

JAVASCRIPT

DOCUMENT OBJECT MODEL

JavaScript

- Niet als andere talen:
 - C# (Microsoft) en .Net (Microsoft)
 - Java (Oracle) en JDK (Oracle)
 - JavaScript (ECMA), DOM (W3C, WHATWG)
- JavaScript is de taal.
- DOM is de API voor manipulatie van de website

DOM

- Design fouten:
 - Browser concurrentie, snelle ontwikkeltijd
 - Ontworpen voor meerdere programmeertalen
- Gevolg: verwarringende API en verouderde delen
 - Vermijd verouderde onderdelen.
 - Documentatie gebruiken!

Browser Object Model (BOM)

- Het *window*-object
 - Geeft toegang tot alle globale objecten en functies

```
window.X === X // altijd true
```

```
window.alert("Test") is hetzelfde als alert("Test")
```
 - Ook: `window.innerHeight`, `window.location`, ...

Document Object Model

- Het *Document* object
 - window.document (of gewoon document)
 - Toegang tot de html elementen
 - document.body = het body-element

```
<input type="button" id="knop" value="Klik!" />
```

```
var knop = document.getElementById("knop");
```

- knop bevat een verwijzing naar het input-element met id “knop”

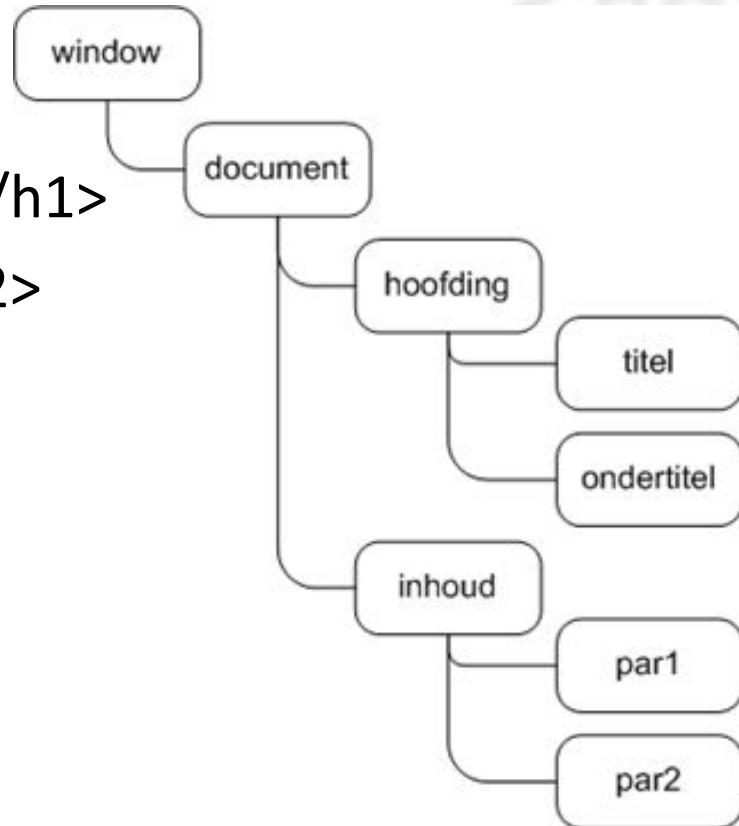
https://www.w3schools.com/js/tryit.asp?filename=tryjs_document_method

Document Object Model

- *Document tree*
 - document.body
 - child = elke aftakking vanaf bovenstaande vertakking;
 - parent = de bovenstaande vertakking;
 - De referenties *self* of *window* verwijzen naar het huidige venster dat de focus heeft;
 - Vanuit het window-object kom je in het document-object;
 - De kinderen van het document zijn de html-elementen.

Document Object Model

```
<body>
  <header id="hoofding">
    <h1 id="titel">Titel van de pagina</h1>
    <h2 id="ondertitel">Ondertitel</h2>
  </header>
  <article id="inhoud">
    <p id="par1">Eerste paragraaf</p>
    <p id="par2">Eerste paragraaf</p>
  </article>
</body>
```



W3C DOM

- W3C DOM
 - Werkt met het name of id-attribuut
 - De drie methodes zijn:

document.getElementById()

document.getElementsByTagName()

document.getElementsByName()

```
var tekst = document.getElementById('gemeente').value;
```

https://www.w3schools.com/js/tryit.asp?filename=tryjs_dom_innerhtml

W3C DOM

```
var tekst = document.getElementsByName("naam").item(0).value;  
var tekst = document.getElementsByName("naam")[0].value;
```

- Attributen opvragen:

```
var titel = document.getElementById("identificatie").title;  
var waarde = document.getElementById("identificatie").value;  
var klasse = document.getElementById("identificatie").class;  
var bron = document.getElementById("identificatie").src;
```

https://www.w3schools.com/js/tryit.asp?filename=tryjs_document_getelementsbytagname

W3C DOM

- CSS-stijlen opvragen:

```
var element = document.getElementById("element-id");
var hoogte = element.style.height;
var kleur = element.style.color;
var achtergrond = element.style.backgroundColor;
var bovenmarge = element.style.marginTop;
```

- **CSS-stijl aan elkaar schrijven in lowerCamelCase zonder het koppelteken**

W3C DOM

- Binnen het W3C-DOM kunnen deze methods verder aangevuld worden met de volgende formuleringen:
 - **childNodes[]**, een reeks van alle knooppunten onder dat bepaalde element, sequentieel genummerd in de volgorde waarin ze in de HTML-code voorkomen.
 - **firstChild**, het eerste onderliggend HTML-element.
 - **lastChild**, het laatste onderliggend HTML-element.

Dynamic HTML

- Tot nu toe: statische websites
- Dynamisch/interactief maken van webpagina's → m.b.v. scripttalen (JavaScript, ...)

DOM-scripting of browseronafhankelijke DHTML

- Om browser- en platformonafhankelijk te werken is het noodzakelijk om enkel met algemeen erkende technologieën te werken, nl.
 - CSS
 - JavaScript
 - het W3C DOM (Document Object Model).

Al deze standaarden worden door de huidige generatie browsers ondersteund.

Event handling

- Het gebruik van JavaScript is event based.
- Event = elke actie die de gebruiker initieert.
- Event handling: handeling uitvoeren nadat het event voorvalt.

```
<input type="button" value="Klik op mij" id="knop"/>
var knop = document.getElementById("knop");
knop.addEventListener("click", toonKlik);
function toonKlik() {
    window.alert("Je hebt op mij geklikt!");
}
https://www.w3schools.com/js/js\_htmldom\_events.asp
https://www.w3schools.com/js/js\_eventlistener.asp
```

Event handling

- Event handlers voor alle **documenten**
 - **onload**
 - Nadat een object is geladen.
 - **onunload** (& onbeforeunload)
 - Indien een object niet langer is geladen.
 - **onresize**
 - Als de afmetingen van het browservenster of van een frame worden aangepast.

Alle events:

<https://developer.mozilla.org/en-US/docs/Web/Events>

Event handling

- Event handlers voor alle **elementen**
 - **onclick**
 - Vooral voor knopachtige elementen, afbeeldingen en hyperlinks.
 - **onmousedown, onmouseup**
 - Als gebruiker met de muis op een element staat en muisknop respectievelijk indruwt en loslaat.
 - **onmouseover, onmouseout**
 - Als gebruiker met de muis naar (en over) een element beweegt, respectievelijk van een element weg beweegt.
 - **onmousemove**
 - Als de gebruiker de muis beweegt.

Event handling

- Event handlers voor **formulieren**
 - **onchange**
 - Bij tags <input>, <select> en <textarea> → als de gebruiker de waarde van deze elementen (de tekst) verandert en vervolgens met de tab toets of met de muis de focus verplaatst.
 - **onsubmit, onreset**
 - Bij <form> tag → als de form verstuurd (submit) of geledigd (reset) wordt.
 - Handler geeft de waarde false terug → form wordt niet verstuurd, geledigd.

Event handling

- Event handlers voor **formulieren**
 - **onfocus**
 - Als een element wordt geselecteerd.
 - **onblur**
 - Als de selectie van een element ongedaan wordt gemaakt.
 - **onkeydown**
 - Als er een toets ingedrukt wordt op het toetsenbord.
 - **onkeyup**
 - Als de toets van het toetsenbord terug wordt losgelaten.
 - **onkeypress**
 - Als na het onkeydown-event, de toets ingedrukt gehouden wordt (enkel printbare karakters).

Pop-up vensters

- *Alert* venster
 - Korte boodschap → gebruiker kan enkel bevestigen door op OK te klikken (meestal gebruikt voor waarschuwingen).
 - De methode heeft als argument een string die de boodschap of waarschuwing bevat.

```
window.alert('U heeft niet alle velden ingevuld');
```

Pop-up vensters

- *confirm* venster
 - Antwoord op een vraag; retourneert true of false, waarop je dan beslissingen kan baseren. Het argument van de functie bevat de gestelde vraag.

```
var status = window.confirm('Wilt u werkelijk afsluiten?');
// status bevat true of false
if (status) {
    doe_iets();
} else {
    doe_iets_anders();
}
```

Pop-up vensters

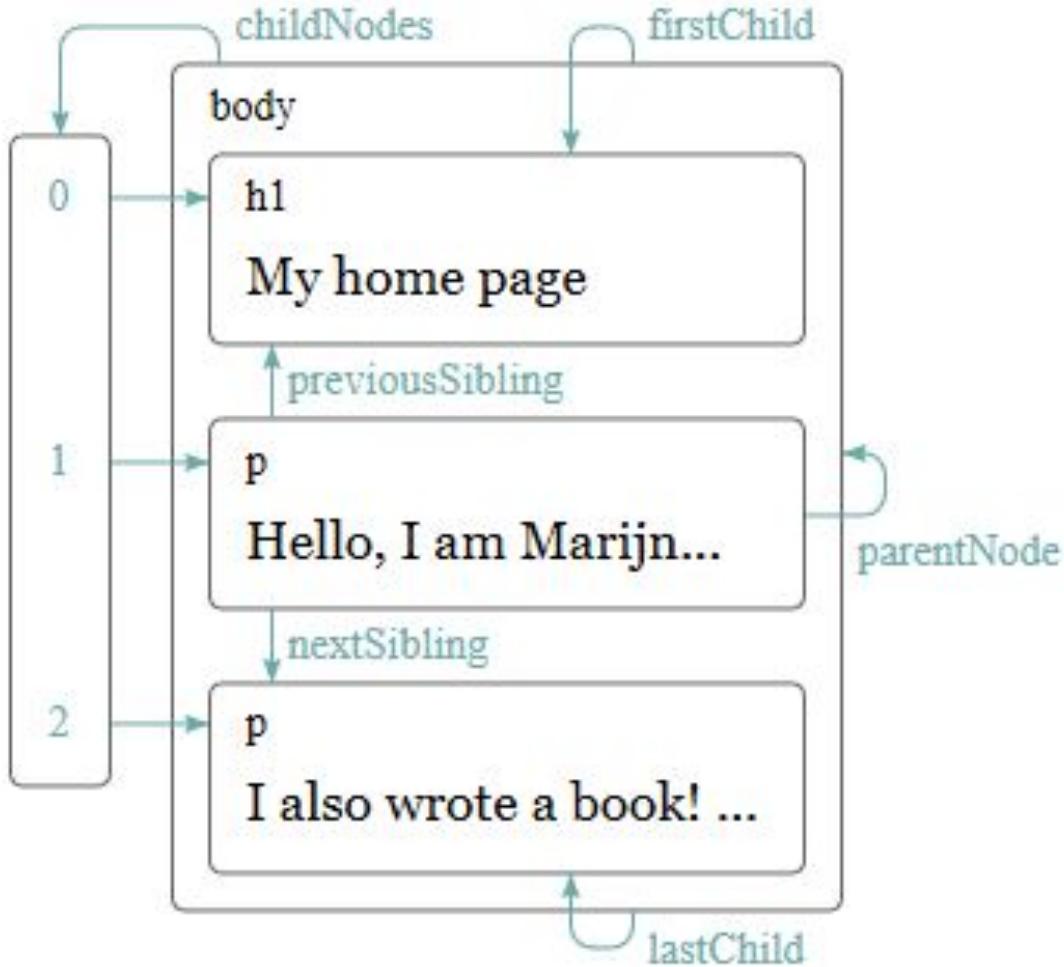
- *prompt* venster
 - Retourneert de ingegeven waarde;
 - indien niets wordt ingegeven, dan wordt null teruggegeven;
 - Het eerste argument van de functie is de gestelde vraag;
 - Het (optionele) tweede argument is de default waarde als de gebruiker op "Annuleren" klikt.

```
var antwoord = window.prompt('Geslacht: ', 'Man');
if (antwoord !== "Man") {
    document.write("Hallo girl!");
} else {
    document.write("Hallo boy!");
}
```

Eigenschappen benaderen

- De typische properties die kunnen benaderd worden zijn:
 - **x.innerHTML**: de HTML-code in het HTML-element
 - **x.innerText**: de tekst in het HTML-element
 - **x.parentNode**: de parent node van een HTML-element
 - **x.childNodes**: de child nodes van een HTML-element
 - **x.firstChild**: het eerste kind van een knooppunt
 - **x.lastChild**: het laatste kind van een knooppunt
 - **x.attributes**: de attributen van een HTML-element
 - **x.nodeName**: het name-attribuut van een HTML-element
 - **x.nodeValue**: het value-attribuut van een HTML-element

Eigenschappen benaderen



DOM methods

- Zoals al besproken in het onderdeel over het W3C DOM zijn er verschillende methodes beschikbaar om een HTML-element te benaderen.
 - *x.getElementById(id)*: via het id-attribuut
 - *x.getElementsByName(name)*: via het name-attribuut
 - *x.getElementsByTagName(tag)*: via het elementtype
 - *x.getElementsByClassName(class)*: via het class-attribuut

Aanpassen van het DOM

- Methodes om de DOM aan te passen:
 - *x.appendChild()*: voegt een onderliggend knooppunt toe
 - *x.removeChild()*: verwijdert een onderliggend knooppunt
 - *x.createElement()*: voegt een HTML-element toe
 - *x.createTextNode()*: voegt een tekstueel knooppunt toe
 - innerText en innerHTML