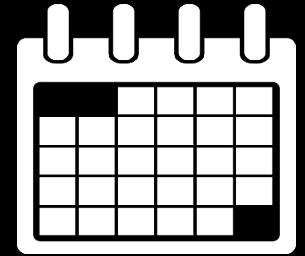


# User Interfaces: JavaScript

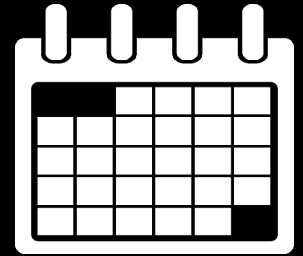
Formuliervalidatie



# 1. Formuliervalidatie

1. Inleiding
2. Werkwijze
3. Elementen selecteren
4. Events afhandelen
5. Valideren
6. Foutbericht

# 2. Gegevens overdragen tussen pagina's



# 1. Formulievalidatie

1. Inleiding
2. Werkwijze
3. Elementen selecteren
4. Events afhandelen
5. Valideren
6. Foutbericht

## 2. Gegevens overdragen tussen pagina's

# Inleiding: voorbeelden

Name:  This field is required

Email:  A valid email address is required

Website:  A valid url is required

Message:  This field is required

Email  Email is required

First name  ✓

Last name  ✓

Password should be atleast 6 characters and contain both lowercase and uppcase letters as well as numbers.

Password  Please use numbers and lower and uppcase letters

Confirm  Please enter the exact same password

Country  ✓

Street address  ✓

Home phone

Office phone

Cell phone

- First name is required.
- Last name is required.
- Invalid phone number.
- Invalid email address.

Create New Contact

First Name:  \*

Last Name:  \*

Phone:  \*

Email:  \*

Attach File | ABC Spelling... \* indicates a required field

Buyer \*  Please input the Buyer name.

Order No.  The correct format is ####-####-####-####, please input again.

Product  Please select the Product name in the list.

Quantity  The quantity is out of the rang 1--100, please input again.

E-mail  The email format is not correct.

Order Date

Remittance Date  The Remittance Date must be greater than or equal to Order Date.

# Inleiding: voorbeelden

Email

Email is required

Username

no spaces please

Password

### Validation Form

Email

Password

Confirm Password

Please enter the same value again.

E-mail

You have not given a correct e-mail address.

URL

The answer you gave was not a correct URL.

Number

The answer you gave was not a correct number.

alphanumeric  
characters and hyphen  
and underscore

### DEMO IFRAME

Username

Password

Confirm password

Passwords does not match

Email

You sure it is valid? The next step in this registration will be sent to the email you enter here.

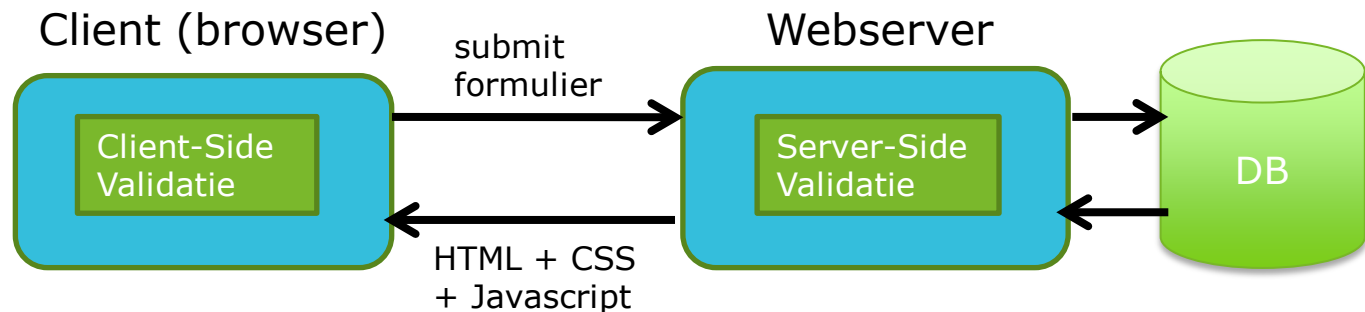
# Inleiding: belangrijke opmerking



Validatie met JavaScript

= **client-side validatie**

- Kan door (malafide) gebruiker omzeild worden!
- Enkel voor gebruikersgemak...



→ Altijd ook **server-side validatie** nodig (zie volgend jaar...)

# Inleiding: soorten validaties

---

- Benodigde velden (*required fields*)
  - Kijken of alle benodigde velden ingevuld zijn.
- Inhoud en formaat van velden:
  - Kijken of een veld voldoet aan een bepaald formaat (e-mail, telefoonnummer, ...)
  - Dikwijls via reguliere expressies (regex)
- Als de velden niet valideren: **submit** mag niet doorgaan...

# Werkwijze

---

1. *Selecteer* het juiste element uit het formulier
2. Kies het *event* waarop je wil reageren  
("click", "focus", "blur", "change", "keyup",  
"mouseover", ...)
3. Schrijf een *event handler* die de controle doet
4. Geef *feedback* (in de pagina via het DOM!)



# **Werkwijze: (belangrijkste) events bij formulieren**

---

- Form events:

Naam	Gaat af...
"submit"	...wanneer op de submit knop wordt geklikt
"reset"	...wanneer op de reset knop wordt geklikt

- Input events: (input, textarea en select)

Naam	Gaat af...
"input"	...bij elke directe wijziging van de inhoud
"change"	...wanneer de inhoud is gewijzigd en het element focus verliest
"blur"	...wanneer het element focus verliest

## **Referenties:**

- HTML forms guide: <https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/Forms>
- Event reference: <https://developer.mozilla.org/en-US/docs/Web/Events>

# Werkwijze: voorbeeld formulier

```
<form id="login" action="dummy.html" method="get">
  <p>
    <label for="naam">Username:</label>
    <input type="text" name="name" id="naam" value="Jos">
  </p>
  <p>
    <label for="wachtwoord">Password:</label>
    <input type="password" name="password" id="wachtwoord">
  </p>
  <input type="submit" name="send" id="verzend" value="Verzend">
</form>
```

Met het 'value' attribuut geef je een 'default' waarde aan het invoerveld.

Username:

Password:

# Elementen selecteren: form

```
1 window.addEventListener("load", init, false);
2
3 function init() {
4     let formulier = document.getElementById("login");
5     console.log(formulier);
6
7     // Alternatief
8     let viaId = document.querySelector("#login");
9     console.log(viaId);
10
11     // Indien geen id
12     let viaTag = document.querySelector("form");
13     console.log(viaTag);
14
15     console.log(viaTag.action);
16 }
```

← Verwijzing naar  
dummy.html

null	<a href="#">formulier.js:5</a>
null	<a href="#">formulier.js:9</a>
▶ <form action="dummy.html" method="get">...</form>	<a href="#">formulier.js:13</a>
<a href="http://localhost:63342/Oefeningen/JavaScript W4 voorbeeldcode/01 VoorbeeldFormulier/dummy.html">http://localhost:63342/Oefeningen/JavaScript W4 voorbeeldcode/01 VoorbeeldFormulier/dummy.html</a>	<a href="#">formulier.js:15</a>

# Elementen selecteren: velden

```
1 window.addEventListener("load", init, false);
2
3 function init() {
4   let veld = document.querySelector("#naam");
5   console.log(veld);
6   console.log(veld.value);
7
8   // alternatieven
9   let byId = document.getElementById("naam");
10  console.log("id " + byId.value);
11
12  let byTagName = document.getElementsByTagName("input")[0];
13  console.log("tag " + byTagName.value);
14 }
```

Verwijzing naar  
invoerveld "naam"

Verwijzing naar de  
inhoud van het  
invoerveld

console:

<input type="text" name="naam" id="naam" value="Jos">	<a href="#">formulier.js:5</a>
Jos	<a href="#">formulier.js:6</a>
id Jos	<a href="#">formulier.js:10</a>
tag Jos	<a href="#">formulier.js:13</a>

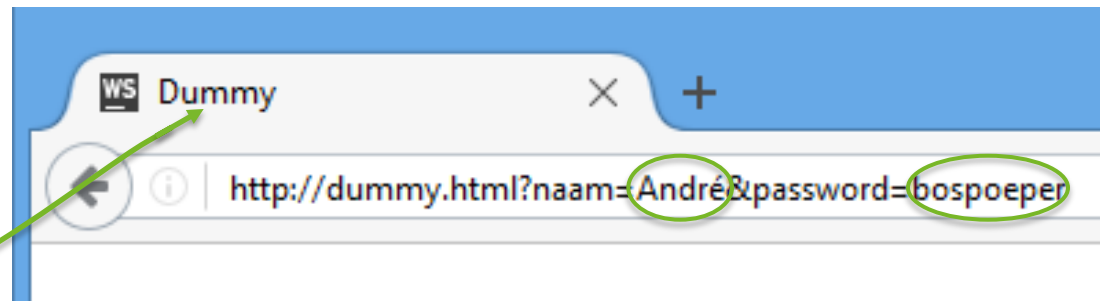
# Elementen selecteren: name-attribuut

```
<form id="login" action="dummy.html" method="get">
  <p><label for="naam">Username:</label>
    <input type="text" name="naam" id="naam" value="André">
  </p>
  <p><label for="wachtwoord">Password:</label>
    <input type="password" name="password" id="wachtwoord">
  </p>
  <input type="submit" id="verzend" value="Verzend">
</form>
```

- Invoerelementen in een formulier geven we meestal een **id** én een **name**.
- De **name** wordt meegestuurd bij de submit:

Username:

Password:



# Events afhandelen: form submit

```
function init() {  
    let formulier = document.querySelector("form");  
    formulier.addEventListener("submit", formSubmit, false);  
}  
  
function formSubmit(event) {  
    let feedback = document.querySelector("#feedback");  
    feedback.innerHTML = "Je hebt geen toegang tot deze website!";  
    event.preventDefault();  
}
```

`<input type="submit"`

## "submit" event:

- Gaat af bij verwerken van een formulier aan de hand van een **submit** knop
- Gaat af voordat de effectieve submit gebeurt
- Je kan de **submit** nog tegenhouden via:  
**event.preventDefault();**
  - Bv: bij geen geldig ingevoerde data (validatie)

# Events afhandelen: form reset

```
function init() {  
    let formulier = document.querySelector("form");  
    formulier.addEventListener("reset", formReset, false);  
}  
  
function formReset(event) {  
    let feedback = document.querySelector("#feedback");  
    feedback.innerHTML = "Nu kan je opnieuw beginnen!";  
}
```

```
<input type="reset"
```

## "reset" event:

- Gaat af bij verwerken van een formulier aan de hand van een **reset** knop
- Gaat af voordat de effectieve reset (= leegmaken van alle invoervelden) gebeurt
- Je kan de **reset** nog tegenhouden via:  
**event.preventDefault();**

# Valideren (vb. 1)

```
function init() {  
    let form = document.querySelector("#login");  
    form.addEventListener("submit", loginSubmit, false);  
}  
  
function loginSubmit(event) {  
    let message = "";  
    let name = document.querySelector("#naam").value;  
    let password = document.querySelector("#wachtwoord").value;  
    if (!name) {  
        message = "Naam niet ingevuld!";  
    }  
    if (!password) {  
        message = "Wachtwoord niet ingevuld!";  
    }  
    if (!name && !password) {  
        message = "Naam en wachtwoord niet ingevuld!";  
    }  
    if (message.length > 0) {  
        document.querySelector("#feedback").innerHTML = message;  
        event.preventDefault();  
    }  
}
```



# Valideren (vb. 2)

```
function init() {
    let form = document.querySelector("#login");
    form.addEventListener("submit", loginSubmit, false);
    let naamVeld = document.querySelector("#naam");
    naamVeld.addEventListener("input", validateNaam, false);
    let wachtwoordVeld = document.querySelector("#wachtwoord");
    wachtwoordVeld.addEventListener("change", validateWachtwoord, false);
}

function validateNaam() {
    let inhoud = document.getElementById("naam").value;
    let feedback = document.getElementById("naamError");
    if (inhoud.length < 2) {
        feedback.innerHTML = "De naam moet minstens 2 tekens lang zijn!";
        return false;
    } else {
        feedback.innerHTML = "";
    }
    return true;
}

function validateWachtwoord() {
    let inhoud = document.getElementById("wachtwoord").value;
    let feedback = document.getElementById("wachtwoordError");
    if (!inhoud) {
        feedback.innerHTML = "Geen wachtwoord ingegeven!";
        return false;
    } else {
        feedback.innerHTML = "";
    }
    return true;
}

function loginSubmit(event) {
    if (!validateNaam() || !validateWachtwoord()) {
        event.preventDefault();
    }
}
```

# Foutbericht

---

- Foutberichten vaak naast het veld of gegroepeerd boven/onder het formulier
- Vaak in het rood
- Achtergrondkleur veld aanpassen
- Labels rood kleuren
- ...



# Foutbericht (html)

- Voorzie op de plaats waar het foutbericht moet getoond worden element (bv. p, span) om het bericht te tonen

```
<!DOCTYPE html>
<html lang="nl">
<head>
  <meta charset="utf-8">
  <title>Keyup</title>
  <script src="script/keyup.js"></script>
  <link href="css/main.css" rel="stylesheet">
</head>
<body>
<form action="https://www.kdg.be">
  <p>
    <label for="naam">Naam: </label>
    <input type="text" id="naam" name="naam">
    <input type="submit" id="verzend" name="submit" value="Go!">
  </p>
  <p id="error"></p>
</form>
</body>
</html>
```

Naam:

De naam moet minstens 2 tekens lang zijn!

# Foutbericht (js)

```
function init() {
    let naamVeld = document.querySelector("#naam");
    let form = document.querySelector("#login");
    naamVeld.addEventListener("input", validateNaam, false);
    form.addEventListener("submit", submitLogin, false);
}

function validateNaam() {
    let inhoud = document.getElementById("naam").value;
    let feedback = document.getElementById("error");
    if (inhoud.length < 2) {
        feedback.innerHTML = "De naam moet minstens 2 tekens lang zijn!";
        return false;
    } else {
        feedback.innerHTML = "";
    }
    return true;
}

function submitLogin(event) {
    if (!validateNaam()) {
        event.preventDefault();
    }
}
```



Submit geblokkeerd

# Automatische validatie/foutbericht

---

HTML5 voorziet zelf in een beperkte validatie obv:

- input type: "text", "number", ...
- Validatie attributen: required, min, max, minlength, maxlength, pattern, ...

Naam:  Nummer:

Naam:  Nummer:    

Vul dit veld in.

Naam:  Nummer:    

Selecteer een waarde die niet lager is dan 100.

# Automatische validatie/foutbericht

---

```
<form action="http://www.kdg.be">
  <p>
    <label for="naam">Naam: </label>
    <input type="text" id="naam" name="naam" placeholder="Voornaam" minLength="2" required>
    <!-- minlength geeft geen 'waterdichte' feedback, dus JS nodig! -->
    <label for="number">Nummer: </label>
    <input type="number" id="number" name="number" min="100" max="999" required>
    <input type="submit" id="verzend" name="submit" value="Go to KdG!">
  </p>
  <p>
    <span id="error"></span>
  </p>
</form>
```

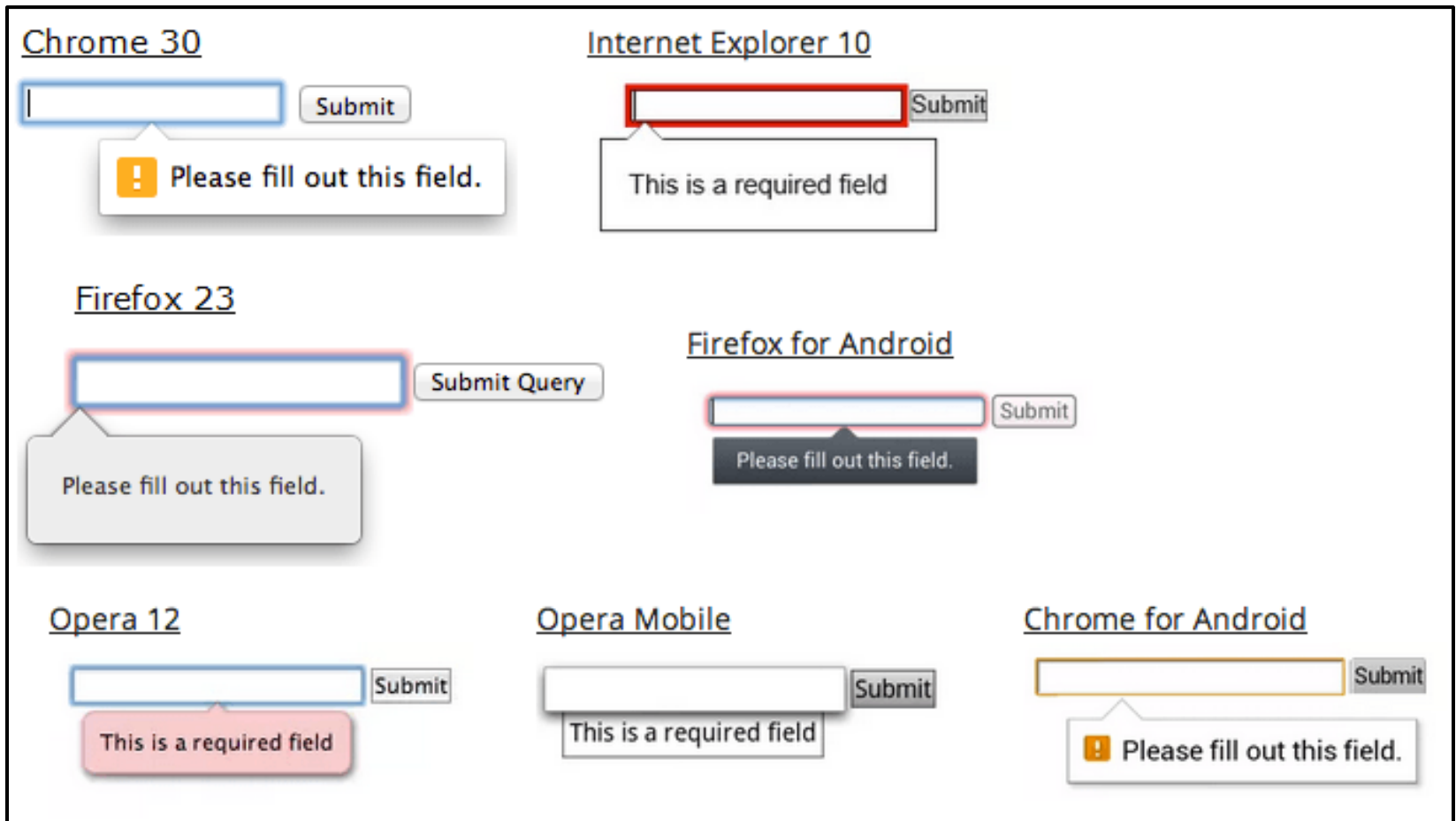
## Meer informatie?

Zie <https://www.the-art-of-web.com/html/html5-form-validation/>

Zie [https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/HTML5/Constraint\\_validation](https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/HTML5/Constraint_validation)

# Automatische validatie/foutbericht

Uitzicht van het foutbericht is browserafhankelijk:



# Oefening validatie 1



- Schrijf en test een JavaScript-functie om te valideren of een invoerveld een geldig Belgisch postnummer van exact 4 cijfers bevat.
  - Tip: minimaal 1000 en maximaal 9999
- Validatie moet uitgevoerd worden bij het klikken op de knop 'Test'

## Voorbeeld validatie (exact 4 cijfers) met Javascript

Postnummer:   Invoer fout

## Voorbeeld validatie (exact 4 cijfers) met Javascript

Postnummer:   Invoer fout

## Voorbeeld validatie (exact 4 cijfers) met Javascript

Postnummer:   Invoer OK



# Oefening validatie 2



- Schrijf en test een JavaScript functie om te valideren of een veld een geboortjaar bevat:
  - Bestaat uit exact 4 cijfers
  - Valt na 1900
  - Kan niet hoger zijn dan het huidige jaar
- De validatie moet gebeuren als het veld verlaten wordt (bv door ergens anders te klikken buiten het invoerveld of door de tab-/enter-toets te gebruiken)

## Validatie Geboortjaar

Voer je geboortjaar in en druk op de <tab> toets

Jaar:

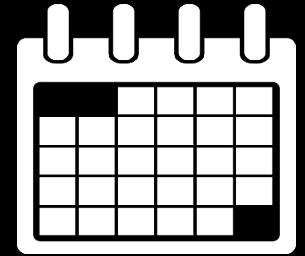
De invoer is foutief

## Validatie Geboortjaar

Voer je geboortjaar in en druk op de <tab> toets

Jaar:

De invoer is juist



# 1. Formulievalidatie

1. Inleiding
2. Werkwijze
3. Elementen selecteren
4. Events afhandelen
5. Valideren
6. Foutbericht

## 2. Gegevens overdragen tussen pagina's

# Probleem

---

- Ik wil gegevens van de ene webpagina naar de andere overbrengen?
- Mogelijke oplossingen:
  - Via een **webserver**
    - we blijven client side, dus kan hier niet!
  - Via **cookies**
    - buiten scope van de cursus!
  - Via de URL, met name door bij de form **method="get"** op te nemen

# Overdracht gegevens

---

- Eerst een HTML document met daarin een formulier voor de invoer van de gegevens:

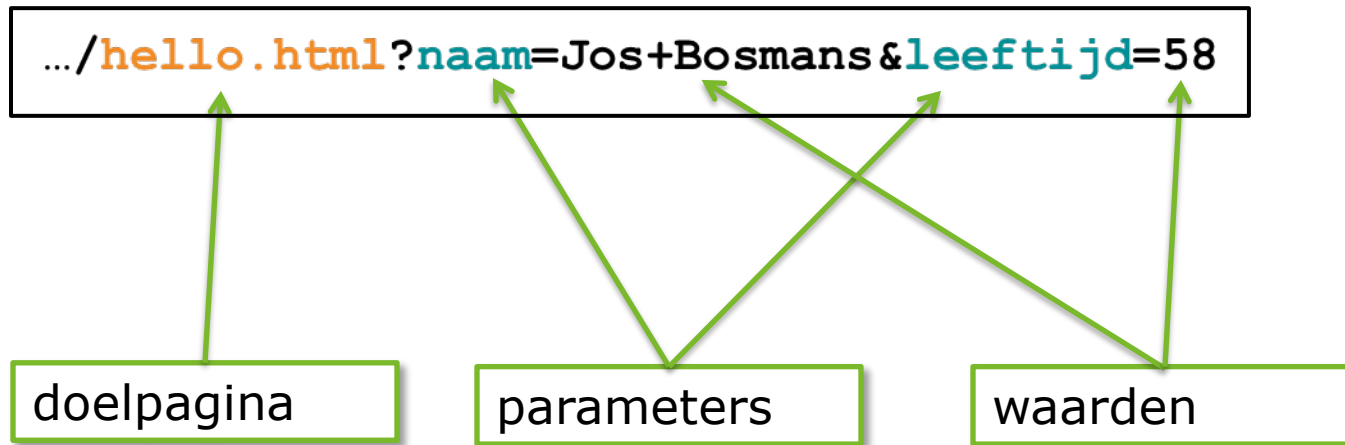
```
<form id="myform" action="html/hello.html" method="get">
  <p>
    <label class="label" for="naam">Naam:</label>
    <input class="input" type="text" id="naam" name="naam" size="25"/>
    <span class="fout" id="naamfout"></span>
  </p>
  <p>
    <label class="label" for="leeftijd">Leeftijd:</label>
    <input class="input" type="text" id="leeftijd" name="leeftijd" size="25"/>
    <span class="fout" id="leeftijdfout"></span>
  </p>
  <p>
    <label class="label" for="verzend">&nbsp;</label>
    <input class="input" type="submit" id="verzend" value="Verzend"/>
  </p>
</form>
```

1. Form: method 'get'
2. Elementen: name-attribuut

# Overdracht gegevens

---

- Veronderstel dat we als naam "Jos Bosmans" en als leeftijd "58" ingeven.
- Bij het klikken op **Verzend** ziet het laatste deel van de URL er zo uit:



# Overdracht gegevens

---

- De body van de pagina `hello.html` waarin we de overgedragen gegevens zullen tonen:

```
<body>
  <h1>Hello <span id="naamId"></span></h1>
  <h2>Je bent dus <span id="leeftijdId"></span> jaar</h2>
</body>
```

- Het enige wat nu nog ontbreekt is?



# Overdracht gegevens

---

- Bij de pagina `hello.html` hoort een JavaScript waarin je het volgende moet doen:

- Haal de querystring van de url in de adresbalk van de browser op

```
const query = window.location.search;
```

- Creëer een object van het type 'URLSearchParams'

```
const searchParams = new URLSearchParams(query);
```

- Haal de waarde van een parameter op obv van de naam/key

```
const paramValue = searchParams.get("paramName");
```

# Oefening validatie 3



- Schrijf een script waarin je de invoer van een naam en een leeftijd controleert. De naam moet minimaal 2 tekens bevatten, de leeftijd moet tussen 0 en 130 liggen.
- Foutmelding moet geven worden bij elke wijziging van de waarde in het invoerveld.
- Als alle invoer correct is moet je op een andere pagina terechtkomen waarin je de ingegeven informatie terug toont.

## Voorbeeld validatie met Javascript

Naam:  De naam moet minimaal 2 tekens lang zijn

Leeftijd: