

Intro Javascript

HTML5 FORMS

Inhoud

Inleiding: HTML Web standaard

Structuur van HTML pagina

HTML Web Forms

- **Form-Structuur**
- **Onderdelen van een form**
- **Javascript in web pagina**
- **Form Controls & Input types**

HTML: Web Pagina standaard

HTML (HyperText Markup Language)

= Standaard voor **structuur** en web-browser **display** van textuele informatie op World Wide Web

- W3C: **World Wide Web Consortium**: beheert Web standaarden
 - Bv in HTML-werkgroep overleggen browserfabrikanten (Microsoft, Google, Mozilla, Opera,...) hoe bv in een webdocument een hyperlink, hoofding,... wordt aangegeven
- 2004: **WHATWG** (Web Hypertext Application Technology Working Group) opgericht (o.a. Apple, Mozilla, Opera)
- In 2011: HTML5 – beheerd door WHATWG
- Specificatie standaard HTML5:
<http://dev.w3.org/html5/spec/Overview.html>

HTML (HyperText Markup Language)

- Is geen programmeertaal, maar een markeertaal
- Web browser kan op basis van HTML markeertaal een webpagina afficheren
- Alle structuurkenmerken aangegeven van tags
 - Bv voor eerste hoofding wordt tag <H1> gebruikt:
 - <h1> Dit is een hoofding</h1>
 - <h1> is openingstag en </h1> is sluittag
- Voorbeeld van een HTML web pagina:

```
<html>
  <head>
    <title>titel van pagina</title>
  </head>
  <body>
    <H1>Here is some text </H1>
  </body>
</html>
```

HTML5 basisStructuur pagina

HTML: Niet case sensitive ≠ Javascript: wél case sensitive!

Tags:

`<!DOCTYPE>` duidt de versie van HTML aan (HTML5)

`<HTML>` start van HTML pagina

`</HTML>` einde van HTML pagina

`<Head>` start van hoofding HTML pagina

`</Head>` einde van hoofding HTML pagina

`<Body>` start van inhoud HTML pagina

`</Body>` einde van inhoud HTML pagina

Voorbeeld HTML5 pagina

```
<!DOCTYPE html>
<html>
  <head>
    <title>Document title</title>
  </head>
  <body>
    <h1>Document heading</h1>
    <p>This is a paragraph of text.</p>
    <p>
      <a href="http://www.google.be/">
        Link text for a link to another-html-document.html
      </a>
    </p>
    <!-- this is a comment -->
  </body>
</html>
```

Test bovenstaande HTML eens uit bv in:

https://www.w3schools.com/html/exercise.asp?fileid=exercise_forms1

<http://next.plnkr.co/edit/?p=preview&preview>

Commentaar in HTML

```
<!-- multi-line comment -->
```

HTML

- Verschil in commentaar-syntax:
 - **HTML:** `<!-- comment -->`
 - **CSS/JS/PHP:** `/* comment */`
 - **Java/JS/PHP:** `// comment`
 - **PHP:** `# comment`

HTML5 <Head> bevat ...

Meta tags

bv `<meta charset="UTF-8">`

Link(s) naar style sheet(s):

bv

```
<link rel="stylesheet" href="styles.css">
```

Link(s) naar Javascript code in extern bestand (.js):

```
<script src="file.js"></script>
```

Javascript code kan rechtstreeks in body or head voorkomen:

```
<script type="text/javascript">  
//Hier komt Javascript code  
</script>
```


HTML5 Forms

HTML Forms BasisStructuur

<form> - begin van form

<input> - informatie kan worden ingegeven in één of meerdere input-velden. Deze velden kunnen een verschillend type hebben

</form> - einde van form

Onderdelen van een Web Form

- Een **Form** is een gebied waar **Form Controls/Elementen** kunnen voorkomen.
- Een gegeven op een form wordt bijgehouden en getoond in een **Field**.
- De waarde zelf is een Field (attribuut): **value**.

Onderdelen van een Web Form

Gebruikers kunnen **Form Controls/ Elementen** invullen of selecteren

Voorbeelden van Form/Control elementen: buttons, checkboxes, text fields, radio buttons, (dropdown)lists, ...

- Een form bevat meestal een **submit** button om de informatie in de formelementen naar de (web)server te sturen.

Voorbeelden van HTML5 Form elementen

Email: `<input type="email" id="useremail">`

Email:

nettutsplus.com is not a legal email address

Required Input: `<input type="text" id="someInput" required>`

Regular Expressions:

```
<input type="text" id="someInput"
      pattern="[A-Za-z]{4}">
```

Form Controls/Elementen

- **Input Boxes** – invullen van tekst en numerieke informatie
- **Option Buttons (Radio Buttons)** – om 1 keuze te kunnen maken uit een voorgedefinieerde lijst
- **Selection Lists(Drop-Down) List Box** – om 1 of meerdere keuzes te kunnen maken uit een lijst van opties
- **Check Boxes** – voor ja/neen opties
- **Text Areas** – om een groter stuk tekst te kunnen ingeven (meerdere lijnen tekst zijn mogelijk)

Text Input (type="text")

Text Field:

- Gebruikers kunnen in een text field 1 lijn tekst ingeven
- Je kan de grootte van het veld zetten via **size**.
- Maximaal aantal karakters kan beperkt worden via **MAXLENGTH**
- Je kan een default waarde zetten via **value**

```
<input type="text" id="textfield" size= "25"  
value="with an initial value">
```

Text Input (type="text")

Voorbeeld 1: Een textField "text1" die 30 karakters breed is

```
<input type="text" id="text1" size="30">
```

Voorbeeld 2: Een textField "text2" die 30 karakters breed is maar zal enkel 20 karakters input toelaten

```
<input type="text" id="text2" size="30"
maxlength="20">
```

Voorbeeld 3: Een textField "text3" die 40 karakters breed is met een default waarde

```
<input type="text" id="text3" size="40" value="We
are not alone">
```



```
<!DOCTYPE html>
<html>
<body>

<h2>The value Attribute</h2>
<p>The value attribute specifies the initial value for an input field:</p>

<form>
First name:<br>
<input type="text" name="firstname" value="John">
<br>
Last name:<br>
<input type="text" name="lastname">
</form>

</body>
</html>
```

Vervolg - voorbeeld HTML5 pagina

The value Attribute

The value attribute specifies the initial value for an input field:

First name:

Last name:

Password Input (type="password")

Password text input is hetzelfde als een text input element, behalve dat de karakters die ingetypt worden door de gebruiker, worden als "bullet" karakters getoond.

Password text wordt gecodeerd over web gestuurd.

- **Voorbeeld 1:** Een passwordField "pass1" van 30 karakters breed

```
<input type="password" id="pass1" size="30">
```

- **Voorbeeld 2:** Een passwordField "pass2" van 30 karakters breed maar die maximum **20 karakters input toelaat:**

```
<input type="password" id="pass2" size="30"  
maxlength="20">
```

Form Field textarea

TextArea is een textField die meer dan 1 lijn kan bevatten en waarbij de tekst kan worden gescrolled.

rows : aantal lijnen (hoogte)

cols : aantal karakters (breedte)

wrap="virtual" : de ingegeven tekst zal automatisch op de volgende lijn komen.

Een default tekst kan worden gedefinieerd

Voorbeeld: Een **textarea** field "comments" die 45 karakters breed is en 6 lijnen hoog.

```
<textarea id="comments" rows="6" cols="45"
wrap="virtual"> De eerste keer dat ik een web-
pagina zag, dacht ik... (wordt vervolgd)
</textarea>
```

Buttons

Een Form heeft minstens 1 Button nodig om de Form-informatie door te sturen (to **submit**) over web naar de server. Forms kunnen ook een **reset-button** bevatten om alle velden terug op de oorspronkelijke waarden te zetten

A submit button:

```
<input type="submit" id="btnSubmit" value="Submit">
```

A reset button:

```
<input type="reset" id="btnReset" value="Reset">
```

Buttons

A submit button:

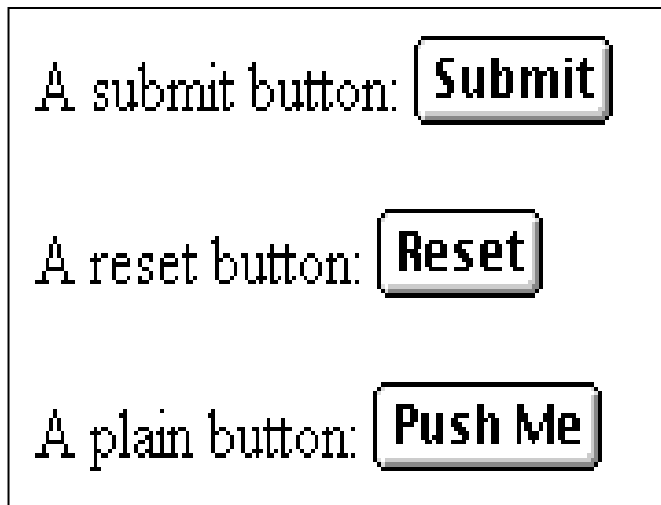
```
<input type="submit" id="Submit" value="Submit">
```

A reset button:

```
<input type="reset" id="Reset" value="Reset">
```

A plain button:

```
<button>Push Me</button>
```



- **submit**: versturen van data naar server
- **reset**: terugzetten naar oorspronkelijke waarden van alle formvelden

Button – Javascript

De button kan ook images bevatten

Om een dynamische actie op de button te plaatsen:

1. Zet een event(gebeurtenis) op de button (**onClick=**)
2. Schrijf een javascript function (bv **berekenBTW()**) die wordt uitgevoerd wanneer er op de knop wordt gedrukt (event onclick)

```
<input type="button" id="calculatebtw"  
value="bereken btw" onclick="berekenbtw()">
```

Voorbeeld Button Form element

```
<!DOCTYPE html>
<html>
<head>
    <title>Button Form voorbeeld</title>
</head>
<body>
    <h2>The Button </h2>
    <form name = "frmButtonForm">
        <button type="button" onclick="alert('Hello world!')">Click
        Me!</button>
    </form>
</body>
</html>
```


Radio buttons (type="radio")

Is een groep of reeks van bijbehorende controls waarvan enkel 1 van de mogelijkheden kan worden aangeduid

Indien één radio button wordt aangeklikt, worden automatisch de andere radio buttons in dezelfde groep uitgezet.

Een groep van radio buttons wordt gedefinieerd door ze dezelfde naam (**name**)

De waarde van een aangeklikte (**checked**) radiobutton is de inhoud van **value**.

De radio button kan bij default aangezet worden via deze **checked** te zetten

Radio buttons (type="radio")

Radio buttons:

```
<input type="radio" id="radiobutton" value="myValue1">male<br>
```

```
<input type="radio" id="radiobutton" value="myValue2" checked>female
```

Radio buttons:

☐ male

☒ female

Checkboxes (type="checkbox")

Checkboxes zijn gelijkaardig aan radiobuttons, maar meer dan 1 checkbox in dezelfde groep kunnen worden aangevinkt (**checked**)

Elke checkbox heeft een unieke naam(**name**)

De waarde van een aangevinkte (**checked**) checkbox is de inhoud van **value**. De waarde van een niet-aangevinkte checkbox is leeg

Een checkbox kan bij default aangevinkt zijn door deze **checked** te definiëren.

Checkboxes (type="checkbox")

A checkbox:

```
<input type="checkbox" id="checkbox"  
value="check" checked>
```

A checkbox: ☒

- Type="checkbox"
- Name: dit form element kan worden gerefereerd in javascript via name
- Value: waarde die wordt teruggegeven wanneer een checkbox is aangevinkt

Drop-down list (combobox)

A menu or list:

```
<select id="select">  
<option value="red">red</option>  
<option value="green">green</option>  
<option value="blue">blue</option>  
</select>
```

A menu or list:

- **size**: het aantal items die zichtbaar zijn in de lijst
- (default is "1")
- **multiple**: indien gespecificeerd, kunnen meerdere items in de lijst tegelijkertijd geselecteerd worden

(Nieuwe) INPUT types in HTML5

http://www.w3schools.com/html/html_form_input_types.asp

- `input type="number"`
- `input type="range"`
- Nog meer types: `email`, `url`, `tel`, ...

Age: `<input type="number" size="6" id="age" min="18" max="99" value="21">
`

Satisfaction: `<input type="range" size="2" id="satisfaction" min="1" max="5" value="3">`

'required' attribute

Voorbeeld:

Naam: `<input type="text" id="name" required>`

‘placeholder’ Attribute

Email Address: `<input type="email" id="email" required placeholder="Enter a valid email address">`

HTML 5 Form Demo

```
<!DOCTYPE html>
<html>
<head>
<title>HTML5 forms</title>
</head>

<body>

<h2>Example HTML5 form</h2>
<form>
  <p>Your Name: <input type="text" name="name" required></p>
  <p>Email Address: <input type="email" name="email" required placeholder="Enter a valid email address"></p>
  <p>Website: <input type="url" name="website" required pattern="https?://.+"></p>
  <p>Age: <input type="number" size="6" name="age" min="18" max="99" value="21"></p>
  <p>Satisfaction: <input type="range" size="2" name="satisfaction" min="1" max="5" value="3"></p>
  <p><input type="submit" value="Submit" name="btnSubmit" /></p>
</form>
</body>
</html>
```

HTML 5 Form Demo-vervolg

Example HTML5 form

Your Name:

Email Address:

Website:

Age:

Satisfaction:

Javascript in HTML web forms

Modules in Javascript

- Ingebouwde Javascript functies en methods
- User-defined functies
 - **Definitie**
 - **Voorbeeld**
 - **User-defined functie als Event-handler**

Modules in JavaScript

Modules in JavaScript

- Functions
- Methods

Behoren tot een object

- JavaScript bevat meerdere voorgedefinieerde functions en methods
parseFloat, parseInt,...

Modules in JavaScript

Functies

- Worden opgestart via een aanroep naar de naam van de functie (function call)
- Ontvangen noodzakelijke informatie via invoerargumenten (parameters)
- Kunnen een waarde teruggeven (return value)

Bv aanroep function:

function-name(parameter1, parameter2,...)

```
let total = parseFloat(inputValue);
```

Modules in JavaScript

Function calls

- Via naam van function
- Left parenthesis
- Arguments staan tussen haakjes en worden gescheiden door een komma

Argumenten kunnen zijn: constants, variables of expressies

Wanneer een waarde wordt teruggegeven door functie (return value), wordt deze toegewezen (assigned) aan een variabele

- Voorbeelden:

```
let total=0.0, s1="1", s2 ="2";  
total += parseFloat("2.3");  
total += parseFloat(s1 + s2);
```

User-Defined Functions

Zelf definiëren van functions

- Alle variabelen in een functie gedeclareerd, bestaan enkel binnen deze functie
- Parameters, argumenten van functie
Waarden die worden meegegeven aan de functie bij aanroep
- Om code te kunnen hergebruiken

User-Function Definitions

syntax van user-defined functiedefinitie :

```
function function-name(parameter-list)  
{  
  declarations and statements;  
}
```

function-name: zelf gekozen functienaam (user-defined)

parameter-list: lijst van namen van variabelen,
gescheiden door komma's

- Moet dezelfde aantal parameters hebben als deze bij functie-aanroep (function call)
- Kan leeg zijn: *function-name*()

declarations and statements

Function-body ("block" van code): bevat 1 of meer lijnen Javascript code

User-Functie Definitie

Terug-geven van waarde door functie via
`return` statement

Een Functie kan ofwel niets, ofwel een waarde teruggeven:

`return` *expression*;

Voorbeeld User-Defined JavaScript functie

```
function name() {  
    statement ;  
    statement ;  
    ...  
    statement ;  
}
```

JS

```
function myFunction() {  
    alert("Hello!");  
    alert("How are you?");  
}
```

JS

User-defined functie als Event-handler (afhandelen van gebeurtenis)

- Bij een gebeurtenis(event) op de web-pagina, kunnen bepaalde acties worden uitgevoerd via Javascript code

Voorbeeld:

```
<Input type="button" value="click me!"  
Onclick="myfunction();">
```

- Javascript functions kunnen worden gebruikt als event handlers
 - Bv wanneer je op een Button klikt met de muis, wordt een Javascript functie aangeroepen

Oefening HTML5 Form

Maak de volgende form met controls:

Who are you?

Name:

Gender: ☐ Male ☐ Female

Oplossing Oefening

```
<html>
<head>
<title>Get Identity</title>
</head>
<body>
<p><b>Who are you?</b></p>
<form method="post" action="">
<p>Name:
<input type="text" id="textfield">
</p>
<p>Gender:
<input type="radio" name="gender" value="m">Male
<input type="radio" name="gender" value="f">Female
</p>
</form>
</body>
</html>
```

Who are you?

Name:

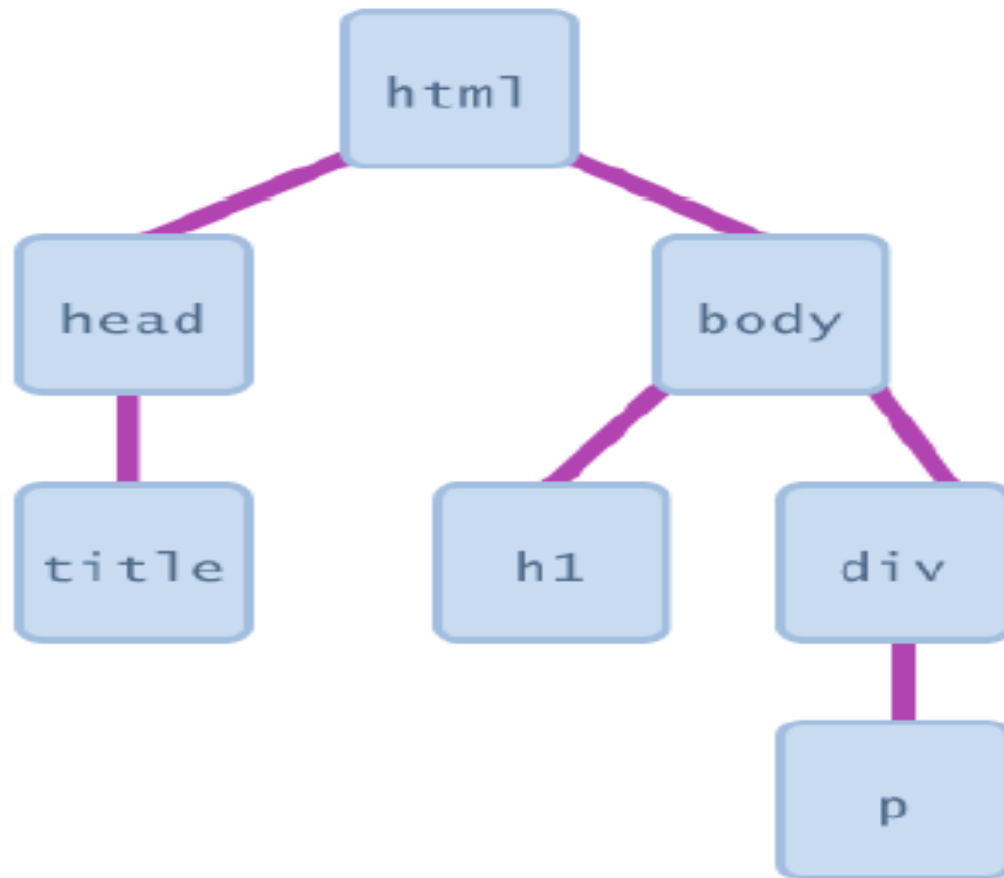
Gender: ☐ Male ☐ Female

Oefening– User-Defined Javascript functies

- Geef het maximum van 2 gehele getallen
 - 2 input text velden en 1 button
 - Converteer naar gehele getalen
 - Wanneer op de button wordt geklikt (onClick event), bepaal maximum met `if {} of Math.max`
 - Afficheer maximum in pagina (of met alert)
-
- Geef het maximum van 3 gehele getallen

DOM (Document Object Model)

Document Object Model (DOM)



HTML

```
<p>  
  Look at this octopus:  
    
  Cute, huh?  
</p>
```

DOM Element Object

Property	Value
tagName	"IMG"
<u>src</u>	"octopus.jpg"
alt	"an octopus"
id	"icon01"

JavaScript

```
var icon = document.getElementById("icon01");  
icon.src = "kitty.gif";
```

Elementen refereren in Javascript via DOM:

document.getElementById

```
<form>
<input type="button" value="Click me!" onclick="changeText();">
<input id="textbox" type="text" />
</form>
```

HTML

```
<script>
function changeText() {
  let textBox = document.getElementById("textbox");
  textBox.style.color = "red";
  textBox.value = "tekst met rode kleur";
}
</script>
```

JS

Voorbeeld document.getElementById

```
<html>
<head>
<title>getElementById</title>
</head>
<body>
<script>
function changeText() {
    let textBox = document.getElementById("textbox");
    textBox.style.color = "red";
    textBox.value = "tekst met rode kleur";
}
</script>
<form>
<input type="button" value="Click me!" onclick="changeText();">
<input id="textbox" type="text" />
</form>
</body>
</html>
```

Elementen refereren in Javascript via DOM: `document.getElementById`

- `document.getElementById` geeft het DOM object van een element terug met een bepaalde id
- Kan van sommige elementen de tekst wijzigen via de `innerHTML` property
- tekst in de form controls kan aangepast worden door de `value` property te wijzigen

Wijzigen van element style: `element.style`

Attribute	Property or style object
color	color
padding	padding
background-color	backgroundColor
border-top-width	borderTopWidth
Font size	fontSize
Font famiy	fontFamily

Datatype opvragen in Javascript: `typeof`

```
let name = expression; JS
```

- Datatypes worden niet cifeerd, JS is zwak getypeerde

```
let clientName = "Connie Client";  
let age = 32;  
let weight = 127.4; JS
```

- Datatypes in Javascript:
 - **Number, Boolean, String, Array, Object, Function, Null, Undefined**

Random-Number Generatie

Random-number generatie geeft een willekeurig getal, gegenereerd door de computer

- **Math.random**

- `let randomValue = Math.random();`

- Geeft Float-waarde between 0 and 1

- Om geheel getal tussen 0 en 6 te krijgen:

- `Math.floor(1 + Math.random() * 6);`

Math.floor (om naar beneden af te ronden naar geheel getal)

Voorbeeld: Dobbelsteen gooien

Button “**Roll Dice**”

Toon het resultaat van de random-gegenereerde waarde (tussen 0 en 6)

Math object

```
let rand1to10 = Math.floor(Math.random() * 10 + 1);  
let three = Math.floor(Math.PI);
```

- methodes: **abs, ceil, cos, floor, log, max, min, pow, random, round, sin, sqrt, tan**
- properties: **E, PI**

Speciale waarden: null en undefined

```
let ned = null;  
let benson = 9;  
// at this point in the code,  
// ned is null  
// benson's 9  
// caroline is undefined  
JS
```

- **undefined** : niet gedeclareerd of bestaat niet
- **null** : bestaat, maar was specifiek toegewezen aan empty of null waarde (objecten)

Referenties

- <https://www.learn-js.org>
- <https://www.w3schools.com/js/default.asp>
- <https://developer.mozilla.org/en-US/docs/Web/JavaScript/>
- <https://javascript.info/>
- <https://developer.mozilla.org/>