

DOM manipulatie



DOM manipulatie

Er zijn verschillende manieren om aan DOM manipulatie te doen. Wij bespreken het gebruik van `innerHTML`, `innerText`, de `classList`, het werken met `attributen` en het `toevoegen /verwijderen van elementen aan andere elementen`.

innerHTML

Met de eigenschap innerHTML kan je zowel de HTML van een element opvragen als aanpassen

[[codepen](#) | [MDN](#) | [w3schools](#)]

innerText

Vergelijkbaar met innerHTML, maar zal enkel de 'menselijk leesbare tekst' ophalen of aanpassen.

Sneller als je pure tekst wil aanpassen en geen html elementen wil invoeren (kan niet).

[[codepen](#) | [MDN](#) | [w3schools](#)]

classList manipuleren

De eigenschap `classList` geeft een lijst weer met de css klassen die toegepast zijn op een element.

Met `add()` en `remove()` kan je css klassen toevoegen of verwijderen.

Met `toggle()` kan je wisselen tussen toevoegen of verwijderen van een class op een element

Met `contains()` kan je nakijken of een specifieke class is toegepast op een element

[[codepen](#) | [MDN](#) | [w3schools](#)]

attributen gebruiken

- `node.getAttribute("naam")` ⇒ haal de waarde van het attribuut "naam" op
[[codepen](#) | [MDN](#) | [w3schools](#)]
- `node.hasAttribute("naam")` ⇒ true als node het attribuut "naam" heeft, anders false
[[codepen](#) | [MDN](#) | [w3schools](#)]
- `node.setAttribute("naam", waarde)` ⇒ kent "waarde" toe aan het attribuut "naam" van element node
[[codepen](#) | [MDN](#) | [w3schools](#)]
- `node.removeAttribute("naam")` ⇒ verwijder attribuut "naam" van node
[[codepen](#) | [MDN](#) | [w3schools](#)]

Toevoegen of aanmaken van Elementen

We kunnen elementen ook

aanmaken met `createElement()`,

verwijderen met `remove()`,

onderliggende elementen verwijderen met `removeChild()`,

achteraan toevoegen aan andere elemente met `append()`,

vooraan toevoegen met `prepend()`,

nakijken of ze onderliggende elementen hebben met `hasChildNodes()` of

een onderliggend element vervangen door een ander element met

`replaceChild()`

createElement, append, prepend en remove

Met `createElement()` kan je een element aanmaken van een gewenst type (div, p, ul, li, ...).

```
let nieuwElement = document.createElement('p')
```

→ maakt een nieuw **p** element aan en houdt dit bij in `nieuwElement`

`parent.append(nieuwElement)` → voegt `nieuwElement` achteraan `parent` toe

`parent.prepend(nieuwElement)` → voegt `nieuwElement` vooraan `parent` toe

`nieuwElement.remove()` → verwijdert `nieuwElement` uit het DOM

Perf matters! Pas het element aan voor je het toevoegt aan het DOM.

<code>createElement</code>	→	[codelab MDN w3schools]
<code>append</code>	→	[MDN]
<code>prepend</code>	→	[MDN]
<code>remove</code>	→	[MDN w3schools]
<code>removeChild</code>	→	[codelab MDN w3schools]