

Inhoud

1	Kennismaken met jQuery	1
	Wat is jQuery?	2
	jQuery – sinds 2006	3
	jQuery in een notendop	3
	Waarom jQuery gebruiken?	5
	Versies van jQuery	7
	Afwegingen	8
	Varianten van jQuery	9
	Indeling van dit boek	10
	Oefenbestanden downloaden	11
	Praktijk - jQuery toevoegen en gebruiken	11
	jQuery insluiten in de pagina	11
	Werken met een Content Delivery Network, CDN	13
	Enkele jQuery basisvoorbeelden	14
	HTML-code	14
	Selecties maken	15
	Het jQuery-object	15
	Chaining	17
	De jQuery API	18
	Elementen selecteren met jQuery	20
	De functie document.ready()	22
	Enkele korte voorbeelden	24
	Oefeningen	26
	Conclusie	26
	Praktijkoefeningen	27

2	API - HTML- en CSS-functies van jQuery	29
	CSS-eigenschappen lezen en schrijven	30
	De functie <code>.css()</code>	30
	Voorbeeld van <code>.css()</code>	31
	Opties meegeven als object	32
	Configuratieobject	33
	<code>.addClass()</code> en <code>.removeClass()</code>	33
	<code>.toggleClass()</code>	35
	<code>.hasClass()</code>	37
	Werken met HTML en attributen	39
	Voorbeeld-HTML	39
	<code>.html()</code>	39
	<code>.text()</code>	40
	<code>.attr()</code>	41
	Object meegeven als parameter	44
	Elementen invoegen en verwijderen uit het DOM	45
	<code>.append()</code> en <code>.prepend()</code>	45
	<code>.before()</code> en <code>.after()</code>	46
	<code>.appendTo()</code> en <code>.prependTo()</code> – Andere manieren van invoegen	47
	<code>.wrap()</code> en <code>.wrapInner()</code> – Elementen omsluiten	48
	<code>.empty()</code> en <code>.remove()</code> – Elementen verwijderen	49
	Formulievelden verwerken met jQuery	50
	<code>.val()</code>	51
	<code>.is()</code>	52
	Keuzerondjes uitlezen	53
	Selectievakjes uitlezen en de functie <code>.each()</code>	55
	Conclusie	57
	Praktijkoefeningen	58
3	jQuery-animatiefuncties	63
	Basisanimatiefuncties	64
	Inleiding	64
	Animatiesnelheid	64
	Standaardcode bij de voorbeelden	65
	<code>.hide()</code> en <code>.show()</code> – Eenvoudige animatie	66
	<code>.toggle()</code>	67
	<code>.slideDown()</code> en <code>.slideUp()</code>	68
	<code>.slideToggle()</code>	69
	Elementen infaden en uitfaden	70
	<code>.fadeIn()</code> en <code>.fadeOut()</code>	70
	<code>.fadeToggle()</code>	70
	<code>.fadeTo()</code> – Zelf transparantie instellen	71

Callbackfuncties na animatie	72
Asynchroon	73
Callbackfunctie	73
Eigen animaties maken met .animate()	75
Parameters voor .animate()	76
Configuratieobject	76
Callback na animatie	77
Reset? Zelf schrijven!	78
Wat kunnen we animeren en hoe?	79
Relatieve notaties	79
Easing gebruiken	81
Meer easingmogelijkheden	83
De plug-in van easings.net gebruiken	84
Geavanceerde animatiefuncties	85
Globale eigenschappen voor animaties	86
Case: tabbladen maken	87
Tabbladen als user interface	87
Stap 1 – de tabs maken	87
Stap 2 – de inhoud van de tabs maken	88
Stap 3 – de tabs vormgeven	88
Stap 4 – de tabs functionaliteit geven	90
Tabs faden	91
Case: een luxe tooltip	92
Stap 1 – de HTML-code	92
Stap 2 – CSS schrijven voor de tooltip	93
Stap 3 – het script schrijven	93
Stap 4 – de tooltip tonen en verbergen	94
Stap 5 – de muis volgen	95
Stap 6 – de browsertooltip verwijderen	96
Conclusie	98
Praktijkoefeningen	98
4 jQuery event handling	101
Eenvoudige event binding en -afhandeling	102
Eenvoudige events	102
.click()	103
.hover()	105
.focus() en .blur()	108
Betere event handling met .on()	110
Fragmentatie	110
Live events met .on()	111
Context selector	112

Live events in lijsten	114
.off()	116
Het jQuery event object	116
Muispositie onderzoeken	117
Het event object inspecteren	118
Conclusie	119
Browser events	120
Formulierevents	120
.focus() en .blur()	120
.select()	121
.change()	122
.submit()	123
Toetsenbordevents	124
Muisevents	125
Conclusie	126
Live event binding	126
Praktijkoefeningen	127
5 jQuery en Ajax	131
Wat is Ajax?	132
Ajax, XML en JSON	132
Ajax in de browser en op de server	133
jQuery en Ajax	133
Ajax – alleen in combinatie met een server	135
Het object XMLHttpRequest	135
HTML-documenten laden met .load()	136
Debuggen van netwerkverkeer	138
Toepassingen	139
Uitbreidingen van .load()	140
Aangegeven fragment laden	140
Gegevens meesturen	141
Callbackfunctie uitvoeren	142
JavaScript same origin policy	142
Geen foutmelding bij .load()	144
jQuery Ajax-functies	145
\$.ajax()	146
De functie .ajax()	147
Opbouw van \$.ajax()	147
Success-callback voor \$.ajax()	148
Error handling	149
Meer parameters voor \$.ajax()	152
Het object jqXHR	152
Enkele veelgebruikte parameters	153

Zijstap – Wat is JSONP?	154
Waarom JSONP?	154
dataType: 'jsonp'	156
Case – werken met openweathermap.org	157
Stap 1 – wat is openweathermap.org?	157
Stap 2 – de interface	159
Stap 3 – het script beginnen	159
Stap 4 – de Ajax-call schrijven	160
Stap 5 – de eerste versie testen	160
Stap 6 – Gegevens tonen in de UI	162
Stap 7 – gegevens aanpassen en UI uitbreiden	163
Stap 8 – foutcontrole inbouwen	165
Error handling bij traditionele JSON-calls	167
Meer API's	168
Standaardinstellingen maken met .ajaxSetup()	170
Ajax-events	171
Toepassingen van Ajax-events	172
Conclusie	173
Praktijkoefeningen	174
6 jQuery plug-ins	177
Kenmerken van plug-ins	178
Plug-ins vinden en downloaden	179
Stappenplan	179
Fork on Github?	181
Kennismaken met plug-ins: Cycle2	181
Stap 1 – De plug-in zoeken en downloaden	182
Stap 2 – De plug-in toevoegen aan de pagina	184
Stap 3 – De plug-in configureren	186
Stap 4 - Methods voor een plug-in	188
Events in de plug-in	190
Conclusie	191
Case – Form Validation	192
Stap 1 – de plug-in downloaden en klaarmaken voor gebruik	193
Stap 2 – het HTML-formulier maken	194
Stap 3 – de plug-in activeren	195
Stap 4 – de plug-in configureren	197
Stap 5 – het formulier verzenden	199
Verdere configuratie	200
Optioneel – formulier valideren met rules	201
Case – jQuery Tip Cards in combinatie met Ajax	203
Tip Cards	203
Ajax	204

Openweathermap.org	205
Stap 1 – de plug-in downloaden	205
Stap 2 – HTML schrijven	205
Stap 3 – Event handler schrijven	207
Stap 4 – De Tip Card opmaken	210
Stap 5 – de plug-in uitbreiden en configureren	211
Meer over plug-ins	212
Praktijkoefeningen	213
7 Werken met jQuery UI	217
Wat is jQuery UI?	218
Andere projecten	218
Onderdelen van jQuery UI	219
Leer de algemene werkwijze	220
Aparte plug-ins	221
jQuery UI downloaden en gebruiken	222
Downloaden	222
Toevoegen aan de pagina	224
Uw eerste widget – de datepicker gebruiken	225
De datumkiezer lokaliseren	226
De gekozen datum uitlezen	228
De component slider en werken met events	229
Configuratieobject voor widgets	229
Een slider maken	230
De slider configureren	231
Events voor de slider	232
Parameters voor events	233
Andere notatie voor event handlers	234
Werken met tabs	235
Thema's voor tabs	236
Opties voor tabs	236
Interacties maken met drag-and-drop	238
De categorie Interactions	238
Draggable	238
Opties voor draggable	240
Dropzones maken	241
De event drop afhandelen	242
Terugkeren ongedaan maken	244
De positie verbeteren	245
Conclusie	247
Werken met thema's	248
Wat is een thema?	248
ThemeRoller	248

Een kant-en-klaar thema downloaden en gebruiken	249
Downloaden	251
Een eigen thema maken	254
Conclusie	256
Conclusie	257
Web	257
Twitter	258
Verder lezen	259
Praktijkoefeningen	260
Index	265

Kennismaken met jQuery

jQuery is een aanvullende JavaScript-bibliotheek die het werken met het DOM sterk vereenvoudigd. Het is geen vervanging van JavaScript. U gebruikt nog steeds JavaScript-technieken als variabelen, lussen en statements. Maar de manier waarop u elementen in webpagina's selecteert en vervolgens bewerkt, is met jQuery stukken eenvoudiger. Door jQuery te gebruiken maakt u snel en betrouwbaar interactieve pagina's, maakt u animaties, kunt u Ajax-functionaliteit gebruiken en nog veel meer. In dit hoofdstuk maakt u eerst kennis met de basis van jQuery en enkele achtergronden. Daarna leert u meer over de verschillende toepassingsgebieden van jQuery.

In dit hoofdstuk:

Wat is jQuery en wat is het niet?

De versies van jQuery en de beschikbare uitbreidingen.

Hoe wordt jQuery toegevoegd aan een pagina?

Op welke manier maakt u selecties met jQuery?

De rol van de jQuery API.

Het belang van de functie `document.ready()`.

Wat is jQuery?

Hoe uitgebreid en hoe functioneel JavaScript ook is, de traditionele DOM API (met methodes als `getElementById()`, `appendChild()` enzovoort) is niet bepaald de meest gebruikersvriendelijke API die bestaat. De notaties zijn ingewikkeld, er bestaan grote verschillen tussen browsers onderling en er is vaak veel code nodig om een relatief eenvoudig resultaat te bereiken. Welkom jQuery!



De Web Development Library

Dit boek maakt deel uit van de Web Development Library (www.webdevelopmentlibrary.nl). In elk deel wordt een op zichzelf staande techniek besproken. Andere, gerelateerde technieken worden bekend verondersteld. In dit boek over jQuery gaan we ervan uit dat u voldoende ervaring hebt met JavaScript. Variabelen, statements, objecten en dergelijke worden niet apart besproken. Kan uw JavaScript-kennis wel een opfrisbeurt gebruiken? Lees dan eerst *Web Development Library – JavaScript* van dezelfde auteur, ISBN 9789059407589.

jQuery is een JavaScript-bibliotheek die de core JavaScript DOM-functies afschermt voor de programmeur. jQuery vervangt deze functies door varianten met een veel eenvoudigere notatie. Bovendien is jQuery veelzijdiger: een handeling die in JavaScript vaak veel regels code vergt (bijvoorbeeld een element langzaam vervagen en daarna onzichtbaar maken) is in jQuery met één statement bereikbaar. Op de achtergrond zorgt jQuery voor het juiste JavaScript in elke browser.

jQuery – sinds 2006

jQuery is in 2006 ontwikkeld door de toen 22-jarige John Resig (ejohn.org). Het is in enkele jaren tijd uitgegroeid tot de meestgebruikte JavaScript-bibliotheek op internet. Miljoenen sites gebruiken jQuery. Het jQuery-project is inmiddels een volwassen organisatie en grote bedrijven als Microsoft, Google en Facebook ondersteunen jQuery met donaties en personeel. Toch is jQuery volkomen gratis gebleven – het is *open source*. U mag de bibliotheek zonder kosten downloaden en gebruiken in uw eigen projecten. Als u wilt kunt u zelfs de broncode aanpassen en optimaliseren – voor zover dat nog mogelijk is – voor eigen gebruik.



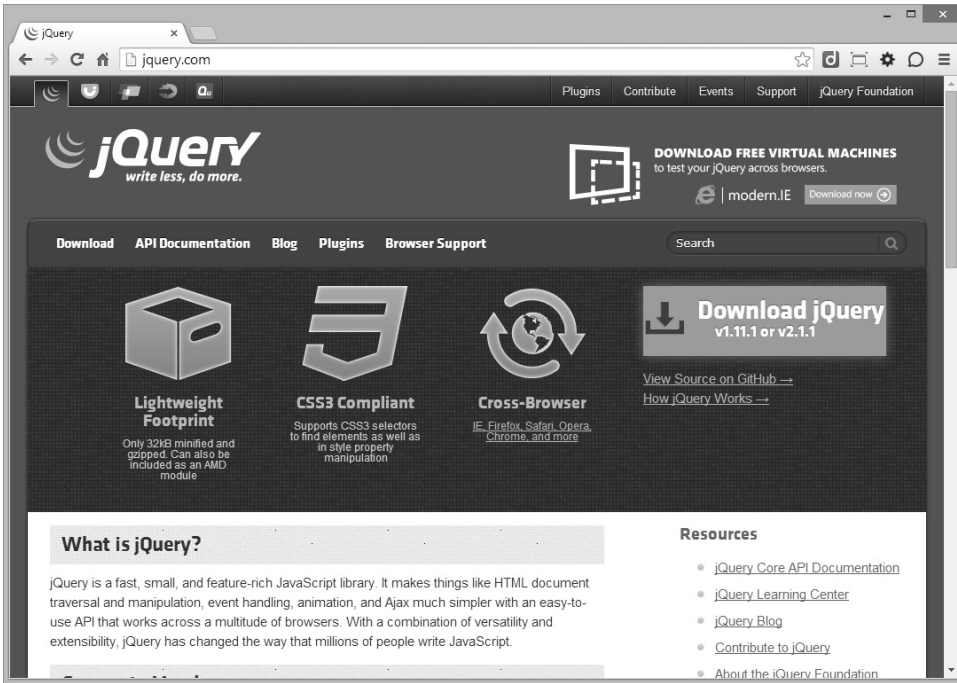
jQuery is geen vervanging van JavaScript

Beginnende programmeurs vragen wel eens of ze JavaScript moeten leren, of direct kunnen beginnen met jQuery. Welnu, jQuery *is* JavaScript. U zult nog steeds moeten weten wat geldige JavaScript-statements zijn, waar accolades en puntkomma's geplaatst moeten worden en hoe functies werken. U hoeft echter niet de JavaScript-functies voor DOM-manipulatie of Ajax te leren. Dit is veel eenvoudiger in jQuery. jQuery is hiermee een *wrapper* rondom JavaScript en heft browsersverschillen op. Maar het is geen vervanging van JavaScript.

jQuery in een notendop

Samenvattend kan het volgende worden gezegd over jQuery:

- jQuery is een open source JavaScript-bibliotheek die u gratis aan uw sites kunt toevoegen.



Afbeelding 1.1 De homepage van jQuery is te vinden op jquery.com. Hier kunt u ook de laatste versie downloaden en vindt u de complete API documentatie.

- Met jQuery worden verschillen tussen de talloze browsers afgevlakt. U hoeft maar één keer script te programmeren om alle browsers goed te bedienen.
- jQuery biedt een universele DOM API waarmee u elementen kunt aanpassen, toevoegen, verwijderen, animeren en verder manipuleren.
- jQuery gebruikt de CSS-selector syntaxis om elementen in de pagina te selecteren en daarna verder te bewerken.
- Daarnaast biedt jQuery basisfuncties voor het uitvoeren van http-aanroepen naar web servers (via het Ajax-mechanisme) en high-level functies voor het werken met arrays, events, objecten en de browser.

Waarom jQuery gebruiken?

Als u bovenstaande lijst hebt doorgenomen, is de waaromvraag waarschijnlijk al grotendeels beantwoord. Door jQuery te gebruiken, kunt u sneller programmeren en met één codebasis alle browsers bedienen. De code is beknopter en daardoor eenvoudiger te lezen en makkelijker te onderhouden.

Zonder compleet te zijn, geven we een beknopt overzicht van enkele zaken die u met jQuery eenvoudig kunt realiseren. Nogmaals, dit zou ook met alleen puur JavaScript gedaan kunnen worden (*alles* wat u met jQuery doet, kan ook in JavaScript gedaan worden), maar met jQuery is het stukken eenvoudiger. Ook de minder gevorderde programmeur kan snel goede resultaten bereiken. U kunt jQuery onder meer inzetten voor:

- Animaties van onderdelen op een pagina: in- en uitschuivende panelen, overlays en pop-ups.
- Menusystemen, tabbladen, tooltips en interactieve panelen maken.
- Formulieren snel opmaken, valideren en verzenden.
- Afbeeldingen vormgeven en slideshows maken.
- Sneller reageren op events: browser-, formulier-, toetsenbord- en muis-events.
- Gegevens naar de webserver versturen en gegevens ophalen zonder de pagina te verlaten (via Ajax).
- Userinterfaces verrijken met onderdelen als kalenders, kleurenkiezers en drag-and-dropfuncties (via plug-ins en de aanvullende bibliotheek jQuery UI).
- Geoptimaliseerde mobiele websites maken met de aanvullende bibliotheek jQuery Mobile.

Werkzaamheden

- Schrijven
-  Yindo
- Training
- Usability
- Programmeren

Schrijven

Ik ben sinds 1989 actief als schrijver van ICT-artikelen en later ook boeken. Eerst voor tijdschriften (ComputerTotaal, HCC Nieuwsbrief, wie kent 'm nog?), later voor uitgeverijen als [Van Duuren Media](#), [Instruct](#) en [SDU/Academic Service](#). Daarnaast deel ik mijn kennis uiteraard via dit blog en schrijf ik op verzoek columns voor nieuwsbrieven en andere sites.

Ik ben specialist op het gebied van (het programmeren voor) internet, Internet ontwikkelingst en website usability. [Mijn profiel op LinkedIn](#).

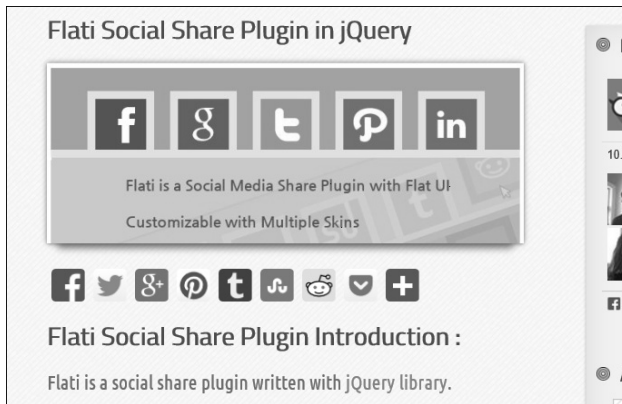
Recente boeken

Afbeelding 1.2 De auteur gebruikt jQuery op zijn site om een eenvoudig tabbladensysteem te maken voor diverse werkzaamheden.



Kan jQuery alles?

jQuery heeft tientallen functies om DOM-elementen te manipuleren en websites te bewerken. Maar uiteraard is het onmogelijk alle functionaliteit die iedereen ooit zou wensen in één library te plaatsen. Daarom hebben de makers van jQuery gezorgd voor een uitbreidbaar model. Er kan functionaliteit aan worden toegevoegd via een systeem van plug-ins. In een plug-in wordt nieuwe functionaliteit aangeboden die één specifieke taak vervult. Denk bijvoorbeeld aan het maken van een diashow op de pagina, het vormgeven van een menusysteem of het bieden van sorteerfunctionaliteit voor tabellen. jQuery kan zodoende niet alles, maar het aantal mogelijkheden is vrijwel onbeperkt.

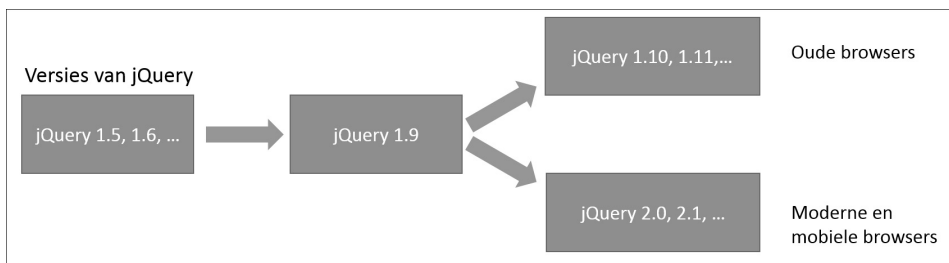


Afbeelding 1.3 Een voorbeeld van een plug-in. Met de Flati Social Share Plugin kunt u snel knoppen op een webpagina plaatsen om de inhoud te delen op sociale netwerken.

Versies van jQuery

Sinds de eerste release van jQuery in 2006 zijn er gestaag updates verschenen. De versienummers lopen zoals gewoonlijk via 1.1, 1.2 enzovoort omhoog. Begin 2013 is echter een belangrijke beslissing genomen door het jQuery-team. Er is een splitsing gemaakt in een 1.xx-branch en een 2.xx-branch.

- jQuery-versies die beginnen met 1.x zijn geschikt voor *alle browsers*, inclusief Internet Explorer 6, 7 en 8 en Firefox 4.
- jQuery-versies die beginnen met 2.x ondersteunen *alleen moderne browsers*, beginnend vanaf Internet Explorer 9. Oudere browsers worden niet meer ondersteund.



Afbeelding 1.4 Van jQuery bestaat op dit moment een 1.x-branch en een 2.x-branch.

Afwegingen

Dit betekent dat u als programmeur een goede afweging moet maken welke versie van jQuery u gaat gebruiken.

- Voor algemene websites, bestemd voor het grote publiek en alle typen computers en browsers, gebruikt u het liefst een jQuery-versie uit de 1.x-reeks, bijvoorbeeld jQuery 1.11.1 (de meest recente versie op het moment van schrijven van dit boek).
- Als u zeker weet dat jQuery alleen in een moderne omgeving wordt gebruikt (bijvoorbeeld in een app op een mobiele telefoon, of in een intranet met alleen moderne browsers), dan kunt u jQuery 2.x gebruiken.
 - Het voordeel van jQuery 2.x is dat de code compacter en sneller is, omdat geen overhead nodig is voor het ondersteunen van oudere browsers. Ook is de code modulair opgebouwd, zodat u desgewenst een eigen jQuery-versie kunt samenstellen waarin alleen de modules zijn opgenomen die u nodig hebt.

Het jQuery-team heeft toegezegd beide takken te blijven onderhouden en verder te ontwikkelen. Dat betekent ook dat na jQuery 1.11 (nu) in de toekomst waarschijnlijk ook versies als jQuery 1.12 en jQuery 1.13 zullen verschijnen. Onafhankelijk hiervan zullen updates in de 2.x-serie gaan verschijnen, als jQuery 2.1, 2.2 en zo verder.