

# Inhoud

<b>1</b>	<b>Kennismaken met JavaScript</b>	<b>1</b>
	Een korte geschiedenis van JavaScript	2
	Kernbegrip – JavaScript core	5
	Voorkennis	8
	Ontwikkelhulpmiddelen voor JavaScript	9
	JavaScript-debuggers	12
	Uw eerste JavaScript	13
	JavaScript-functies	16
	De debugger gebruiken	20
	JavaScript-code in extern bestand	25
	Conclusie	27
	Praktijkoefeningen	27
<b>2</b>	<b>Statements, gegevenstypen en variabelen</b>	<b>31</b>
	De syntaxis van JavaScript	32
	Statements	32
	Werken met variabelen	35
	Gereserveerde woorden	37
	Commentaar	38
	Gevenstypen	39
	Getallen converteren met parseInt() en parseFloat()	41
	Tekenreeksen of strings	44
	Verschillende stringfuncties	47
	Booleaanse waarden	49
	Objecttypen	50
	Conclusie	50
	Praktijkoefeningen	51

<b>3</b>	<b>Operatoren</b>	<b>53</b>
	Variabelen bewerken met operatoren	54
	Bewerkingsvolgorde	65
	Praktijkoefeningen	68
<b>4</b>	<b>Beginnen met functies, arrays en objecten</b>	<b>71</b>
	Complex gegevenstype 1 – Functies	72
	Structuur van een functie	73
	Parameters doorgeven	77
	Waarden retourneren	80
	Complex gegevenstype 2 - Arrays	83
	Arraymethoden	87
	Complex gegevenstype 3 - Objecten	92
	Waarden van objecten uitlezen	97
	Conclusie	99
	Praktijkoefeningen	100
<b>5</b>	<b>Program flow controleren</b>	<b>103</b>
	Inleiding – verschillende typen lussen	104
	If-else	104
	While()	109
	Het statement for()	111
	De statements break, continue en return	114
	Het statement for-in	115
	Conclusie	117
	Praktijkoefeningen	118
<b>6</b>	<b>JavaScript-events en event handlers</b>	<b>121</b>
	Wat zijn events?	122
	Voorbeelden van events en event handlers	126
	Klikken op knoppen afvangen	137
	Inhoud van een tekstvak ophalen	140
	Toetsenbordevents afvangen	142
	Conclusie	146
	Praktijkoefeningen	147

<b>7</b>	<b>Werken met het DOM</b>	<b>151</b>
	Wat is het DOM?	152
	Elementen in het DOM selecteren	155
	Elementen in het DOM toevoegen en verwijderen	164
	Overige DOM-functies	173
	Praktijkoefeningen	174
<b>8</b>	<b>Gevorderd gebruik van functies</b>	<b>177</b>
	Functies met een variabel aantal parameters	178
	Immediately Invoked Function Expressions	183
	De functies call() en apply()	193
	Werken met closures	197
	Conclusie	203
	Praktijkoefeningen	204
<b>9</b>	<b>Modulair programmeren in JavaScript</b>	<b>207</b>
	Waarom modulair programmeren?	208
	Kennismaken met design patterns	209
	Patroon 1 – Code groeperen in object literals	210
	Patroon 2 – Prototype pattern	214
	Patroon 3 – Revealing module pattern	220
	Overerving in JavaScript	223
	Conclusie	228
	Verder lezen	229
	Praktijkoefeningen	230

# Kennismaken met JavaScript

*HTML is al ruim vijftientig jaar de standaard voor het maken van websites. HTML kan echter niet alles. In HTML wordt alleen de structuur van pagina's beschreven. JavaScript is de aanvullende programmeertaal om HTML interactief te maken. Het is de populairste programmeertaal op internet. Elke browser heeft een ingebouwde JavaScript-motor, waardoor moderne webapps mogelijk worden. JavaScript staat daarmee aan de basis van elke techniek die de moderne web developer moet kennen. Of u later nu aan de slag gaat met Angular, webapps gaat maken met PhoneGap of React, of uw eigen bibliotheekje met helperfuncties maakt: zonder JavaScript bent u nergens. Dit inleidende hoofdstuk toont de algemene kenmerken van JavaScript en laat zien welke tools u nodig hebt om succesvol met JavaScript aan de slag te kunnen gaan. Natuurlijk schrijft u alvast een eerste JavaScript voor snel resultaat.*

**In dit hoofdstuk:**

*Een korte geschiedenis van JavaScript.*

*Waarvoor wordt JavaScript gebruikt?*

*Belangrijke begrippen die u moet kennen bij het werken met JavaScript.*

*Welke tools hebt u nodig bij het programmeren?*

*Hoe JavaScript en HTML gecombineerd worden in webapps.*

*Een eerste script schrijven en de tags `<script>...</script>`.*

*Kennismaken met JavaScript-debugging.*

# Een korte geschiedenis van JavaScript

## Brendan Eich

JavaScript is oorspronkelijk in 1995 ontwikkeld door Brendan Eich, die bij Netscape werkte. Netscape is het bedrijf dat een van de oerbrowsers voor internet maakte, Netscape Navigator. Ook toen al was JavaScript bedoeld als uitbreiding van HTML om meer interactiviteit op webpagina's mogelijk te maken. De combinatie van HTML en JavaScript stond destijds bekend onder de naam *Dynamic HTML* (DHTML).

De browsers uit die tijd (Internet Explorer 4 en Netscape 4) boden ondersteuning voor de combinatie van HTML en JavaScript in webpagina's. Ondertussen zijn we natuurlijk vele browserversies verder. JavaScript is uitgegroeid tot een van de belangrijkste pijlers binnen de browser en in moderne webapplicaties ('webapps'). Zonder JavaScript zouden geen Facebook, Gmail, e-commerce of internetbankieren bestaan. Alle huidige browsers (Internet Explorer, Firefox, Chrome, Safari enzovoort) kunnen JavaScript uitvoeren.

## ECMAScript, JavaScript en versienummers

In de loop der jaren is het versienummer van JavaScript steeds licht opgehoogd. Eerst liep het versienummer omhoog met het verschijnen van een nieuwe browserversie. Nu is de ontwikkeling van JavaScript losgekoppeld van nieuwe versies van de browser. JavaScript is inmiddels omgevormd tot een officiële, genormeerde en gestandaardiseerde programmeertaal. Dit is gedaan door de structuur en inhoud van de taal te laten goedkeuren en standaardiseren door de organisatie European Computer Manufacturers Association (ECMA). Hiermee is de ontwikkeling van JavaScript nu in handen van een onafhankelijk instituut en niet meer van één browserfabrikant.

Vroeger spraken we dus van JavaScript 1.0, 1.5 enzovoort, tegenwoordig wordt gesproken van ECMAScript 5 en de toekomstige versie ECMAScript 6 (vaak afgekort tot ES6).

---



### Wat doet ECMA?

ECMA heeft tot taak technische standaarden te publiceren en hierop toe te zien. ECMA houdt zich niet alleen bezig met JavaScript, maar ook met andere computergerelateerde standaarden, zoals het bestandsformaat voor cd-roms, de specificaties van de programmeertaal C# en het bestandsformaat Office Open XML. Wilt u hier meer over weten, bezoek dan [www.ecma-international.org](http://www.ecma-international.org).

---

De JavaScript-specificatie die we in dit boek bespreken is vastgelegd in een document met het nummer 262 (en daarvan inmiddels de vijfde versie). Daarom heet JavaScript officieel *ECMAScript-262 5th Edition*. Maar voor het gemak heeft iedereen het altijd gewoon over JavaScript. Dat doen we ook in dit boek.

---



### Nieuwe versie van JavaScript

JavaScript is een taal die continu in ontwikkeling is. Ondertussen is ook een officiële volgende versie van JavaScript verschenen. Deze heeft verschillende namen: EcmaScript 6 (of kortweg ES6), EcmaScript 2015 of EcmaScript Next. Deze versie wordt echter niet algemeen ondersteund door alle browsers. Met name mobiele browsers (oudere Android-versies en IOS tot en met IOS9) laten het afweten. In dit boek leert u JavaScript dat overal goed werkt. Wilt u meer weten over ES6, lees dan ná dit boek ook *Web Development Library - EcmaScript 6 en TypeScript* (ISBN 9789059409323).

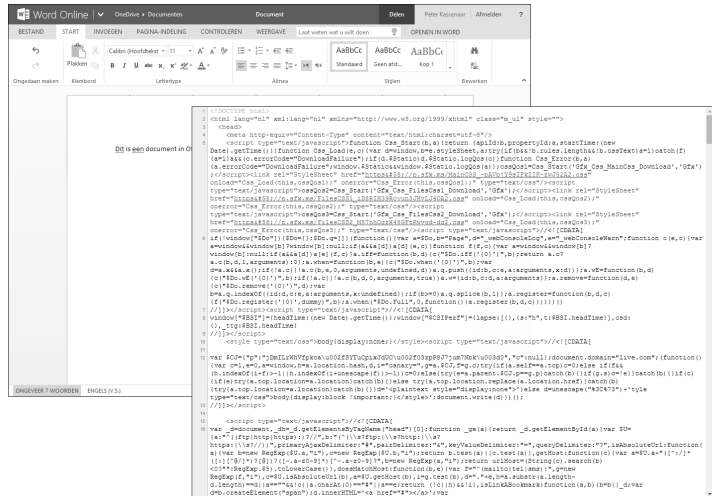
---

## Waarvoor wordt JavaScript gebruikt?

We hebben al globaal aangegeven dat JavaScript wordt gebruikt om gedrag of interactie aan webpagina's toe te voegen. Maar wat wordt daar dan precies mee bedoeld? Denk bijvoorbeeld aan de volgende toepassingen. Er zijn er nog veel meer, maar dit is alvast een begin:

- **Formuliervalidatie** JavaScript is erg geschikt om de ingevulde gegevens in een webformulier op een pagina te controleren voordat het formulier wordt verzonden. Omdat deze controle op de computer van de gebruiker plaatsvindt, gaat dit veel sneller dan controle op de webserver na het versturen. Door het gebruik van JavaScript is dus geen *roundtrip* nodig naar de server en kan via een lokale melding direct worden aangegeven dat iemand bijvoorbeeld een ongeldig e-mailadres heeft ingevuld.
- **Uitrolmenu's en afbeeldingen** Met JavaScript kunnen menu's en afbeeldingen tijdens het gebruik van de pagina worden vervangen. Dit kan bijvoorbeeld van pas komen bij fotocarrousel of uitklapmenu's.
- **Aanpassingen van stijlen en animatie** JavaScript kan in een pagina de aanwezigheid, positie en inhoud van elk element (teksten, afbeeldingen enzovoort) ophalen en manipuleren. Zo kunnen bijvoorbeeld kaders op een pagina vloeiend open- en dichtschuiven, menu's dynamisch worden uitgebreid, muisklikken van de rechtermuisknop worden afgevangen en aangepast en zo verder. Er zijn tal van kant-en-klare JavaScript-bibliotheken beschikbaar waarin al vele animatiefuncties zijn voorgeprogrammeerd. Deze kunt u op de pagina laden en (bijna) direct gebruiken. *jQuery* is hiervan een van de bekendste.
- **Ajax-webapplicaties** U hebt misschien de term Ajax wel eens gehoord. Het staat voor *Asynchronous JavaScript And XML*. Dit wil zeggen dat na het laden van de pagina asynchroon delen

van de pagina ververst of aangepast kunnen worden. Tegenwoordig wordt hiervoor JSON gebruikt in plaats van XML. Het hele idee van applicaties op het web zoals Office 365, Facebook, Gmail en Twitter is gebaseerd op gegevensuitwisseling op de achtergrond (*asynchroon*) met JavaScript en JSON. Zonder JavaScript zou het web in zijn huidige vorm niet bestaan!



**Afbeelding 1.1** Office Online (Office 365) is geheel geschreven in JavaScript. Het wordt ook wel ‘de grootste JavaScript-applicatie ter wereld’ genoemd.

## Kernbegrip – JavaScript core

Het is belangrijk om te weten dat de *taal* JavaScript uit een relatief kleine set instructies bestaat. Er zijn opdrachten om te werken met variabelen, lussen, teksten, arrays en objecten, maar verder is er in vergelijking met andere programmeertalen niet zo veel bijzonders. JavaScript bevat bijvoorbeeld geen opdrachten voor invoer en uitvoer, er zijn geen netwerkmogelijkheden of mogelijkheden voor het werken met lokale bestanden. De browser is een ‘sandbox’. In talen als Java, PHP of C# zijn dergelijke zaken wel opgenomen.



In JavaScript worden dit soort uitgebreide handelingen overgelaten aan de zogenoemde *hosting environment*. En op internet is de webbrowser die host waarin JavaScript draait. JavaScript maakt bijvoorbeeld gebruik van het browservenster om teksten op de pagina te tonen en te manipuleren. Als JavaScript communiceert met een script op de webserver, maakt het gebruik van de browsermogelijkheden om netwerkverbindingen en (Ajax-)aanroepen op te zetten. De *taal* JavaScript zelf biedt hiervoor geen voorzieningen.

## Indeling van dit boek

Dit boek bestaat uit twee delen:

- **Deel 1 – Kernmogelijkheden van JavaScript** De hoofdstukken 1 tot en met 7 gaan over de kernmogelijkheden van JavaScript. U leert de taal goed kennen door eenvoudige programma's te schrijven met de gereserveerde JavaScript-woorden. U maakt kennis met variabelen, lussen en overige JavaScript-syntaxis. Dit is de kern van elke programmeertaal. JavaScript is hierop geen uitzondering. Als u deze onderdelen goed beheerst, kunt u JavaScript in tal van omgevingen toepassen. U leert ook beknopt werken met het DOM (de webpagina).
- **Deel 2 – Gevorderde JavaScript-onderwerpen** In de hoofdstukken 8 en 9 gebruikt u de onderwerpen uit deel 1 om uw JavaScript-kennis verder uit te breiden. U leert meer over het uitgebreid werken met functies en ziet hoe JavaScript op een objectgeoriënteerde manier ingezet kan worden. Beginnende programmeurs zullen deze technieken misschien niet direct inzetten. Maar zodra u wat grotere programma's gaat schrijven, of scripts van internet download en wilt uitbreiden, komt deze kennis goed van pas.



## Meer dan de browser

Webbrowsers zoals Chrome, Firefox, Internet Explorer en Safari zijn zonder twijfel de bekendste omgevingen om JavaScript-toepassingen uit te voeren. Maar er zijn meer varianten van JavaScript. Omdat het een gestandaardiseerde taal is, zijn er veel afgeleiden ontwikkeld. Zo zijn Adobe Flex en Flash ActionScript ook dialecten van JavaScript. Daarbij is Flash Player de hosting environment waarin het programma wordt uitgevoerd. Inmiddels is node.js een bekende omgeving om JavaScript op de server uit te voeren (zie [nodejs.org](http://nodejs.org) voor meer informatie). In dit boek gebruiken we overigens altijd een webbrowser, omdat die voor iedereen die met JavaScript aan de slag wil direct toegankelijk is. U hoeft er niks extra voor te installeren of te downloaden.

---

## Oefenbestanden downloaden

Alle codevoorbeelden en -fragmenten die in de tekst worden genoemd zijn als voorbeeldbestanden te downloaden. Het adres hiervoor is [www.webdevelopmentlibrary.nl](http://www.webdevelopmentlibrary.nl) (klik op **Downloads**). Soms gaat het maar om enkele regeltjes code, maar dat kan net genoeg zijn om u op weg te helpen of om als startpunt te dienen voor uw eigen experimenten. De voorbeelden zijn verdeeld in mappen per hoofdstuk.

## Voorkennis

Kan iedereen JavaScript gebruiken? Worden aan de JavaScript-programmeur speciale eisen gesteld? We geven kort aan welke voorkennis nodig is en op welke wijze u uw kennis eventueel kunt bijspijkeren.

Om JavaScript te kunnen gebruiken is in principe weinig voorkennis nodig. JavaScript staat als leesbare platte tekst in de broncode van het webdocument of in een apart (gekoppeld) scriptbestand.

Omdat JavaScript een programmeertaal is, is het zonder meer een voordeel als u enige ervaring hebt met programmeren. Zelfs met uitsluitend wat basiskennis van PHP of eenvoudig Java hebt u al een voorsprong. De statements, de code, de controlestructuren en de syntaxis zult u wat sneller onder de knie kunnen krijgen. Hebt u eerder vooral andere scripts en misschien wat jQuery-code gekopieerd en geplakt, dan leert u nu eindelijk wat al die haakjes, puntkomma's en accolades betekenen. Hebt u nog geen programmeerervaring, dan is dat geen enkel probleem. Begin gewoon te oefenen in het volgende hoofdstuk, dan hebt u aan het einde van het boek JavaScript goed onder de knie.

## Bekendheid met HTML en CSS

We gaan ervan uit dat u bekend bent met HTML. Als u vertrouwd bent met tags, id's, attributen en de andere elementen waaruit een webpagina is opgebouwd, zult u deze snel kunnen aanpassen om ze zo met behulp van JavaScript op de pagina te manipuleren.

Kent u dit nog niet, lees dan eerst een ander boek, waarin de basisbeginselen van HTML uiteengezet worden, bijvoorbeeld *Web Development Library HTML5* (ISBN 9789059408081) en *Web Development Library - CSS3* (ISBN 9789059408098).

In dit boek staan we niet verder stil bij de HTML- en CSS-syntaxis van elementen. Zij worden bekend verondersteld.

```
1 <!DOCTYPE HTML>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>Titel van het document</title>
6 <style>
7   /* alle stijlen in het document */
8 </style>
9 </head>
10
11 <body>
12 <div id="container">
13   <!-- inhoud van het document -->
14 </div>
15 </body>
16 </html>
17
```

**Afbeelding 1.2** Bekendheid met de notatie van HTML en CSS is een vereiste. We gebruiken in dit boek het HTML5-documenttype.

## Wat hoeft u niet te weten?

U hoeft niet iets te weten van serversided programmeertalen zoals PHP, Java of C#. Ook hoeft u geen beschikking te hebben over een webserver of eigen domein. In dit boek gaan we uit van clientsided JavaScript. Dit betekent dat scripts op letterlijk elke pagina gebruikt kunnen worden. Het script maakt deel uit van de webpagina. In de meeste gevallen is het zelfs niet nodig dat u online bent voor het uitvoeren van de oefeningen. Een editor en een browser zijn veelal voldoende.

## Ontwikkelhulpmiddelen voor JavaScript

JavaScript is gewoon platte tekst. In principe kunt u daarom met Kladblok (Windows) of Teksteditor (Mac) al aan de slag. Maar in de praktijk is het wel erg handig om met een gespecialiseerde editor aan de slag te gaan. We noemen er in deze paragraaf enkele, zodat u zelf een keuze kunt maken. Naast het schrijven van JavaScript is het opsporen en verhelpen van fouten in een script

erg belangrijk. Hiervoor zijn inmiddels gelukkig ook goede hulpmiddelen beschikbaar. Deze komen aan het eind van de paragraaf aan de orde.

---



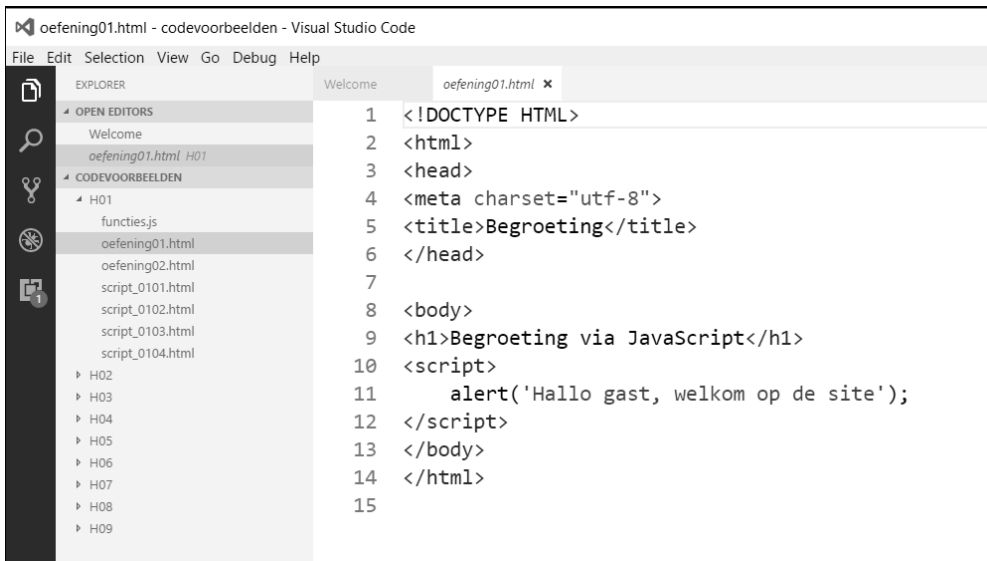
### WYSIWYG-editors ongeschikt

Webontwikkelaars die uit de vormgevingshoek afkomstig zijn, gebruiken graag een editor waarmee webpagina's op visuele wijze worden vormgegeven. Zij werken bijvoorbeeld in de ontwerpweergave van Adobe Dreamweaver of gebruiken aparte tools als Adobe Muse of Microsoft Expression Web. Het samenvattende begrip hiervoor is *What You See Is What You Get*-editors (*WYSIWYG*). Voor het schrijven van JavaScript zijn die niet geschikt. U zult echt met de code zelf aan de slag moeten om de beste resultaten te bereiken. In dit boek worden visuele editors niet gebruikt.

---

- **Visual Studio Code** Microsoft heeft sinds enige jaren een uitstekende, gratis, open source editor beschikbaar. Deze heet Visual Studio Code (niet te verwarren met het grote, betaalde Visual Studio) en is beschikbaar voor Mac, Windows en Linux. Uw exemplaar is te downloaden vanaf [code.visualstudio.com](http://code.visualstudio.com).
- **JetBrains WebStorm** WebStorm is een editor die zich profileert als 'speciaal gemaakt voor JavaScript'. Er zijn uitgebreide voorzieningen aanwezig voor het profileren van documenten, herschrijven van code, testen en het automatisch springen naar functies. Zie [www.jetbrains.com/webstorm/](http://www.jetbrains.com/webstorm/) voor meer informatie en een 30-dagenversie.
- **Sublime Text** Een andere bekende open source editor is Sublime Text, te vinden op [www.sublimetext.com](http://www.sublimetext.com). Deze editor is van zichzelf een tamelijk 'kaal' product, maar kan exact naar wens worden ingesteld met tientallen plug-ins voor

kleurcodering, automatisch formateren, FTP, code hints en testing-tools.



**Afbeelding 1.3** *Visual Studio Code is een goede, gratis editor. Erg geschikt voor programmeren in JavaScript.*

Maar zoals gezegd kunt u met elke editor die een document als platte ASCII-tekst kan opslaan uit de voeten. Kan geen van de hiervoor genoemde editors u bekoren, dan kunt u het proberen met Atom, Sublime Tekst, Brackets, Notepad++ of een andere editor. Als die ook niet bevallen, dan weten wij het ook niet meer. Veel plezier met Kladblok in dat geval.