

---

## 5 Formulieren

We kunnen pas van een dynamische website spreken als verschillende invoeringen door de client tot verschillende resultaten en reacties leiden. Omdat een muisklik alleen niet veel toepassingsmogelijkheden biedt, moeten er formulieren gebruikt worden. HTML5 biedt de webprogrammeur een hele waaier aan formulierelementen.

Formulieren kunnen in elk normaal HTML-document worden opgenomen. Je hoeft hiervoor geen speciale voorzorgsmaatregelen te treffen of acties te ondernemen. Formulieren kunnen prima gecombineerd worden met gewone tekst., opmaakcode, tabellen, enz. Op de plek waar je in het webdocument een formulier wilt opnemen, plaats je het element `<form>...</form>`.

Alle formulierobjecten staan binnen dit element. Binnen het formulierelement kunt u desgewenst allerlei andere opmaakcodes opnemen (alineas, lijsten, tabellen, ...). De basisopbouw van een formulier ziet er als volgt uit:

```
<form action="filename" method="post/get">
  formulierobjecten
</form>
```

Het element `<form>` bevat altijd een aantal attributen die de webserver vertellen hoe het formulier moet worden verwerkt. Die attributen beschrijven de methode waarop de gegevens verstuurd worden en de actie die de server moet ondernemen om de gegevens verder te verwerken. De attributen heten **method** en **action**.

Het method attribuut geeft aan hoe de form-data verstuurd moeten worden. De data kan verzonden worden als URL-variabelen (zoals bij Google zoekopdrachten) of via een HTTP POST request.

Je kan met de methode GET maximaal 8192 karakters meezenden. Alle gegevens uit het formulier worden als een lange sliert gegevens achter de URL gepakt (de querystring). Op het moment dat je veel gegevens wilt versturen (zoals textarea's of paswoorden) is het verstandiger om de POST methode te gebruiken.

Het attribuut action geeft letterlijk aan welke action de server met de gegevens die je hebt gepost moet uitvoeren. De URL geeft aan waar het script zich bevindt dat de gegevens verwerkt. Met dit attribuut kan je twee kanten op:

- Je stuurt de data naar een script dat de gegevens verwerkt, bijvoorbeeld `action="http://www.vives.be/verwerk.aspx"` (deze manier van werken zien we in het vak ASP.NET)
- Je stuurt de data naar een e-mailadres, waardoor het attribuut wordt: `action=mailto:gegevens@vives.be?subject=gegevensformulier`.  
Je hoeft bij het verzenden via e-mail niet over een script te beschikken en krijgt zelf de resultaten van het formulier per e-mail geleverd. Deze vorm is gemakkelijk te gebruiken maar wordt uit veiligheidsoverwegingen niet meer door alle browsers ondersteund.

Wij zullen voorlopig action niet benoemen, we laten het dus leeg: `action=""`.

---

## 5.1 Formulierobjecten

### 5.1.1 Tekstvak

Een gebruiker kan een tekstvak gebruiken om tekst in te voeren. Deze tekst kan vervolgens door de webapplicatie verwerkt worden.



Figuur 10 Een tekstvak, standaard styling

```
<input type="text"
  name="invoerveld"
  size="aantal karakters zichtbaar"
  maxlength="max lengte"
  value="startwaarde" />
```



```
<input type="text" name="postcode" size="4"
  maxlength="4"/>
```

### 5.1.2 Paswoordvelden

```
<input type="password"
  name="data-id"
  size="aantal karakters zichtbaar"
  maxlength="max lengte"
  value="startwaarde" />
```



```
<input type="password" name="paswoord" size="10"
  maxlength="10" />
```

### 5.1.3 Email Address & Web Address

Deze twee input types zijn ontwikkeld met mobiele browsers in het achterhoofd, en voornamelijk voor smartphones waar tekstinvoer op het scherm plaatsvindt. Als voorbeeld nemen we de *iPhone* browser. Zodra de gebruiker de focus in een invoerveld zet, zal de weergave van het toetsenbord veranderen:

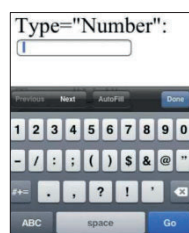
- emailadres: elk e-mailadres bestaat uit minimaal een punt en @ teken en spaties zijn niet toegelaten. Het toetsenbord zal daarom de twee tekens weergeven en maakt de spatiebalk kleiner dan in de standaard weergave.
- webadres: de spatiebalk is weggelaten, de punt en slash zijn weergegeven en tevens een knop om een achtervoegsel (.com, .org, .net) mee toe te voegen.

In desktopbrowsers is deze functionaliteit uiteraard niet aanwezig en zal er geen visueel verschil zijn in vergelijking met een standaard invoerveld. Het is echter wel zo dat het vervangen van een tekstveld naar een email of url invoerveld voordelen heeft met validatie.

|   |  |
|---|--|
| <pre>&lt;input type="email" name="txtEmail" /&gt;</pre>                           | <pre>&lt;input type="url" name="txtUrl" /&gt;</pre>                                |
|  |  |

Niet alle attributen werden opgenomen, bekijk **de volledige** lijst op w3schools<sup>23</sup>.

## 5.1.4 Number



Voor het ingeven van nummers zijn er een tweetal input types in HTML5 toegevoegd:

```
<input type="number" name="txtNumber" />
<input type="range" name="txtRange" />
```

Voorheen werd hier een standaard tekstveld voor gebruikt. Wanneer een mobiele browser dit input type tegenkomt wordt het toetsenbord aangepast: cijfers in de plaats van letters. Desktop browsers geven het op twee manieren weer: *Opera* en *Chrome* voegen spinboxes (pijl omhoog en omlaag) aan de rechterkant van het invoerveld toe. *Internet Explorer*, *Firefox* en *Safari* geven het als een standaard tekstveld weer.



Figuur 11 Illustratie spinbox

Het verschil tussen number en range zit hem voornamelijk in de weergave. *Opera*, *Safari* en *Chrome* geven in de plaats van een invoerveld een schuifbalk. *Internet Explorer* en *Firefox* geven het als een standaard tekstveld weer.



Figuur 12 Illustratie schuifbalk

<sup>23</sup> [https://www.w3schools.com/html/html\\_form\\_input\\_types.asp](https://www.w3schools.com/html/html_form_input_types.asp)

---

Aan beide input types kunnen de volgende attributen worden meegegeven:

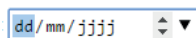
- Min: de laagst mogelijke waarde
- Max: de hoogst mogelijke waarde
- Step: met elke muisklik wordt deze waarde bij de huidige waarde opgeteld of afgetrokken.
- Value: de beginwaarde die wordt weergegeven.

### 5.1.5 Data en tijden invoeren

Er zijn totaal een zestal input types waarmee data gekozen kan worden:

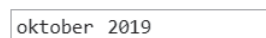
- `<input type="date" name="" />`

De dropdown button geeft een kalender weer waarin een maand, jaar en dag gekozen kan worden.



- `<input type="month" name="txtMaand" />`

Dezelfde kalender wordt weergegeven als bij date, maar in het invoerveld wordt alleen het jaar en maand weergegeven.




- `<input type="week" name="txtWeek" />`

Dezelfde kalender wordt weergegeven als bij date, maar in het invoerveld wordt alleen het jaar en week weergegeven (dit werkt voorlopig enkel in Chrome/Opera en Microsoft Edge bij het schrijven van de cursus).

- `<input type="time" name="txtTime" />`

Met een spinbox of door het zelf in te voeren kan de tijd worden aangegeven.



### 5.1.6 Searchbox

Als je het search invoerveld vergelijkt met standaard tekstvelden, zal er weinig verschil te merken zijn. De context van de invoer blijft namelijk hetzelfde, alleen de weergave in sommige browsers verschilt. *Safari* en *Chrome* tonen een 'X'-icoontje aan de rechterkant van het invoerveld. Hiermee kan in één klik de inhoud van het invoerveld leeggemaakt worden.

```
<input type="search" name="txtSearch" />
```



### 5.1.7 Tekstvak met meerdere regels

Een tekstvak waarin je meerdere regels in kan typen noemen we een *textarea*.



Figuur 13 Illustratie textarea

```
<textarea rows="aantal rijen"
  cols="aantal karakters breed"
  name="date-id"
  wrap="virtual">
  eventueel begintekst
</textarea>
```



```
<textarea rows="5" cols="50" name="Opmerkingen"
  wrap="virtual">
  Bij opmerkingen gelieve dit invoerveld te gebruiken
</textarea>
```

### 5.1.8 Checkbox

Een checkbox gebruik je om een binaire keuze te vragen of geven aan de gebruiker.



Figuur 14 Een checkbox met standaard styling

```
<input type="checkbox"
  name="data-id"
  value="waarde die wordt overgedragen"
  checked = "checked" />
```



```
Kent u deze sporten ?<br/>
<input type="checkbox" name="voetbal" value="v"
  checked="checked" />
Voetbal<br/>
<input type="checkbox" name="tennis" value="t" />
Tennis
```

### 5.1.9 Radiobutton

Een radiobutton gebruik je om input van de gebruiker te vragen, en wanneer er maar één antwoord mag geselecteerd worden.



Figuur 15 Illustratie radiobutton

```
<input type="radio"
  name="data-id"
  value="waarde die wordt overgedragen"
  checked = "checked" />
```



```
Hoe zal u betalen ? <br>
<input type="radio" name="betaalmiddel" value="c" />
Cash
<input type="radio" name="betaalmiddel" value="v" />
Visa
<input type="radio" name="betaalmiddel" value="b"
checked="checked" />
Bancontact
```

**Opmerking:** het name-attribuut heeft voor alle drie de radiobuttons dezelfde waarde.

### 5.1.10 Keuzelijst

Een keuzelijst laat toe om (potentieel) meerdere opties te selecteren.

```
<select name="data-id" size="aantal menuopties zichtbaar"
multiple="multiple">
  <option value="waarde1" selected="selected">keuzel</option>
  <option value="waarde2">keuze2</option>
</select>
```



```
Welke programmeertalen kent u ?<br/>
<select name="taal" size="3">
  <option value="java">JAVA </option>
  <option value="delphi">Delphi</option>
  <option value="c">C#.NET</option>
  <option value="asp">ASP.NET</option>
  <option value="vb">VB.NET</option>
</select>
```

### 5.1.11 Verborgenvelden

Onder een verborgen veld verstaan we een formulierelement dat niet op het beeldscherm is te zien, maar wel wordt overgedragen.



```
<input type="hidden"
name="data-id"
value="waarde" />
```

## 5.2 Nieuwe attributen voor <input> element

### 5.2.1 Autofocus

Met autofocus is het mogelijk om aan te geven naar welk invoerveld de browser moet gaan, zodra de pagina is geladen. Stel dat een pagina alleen een zoekbalk bevat (zoals *Google* bijvoorbeeld), dan is het

handig om autofocus te gebruiken. Dit scheelt één extra handeling, aangezien de gebruiker niet het invoerveld hoeft aan te klikken. Als er op meerdere invoervelden het autofocus attribuut wordt geplaatst, zal de functionaliteit alleen op het laatst aangegeven veld worden uitgevoerd.

```
<input type="text" name="txtNaam" autofocus="autofocus" />
```

Tot nu toe moest er *JavaScript* code aan te pas komen om deze functionaliteit aan te bieden. In het onderstaande voorbeeld is het formulier weggelaten, maar het is belangrijk om te weten dat aan het invoerveld `id="element"` wordt meegegeven):

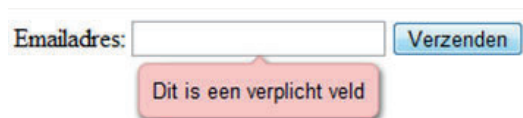
```
<script>
  window.addEventListener('load', (event) => {
    document.getElementById('element').focus();
  });
</script>
```

## 5.2.2 Required

Voor het registreren op een website zijn er altijd een aantal waarden, die verplicht zijn; o.a. naam, e-mailadres en wachtwoord. Voor dit soort situaties is er in *HTML5* het required attribuut toegevoegd. Als dit attribuut aan een input type wordt meegegeven betekent dit dat de gebruiker verplicht is om het in te voeren. Voorlopig ondersteunen enkel Opera en *Chrome* dit: als het veld niet is ingevuld en de gebruiker drukt op de *submit* knop, zal de focus op het niet ingevulde invoerveld worden gezet.

```
<input type="text" name="txtEmail" required="required" />
```

Waarden in andere invoervelden blijven ingevuld. Voorheen moest de ontwikkelaar zelf een stuk code in *JavaScript* of *ASP.NET* schrijven om te kijken of de verplichte velden ingevuld waren. In *HTML5* wordt het als volgt aangegeven:



Figuur 16 Verplicht input veld in HTML5

## 5.2.3 Placeholder

Met het placeholder attribuut kan standaard inhoud meegegeven worden aan een invoerveld. Zodra er in het invoerveld geklikt wordt zal de tekst verdwijnen.

```
<input type="text" name="txtNaam"
  placeholder="gelieve naam in te geven" />
<input type="submit" />
```

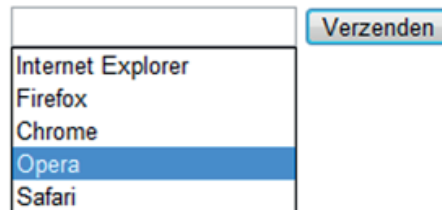


Figuur 17 Placeholder waarde in <input >

### 5.2.4 List

Met behulp van het list attribuut koppel je een lijst met data aan een bepaald invoerveld toe. Wat het doet is een normaal invoerveld samenvoegen met een drop-down lijst (oftewel het <select> input type). Het list attribuut is ideaal om waarden voor te leggen aan de gebruiker. Bijvoorbeeld; top 10 zoektermen of alle e-mailadressen van de werknemers.

Als voorbeeld een standaard invoerveld met het list attribuut. De gebruiker kan zelf een waarde intypen, maar kan ook één van de waarden uit de lijst kiezen. Werkt voorlopig enkel in Opera.



Figuur 18 List voorbeeld

Code voor bovenstaand voorbeeld is als volgt:

```
<input list="browsers" name="browser" />
<datalist id="browsers">
  <option value="Internet Explorer">
  <option value="Firefox">
  <option value="Chrome">
  <option value="Opera">
  <option value="Safari">
</datalist>
<input type="submit" />
```

### 5.2.5 Multiple

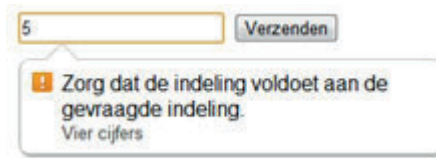
Met dit attribuut wordt aangegeven dat het mogelijk is om meerdere waarden in één invoerveld in te voeren. Dit is bijvoorbeeld handig bij het file input type, waarmee er bestanden op de computer kunnen geselecteerd worden. Om de functionaliteit van dit attribuut te gebruiken moet het in het input type element worden geplaatst:

```
<input type="text" multiple="multiple" />
```

### 5.2.6 Pattern

Met het pattern attribuut kan aangegeven worden waar een invoerveld aan moet voldoen. Stel dat men een formulier heeft waar de gebruiker een postcode moet invullen. Een Belgische postcode bestaat altijd uit vier cijfers. Aan het invoerveld zou dan meegegeven kunnen worden dat er totaal vier cijfers verwacht worden of exact twee letters. Validatie van de waarden wordt hierdoor een stuk makkelijker.





Figuur 19 Illustratie pattern op <input>



```
<input type="text" pattern="[0-9]{4}"  
title="Vier cijfers" />
```

Hoe je dergelijke patronen moet opstellen, is geen materie binnen deze cursus. Het komt wel aan bod binnen het vak webprogrammeren (andere voorbeelden: <http://html5pattern.com/>).

## 5.3 Formulieren verzenden

In de voorgaande paragrafen heb je de belangrijkste HTML-opdrachten voor het maken van een formulier gezien. Eén aspect is tot nu toe nog niet behandeld: hoe verzend je een formulier?

Hiervoor zijn speciale formulierobjecten beschikbaar: *buttons*. Er zijn drie hoofdtypen:

- Een button waarmee de informatie die met het formulier verzameld is, verstuurd wordt; dit heet Submit. De code is: `<input type="submit" />`
- Een andere hoofdtype is een button waarmee alle invoervelden in het formulier gewist worden. Dit type heet Reset. De code is: `<input type="reset" />`
- Een derde type is een neutrale button. Je zal acties voor deze knop zelf moeten programmeren met JavaScript (zie cursus deel: Javascript).

Voor alle typen is het attribuut `value` beschikbaar, waarmee je tekst op een button kunt plaatsen.

```
<form action="filename" method="post/get">  
  formulierobjecten  
  <input type="submit" name="knop" value="Verzenden" />  
</form>
```

Wanneer een formulier verstuurd wordt, moet men aangegeven volgens welke METHOD dit zal gebeuren. Hierbij hebben we de keuze tussen POST of GET. Met POST worden de gegevens naar de server gestuurd als onderdeel van de HTTP-headers, met GET als onderdeel van de URL. Welke actie er moet gebeuren op de verzonden data, wordt vermeld na ACTION. Dit attribuut laten we voorlopig leeg. Het zullen we later gebruiken binnen het domein webprogrammeren (ASP.NET CORE). Met de HTML-code `<input type="submit">` krijgt men een verzendknop. Let op de volgende attributen. `id` is de identifier van het knopvlak. `value` geeft een waarde aan het knopvlak, belangrijk voor de latere identificatie. Als men een aantal van deze knoppen in een formulier gebruikt, kan men met `value` vaststellen welke knop is aangeklikt.

## 5.4 Voorbeeld van een formulier

The screenshot shows a web form with the title "Inlichtingen". It contains the following elements:

- Name:** A text input field with the placeholder text "Name".
- Email:** A text input field with the placeholder text "info@katho.be".
- Website:** A text input field with the placeholder text "http://www.example.com".
- Number:** A number input field with a spinner and the placeholder text "Even num".
- Range:** A range input field with a slider and the placeholder text "0 - 10".
- Message:** A text area with the placeholder text "Message".
- Send Message:** A button.

Figuur 20 Voorbeeldformulier

**Opmerking:** de formulierelementen staan niet mooi onder elkaar. Hoe we dit correct afhandelen komt aan bod in de cursus CSS.

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Voorbeeld formulier</title>
  </head>
  <body>
    <h2>Inlichtingen</h2>
    <form action="" method="post">
      <label for="name"> Name:</label>
      <input type="text" name="name" required placeholder="Name" />
      <br/>
      <label for="email">Email:</label>
      <input type="email" name="email" required
        placeholder="info@vives.be" />
      <br/>
      <label for="website">Website:</label>
      <input type="url" name="website" required
        placeholder="http://www.example.com" />
      <br/>
      <label for="number">Number:</label>
      <input type="number" name="number" min="0" max="10"
        step="2" required placeholder="Even num < 10"/>
      <br/>
      <label for="range">Range:</label>
      <input type="range" name="range" min="0" max="10" step="2" />
      <br/>
      <label for="message">Message:</label>
```

---

```
<textarea name="message" required></textarea>
<br/>
<input type="submit" value="Send Message" />
</form>
</body>
</html>
```