

Webmediaproductie 5

PHP - Forms

Kristof Michiels

Forms?

HTML Forms

Laten interactie toe tussen bezoekers en toepassing

- Zoekformulieren
- Loginformulieren
- Ratingsystemen
- Gebruikerssettings
- Beheerformulieren

HTML Forms

- Informatie creëren, updaten, verwijderen
- Bestanden opladen
- Heel belangrijk onderdeel van de UI, de gebruikersinterface
- Cruciaal om dit gebruiksvriendelijk te maken!
- Bepaalt mee het gezicht van je toepassing

Forms: een nieuw PHP project

(met tegelijk nog een beetje herhaling van de concepten)

Een nieuw project

- Maak een nieuwe folder aan
- Neem kopieën van de folders templates en classes uit de vorige oefening
- Maak een lege views folder aan
- Maak in de root van je project een nieuwe index.php aan met volgende code:

```
<?php
    include_once "classes/Pagina_Data.class.php";
    $paginaData = new Pagina_Data();
    $paginaData->titel = "HTML Forms met PHP";
    $paginaData->content = "<nav>Hier komt binnenkort de navigatie...</nav>";
    $paginaData->content .= "<div>...straks komt hier een form</div>";
    $pagina = include_once "templates/pagina.php";
    echo $pagina;
```

Dynamische navigatie aanmaken

- We gaan 2 verschillende forms aanmaken
- We zullen een site navigatie nodig hebben om te navigeren tussen de beide forms
- Maak een nieuwe views/navigatie.php aan

```
<?php
    return "
        <nav>
            <a href="index.php?pagina=zoeken">Zoeken op Google</a>
            <a href="index.php?pagina=quiz">Dynamische quiz</a>
        </nav>
    ";
```

Dynamische navigatie aanmaken

We passen index.php aan om de navigatie te tonen

```
<?php
    include_once "classes/Pagina_Data.class.php";
    $paginaData = new Pagina_Data();
    $paginaData->titel = "HTML Forms met PHP";
    $paginaData->content = include_once "views/navigatie.php";
    $paginaData->content .= "<div>...straks komt hier een form</div>";
    $pagina = include_once "templates/pagina.php";
    echo $pagina;
```


Pagina views aanmaken voor de form

- We maken 2 views aan: zoeken.php en quiz.php
- Ze bevinden zich allebei in de views folder

```
<?php
//code voor views/zoeken.php
return "Zal weldra het zoekformulier tonen";
```

```
<?php
//code voor views/quiz.php
return "De quiz komt hier te staan";
```

pagina views tonen via index.php

We moeten enkele aanpassingen maken aan onze index.php file

```
<?php
    include_once "classes/Pagina_Data.class.php";
    $paginaData = new Pagina_Data();
    $paginaData->titel = "HTML Forms met PHP";
    $paginaData->content = include_once "views/navigatie.php";
    $navigatieIsClicked = isset($_GET['pagina']);
    if ( $navigatieIsClicked ) {
        $bestandTeLaden = $_GET['pagina'];
    } else {
        $bestandTeLaden = "zoeken";}
    $paginaData->content .=include_once "views/$bestandTeLaden.php";
    $pagina = include_once "templates/pagina.php";
    echo $pagina;
```

Een eenvoudig zoekformulier

- HTML forms worden aangemaakt met `<form>` elementen
- Er zijn nog andere HTML elementen die speciaal ontwikkeld zijn voor forms
- Er zijn verschillende HTML elementen die specifiek gemaakt zijn voor forms
- Meest essentiële: het `<input>` element
- Een `<form>` element is zoals een `<a>`: action is te vergelijken met href-eigenschap
- Na klikken op submit knop zal de browser een nieuwe URL aanvragen. Dus net zoals klikken op een a-element

Een eenvoudig zoekformulier

We passen zoeken.php als volgt aan:

```
<?php
    return "
        <form method="get" action="http://google.be/search">
            <input type="text" name="q">
            <input type="submit" value="zoek via Google">
        </form>";
```

Een eenvoudig zoekformulier

- het action-kenmerk: daar is waar de browser heen gaat wanneer de form wordt doorgestuurd
- Er wordt nog een URL variabele aan toegevoegd (hier q, de naam van het input element) alsook een waarde
- Dit is hoe forms kunnen werken: ze kunnen URL variabelen encoderen in HTTP requests
- URL variabelen kunnen dan via PHP worden gebruikt

```
<?php
    return "
        <form method="get" action="http://google.be/search">
            <input type="text" name="q">
            <input type="submit" value="zoek via Google">
        </form>";
```

Het method kenmerk

- URL variabelen die kunnen gezien worden in de URL => via de GET method
- Er is een limiet van ongeveer 2000 karakters
- Pagina's kunnen gemakkelijk gebookmarked worden => perfect voor sitenavigatie

```
<?php
    return "
        <form method="get" action="http://google.be/search">
            <input type="text" name="q">
            <input type="submit" value="zoek via Google">
        </form>";
```

Even iets anders: PHP functies

- Een krachtige eigenschap van PHP is de mogelijkheid om functies te creëren en uit te voeren vanuit je code
- Een functie is een blok met code die je kan declareren binnen je code en die je op een later moment kan oproepen
- Syntax: eerst declareren met een function keyword gevolgd door de functienaam
- Mag cijfers en letters bevatten alsook underscores. Mag niet beginnen met een cijfer
- Functiecode zit vervat in accolades

```
function functieNaam () {  
    //functie statements komen hier  
}
```

PHP functies

- Een functie wordt pas uitgevoerd als hem expliciet oproept
- Je kan een functie uiteraard verschillende keren oproepen

```
<?php
    function geefParagraaf(){
        echo "<p>Deze paragraaf komt via een functie</p>";
    }
    p();
    p();
```


PHP functies

- Hier heeft de functie geen echo meer, maar gebruiken we return
- Veel beter om 1 enkele return te gebruiken in je code
- Goede vuistregel: geen echo gebruiken in een functie

```
<?php
    function geefParagraaf(){
        return "<p>Deze paragraaf komt via een functie</p>";
    }
    $output = geefParagraaf();
    $output .= "<h1>Een header h1</h1>";
    $output .= geefParagraaf();
    echo $output;
```

flexibiliteit met functieargumenten

- Functies kunnen voorzien zijn van argumenten
- Hier declareren we een variabele (\$content) binnen de haakjes in de functiedeclaratie
- De waarde wordt aan \$content gegeven telkens de functie wordt uitgevoerd
- Functieargumenten stellen je in staat 1 functie te gebruiken met veel verschillende waardes

```
<?php
    function geefParagraaf($content) {
        return "<p>$content</p>";
    }
    $output = geefParagraaf("Ik wil deze tekst in mijn eerste paragraaf");
    $output .= geefParagraaf("...en deze tekst in mijn tweede");
    echo $output;
```

Een form aanmaken voor een quiz

In de views folder maken we een file aan en noemen ze quiz-form.php

```
<?php
    return "
        <form method="post" action="index.php?pagina=quiz">
            <p>Is PHP leren moeilijk?</p>
            <select name="antwoord">
                <option value="ja">Ja, het is moeilijk</option>
                <option value="nee">Nee, toch niet</option>
            </select>
            <input type="submit" name="quiz-submitted" value="post">
        </form>";
```

Een form aanmaken voor een quiz

- We gebruiken een select HTML element, met geneste option elementen.
- We gebruiken hier de POST method. Geen 2000 karakterlimiet meer, ook gebruikt om files op te laden
- Bovendien: niet meer zichtbaar in de URL
- Worden verborgen opgestuurd. Perfect voor grotere forms, of voor gevoelige informatie,...

\$_POST Superglobal array

- We gaan nu quiz.php updaten (code op volgende slide)
- PHP heeft ook een superglobal array genaamd \$_POST
- Deze kan gebruikt worden om de URL variabelen op te vragen in het geval van een POST method
- Wanneer je index.php?pagina=quiz intikt, dan zie je de quiz

\$_POST Superglobal array

```
<?php
    $quizIsSubmitted = isset( $_POST['quiz-submitted'] );
    if ($quizIsSubmitted ){
        $antwoord = $_POST['antwoord'];
        $output = toonQuizAntwoord($antwoord);
    } else {
        $output = include_once "views/quiz-form.php";
    }
    return $output;
function toonQuizAntwoord($antwoord){
    $terugSturen = "<p>Je klikte op $antwoord</p>";
    $terugSturen .= "<p><a href='index.php?pagina=quiz'>Quiz opnieuw doen?</a></p>";
    return $terugSturen;
}
```

`$_POST` Is een Array

- Net als `$_GET` is ook `$_POST` een superglobal array
- Een array is een datatype dat meerdere items kan bevatten
- Elk item is opgeslagen onder een indexnaam
- `print_r()` is een functie om alle elementen in een array te inspecteren. Het resultaat zal ongeveer zijn zoals in volgend codevoorbeeld

\$_POST Is een Array

```
<?php
    $mijn['naam'] = "John";
    $mijn['geboortejaar'] = 1985;
    $mijn['lichaamslengte'] = "184cm";
    $out = "<pre>";
    $out .= print_r($mijn, true);
    $out .= "</pre>";
    echo $out;
```

```
Array (
    [naam] => John
    [geboortejaar] => 1985
    [lichaamslengte] => 184cm
)
```


terug naar quiz.php

- Quiz.php updaten
- Surfen naar `index.php?page=quiz`
- Je kan op deze manier het antwoord zien, maar ook alle items in de `$_POST` array
- Hier het resultaat, op volgende slide de code

```
Array (  
  [antwoord] => ja  
  [quiz-submitted] => post  
)
```

terug naar quiz.php

```
<?php
    $quizIsSubmitted = isset( $_POST['quiz-submitted'] );
    if ( $quizIsSubmitted ){
        $antwoord = $_POST['antwoord'];
        $output = toonQuizAntwoord($antwoord);
        $output .= "<pre>";
        $output .= print_r($_POST, true);
        $output .= "</pre>";
    } else {
        $output = include_once "views/quiz-form.php";
    }
    return $output;
function toonQuizAntwoord( $antwoord ){
    $terugSturen = "<p>Je klikte op $antwoord</p>";
    $terugSturen .= "<p><a href='index.php?pagina=quiz'>Quiz opnieuw doen?</a></p>";
    return $terugSturen;}
```

if else

Hebben de volgende gedaante

```
if ( Booleaanse expressie ) {  
    // code blok  
}
```

```
if ( booleaanse expressie ) {  
    // code blok  
} else {  
    // andere blok code  
}
```

Het quiz antwoord evalueren

- We updaten nu quiz.php
- Om te checken of 2 waarden gelijk zijn gebruiken we ===
- Soms zie je ook ==. Is bijna hetzelfde: <http://php.net/manual/en/language.operators.comparison.php>

```
function toonQuizAntwoord($antwoord){  
    $terugSturen = "<p>Je klikte op $antwoord";  
    if ( $antwoord === "ja" ){  
        $terugSturen .= " - Oefenen oefenen oefenen!";  
    }  
    $terugSturen .= "</p>";  
    $terugSturen .= "<p><a href='index.php?pagina=quiz'>Quiz opnieuw doen?</a></p>";  
    return $terugSturen;  
}
```

Oefening

- Voeg aan de voorbeeldapp een derde form toe die geld converteert van een bedrag in euro naar de andere geselecteerde munt
- Kies vrij (met een select) 5 munteenheden waartussen kan worden gekozen
- Gebruik de wisselkoersen van vandaag. Je kan deze koersen met wat Googlewerk meteen vinden
- Werk in je code zo veel mogelijk met functies
- Zorg ervoor dat je de folder en gekozen bestandsstructuur goed begrijpt
- Style met CSS

WMP5 - PHP - Forms

kristof.michiels01@ap.be