

Week 10 en 11

Javascript Deel 5



**DE HOGESCHOOL
MET HET NETWERK**

Elfde-Liniestraat 24, 3500 Hasselt, www.pxl.be



Events

Wat zijn events?

- Gebeurtenissen, 'dingen' die gebeuren met de HTML-elementen op een webpagina, zoals bijvoorbeeld:
 - het beëindigen van het laden van de webpagina
 - het veranderen van de inhoud van een invulveld
 - het klikken op een knop
 - het hoveren met de muisaanwijzer over een afbeelding
 - ...

Voorbeelden

Event	Uitleg
change	Voert een script uit wanneer een element wordt aangepast.
click	Voert een script uit wanneer op een element wordt geklikt.
dblclick	Voert een script uit wanneer op een element wordt gedubbelklikt.
mouseover	Voert een script uit wanneer over een element wordt gegaan met een muisaanwijzer.

Voorbeelden

Event	Uitleg
keydown	Voert een script uit wanneer een toetsenbordknop wordt ingedrukt.
load	Voert een script uit wanneer de browser klaar is met het laden van de webpagina.
copy	Voert een script uit wanneer content van een element wordt gekopieerd.
submit	Voert een script uit wanneer een formulier wordt gesubmitteerd.

Targets

- Events hebben altijd een target!
- Het target is het HTML-element waarop het event van toepassing is.
- Als we een event willen verwerken, moeten we zowel de eventnaam gebruiken als het target.

Event handler / Event listeners / Callbacks

Wat is een event handler?

- Met een event handler kunnen we ervoor zorgen dat onze toepassing op een event reageert.
- De event handler is meestal een functie binnen Javascript waarin andere Javascript-functies of -taken staan die worden uitgevoerd zodra het event plaatsvindt.

Drie manieren om een event handler te schrijven

- Als eventattribuut in de HTML-code
 - Dit is een oude manier waarbij je HTML- en JS-code mixt.
 - Deze manier wordt niet meer gebruikt!
- In Javascript
- Via de functie `addEventListener()` in Javascript
 - *Naar deze manier gaat onze voorkeur uit!*

Als eventattribuut in HTML

- We kunnen een event (voorafgegaan door het woord 'on') als eventattribuut toevoegen aan een HTML-element en zo het event toewijzen aan dat HTML-element.

```
// HTML-code  
<h1 id="titel" onclick="veranderTekst()">Dit is een titel!</h1>
```

```
// JS-code  
function veranderTekst() {  
    document.getElementById("titel").innerHTML = "Een nieuwe titel!";  
}
```

In javascript

- We kunnen via deze tweede manier ook een event (voorafgegaan door het woord 'on') volledig toewijzen aan een HTML-element in ons script zelf.

```
document.getElementById("knop").onclick = toonDatum;  
function toonDatum() {  
    document.getElementById("datumparagraaf").innerHTML = Date();  
}
```

Via de functie `addEventListener()`

- De functie `addEventListener()` is een nieuwere en modernere manier om event handlers te schrijven in Javascript.
- Deze functie koppelt een event aan een target zonder invloed te hebben op andere, bestaande event handlers. Het is met deze functie ook mogelijk om meerdere event handlers toe te voegen aan één target. Het target kan daarnaast eender welk DOM-object zijn en beperkt zich niet alleen tot HTML-elementen.

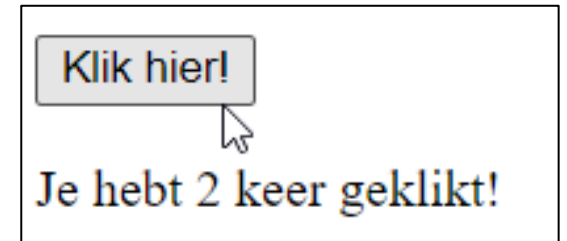
Via de functie `addEventListener()`

- De functie heeft minimaal twee belangrijke onderdelen:
 1. Het event (niet meer voorafgegaan door het woord 'on')
 2. De functie die wordt uitgevoerd wanneer het event plaatsvindt

Voorbeeld

Bij het klikken op de knop hoort een nummer op dat het aantal clicks weergeeft.

```
//HTML-code  
<button id="knop">Klik hier!</button>  
<p id="clicksparagraaf"></p>
```



Voorbeeld

```
// JS-code
let nummerClicks = 0;

document.getElementById("knop").addEventListener("click", aantalClicks);

function aantalClicks() {
    nummerClicks++;
    document.getElementById("clicksparagraaf").innerHTML = "Je hebt " +
    nummerClicks + " keer geklikt!";
}
```

Voorbeelden

Voorbeeld 2 – MouseEvents

Voorbeeld 3 – Inhoud van tekstvak hergebruiken