***Hoofdstuk 14: the document object model***

De HTML en CSS wordt door de browser omgeschreven naar de DOM structuur, die live met JavaScript aan te passen is.



Elke box is een JS object waar we iets mee kunnen. Dit is het **document object model (DOM).** Elke **Node** refereert weer naar zijn eigennode (**children**). Het document is de **root. Leaves** zijn nodes zonder children (laatst geneste node). Elke node heeft bovendien een eigen unieke code 🡪 de **nodeType property** (zo zijn elementen code 1 en tekst code 3, etc.).

Doordat de DOM ook bijv. met XML werkt, werkt het niet optimaal. Child nodes zijn bijvoorbeeld in een array, maar slice en map werken niet.

Met de DOM kan je vrijwel alles targeten:



*function talksAbout(node, string) {*

*if (node.nodeType == Node.ELEMENT\_NODE) {*

*for (let i = 0; i < node.childNodes.length; i++) {*

*if (talksAbout(node.childNodes[i], string)) {*

*return true;*

*}*

*}*

*return false;*

*} else if (node.nodeType == Node.TEXT\_NODE) {*

*return node.nodeValue.indexOf(string) > -1;*

*}*

*}*

*console.log(talksAbout(document.body, "book"));*

*// → true*

Je kan:

**- getElementByTagName(‘’a’’)[n]**

**- getElementById**

**-getElementBtClassName**

**- querySelectorAll (IS ALS ENIGE NIET LIVE GEUPDATET!!)**

En hebt:

**- remove**

**- appendChild**

**- insertBefore**

**- replaceChild**

**- createTextNode (creëert een tekst element in HTML)**

**- createElement**

*<p>The <img src="img/cat.png" alt="Cat"> in the*

*<img src="img/hat.png" alt="Hat">.</p>*

*<p><button onclick="replaceImages()">Replace</button></p>*

*<script>*

*function replaceImages() {*

*let images = document.body.getElementsByTagName("img");*

*for (let i = images.length - 1; i >= 0; i--) {*

*let image = images[i];*

*if (image.alt) {*

*let text = document.createTextNode(image.alt);*

*image.parentNode.replaceChild(text, image);*

*}*

*}*

*}*

*</script>*

Hierbij moet de loop terugtellen, omdat de DOM live verandert.

**Je kan met attribute namen het HTML element pakken 🡪 handig omdat dit alleen bij Javascript kan. Dus geen classes en ID’s gebruiken.**

Verschillende soorten positionering properties binnen javascript:

**- offsetWidth**

**- offsetHeight**

**- clientHeight (zonder de border meegenomen)**

**- clientWidth**

**- getBoundingClientRect (geeft top, bottom, left en right properties binnen een object)**

*<p><span id="one"></span></p>*

*<p><span id="two"></span></p>*

*<script>*

*function time(name, action) {*

*let start = Date.now(); // Current time in milliseconds*

*action();*

*console.log(name, "took", Date.now() - start, "ms");*

*}*

*time("naive", () => {*

*let target = document.getElementById("one");*

*while (target.offsetWidth < 2000) {*

*target.appendChild(document.createTextNode("X"));*

*}*

*});*

*// → naive took 32 ms*

*time("clever", function() {*

*let target = document.getElementById("two");*

*target.appendChild(document.createTextNode("XXXXX"));*

*let total = Math.ceil(2000 / (target.offsetWidth / 5));*

*target.firstChild.nodeValue = "X".repeat(total);*

*});*

*// → clever took 1 ms*

*</script>*

Zo kan je CSS direct manipuleren met javascript:

*<p id="para" style="color: purple">*

*Nice text*

*</p>*

*<script>*

*let para = document.getElementById("para");*

*console.log(para.style.color);*

*para.style.color = "magenta";*

*</script>*

**requestAnimationFrame** gebruiken om animaties met javascript te doen. Anders zou de browser te vaak updaten en zou het scherm vastlopen.