



Hoofdpunten

JavaScript is een object-based programmeertaal. Dit in tegenstelling tot object-oriented programmeertalen (OOP) zoals JAVA die alle eigenschappen van OOP ondersteunen. Zie [Wikipedia](#).

De oudere versies van JavaScript waren prototype based, de nieuwere versies ondersteunen ook het gebruik van classes hoewel dit strikt genomen een 'syntactic sugar' (een suikerlaagje om de syntax, oftewel een nieuwe manier om het meer te laten lijken op andere programmeertalen) van het oude prototypemodel is.

JavaScript valt onder ECMAScript, dit is de 'officiële' benaming. Vandaar de ES in de nieuwe versies.

De eerste versie van JavaScript heette 'Mocha'.

JavaScript en JAVA hebben niets met elkaar te maken, het was een marketingtruc om deze benaming te 'lenen'.

Ieder jaar verschijnen er nieuwe features en functies. Het jaartal zit in de versie verwerkt (ES2015 is in 2015 officieel uitgekomen). Theoretisch is er elk jaar een nieuwe release. Zie [ECMAScript editions](#).

Iedere browser ondersteunt JavaScript, maar niet iedere browser ondersteunt de laatste features en nieuwe functies. JavaScript-programma's moeten daarom steeds met alle browsers getest worden die minimaal ondersteund moeten worden volgens de opdrachtschrijving. Maar eigenlijk zouden ontwikkelaars uit principe geen opdrachten moeten accepteren om code voor verouderde versies van browsers te genereren, deze bevatten immers te veel risico's en kwetsbaarheden

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 formulervelden, het invullen of veranderen van formulervelden, en op mobiele devices, het uitvoeren van 'swipe' en 'touch' handelingen.

JavaScript is platformonafhankelijk en kan zowel in iedere browser uitgevoerd worden als onder ieder operating system, dus ook op webserverns.

JavaScript is officieel een scripting language, dat houdt in dat het (in de vorm waarin het geschreven is) uitgevoerd (geïnterpreteerd) wordt door de omgeving waar het draait, terwijl een programming language gecompileerd wordt (omgezet in machinetaal of bytecode) en vervolgens zelfstandig uitgevoerd kan worden (bijvoorbeeld als .exe-bestand onder Windows).