

# Welkom bij de Web Development Library

Geen enkel ander onderwerp staat de laatste jaren zo in de belangstelling als het ontwikkelen van goede websites en webapps. Bedrijven en organisaties nemen geen genoegen meer met eenvoudige HTML-websites of de standaardlay-out van een WordPress-site. Dit betekent dat ook de gereedschapskist van elke webdeveloper steeds beter gevuld moet zijn. Welkom bij de Web Development Library!

De serie *Web Development Library* bestaat uit een reeks compacte boeken, waarin op een wat hoger niveau onderwerpen uit het vakgebied webdevelopment worden behandeld. Elk boek bespreekt één onderwerp. Voorkennis over gerelateerde technieken wordt hierbij als bekend verondersteld. Beschikt u nog niet over alle voorkennis? Dan wordt dit opgelost in een ander deel uit de reeks.

De serie bevat op die manier alles wat u nodig hebt om zelf moderne websites en -apps te kunnen maken. De Web Development Library kan worden gebruikt bij zelfstudie, in opleidingstrajecten of bij leergangen. De reeks bestaat bij aanvang uit ongeveer vijf titels, maar wordt snel uitgebreid.

Onderwerpen die in de serie aan de orde komen (elk in een eigen uitgave) zijn onder meer

- JavaScript
- jQuery
- jQuery Mobile
- PhoneGap
- AngularJS

Kijk op [www.webdevelopmentlibrary.nl](http://www.webdevelopmentlibrary.nl) of [www.vanduurenmedia.nl](http://www.vanduurenmedia.nl) voor de nieuwste titels.

# Inhoud

<b>1</b>	<b>Kennismaken met JavaScript</b>	<b>1</b>
	<b>Een korte geschiedenis van JavaScript</b>	<b>2</b>
	Brendan Eich	2
	ECMAScript, JavaScript en versienummers	2
	Waarvoor wordt JavaScript gebruikt?	3
	<b>Kernbegrip – JavaScript core</b>	<b>5</b>
	Indeling van dit boek	6
	Oefenbestanden downloaden	7
	<b>Voorkennis</b>	<b>8</b>
	Bekendheid met HTML en CSS	8
	Wat hoeft u niet te weten?	9
	<b>Ontwikkelhulpmiddelen voor JavaScript</b>	<b>9</b>
	<b>JavaScript-debuggers</b>	<b>12</b>
	<b>Uw eerste JavaScript</b>	<b>13</b>
	Commentaar gebruiken	14
	<b>JavaScript-functies</b>	<b>16</b>
	Parameters	16
	Een inline event handler schrijven	18
	<b>De debugger gebruiken</b>	<b>21</b>
	<b>JavaScript-code in extern bestand</b>	<b>25</b>
	<b>Conclusie</b>	<b>26</b>
	<b>Praktijkoefeningen</b>	<b>27</b>
<b>2</b>	<b>Statements, gegevenstypen en variabelen</b>	<b>29</b>
	<b>De syntaxis van JavaScript</b>	<b>30</b>
	<b>Statements</b>	<b>30</b>
	Structuur van statements	31
	Hoofdletters en kleine letters	32
	<b>Werken met variabelen</b>	<b>33</b>
	De naamgeving van variabelen	33
	<b>Gereserveerde woorden</b>	<b>35</b>

<b>Commentaar</b>	<b>36</b>
<b>Gegevenstypen</b>	<b>36</b>
Primitieve- of enkelvoudige gegevenstypen	37
Strongly typed en loosely typed	37
Numbers	38
<b>Getallen converteren met parseInt() en parseFloat()</b>	<b>38</b>
Verkorte schrijfwijze: nesting	41
<b>Tekenreeksen of strings</b>	<b>41</b>
Lege string	42
Speciale tekens in strings	42
Escapetekens	43
Een backslash tonen	44
<b>Vershillende stringfuncties</b>	<b>44</b>
<b>Booleaanse waarden</b>	<b>46</b>
<b>Objecttypen</b>	<b>47</b>
<b>Conclusie</b>	<b>48</b>
<b>Praktijkoefeningen</b>	<b>48</b>
<b>3 Operatoren</b>	<b>51</b>
<b>Variabelen bewerken met operatoren</b>	<b>52</b>
Toewijzingsoperatoren	53
Verkorte schrijfwijze	53
Nog kortere schrijfwijze: increment en decrement	54
Wiskundige operatoren	55
Stringoperatoren	56
Logische operatoren	56
Vergelijkingsoperatoren	57
De operatoren == en ===	58
De voorwaardelijke operator ?, :	60
De operator typeof	61
<b>Bewerkingsvolgorde</b>	<b>63</b>
Vorrangsregels of operator precedence	63
Voorbeelden	64
Bij twijfel, gebruik haakjes	65
<b>Praktijkoefeningen</b>	<b>65</b>

<b>4</b>	<b>Beginnen met functies, arrays en objecten</b>	<b>69</b>
	<b>Complex gegevenstype 1 – Functies</b>	<b>70</b>
	Herhaald taken uitvoeren	70
	Parameters	71
	<b>Structuur van een functie</b>	<b>72</b>
	Andere notatie	73
	Anonieme functies	73
	Functies aanroepen	74
	<b>Parameters doorgeven</b>	<b>75</b>
	Naamgeving van parameters binnen en buiten de functie	75
	Regels voor parameters	76
	<b>Waarden retourneren</b>	<b>77</b>
	Eén waarde retourneren	78
	Meerdere waarden retourneren	79
	<b>Complex gegevenstype 2 - Arrays</b>	<b>80</b>
	Arrayelementen uitlezen en toevoegen	81
	Arrays in de debugger	81
	Lengte van array	83
	<b>Arraymethoden</b>	<b>83</b>
	.join()	84
	.reverse()	85
	.sort()	85
	.push()	86
	.pop()	87
	Overige arraymethoden	87
	<b>Complex gegevenstype 3 - Objecten</b>	<b>88</b>
	Eigenschappen, namen en waarden	88
	Accolades	89
	Complexe objecten	90
	this	91
	Classes	91
	<b>Waarden van objecten uitlezen</b>	<b>92</b>
	<b>Conclusie</b>	<b>94</b>
	<b>Praktijkoefeningen</b>	<b>95</b>

<b>5</b>	<b>Program flow controleren</b>	<b>97</b>
	<b>Inleiding – verschillende typen lussen</b>	<b>98</b>
	<b>If-else</b>	<b>98</b>
	Accolades	100
	else	100
	Veelgemaakte fout: toekenning in plaats van vergelijking	101
	Nogmaals: de vergelijksoperator	101
	Conclusie	102
	<b>While()</b>	<b>103</b>
	<b>Het statement for()</b>	<b>105</b>
	Parameters voor de for-lus	105
	Voorbeeld – de tafel van tien met for()	106
	<b>De statements break, continue en return</b>	<b>108</b>
	<b>Het statement for-in</b>	<b>109</b>
	<b>Conclusie</b>	<b>111</b>
	<b>Praktijkoefeningen</b>	<b>112</b>
<b>6</b>	<b>JavaScript-events en event handlers</b>	<b>115</b>
	<b>Wat zijn events?</b>	<b>116</b>
	Procedureel programmeren	116
	Eventgeoriënteerd programmeren	117
	Naamgeving van events	118
	Target	118
	Event handlers of callbacks	118
	De functie addEventListener()	119
	<b>Voorbeelden van events en event handlers</b>	<b>120</b>
	Controleren of het document geladen is	120
	Muisevents afvangen	122
	De parameter e gebruiken in event handlers	125
	Eigenschappen van de event e analyseren	127
	<b>Klikken op knoppen afvangen</b>	<b>130</b>
	Alternatieve notatie voor de event handler	132
	<b>De inhoud van een tekstvak ophalen</b>	<b>132</b>
	<b>Toetsenbordevents afvangen</b>	<b>135</b>
	<b>Conclusie</b>	<b>138</b>
	<b>Praktijkoefeningen</b>	<b>138</b>

<b>7</b>	<b>Werken met het DOM</b>	<b>143</b>
	<b>Wat is het DOM?</b>	<b>144</b>
	Begrippen	145
	<b>Elementen in het DOM selecteren</b>	<b>147</b>
	Selecteren via Id	148
	Selecteren via Type	149
	Selecteren via CSS-klasse	150
	querySelector() en querySelectorAll() – selecteren met CSS-selectors	152
	Voorbeeld – radiobuttons selecteren en uitlezen	154
	<b>Elementen in het DOM toevoegen en verwijderen</b>	<b>156</b>
	Elementen maken met document.createElement()	156
	Textnodes maken	158
	Elementen invoegen in het DOM	158
	.appendChild()	158
	.insertBefore()	160
	.removeChild()	161
	.replaceChild()	163
	<b>Overige DOM-functies</b>	<b>164</b>
	<b>Praktijkoefeningen</b>	<b>166</b>
<b>8</b>	<b>Gevorderd gebruik van functies</b>	<b>169</b>
	<b>Functies met een variabel aantal parameters</b>	<b>170</b>
	De parameter arguments	171
	Parametertype controleren	172
	arguments[] is geen echte array	173
	<b>Immediately Invoked Function Expressions</b>	<b>174</b>
	Iffy	174
	Uitvoeringscontext en global scope	175
	Haakjes rondom functiedefinitie	176
	Parameters meegeven aan een iffy	177
	Iffy gebruiken om module te maken	178
	De module uitbreiden	179
	Een mini-jQuery maken	180
	De HTML-code	182
	De JavaScript-code	182
	Mogelijke verbeteringen	183
	<b>De functies call() en apply()</b>	<b>184</b>
	.call()	185
	.apply()	186
	Nuttige toepassing van .call()	187

<b>Werken met closures</b>	<b>188</b>
Geen closure	189
Wel een closure	190
Praktisch voorbeeld van een closure	191
<b>Conclusie</b>	<b>194</b>
<b>Praktijkoefeningen</b>	<b>195</b>
<b>9 Modulair programmeren in JavaScript</b>	<b>199</b>
<b>Waarom modulair programmeren?</b>	<b>200</b>
<b>Kennismaken met design patterns</b>	<b>200</b>
<b>Patroon 1 – Code groeperen in object literals</b>	<b>202</b>
Oude situatie – alle functies en variabelen in global scope	203
Nieuwe situatie – een object literal	204
Voordelen en nadelen van het object literal-patroon	205
<b>Patroon 2 – Prototype pattern</b>	<b>205</b>
Een klasse Auto maken	206
Het prototype uitbreiden	207
Functies in de klasse zelf schrijven?	208
Standaardobjecten uitbreiden met eigen functies	209
Potentieel gevaar	210
Voordelen en nadelen van het prototype pattern	210
<b>Patroon 3 – Revealing module pattern</b>	<b>211</b>
Revealing modules	211
Revealing module pattern kenmerken	213
Eén instantie	214
<b>Overerving in JavaScript</b>	<b>214</b>
Prototype instellen	216
Testen	217
Override	217
<b>Conclusie</b>	<b>219</b>
<b>Verder lezen</b>	<b>219</b>
<b>Praktijkoefeningen</b>	<b>220</b>
<b>Index</b>	<b>223</b>

# Kennismaken met JavaScript

*HTML is al ruim twintig 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 jQuery, webapps gaat maken met PhoneGap 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, Twitter, 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, maar 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 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.

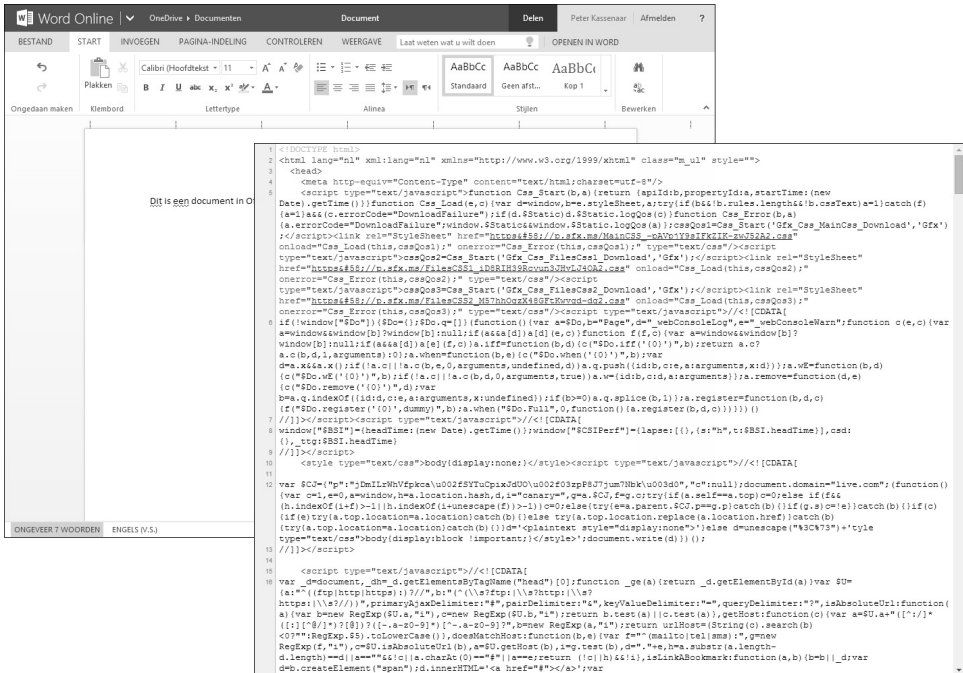
## Waarvoor wordt JavaScript gebruikt?

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 controle-

ren 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-mail-adres heeft ingevuld.

- **Dynamische menu'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 ververs of aangepast kunnen worden door gebruik te maken van XML en JavaScript. 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 XML. 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 netwerk mogelijkheden of mogelijkheden voor het werken met bestanden. 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). Kortom, de browser is zonder twijfel de bekendste, maar zeker niet de enige omgeving waarin JavaScript wordt ingezet. 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 *Handboek (X)HTML, CSS en JavaScript* van dezelfde auteur. 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

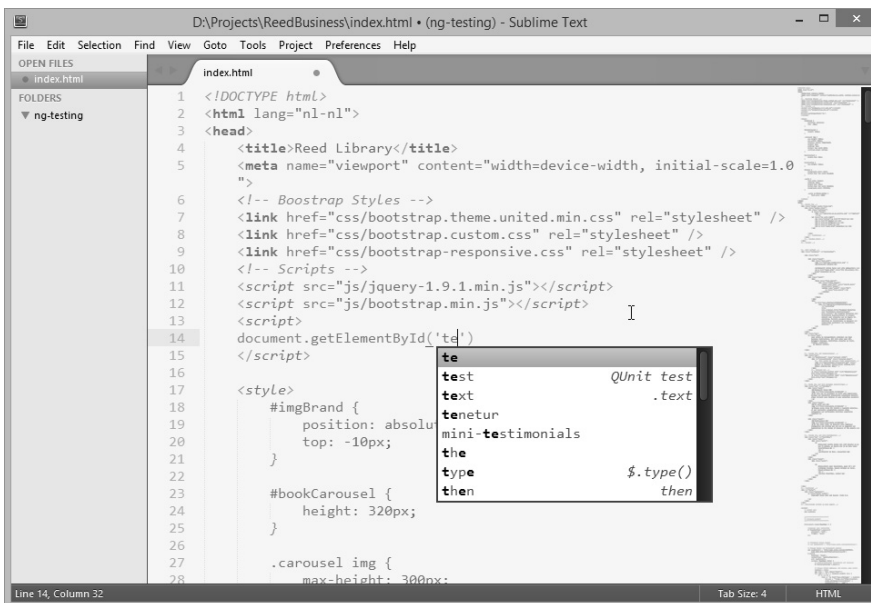
Webontwikkelaars die uit de vormgevingshoek afkomstig zijn, gebruiken graag een editor waarmee webpagina's op visuele wijze worden vormgegeven. Zij werken bijvoorbeeld in de ontwerpweergave van Adobe Dreamweaver of gebruiken aparte tools als Adobe Muse of Microsoft Expression Web. Het samenvattende begrip 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.

---

- **Adobe Dreamweaver** Dreamweaver is een tool die vooral bij Adobe-gebruikers bekend is. Adobe is onder meer de maker van Photoshop, Indesign en het PDF-bestandsformaat. In Dreamweaver kunt u in de codeweergave scripts schrijven met ondersteuning voor kleurcodering, codehints en hulp bij het oplossen van fouten in functies. Download een trialversie van Dreamweaver vanaf [www.adobe.com/go/trydreamweaver](http://www.adobe.com/go/trydreamweaver).
- **Microsoft Visual Studio** Als u meer in de Microsoft-hoek zit, ligt het gebruik van Visual Studio voor de hand. Ook dit is een totaaloplossing, voor het maken van Windows- en Windows Phone-apps, maar is ook uitstekend geschikt voor webapplicaties. Er is een gratis versie van Visual Studio verkrijgbaar op [www.visualstudio.com/en-us/products/visual-studio-express-vs.aspx](http://www.visualstudio.com/en-us/products/visual-studio-express-vs.aspx).
- **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** Erg populair de laatste tijd is de editor Sublime Text (voor Windows en Macintosh). Het is een editor die 'kaal' niet zo bijzonder is, maar door een enorm landschap van plugins en snippets is hij uit te breiden met alle mogelijke snuffjes. Vooral in de opensourcehoek is Sublime Text erg populair. U vindt Sublime Text op [www.sublimetext.com](http://www.sublimetext.com).



Afbeelding 1.3 Sublime Text is een tamelijk nieuwe webeditor.

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 Notepad++, Eclipse, NetBeans, CoffeeCup of een andere editor. Als die ook niet bevallen, dan weten wij het ook niet meer. Veel plezier met Kladblok in dat geval.