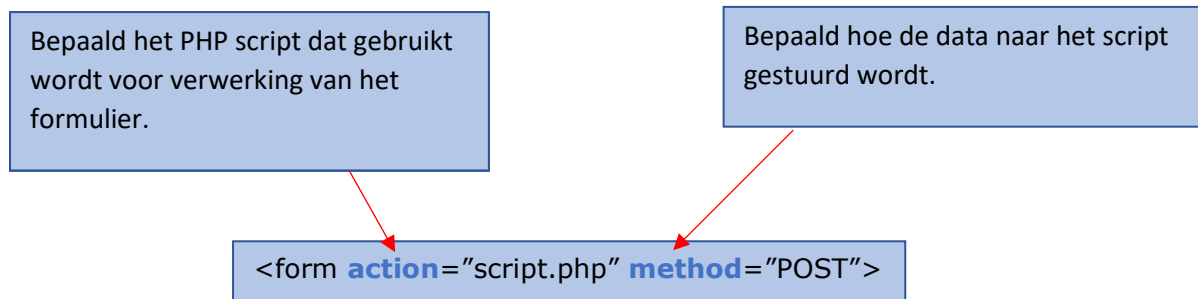


Acties uitvoeren

Het verwerken van gegevens die zijn verzonden vanuit een HTML-formulier op een webpagina is misschien wel de meest fundamentele taak die wordt uitgevoerd door een PHP-script.



PHP biedt een speciale 'superglobal' `$_POST`-arrayvariabele waarin formuliergegevens worden opgeslagen die zijn ingediend via de **POST**-methode. Het maakt een array-element met dezelfde naam als elke verzonden formulier-elementnaam, waarbij de waarde wordt opgeslagen die ze bevatten.

Bijvoorbeeld:

```
<input type="text" name="email">
```

Wanneer het formulier wordt ingediend, maakt PHP een arrayelement met de naam **`$_POST['email']`** dat de waarde bevat die is ingevoerd in de invoer van de gebruiker.

- 1 Maak een HTML-document met een formulier met drie tekst-invoervelden voor het indienen van gegevens.

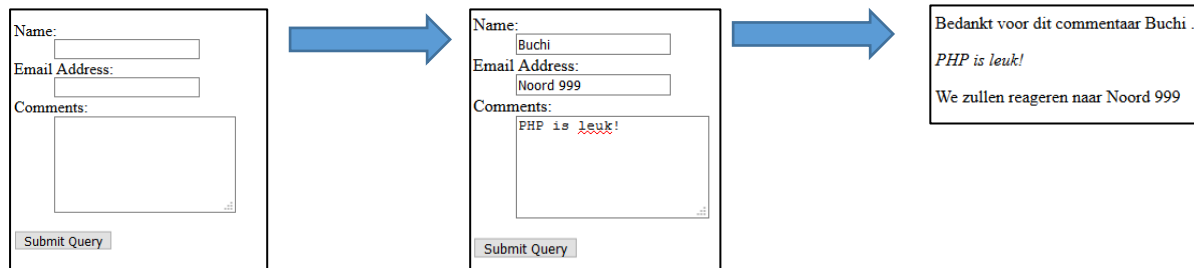
```
<form action="action_handler.php" method="POST">
<dl>
  <dt>Name:
    <dd><input type="text" name="name">
  <dt>Email Address:
    <dd><input type="text" name="mail">
  <dt>Comments:
    <dd><textarea rows="5" cols="20" name="comment"></textarea>
</dl>
<p><input type="submit"></p>
</form>
```

- 2 Sla het document op in de map **/htdocs** als **action.php**.

- Maak vervolgens een HTML-document aan waarin doormiddel van een script formulierwaarden worden toe gewezen aan gelijknamige PHP-variabelen voor uitvoer.

```
<?php
$name = $_POST['name'];
$mail = $_POST['mail'];
$comment = $_POST['comment'];
echo "<p>Bedankt voor dit commentaar $name ...</p>";
echo "<p><i>$comment</i></p>";
echo "<p>We zullen reageren naar $mail</p>";
?>
```

- Sla het document op in **/htdocs** als **action_handler.php**. Open vervolgens **action.php** via localhost en verzend enkele gegevens om te zien dat deze wordt weergegeven op de reactiepagina die wordt geproduceerd door de PHP-script.



Controleren van ingestelde waarden

PHP kan eenvoudig garanderen dat de gebruiker gegevens heeft ingevoerd in ingediende HTML-formulervelden met behulp van de ingebouwde functie **isset()**.

De functie neemt de naam van een variabele als zijn argument en geeft alleen **TRUE** (WAAR) als resultaat als die variabele geen **NULL** is.

- Maak een HTML-document met een formulier met **radio buttons** (keuzerondjes) en een enkel veld voor het indienen van gegevens.

```
<form action="isset_handler.php" method="POST">
<fieldset>
  <legend>Wat voor soort taal is PHP?</legend>
  Scripting<input type="radio" name="definition" value="Scripting"><br>
  Markup<input type="radio" name="definition" value="Markup"><br>
  Programming<input type="radio" name="definition" value="Programming">
</fieldset>
<p><input type="submit"><p>
</form>
```

- Sla het document op in **/htdocs** als **isset.php**

3

Maak vervolgens een HTML-document met een PHP-script om een variabele met een ingezonden waarde te initialiseren - of met **NULL** als er geen waarde is ingediend.

```
<?php
if ( isset( $_POST['definition']))
{
    $definition = $_POST['definition'];
}
else
{
    $definition = NULL;
}

# Antwoord instructies
if ( $definition != NULL )
{
    if ( $definition != 'Scripting' )
    {
        echo "$definition is Niet correct";
    }
    else
    {
        echo "$definition is Correct";
    }
}
else
{
    echo 'Je moet een antwoord selecteren';
}
?>
```

4

Sla het document op in **/htdocs** als **isset_handler.php**. Open vervolgens **isset.php** via localhost en verzend het formulier om de antwoorden te zien.

Wat voor soort taal is PHP?

Scripting ☐

Markup ☐

Programming ☐

Submit Query

Je moet een antwoord seleteren

Scripting is Correct

Controleren van ingestelde waarden

Om ervoor te zorgen dat de gebruiker iets in een tekstveld heeft ingevoerd, biedt PHP een **empty()** functie die een variabel argument accepteert en **TRUE** (WAAR) retourneert als de waarde ervan een lege tekenreeks, nul, **NULL** of **FALSE** (ONWAAR) is.

Om ervoor te zorgen dat de gebruiker gegevens in numerieke indeling heeft ingevoerd, biedt PHP een **is_numeric()** functie die een variabel argument accepteert en alleen **TRUE** retourneert als de waarde een getal is.

Meer specifieke gegevensvalidatie kan worden uitgevoerd met behulp van de PHP **preg_match()** functie die een *"reguliere expressie"* argument accepteert en een variabel argument waarin een patroon wordt gezocht. Dit wordt meestal gebruikt om ervoor te zorgen dat een e-mailadres de verwachte indeling heeft.

- 1 Maak een HTML-document met een formulier met twee tekstvelden voor het indienen van gegevens.

```
<form action="valid_handler.php" method="POST">
  <fieldset>
    <legend>Voer een hoeveelheid en email address in</legend>
    <p>Hoeveelheid : <input type="text" name="quantity"></p>
    <p>Email adres : <input type="text" name="email"></p>
  </fieldset>
  <p><input type="submit"></p>
</form>
```

- 2 Sla het document op in **/htdocs** als **valid.php**

- 3 Maak vervolgens een HTML-document met een PHP-script om een variabele met een ingediende waarde te initialiseren - of met **NULL** als er geen waarde is opgegeven.

```
# Empty check.
if ( !empty ( $_POST['quantity'] ) )
{
    $quantity = $_POST['quantity'] ;
    # Format check.
    if ( !is_numeric( $quantity ) )
    {
        $quantity = NULL ;
        echo 'Hoeveelheid moet een numerieke waarde zijn.<br>' ;
    }
}
```

```

    }
}
else
{
    $quantity = NULL ;
    echo 'Je moet een hoeveelheid invoeren<br>' ;
}
# Empty check.
if ( !empty ( $_POST['email'] ) )
{
    $email = $_POST['email'] ;
    # Format check.
    $pattern = '/\b[\w.-]+\@([\w.-]+\.[A-Za-z]{2,6})\b/' ;
    if ( !preg_match( $pattern , $email ) )
    {
        $email = NULL ;
        echo 'Email adres heeft een incorrect formaat' ;
    }
}
else
{
    $email = NULL ;
    echo 'Je moet een email adres invoeren' ;
}
# Success response.
if ( ( $quantity != NULL ) && ( $email != NULL ) )
{
    echo "Email : $email<br>Hoeveelheid : $quantity " ;
}

```

4

Sla het document op in **/htdocs** als **valid_handler.php**. Open vervolgens **valid.php** via localhost en verzend het formulier om de antwoorden te zien.

Voer een hoeveelheid en email address in

Hoeveelheid :

Email adres :

Email : juancho@hotmail.com
 Hoeveelheid : 7

Verborgen gegevens versturen

Naast gegevens die door de gebruiker zijn ingevoerd in invoervelden voor HTML-formulier, kunnen gegevens die zijn gegenereerd door PHP worden ingediend met verborgen velden in een HTML-formulier. Net als reguliere invoervelden, zenden deze ook een *name* = *value* paar aan het formulier handler script maar hier wordt de waarde geleverd door PHP, niet door de gebruiker.

Gegevens die worden door gestuurd met behulp van verborgen formulervelden kunnen doorgaans de aanmeldingsnaam van de gebruiker bevatten en een tijdstempel van de inzending.

- 1 Maak een HTM-document met een PHP-script om de tijdzone te definiëren en twee variabelen te initialiseren.

```
<?php
date_default_timezone_set('UTC') ;
$time = date( ' H:i , F j ' ) ;
$user = 'Janchi' ;

echo '
<form action="hidden_handler.php" method="POST">
<fieldset>
<legend>Send us your comments</legend>
<textarea rows="5" cols="20" name="comment"></textarea>
<input type="hidden" name="user" value=" ' . $user . ' ">
<input type="hidden" name="time" value=" ' . $time . ' ">
</fieldset>
<p><input type="submit" ></p>
</form> ' ;
?>
```

- 2 Sla het document op in **/htdocs** als **hidden.php**. Open vervolgens voer het script uit om de responses te zien.

3

Maak vervolgens een HTML-document met een PHP script om een variabele met een ingezonden waarde te initialiseren - of met **NULL** als er geen waarde is ingediend.

```
<?php
# Controleer of leeg is
if ( !empty ( $_POST['comment'] ) )
{
    $comment = $_POST['comment'] ;
}
else
{
    $comment = NULL ;
    echo 'You must enter a comment' ;
}

# Controleer of waardes zijn toegekend
$time = ( !isset ( $_POST['time'] ) ) ? NULL : $_POST['time'] ;
$user = ( !isset ( $_POST['user'] ) ) ? NULL : $_POST['user'] ;

# Geef Succes response
if ( ( $comment != NULL ) && ( $time != NULL ) && ( $user != NULL ) )
{
    echo "<p>Comment received : \" $comment \"<br>From $user at $time
</p>" ;
}
?>
```

4

Sla het document op in **/htdocs** als **Hidden_handler.php**. Open het script **hidden.php** via *localhost* om de responses te zien.

Send us your comments

PHP is Geweldig!

Submit Query

Comment received : " PHP is Geweldig! "

From Janchi at 11:32 , October 1

Aanmeldingen verwerken

PHP biedt een speciale "superglobal" **\$_SERVER** array variabele met een **REQUEST_METHOD** element waarin het type verzoek wordt opgeslagen. Wanneer een formulier de **POST**-methode gebruikt voor submitatie, worden twee verschillende typenmeldingen naar het PHP-script gemaakt wanneer een pagina wordt geladen. Wanneer een formulier de **POST**-methode gebruikt voor submitaties, worden twee verschillende

typenmeldingen naar het PHP-script gemaakt wanneer een pagina wordt geladen. Daarom kan de **REQUEST_METHOD** worden onderzocht om te bepalen welke koers moet worden gevolgd.

- 1 Maak een HTML-document met een PHP-script dat de methode voor het opvragen van pagina's onderzoekt.

```
if ( $_SERVER[ 'REQUEST_METHOD' ] != 'POST' )
{
    echo '
    <form action="single.php" method="POST">
    <fieldset>
    <legend>Send us your comments</legend>
    <textarea rows="5" cols="40" name="comment"></textarea>
    </fieldset>
    <p><input type="submit" ></p>
    </form> ' ;
}
else
{
    if ( !empty ( $_POST['comment'] ) )
    {
        $comment = $_POST['comment'] ;
        echo "Comment : $comment " ;
    }
    else
    {
        $comment = NULL ;
        echo 'You must enter a comment' ;
    }
}
```

- 2 Sla het document op in **/htdocs** als **single.php**. Open vervolgens voer het script uit om de responses te zien.

Send us your comments

Submit Query

You must enter a comment

Send us your comments

PHP kan makkelijk formulieren tonen invoer
verwerken in een enkele script.

Submit Query

Comment : PHP kan makkelijk formulieren tonen invoer verwerken in een enkele script.

Kleverige formulieren maken

Wanneer een gebruiker probeert een formulier in te dienen waarin een vereist veld niet is ingevuld, is het wenselijk om de vermeldingen in de velden die ze hebben ingevuld te behouden door het formulier "plakkerig" te maken.

Het toewijzen van de relevante ingediende veldgegevens uit de array **\$ _POST** aan de HTML-tag's **value** eigenschap met **echo** behoudt de invoer wanneer het formulier opnieuw wordt weergegeven om te worden voltooid. Als voorzorgsmaatregel moet de instructie de functie **isset()** gebruiken om te controleren of het veld feitelijk is voltooid voordat het wordt ingediend.

Een foutbericht voor elk onvolledig veld kan worden toegewezen aan een array om weglatingen te melden wanneer het formulier opnieuw wordt weergegeven, of een bericht dat het succes bevestigt wanneer alle velden zijn voltooid.

- 1 Maak een HTML-document met een sticky-formulier met twee tekstvelden voor naam en e-mailadres.

```
<?php
# Check form submitted.
if ( $_SERVER[ 'REQUEST_METHOD' ] == 'POST' )
{
    # Initialize an error array.
    $errors = array();

    # Check for a name & email address.
    if ( empty( $_POST[ 'name' ] ) ) { $errors[] = 'name' ; }
    else { $name = trim( $_POST[ 'name' ] ) ; }

    if ( empty( $_POST[ 'email' ] ) ) { $errors[] = 'email' ; }
    else { $email = trim( $_POST[ 'email' ] ) ; }
```

```
# Report result.
if( !empty( $errors ) )
{
    echo 'Error! Please enter your ' ;
    foreach ( $errors as $msg ) { echo " - $msg " ; }
}
else { echo "Success! Thanks $name " ; }
}
?>
```

```
<!-- Display body section with sticky form. -->
<form action="sticky.php" method="POST">
<p>Name: <input type="text" name="name" value="<?php if
(isset($_POST['name'])) echo $_POST['name']; ?>"> </p>
<p>Email: <input type="text" name="email" value="<?php if
(isset($_POST['email'])) echo $_POST['email']; ?>"> </p>
<p><input type="submit"></p>
</form>
```

- 2 Sla het document op in **/htdocs** als **sticky.php**. Open vervolgens voer het script uit om de responses te zien.



Omringende formulieren

PHP staat toe dat meerdere bestanden samen worden gevoegd om een enkele webpagina te maken. Dit betekent dat dynamische formulieren kunnen worden omgeven door statische inhoud van kop- (**header**) en voetteksten (**footer**) die gemakkelijk kunnen worden hergebruikt om andere pagina's binnen een website te maken.

De functie PHP **include()** neemt de **URL** van een ander bestand dat moet worden opgenomen in een webpagina als het argument. Typisch inbegrepen statische inhoudsbestanden zijn HTML en bevinden zich in een map "*includes*" in de map **/htdocs** van uw webserver.

PHP heeft ook een **require()** functie. Deze werkt op de zelfde manier als de **include()** functie met een klein verschil. Als een **include()** functie mislukt wordt er een waarschuwing gegeven maar de browser gaat verder met de uitvoer van het script. Bij de **require()** functie wordt er een error melding getoond en de browser stopt met de uitvoer van het script.

- 1 Maak een HTML-document met de inhoud van de statische (header) koptekst en een PHP-fragment om de paginatitel te schrijven.

```
<!DOCTYPE HTML>
<html lang="en">
<head>
<meta charset="UTF-8">
<title><?php echo $page_title; ?></title>
<link rel="stylesheet" href="includes/style.css">
</head>

<body>
<header><h1>Page Header</h1></header>
```

- 2 Maak een HTML-document met voettekstinhoud (footer).

```
<footer><p>Page Footer</p></footer>
</body>
</html>
```

- 3 Maak een map met de naam *"includes"* in **/htdocs** en sla beide documenten op als **header.html** en **footer.html**.

- 4 Maak een CSS-stijlblad om de kop- en voettekstinhoud (en regels voor de presenteren van de latere voorbeelden)

```
header > h1 { border-bottom: 1px dashed black ; font-style: italic ; font-size: x-
large; }
footer > p { border-top: 1px dashed black ; font-style: italic ; }
table { border-spacing:5px; width:530px;}
th { color:#FFF; background:#000; text-align:left; }
td { border-bottom:1px solid black; background:#F0F0F0; padding:3px; text-
align:left; vertical-align:top;}
p#err_msg { color:#F00; font-weight:bold; }
```

- 5 Sla het stijlblad op in de map / includes.

- 6 Maak een PHP-script dat de titel van de webpagina instelt en een formulier omgeeft met kop- en voettekst.

```
<?php
$page_title = 'PHP Include' ;
include ( 'includes/header.html' ) ;

echo '<form action="include.php" method="POST">
<p>Name: <input type="text" name="name"> </p>
<p>Email: <input type="text" name="email" ></p>
<p><input type="submit"></p>
</form>' ;
include ( 'includes/footer.html' ) ;
?>
```

- 7 Sla het PHP-script in **/htdocs** op als **include.php** en open het vervolgens via **localhost** om de paginatitel en een formulier omgeven door statische inhoud te bekijken.

Page Header

Name:

Email:

Page Footer

Verbonden informatie toevoegen

PHP gedefinieerde '*superglobal*'-variabelen, zoals de **\$_SERVER** array variabele die de paginaverzoekmethode opslaat en de **\$_SESSION** array variabele die gebruikersgegevens opslaat over pagina's van een website totdat de gebruiker stopt, zijn toegankelijk in alle scopes van elk script.

Wanneer een HTML-formulier gegevens verzendt via de **POST**-methode, slaat de array **\$_POST** de formulierwaarden op in arrayelementen met dezelfde naam. De waarde van een ingediend veld met de naam "stad" is dus toegankelijk via **\$_POST['stad']**.

Op dezelfde manier slaat de array **\$_GET** de waarden op in array-elementen wanneer gegevens worden ingediend met de methode **GET**. De waarde van de ingediende gegeven met de naam "id" is dus toegankelijk via **\$_get['id']**. Dit wordt meestal gebruikt om gegevens naar een script te verzenden door een *name = value* toe te voegen aan de URL in een HTML-hyperlink zoals dit:

```
<a href="www.voorbeeld.com/script.php?id=1">
```

Het script dat de gegevens ontvangt, benoemt door de hyperlink, kan dan de gepasseerde waarde gebruiken om een antwoord te bepalen.

Informatie verstuurd met de **POST** methode zijn niet zichtbaar in the URL. die met de **GET** methode wel.

- 1 Maak een HTML-document met een PHP-script dat een element met de naam "id" zoekt binnen de array **\$_GET**.

```
<?php
# Handle the submitted link data.
if ( isset( $_GET['id'] ) )
{
    $id = $_GET['id'] ;
    switch( $id )
    {
        case 1 : echo 'Konijn selected<hr>' ; break ;
        case 2 : echo 'Hond selected<hr>' ; break ;
        case 3 : echo 'Poes selected<hr>' ; break ;
    }
}
# Display hyperlinks with appended id values.
echo '<h1>Selecteer een vriend</h1>' ;
echo ' <p><a href="link.php?id=1">Konijn</a> | ' ;
echo ' <a href="link.php?id=2">Hond</a> | ' ;
echo ' <a href="link.php?id=3">Kat</a></p>' ;
?>
```

- 2 Sla het PHP-script in **/htdocs** op als **link.php** en open het vervolgens via **localhost** om de paginatitel en klik op de koppelingen om de antwoorden voor de doorgegeven waarden te bekijken.



Locatie verplaatsen

PHP biedt een `header()` -functie die een argument accepteert om een onbewerkte HTTP-"header" op te geven die naar de webbrowser moet worden verzonden. Meest nuttig is dat de PHP-header `()` -functie de "Locatie" -header kan instellen om de browser naar een nieuwe pagina te leiden met behulp van deze syntaxis:

```
Header('Location: www.voorbeeld.com/page.php');
```

Gebruikersgegevens kunnen worden bewaard bij het verplaatsen naar een nieuwe pagina door deze eerst op te slaan in de speciale array-array PHP **\$_SESSION**. Dit is pas mogelijk wanneer de functie **session_start()** is aangeroepen in het script.

Variabele waarden kunnen vervolgens worden opgeslagen in elementen met dezelfde naam van de array **\$_SESSION**, zodat ze beschikbaar

worden vanaf andere pagina's op de website totdat de gebruiker de site afsluit of totdat de functie **session_destroy()** in het script wordt aangeroepen.

- 1 Maak een HTML-document met een PHP-script dat een element met de naam "id" binnen de array **\$_POST** zoekt.

```
<?php

if ( isset( $_POST['id'] ) )
{
    $id = $_POST['id'] ;
    if( $id== 123 )
    {
        session_start() ;
        $_SESSION['id'] = $id ;
        header( 'Location: location.php' ) ;
        exit() ;
    }
    else
    {
        echo "<p>$id is an incorrect ID!</p>" ;
    }
}
echo '
<form action="header.php" method="POST">
<fieldset>
<legend>Enter Your User ID</legend>
<p>ID : <input type="text" name="id"> </p>
</fieldset>
<input type="submit">
</form> ' ;
?>
```

- 2 Sla het PHP-script in **/htdocs** op als **header.php**

- 3 Maak een doel-HTML-document met daarin een PHP-script om de behouden \$ array-waarde \$ _SESSION weer te geven.

```
<?php
session_start() ;
if ( isset( $_SESSION['id'] ) )
{
    $id = $_SESSION['id'] ;
    echo "Welcome user ID # $id" ;
}
?>
```

4

Sla dit document op als **location.php** in **/htdocs** en open **header.php** via **localhost** en geef een id-waarde mee om de resultaten te zien.

