



Hoofdpunten

JavaScript is een 'high-level' programmeertaal, wat betekent dat het redelijk dicht in de buurt komt van geschreven tekst. Door gebruik te maken van bekende Engelse termen (for, while, switch, function, if, else etc) is de code goed leesbaar en het werken in JavaScript is snel te leren. Een nadeel is dat het voor computers weer lastiger is om te begrijpen, dus code in JavaScript is vaak langzamer dan 'low-level' talen.

JavaScript is een interpreter taal, wat betekent dat de code regel voor regel wordt gelezen en uitgevoerd, en direct stopt als er een fout in de code staat. Dit in tegenstelling tot compiler talen die eerst de code omzetten in een machineleesbare variant en op dat moment al melden dat er fouten in de code staan. Bij JavaScript gebeurt dat pas bij het uitvoeren van de code. Dit heet JIT (Just-In-Time) compileren. Een gevolg is dat een deel van de code kan werken en dan plots stopt. In de console kun je dan zien op welke regel de code is gestopt, en soms met een hint naar de oorzaak.

JavaScript is object-based, wat verschilt van object-oriented. De basisprincipes van het werken met objecten zoals het groeperen van variabelen en het koppelen van methoden (functies) komt in JavaScript wel voor, bv. `console.log("test")` is de methode `log` van het object `console`. Het helemaal zelf definiëren van nieuwe objecten is beperkt tot het maken van uitbreidingen op het *prototype* object. Wanneer met de Class instructie dus een nieuw object wordt toegevoegd, is dat niet, zoals bij een object-oriënted taal een geheel nieuw object, maar een object van het type *prototype*. Alle eigenschappen van *prototype* zijn aanwezig in ieder nieuw object.

JavaScript valt onder ECMAScript. Dat is de 'officiële' definitie van de talen waar JavaScript ook onder valt. Iedere versie begint met ES. De laatste 'grote' versie was versie 6, ES2015. Ieder jaar verschijnen er nieuwe features en functies. Het jaartal zit in de versie verwerkt. De 14de editie was ES2023.

Ook al is JavaScript een officieel gedefinieerde taal, toch maakt ieder browserfabrikant hier aanpassingen op, om de eigen browser beter te laten werken. JavaScript-programma's moeten daarom steeds op alle browsers getest worden die minimaal ondersteund moeten worden volgens de opdrachtschrijving. Het is lang zo geweest dat Internet Explorer radicaal anders werkte dan andere browsers, maar ondertussen maakt ook Microsoft Edge gebruik van de chromium engine waardoor de 'grote' browsers Chrome, Firefox en Edge elkaar weinig meer ontlopen. Toch is het verstandig om met de opdrachtgever afspraken te maken welke browsers gebruikt moeten worden in de testfase, want het is onmogelijk om alle browsers en browserversies te testen.

JavaScript is event driven, dat houdt in dat de taal vooral geschreven is om te reageren op acties van de gebruiker, zoals het oproepen van een pagina, het scrollen, het schuiven of klikken met de muis, het intoetsen van toetsen op het toetsenbord, het aanklikken van

formulievelden, en op mobiele devices, het uitvoeren van 'swipe'- en 'touch'-handelingen. Daarnaast zijn er events die vanuit de browser of van andere apparaten kunnen worden opgeroepen, zoals timers, push-berichten, scherm-slaapstand of andere binnenkomende informatie zonder gebruikersinteractie.

Referenties:

[Wikipedia](#)

[ECMAScript editions](#)