

JavaScript-events en event handlers

Overzicht

- Wat zijn events en hoe kunt u er met code op reageren
- De typen events van JavaScript
- Hoe event handlers geschreven worden voor diverse typen events

Wat zijn events?

- **Procedureel programmeren**

- Toepassing wordt gestart en doorloopt van A tot Z de volgorde die is vastgesteld

- **Event georiënteerd programmeren**

- Wachten op een actie van de gebruiker (*event driven*)
 - Klikken op een knop
 - Tekst selecteren
 - Tekst invoeren
 - Muis over een gebied of afbeelding bewegen
 - ...

Synchroon en asynchroon

- **Synchroon**

- Programma kan pas verder met een volgende taak als de eerste taak is afgehandeld

- **Asynchroon**

- Event wordt op de achtergrond afgehandeld
- De gebruiker kan nog steeds op een andere knop klikken

Naamgeving van events

- Elk event heeft zijn eigen naam
 - click
 - change
 - load
 - mouseover
 - ...
- Eigen events schrijven is mogelijk (*custom events*)

Target

- Het element waarop de event van toepassing is
- Zowel eventnaam als target moet gebruikt worden
 - Event **click** voor de knop **btnSubmit**
 - Event **focus** voor het tekstvak **txtFirstName**
 - ...

Event handlers

- **Andere namen:** event listeners, callbacks
- een functie waarin andere JavaScript-functies of –opdrachten staan die worden uitgevoerd zodra de event optreedt

De functie `addEventListener()`

- Een andere manier om event handlers te schrijven
- Word toegepast op een geselecteerd element
- 2 parameters
 - De event die wordt afgevangen
 - De functie die wordt uitgevoerd als de event optreedt

Toetsenbordevents afvangen

- Onderzoeken welke toets is ingedrukt
 - velden waarin alleen getallen, of juist alleen letters ingevuld mogen worden
 - onderzoeken of de gebruiker op 'Enter' heeft gedrukt
 - controleren of de invoer aan een bepaalde lengte voldoet
- 3 events
 - **keydown**: start toets indrukken
 - **keypress**: toets ingedrukt
 - **keyup**: toets loslaten
- Als parameter wordt er de event (*keyboardevent*) zelf meegegeven

```
if (top !== self) {  
  function calcWidth() {  
    var wW = 0;  
    if (typeof window.innerWidth == 'number') {  
      wW = window.innerWidth;  
    } else if (document.documentElement && document.  
      wW = document.documentElement.clientWidth;  
    } else if (document.body && document.body.c  
      wW = document.body.clientWidth;  
    }  
    if (sH = document.documentElement.scrollHeight  
      var WH = window.innerHeight || document.  
      sW = !document.all && (sH > WH) ?  
        ('menu', 'width
```

Demo

Werken met het DOM

Overzicht

- De begrippen die u moet kennen bij het werken met het DOM
- De beschikbare JavaScript-functies om elementen in het DOM te selecteren
- Nieuwe elementen maken en aan het DOM toevoegen
- Elementen verwijderen of vervangen in het DOM

Elementen in het DOM selecteren

- **Selecteren door de id van het element op te geven**
 - `document.getElementById("");`
- **Selecteren door de name van het element op te geven**
 - `document.getElementsByName("");`
- **Selecteren door een tagnaam op te geven**
 - `document.getElementsByTagName("");`
- **Selecteren door de CSS-klasse op te geven**
 - `document.getElementsByClassName("");`
- **Selecteren door CSS-selector te gebruiken**
 - `document.querySelector("");` / `document.querySelectorAll("");`

Selecteren via ID/Name

- Selecteren op het HTML-attribuut id/name
- Meestal selecteren op ID; bij keuzerondjes meestal Name
- Uniek element

Selecteren via Type

- Selecteren op type of tagnaam
- Verzameling/array van elementen
- Hoofdletter onafhankelijk

Selecteren via CSS-klasse

- Selecteren op CSS-klasse
- Verzameling/array van elementen
- Kunnen meerdere klasse namen opgegeven worden

Selecteren via CSS-selectors

- Selecteren met CSS-selectors
 - **#menu** – Selecteert het element id="menu" in de pagina
 - **.warning** – Selecteert alle elementen met de klasse 'warning' in de pagina
 - **div** – Selecteert alle elementen <div> in de pagina
 - **#menu ul li** – Selecteert alle listitems in de lijst binnen het element menu
 - **div, .geel** – Selecteert alle <div>'s in de pagina én alle elementen met de klasse geel
 - ...

```
if (top !== self) {  
  function calcWidth() {  
    var wW = 0;  
    if (typeof window.innerWidth == 'number') {  
      wW = window.innerWidth;  
    } else if (document.documentElement && document.  
      wW = document.documentElement.clientWidth;  
    } else if (document.body && document.body.c  
      wW = document.body.clientWidth;  
    }  
    if (sH = document.documentElement.scrollHeight  
      var WH = window.innerHeight || document.  
      sW = !document.all && (sH > WH) ?  
        ('menu', 'width
```

Demo

Elementen maken

- **Document.createElement():** Maak een nieuw element van het aangegeven type. U gebruikt deze opdracht bijvoorbeeld om vanuit script een nieuwe div, een nieuwe paragraaf, ... te maken
- **Document.createTextNode():** Maak een tekstnode (*een stuk kale tekst*), dat in de paragraaf of in de div geplaatst kan worden
- **LET OP!** Na het aanmaken van een element moet dit nog in de DOM geplaatst worden. Dit gebeurt niet automatisch

Elementen invoegen in het DOM

- **appendChild():** Voegt een node toe als laatste node aan een geselecteerd element
- **insertBefore():** Voegt een node toe aan een geselecteerd element, en maakt het bovendien mogelijk aan te geven op welke plek (*dus: voor welke andere node*) de betreffende node wordt ingevoegd

Elementen verwijderen uit het DOM

- **removeChild():** verwijderd een element uit het DOM.
 - Niet aanroepen voor het element zelf, maar voor de 'parent' van het betreffende element
 - Het te verwijderen element wordt als parameter meegegeven

Elementen vervangen

- **replaceChild():** vervangt een element in het DOM.
 - **eerste parameter:** nieuwElement
 - **tweede parameter:** oudElement

Overige DOM-functies

- `createDocumentFragment()`
- `hasFeature()`
- `parentNode`
- `firstChild`
- `lastChild`
- `nextSibling`
- `offsetParent`

Overige DOM-functies

- `previousSibling`
- `cloneNode()`
- `hasChildNodes()`
- `getAttribute()`
- `removeAttribute()`
- `setAttribute()`


```
if (top !== self) {
function calcWidth() {
    var wW = 0;
    if (typeof window.innerWidth == 'number') {
        wW = window.innerWidth;
    } else if (document.documentElement && document
        wW = document.documentElement.clientWidth;
    } else if (document.body && document.body.c
        wW = document.body.clientWidth;
    }
    if (sH = document.documentElement.scrollHeight
        var WH = window.innerHeight || document
        sW = !document.all && (sH > wH) ?
        ('menu', 'width
```

Demo