In deze paragraaf ga je leren wat events zijn, een stukje geschiedenis van JS en wat frameworks zijn.

Als je wilt controleren of een bepaalde actie uitgevoerd wordt kunnen events erg van pas komen. Een paar veel gebruikte events zijn: onchange, onclick, onmouseover, onmouseout, onkeydown en onload. Dit zijn allemaal acties die kunnen voorkomen bij elementen. Als zo’n actie plaatsvindt wordt dit als ‘trigger’ gebruikt voor het uitvoeren van code. Je kunt bijvoorbeeld andere informatie geven als iemand op een knop klikt.

|  |  |
| --- | --- |
| Event | Beschrijving |
| Onchange | Het element is aangepast |
| Onclick | Het element is aangeklikt |
| Onmouseover | De cursor is over het element gegaan |
| Onmouseout | De cursor is van het element af gegaan |
| Onkeydown | Een knop is ingeklikt (fysiek) |
| Onload | Het element is geladen |

Deze events kun je koppelen aan een element. Zo kun je een actie in de browser koppelen aan een stuk JS code. Om deze events ‘op te vangen’ kun je een event listener gebruiken, dit is een stuk code dat controleert wat er met het element gebeurt. Een voorbeeld is:

HTML

#CODE

<html>

<head>

<!-- stel karakterset in en verwijder waarschuwing -->

<meta charset="UTF-8" />

</head>

<body>

<header>

</header>

<div id='parent'>

<p id='element'>

Klik hier!

</p>

</div>

<footer>

</footer>

<script src="js.js"></script>

</body>

</html>

JS

#CODE

function veranderInhoud(element){

console.log(element);

element.textContent = "Het event heeft plaatsgevonden.";

alert('triggered');

}

document.getElementById('element').addEventListener("click", function(){

veranderInhoud(this);

})

Als de <p> met id ‘element’ aangeklikt wordt zal de tekst ervan veranderen. De event listener wordt dus toegevoegd aan het element en voert een functie uit als het event plaatsvindt.

JS kent verschillende versies en deze werken niet allemaal op elke browser. Als je nieuwe mogelijkheden wilt gebruiken moet je hier dus op letten. JS, ook ECMAScript genoemd kwam uit in 1997 en heeft meerdere updates gehad. De latere versies voegen vooral handige functies toe om het leven van een programmeur te verfijnen.

|  |  |
| --- | --- |
| Versie | Belangrijke veranderingen |
| ES5 | Betere ondersteuning voor JSON en handige functies voor het werken met variabelen. |
| ES6 | Meer verbeteringen voor het omgaan met variabelen in specifieke manieren. Betere omgang met arrays. Extra notatiewijze voor functies. |

Op dit moment (2018) worden versies na ES6 nog nauwelijks ondersteund. De recentere updates hebben vooral verbeteringen voor performance.

JS kent ook nog uitbereidingen/aanpassingen oftewel frameworks en libraries. Dit zijn systemen die gemaakt zijn met JS om het schrijven van JS makkelijker te maken. Ze voegen vaak functies toe en zorgen voor minder schrijfwerk. Het verschil tussen een framework en library is dat een framework een ‘skelet’ vormt voor jouw applicatie, terwijl een library een verzameling is van afkortingen.

Bekende frameworks zijn: Angular en Vue. Bekende libraries zijn: jQuery en React.

Het is makkelijker om een library te leren dan het is om een framework te leren. Frameworks hebben meer eigen conventies en restricties. Een library is meer een andere manier om hetzelfde te schrijven, terwijl een framework meer een schrijfstijl is.

#VRAGEN

1. Maak zelf een JS event.
2. Zijn nieuwe versies van JS al goed ondersteund?
3. Beschrijf het verschil tussen een library en framework in jouw eigen woorden.

#ANTWOORDEN

1. Eigen antwoord
2. Nee, dit duurt een aantal jaar en sommige browsers zullen nauwelijks bijwerken.
3. Een mogelijk antwoord is: “Een framework is een schrijfwijze en een library is een verzameling afkortingen.”.