

jQuery Core en jQuery UI

Peter Kassenaar
info@kassenaar.com

Introductie

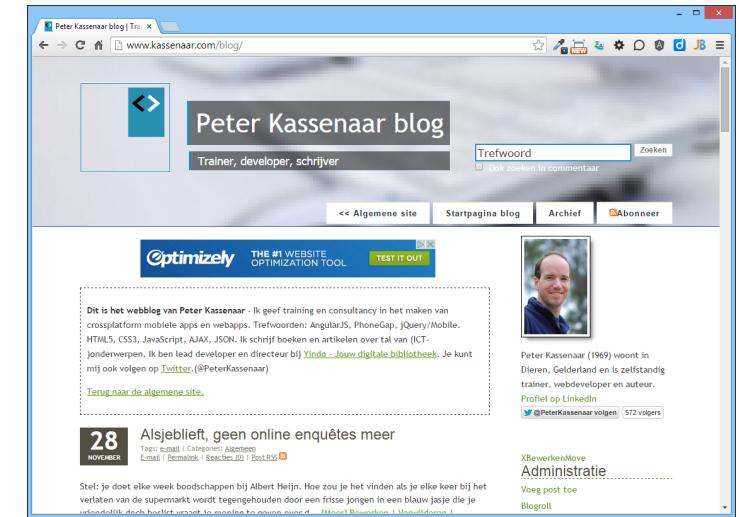
Peter Kassenaar;

- Auteur, programmeur, docent
- Eigen bedrijf (1996)

www.kassenaar.com/blog

info@kassenaar.com

Twitter: [@PeterKassenaar](https://twitter.com/PeterKassenaar)



Over jullie...



Voorkennis Javascript?

Voorkennis andere (web)talen?

Verwachtingen van de cursus?

Concrete projecten?

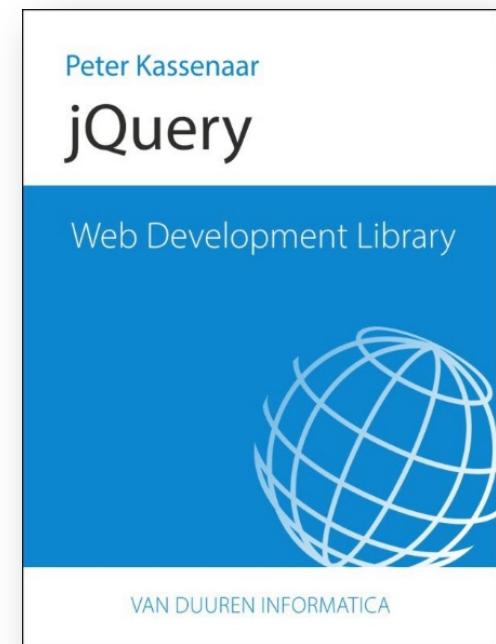
Materialen

Handouts

Oefeningen

Boek

Websites



Indeling

Basisbegrippen

Het Document Object Model (DOM)

Functies en de jQuery API

jQuery toepassen

jQuery UI – achtergronden en toepassingen

Dag 1: Geschiedenis, Selectors, basisfuncties

Dag 2: Meer API-functies, plug-ins en AJAX

Dag 3: jQuery UI, widgets toepassen

Vragen?

De plaats van JavaScript

De
gebruikersinterface
(User Interface) wordt
samengesteld op
basis van:

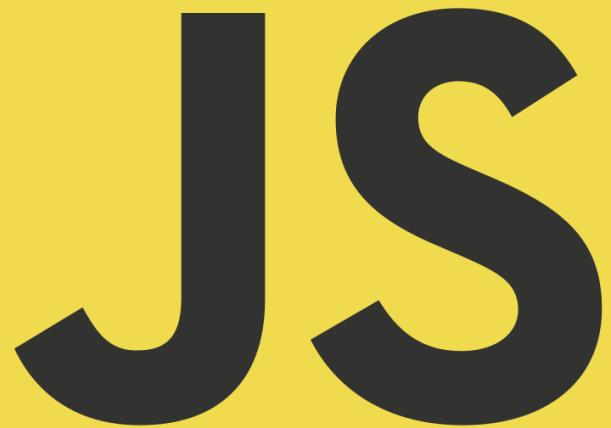


Javascript – the ugly Stepchild

- Javascript's been around forever but...
 - Minimally used for its first 10 years
 - Few app developers were **really** proficient with Javascript
- Browser Incompatibilities
 - Varying Dom/CSS implementations
 - Varying event management interfaces
 - Varying feature support

Conclusie

JavaScript is OK...



A large, bold, black "JS" logo centered on a solid yellow rectangular background.

...maar de browsers zijn de boosdoeners!



Door: eigen interpretaties van de standaard en
browserspecifieke uitbreidingen



“jQuery makes JavaScript suck less...”

“jQuery is what the HTML DOM should have been!”

Uitgangspunten

- Enige Javascript voorkennis bekend
verondersteld
 - de browser is de *container* waarbinnen het
jQuery/JavaScript wordt uitgevoerd





The screenshot shows the official jQuery website homepage. At the top, there's a navigation bar with links for "Plugins", "Contribute", "Events", "Support", and "jQuery Foundation". Below the navigation, the main content area features the jQuery logo ("write less, do more.") and information about the "jQuery conference PORTLAND | JUNE 13 - 14 Oregon Convention Center Members - \$50 Off".

The central part of the page highlights three key features: "Lightweight Footprint" (represented by a 3D cube icon), "CSS3 Compliant" (represented by a stylized '3' icon), and "Cross-Browser" (represented by a globe icon). To the right, there's a prominent orange button with the text "Download jQuery v1.10.0 or v2.0.1".

On the left side, there's a sidebar with the heading "What is jQuery?" and a brief description of the library's purpose. On the right side, there's a "Resources" section containing links to various parts of the jQuery ecosystem, such as the Core API Documentation, Learning Center, Blog, and Bug tracker. A "TRY jQuery" button is located at the bottom right.

Waarom jQuery?

- jQuery offers a *common Browser API*
 - Wraps the HTML DOM in a high level API
 - Provides browser independence
 - Greatly simplifies many common tasks intuitively
- By using jQuery:
 - Reduce amount of Javascript code drastically
 - Produce more robust Javascript code
 - Produce unobtrusive Javascript

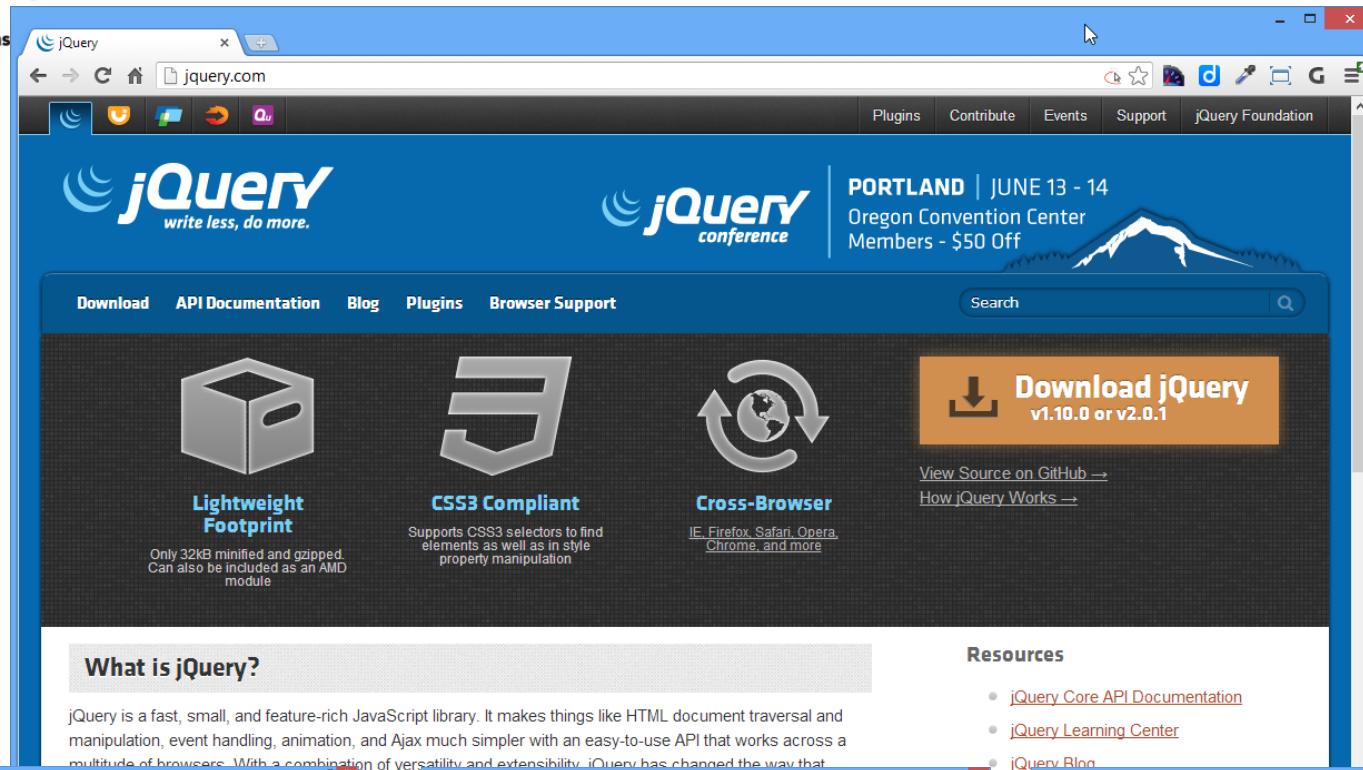


-

“jQuery puts the fun back into client scripting”

jQuery varianten

- jQuery core, <http://jquery.com/>
 - Core functionality selecting, manipulation, Ajax
- jQuery UI, <http://jqueryui.com/>
 - Rich UI-effects, widgets
- jQuery mobile, <http://jquerymobile.com/>
 - jQuery for mobile devices
- jQuery Plugins, <http://plugins.jquery.com/>
 - Extending the power of jQuery



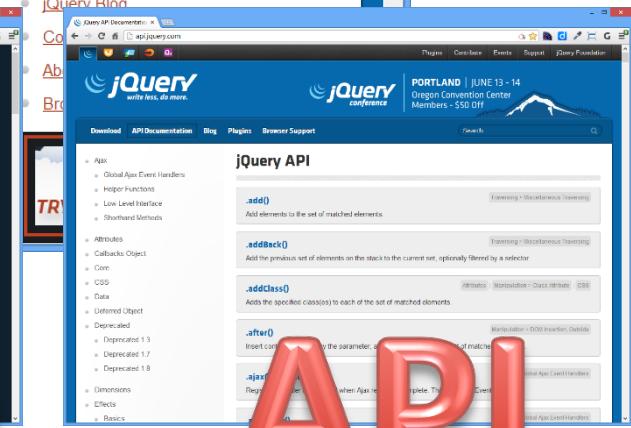
The screenshot shows the official jQuery website at jquery.com. The header includes the jQuery logo and the tagline "write less, do more." A banner for the "jQuery conference PORTLAND | JUNE 13 - 14" is displayed, along with a "Members - \$50 Off" discount. The main content area features several key features of jQuery: "Lightweight Footprint" (represented by a small cube icon), "CSS3 Compliant" (represented by a stylized '3 icon), and "Cross-Browser" (represented by a globe icon). A prominent orange button allows users to "Download jQuery v1.10.0 or v2.0.1". Below the main features, there are links to "View Source on GitHub" and "How jQuery Works". The footer contains a "Resources" section with links to the Core API Documentation, Learning Center, and Blog.



The screenshot shows the official jQuery UI website at jqueryui.com. The header includes the jQuery UI logo and the tagline "user interface". A banner for the "jQuery conference PORTLAND | JUNE 13 - 14" is displayed, along with a "Members - \$50 Off" discount. The main content area features sections for "Interactions" (Draggable, Resizable, Selectable, Sortable) and "Widgets" (Accordion, Autocomplete, Button, Datepicker, Dialog, Menu, Progressbar, Slider, Spinner). A "What's New in jQuery UI 1.10?" section details changes in version 1.10, including bug fixes and improved accessibility. A "Developer Links" section provides links to Source Code (GitHub), API Documentation, and Bug Tracker.



The screenshot shows the official jQuery Mobile website at jquerymobile.com. The header includes the jQuery Mobile logo and the tagline "write less, do more.". A banner for the "jQuery conference PORTLAND | JUNE 13 - 14" is displayed, along with a "Members - \$50 Off" discount. The main content area features sections for "jQuery MOBILE | RELEASE" (describing it as a touch-optimized web framework for smartphones and tablets) and "Seriously cross-platform with HTML5" (explaining how it uses HTML5 to provide a native-like experience across multiple platforms). A "Developer Links" section provides links to Source Code (GitHub), API Documentation, and Bug Tracker.



The screenshot shows the official jQuery API documentation website at api.jquery.com. The header includes the jQuery logo and the tagline "write less, do more.". A banner for the "jQuery conference PORTLAND | JUNE 13 - 14" is displayed, along with a "Members - \$50 Off" discount. The main content area features a "jQuery API" section listing various methods such as .add(), .addBack(), .after(), .ajax(), .attr(), .before(), .clone(), .css(), .data(), .die(), .empty(), .eq(), .filter(), .first(), .get(), .getJSON(), .has(), .html(), .is(), .length(), .load(), .map(), .not(), .one(), .on(), .parent(), .prev(), .remove(), .size(), .text(), .trigger(), .wrap(), and .wrapAll(). Each method is accompanied by a brief description and examples.

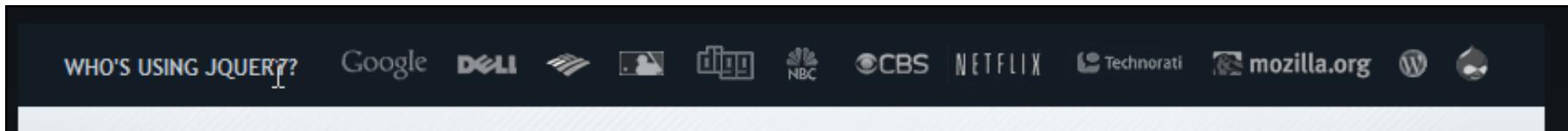
UI Mobile API

What is jQuery?

- *Javascript HTML DOM Manipulation library*
- Light weight (~80k compressed)
- Powerful and extremely practical functionality
- Easy to learn and intuitive to use
- *CSS 3 based selector syntax*
 - ID's, Classes, Elements

What is jQuery?

- Easy to extend through plug-in API
- 100's of plug-ins for just about anything
- Popular and getting more popular all the time



Toepassingen van jQuery

- jQuery gebruiken voor:
 - browserverschillen opheffen
 - animaties maken
 - Ajax-functies toepassen
 - events opvangen
 - elementen manipuleren
 - stijlen aanpassen
 - interactieve webapplicaties
 - Verplaatsing van logica van server → client

Praktijk: jQuery gebruiken

Add script reference to HTML:

```
<script src="scripts/jquery.js"  
       type="text/javascript"></script>
```

Select elements and apply behavior:

```
<script type="text/javascript">  
  
    $( "#mijnDiv" )                      // select  
        .addClass( "myLayout" )            // apply layout  
        .click( function() {} ) ;         // attach behavior  
  
</script>
```

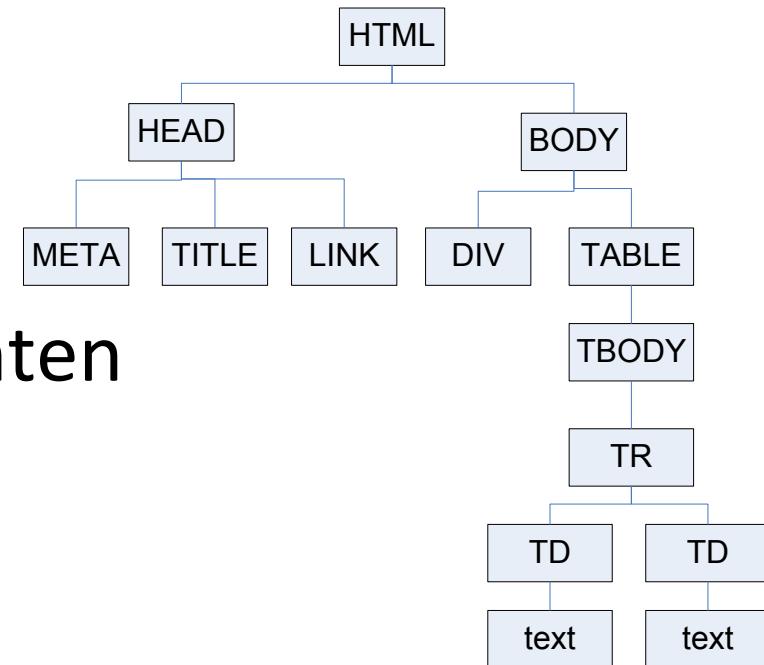
jQuery statements ontleden

selector	action	parameters
jQuery('p')	.css	('color', 'blue');
\$('p')	.css	('color', 'blue');

- *selector* is altijd `jQuery`, of kortweg `$`
- *Action* is één van de opdrachten uit de jQuery API
- *parameters* is een lijst van 0, 1, of meer opdrachten
 - kommagescheiden notatie
 - JSON-notatie

Het DOM

- Wat selecteer je?
 - Onderdelen uit het DOM!
- Gebruik de CSS-syntaxis voor selectie van elementen
 - elementen
 - classes
 - id's



Make sure the Page is ready

- All elements needs to be loaded before we can manipulate them
- Always use the document-ready function

```
$ (document) .ready (function () {  
    alert ('Document is ready...');  
    // all (well, most) other functions, hooks, etc.  
});
```

jQuery toepassing

- altijd starten met hetzelfde basisprincipe:

```
$('document').ready(function() {  
    // hierbinnen alle eigen javascript code plaatsen  
});
```

- Dus bijvoorbeeld in je HTML:

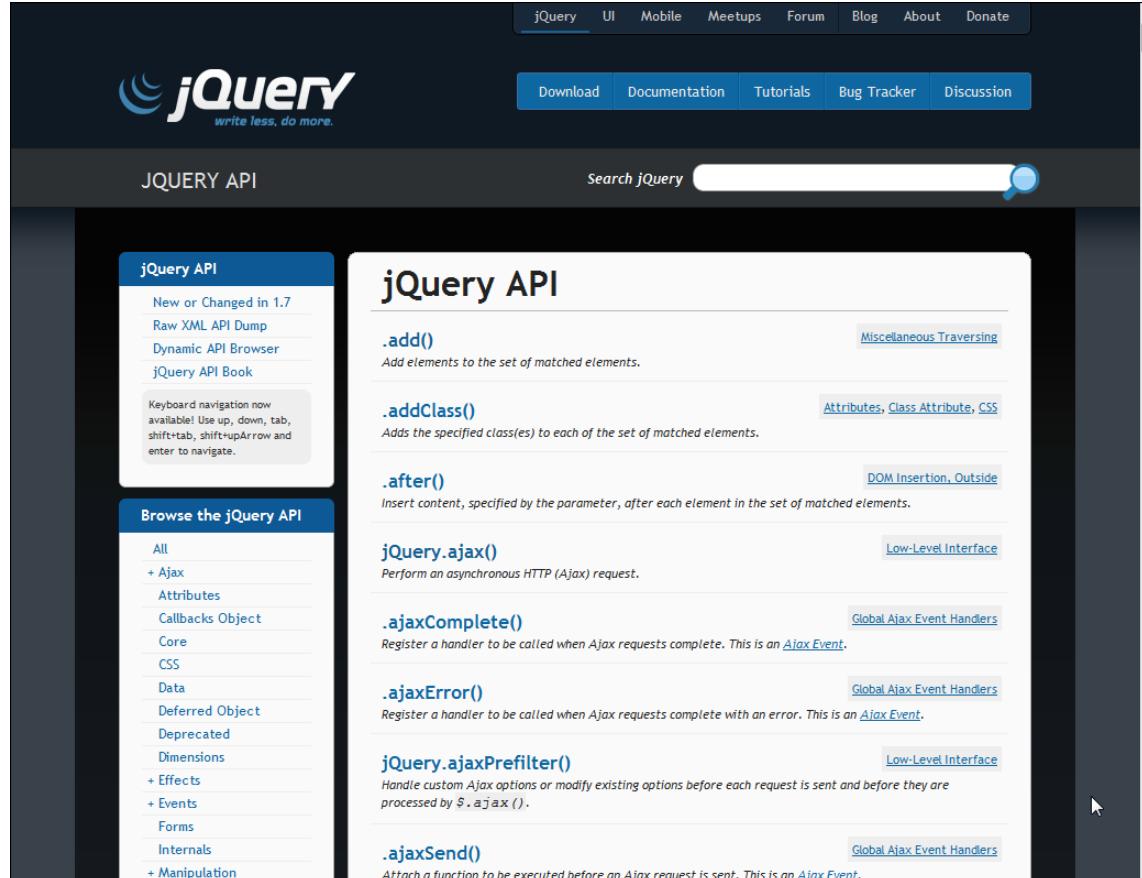
```
--  
23 <body>  
24 <h2>Dit is een koptekst</h2>  
25 <p>Dit is de eerste paragraaf</p>  
26 <p>Dit is de tweede paragraaf</p>  
27 <p>Dit is de derde paragraaf</p>  
28 <button id="btnHide">Laat de paragrafen verdwijnen!</button>  
29 </body>
```

- En je jQuery-JavaScript

```
13 <script>  
14 $(document).ready(function(e) {  
15     $('#btnHide').on('click',function(){  
16         $('p').fadeOut('slow');  
17     });  
18 });  
19  
20 </script>
```

de jQuery API

- Alle mogelijkheden voor `.action()` worden beschreven in de *jQuery API*
- Leer/gebruik deze!



The screenshot shows the official jQuery API documentation page. At the top, there's a navigation bar with links for "jQuery", "UI", "Mobile", "Meetups", "Forum", "Blog", "About", and "Donate". Below that is another row with "Download", "Documentation", "Tutorials", "Bug Tracker", and "Discussion". The main content area has a search bar labeled "Search jQuery" with a magnifying glass icon. On the left, a sidebar titled "jQuery API" lists "New or Changed in 1.7", "Raw XML API Dump", "Dynamic API Browser", and "jQuery API Book". It also includes a note about keyboard navigation and a "Browse the jQuery API" section with a list of categories: All, Ajax, Attributes, Callbacks Object, Core, CSS, Data, Deferred Object, Deprecated, Dimensions, Effects, Events, Forms, Internals, and Manipulation. The main content area is titled "jQuery API" and lists several methods with their descriptions and category links:

- .add()** (Miscellaneous Traversing)
- .addClass()** (Attributes, Class Attribute, CSS)
- .after()** (DOM Insertion, Outside)
- jQuery.ajax()** (Low-Level Interface)
- .ajaxComplete()** (Global Ajax Event Handlers)
- .ajaxError()** (Global Ajax Event Handlers)
- jQuery.ajaxPrefilter()** (Low-Level Interface)
- .ajaxSend()** (Global Ajax Event Handlers)

Tools

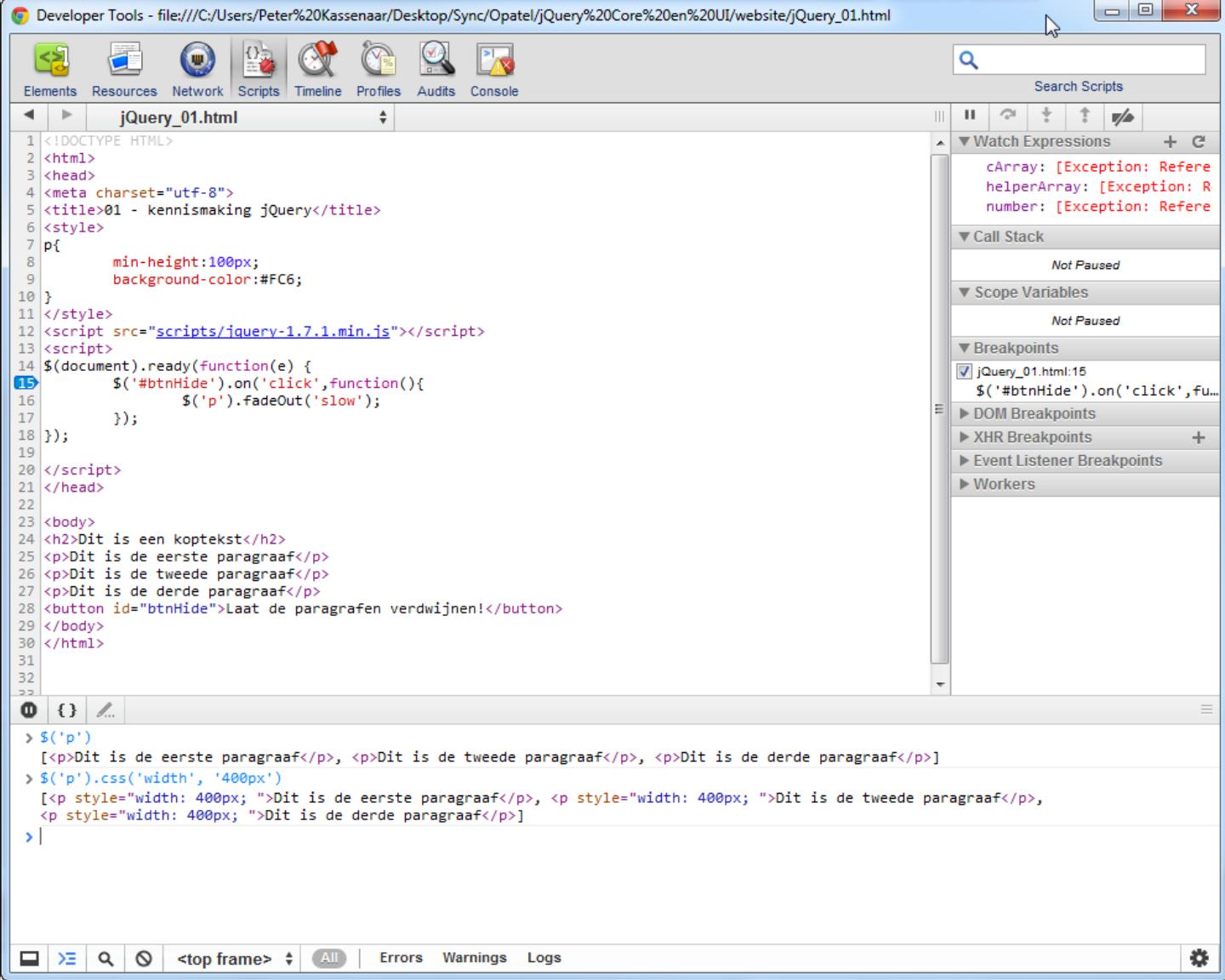


Extreem belangrijk: debugging

- Chrome Developer Tools
- Mozilla Firefox – Firebug



Developer Tools - file:///C:/Users/Peter%20Kassenaar/Desktop/Sync/Opatel/jQuery%20Core%20en%20UI/website/jQuery_01.html



The screenshot shows the Google Chrome Developer Tools interface. The left pane displays the HTML code for 'jQuery_01.html'. The code includes a CSS rule for a paragraph and a JavaScript function that hides paragraphs when a button is clicked. A breakpoint is set on the line where the fadeOut method is called. The right pane shows the 'Scripts' panel with sections for Watch Expressions, Call Stack, Scope Variables, Breakpoints, DOM Breakpoints, XHR Breakpoints, Event Listener Breakpoints, and Workers. The 'Breakpoints' section has a checked entry for the current line. The bottom pane shows the 'Console' output, which includes the result of running a CSS width rule on the paragraphs.

```
<!DOCTYPE HTML>
<html>
<head>
<meta charset="utf-8">
<title>01 - kennismaking jQuery</title>
<style>
p{
    min-height:100px;
    background-color:#FC6;
}
</style>
<script src="scripts/jquery-1.7.1.min.js"></script>
<script>
$(document).ready(function(e) {
    $('#btnHide').on('click',function(){
        $('p').fadeOut('slow');
    });
});
</script>
</head>
<body>
<h2>Dit is een koptekst</h2>
<p>Dit is de eerste paragraaf</p>
<p>Dit is de tweede paragraaf</p>
<p>Dit is de derde paragraaf</p>
<button id="btnHide">Laat de paragrafen verdwijnen!</button>
</body>
</html>
```

\$('p')
[<p>Dit is de eerste paragraaf</p>, <p>Dit is de tweede paragraaf</p>, <p>Dit is de derde paragraaf</p>]
\$('p').css('width', '400px')
[<p style="width: 400px; ">Dit is de eerste paragraaf</p>, <p style="width: 400px; ">Dit is de tweede paragraaf</p>, <p style="width: 400px; ">Dit is de derde paragraaf</p>]

Debugging techieken

- Breekpunten instellen
- Elementen inspecteren
- Waarden aanpassen



Verdere selectie

“Beheersing van selector-opties is de sleutel tot jQuery-succes”

Filters:

:even

:odd

:checked

:input

...

Selectie van elementen

HTML-code	JQuery-selector	Omschrijving
<div id="telefoonLijst">	\$('#telefoonLijst')	Selecteer het element 'telefoonLijst' in het document.
<p class="error">	\$('.error')	Selecteer alle elementen in het document met de CSS-klasse 'error'
<ul id="menu"> 	\$('#menu li a')	Selecteer alle hyperlinks binnen lijstelementen van het element #menu in het document

Complexe selectie

- Selectie op attributen:
 - `$ (" [href='#'] ")` → alle tags met een href-attribuut dat gelijk is aan #
 - `$ (" [src$='.jpg'] ")` → alle tags met een src-attribuut dat eindigt op .jpg
- Of op positie in een lijst:
 - `$ ("ul li:first")` → alle eerste li-tags in elke lijst
 - `$ ("tr td:last")` → elke laatste cel in tabelrij
 - `$ ("tr td:eq(2)")` → elke tweede cel binnen tabelrij
 - `$ ("tr:even")` → elke even tabelrij
 - `$ ("ul li:odd")` → elke oneven li-tag binnen een ul

- Of op type formulierelement:
 - `$(":text")` → alle input elementen van het type tekst
 - `$(":password")` → alle input elementen van het type password
 - `($('selected')` → alle elementen die geselecteerd zijn
 - `($('checked')` → alle selectievakjes die geselecteerd zijn

Naslag: cheat sheets

- zoek op trefwoord ‘jQuery Cheat Sheet’

JQUERY 1.5

VISUAL CHEAT SHEET

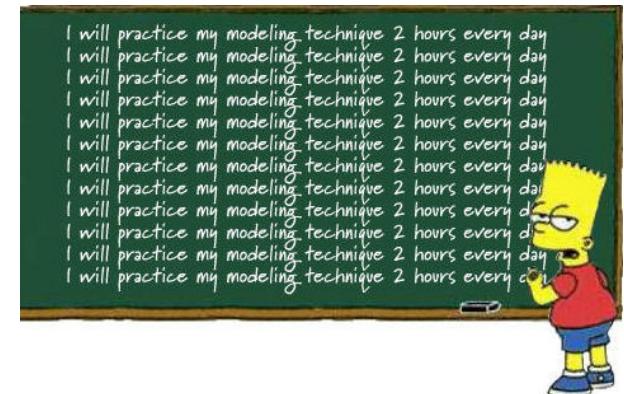
● ○ ○ ★ = NEW IN JQUERY 1.5 / f(x) = FUNCTION / a = ARRAY / jQ = JQUERY / El = ELEMENT / 0-1 = BOOLEAN / Obj = OBJECT / Num = NUMBER / Str = STRING

SELECTORS * CORE * ATTRIBUTES * CSS * TRAVERSING * MANIPULATION * EVENTS * EFFECTS * AJAX * UTILITIES * DEFERRED OBJECT
 Designed by Antonio Lupetti © 2011 • <http://woorkup.com> • <http://twitter.com/w0ork> | jQuery is © of John Resig and the jQuery Team.

Selectors / 1. Basic <ul style="list-style-type: none"> All Selector ("") El(s) Selects all elements. Class Selector ("class") El(s) Matches all elements with the given name. Element Selector ("element") El(s) Selects all elements with the given tag name. ID Selector ("#id") El Selects a single element with the given id attribute. Multiple Selector ("selector1, selector2, selectorN") El(s) Selects the combined results of all the specified selectors.	Selectors / 2. Hierarchy <ul style="list-style-type: none"> Child Selector ("parent > child") El(s) Selects all direct child elements specified by "child" of elements specified by "parent". Descendant Selector ("ancestor descendant") El(s) Selects all elements that are descendants of a given ancestor. Next Adjacent Selector ("prev + next") El(s) Selects all next elements matching "next" that are immediately preceded by a sibling "prev". Next Siblings Selector ("prev ~ siblings") El(s) Selects all sibling elements that follow after the "prev" element, have the same parent, and match the filtering "siblings" selector. 	Selectors / 3. Basic Filter <ul style="list-style-type: none"> animated Selector El(s)
Selectors / 4. Content Filter <ul style="list-style-type: none"> :contains() Selector El(s) Selects all elements that contain the specified text. :empty Selector El(s) Selects all elements that have no children (including text nodes). :has() Selector El(s) Selects elements which contain at least one element that matches the specified selector. :parent Selector El(s) Selects all elements that are the parent of another element, including text nodes. 	Selectors / 5. Attribute <ul style="list-style-type: none"> [name]=value El(s) Selects elements that have the specified attribute with a value either equal to a given string or starting with that string followed by a hyphen (-). [name*=value] El(s) Selects elements that have the specified attribute with a value containing the a 	Selectors / 6. Child Filter <ul style="list-style-type: none"> :first-child Selector El(s) Selects all elements that are the first child of their parent. :last-child Selector El(s) Selects all elements that are the last child of their parent. :nth-child Selector El(s) Selects all elements that are the nth-child of their parent. :only-child Selector El(s) Selects all elements that are the only child of their parent.
Selectors / 7. Visibility Filter <ul style="list-style-type: none"> :hidden Selector El(s) Selects all elements that are hidden. :visible Selector El(s) Selects all elements that are visible. 	Selectors / 8. Form <ul style="list-style-type: none"> :button Selector El(s) 	Core / 1. The jQuery Function <ul style="list-style-type: none"> jQuery() jq Accepts a string containing a CSS selector which is then used to match a set of elements. jQuery.sub() ★ jq Creates a new copy of jq whose properties and methods can be modified without affecting the original jQuery object. jQuery.when() ★ Deferred Provides a way to execute callback functions based on one or more objects, usually Deferred objects that represent asynchronous events. jQuery.noConflict() Obj Relinquish jQuery's control of the \$
Core / 2. DOM <ul style="list-style-type: none"> .get() [index] El a Retrieves the DOM elements matched by the jQuery object. .index() Num Search for a given element from among the matched elements. .length Num The number of elements in the jQuery object. .selector Str A selector representing selector originally passed to jQuery(). .size() Num Return the number of DOM elements matched by the jQuery object. .toArray() a Retrieve all the DOM elements contained in the jQuery set, as an array. 	Core / 3. Data <ul style="list-style-type: none"> .queue() [queueName], newQueue jq Show the queue of functions to be executed on the matched elements. .data(obj) jq Store arbitrary data associated with the matched elements. .removeData([name]) jq Remove a previously-stored piece of data. .dequeue([queueName]) jq Execute the next function on the queue for the matched elements. 	Core / 4. Interoperability <ul style="list-style-type: none"> jQuery.fn.extend(object) jq Extends the jQuery element set to provide new methods (used to make a typical jQuery plugin). jQuery.extend(object) jq

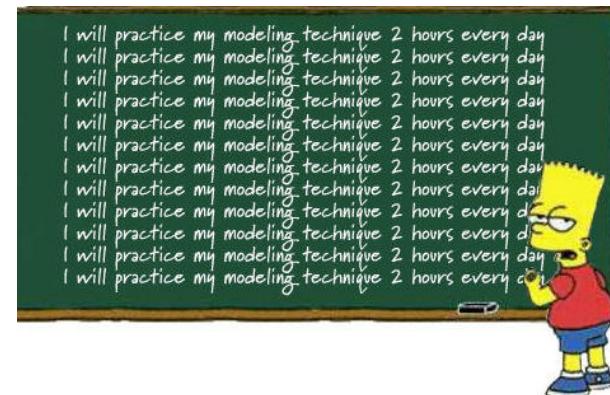
Oefening

- Maak een template/kapstokpagina waarin jQuery geladen wordt.
- Gebruik: `$(document).ready()` en toon een `alert()` zodra DOM gereed is



Oefening

- Maak kennis met Chrome Developer Tools of Firebug.
- Zet breekpunten in de code en ga er stapsgewijs doorheen
- Selecteer elementen via de console en pas rechtstreeks eigenschappen aan.



jQuery: na de selectie, de actie

- Vaak als anonymous functions:

```
$(document).ready(function() {  
    $('button').click(function() {  
        $('p').fadeOut('slow');  
    });  
});
```

- Voorbeelden van event-binding:

- `$(‘selector’).on(‘click’, function() {...});`
- `$(‘selector’).on(‘focus’, function() {...});`
- `$(‘selector’).on(‘keypress’, function() {...});`

Vroeger: (< jQuery 1.7)

- Elke event apart benoemd
 - \$(selector).click(function() { ... }) ;
 - \$(selector).focus(function() { ... }) ;
 - \$(selector).mouseover (function()
{ ... }) ;
- Tegenwoordig: gebruik de algemene event
handler .on()
- <http://api.jquery.com/on/>

jQuery API

New or Changed in 1.7
Raw XML API Dump
Dynamic API Browser
jQuery API Book

Browse the jQuery API

All
+ Ajax
Attributes
Callbacks Object
Core
CSS
Data
Deferred Object
Deprecated
Dimensions

.on()

Categories: [Events](#) > [Event Handler Attachment](#)

.on(events [, selector] [, data] , handler(eventObject))

Returns: [jQuery](#)

Description: Attach an event handler function for one or more events to the selected elements.

.on(events [, selector] [, data], handler(eventObject)) version added: [1.7](#)

events One or more space-separated event types and optional namespaces, such as "click" or "keydown.myPlugin".

selector A selector string to filter the descendants of the selected elements that trigger the event. If the selector is null or omitted, the event is always triggered when it reaches the selected element.

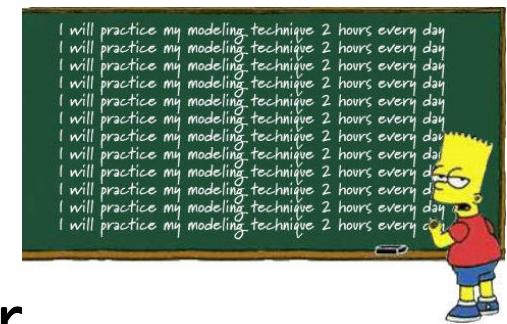
data Data to be passed to the handler in [event.data](#) when an event is triggered.

Ook beschikbaar:

- `.off()` – verwijdert een event handler
- `.one()` – voert event eenmalig uit en verwijdert daarna zichzelf

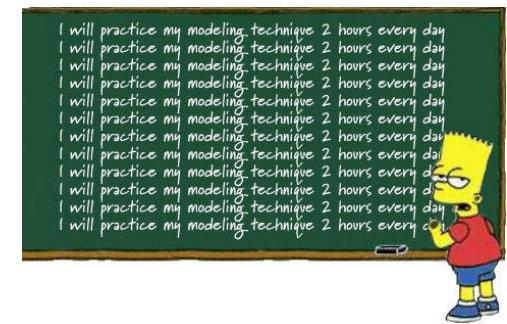
Oefening

- Maak een script waarbij na een klik op de knop alle links in de pagina een andere achtergrondkleur krijgen.
- Maak een tabel. Kleur de tabel rijen om-en-om (even/oneven)
- Geef de eerste rij een andere kleur.
- Maak een eventhandler .on () voor knop. Verwijder deze met .off (). Test dit.



Oefening

- Maak een script waarbij na afloop van een animatie dit in een paragraaf wordt getoond.
 - Tip: gebruik een callback-functie



Voorbeeld: jQuery effecten

- `.show()`
 - argumenten: slow, fast, ms
 - callback (functie die wordt uitgevoerd na afloop van het effect)
- `.hide()`
 - argumenten: slow, fast, ms
 - callback (functie die wordt uitgevoerd na afloop van het effect)

```
18 $(document).ready(function(e) {  
19     // 'Algemene voorwaarden' tonen  
20     $('#btnVoorwaarden').on('click', function() {  
21         $('.voorwaarden').show('fast');  
22     });  
23     // 'Algemene voorwaarden' verbergen  
24     $('#btnVerbergVoorwaarden').on('click', function() {  
25         $('.voorwaarden').hide('fast');  
26     });  
27 });  
28 );
```

Oefening: bundel het tonen en verbergen van de voorwaarden in één functie.



Toon voorwaarden Verberg voorwaarden

Algemene voorwaarden

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popul

Optioneel: callback-functie

- wordt uitgevoerd na afloop van de animatie/hoofdfunctie

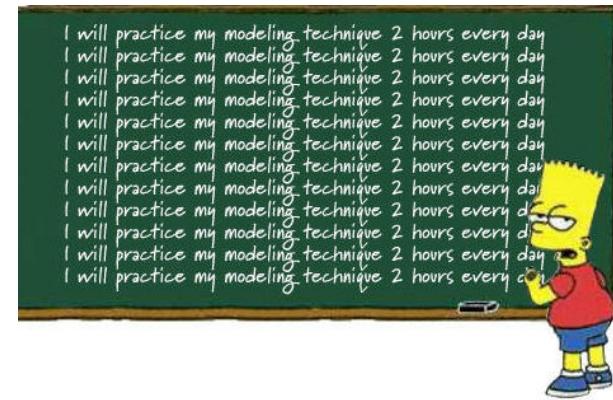
```
18 $(document).ready(function(e) {
19     // 'Algemene voorwaarden' tonen
20     $('#btnVoorwaarden').on('click', function(){
21         $('.voorwaarden').show(
22             'fast',      // parameter met de snelheid
23             function(){ // callback-functie die wordt uitgevoerd na afloop v/d animatie
24                 alert('voorwaarden zijn zichtbaar');
25             }
26         );
27     });
});
```

Meer API-animatiefuncties:

- `.fadeTo()` (een bepaalde transparantie)
- `.toggle()`
- `.slideToggle()`
- `.slideDown()`
- `.slideUp()`

Oefening

- Combineer deze animatiefuncties met diverse events die je nu kent
- Gebruik:
 - de jQuery-notatie `.on()`
 - de API-documentatie
 - Standaard JavaScript-technieken als `if...else`, `for()` en variabelen



De DOM-structuur beheren

- `.text()` en `.html()`
 - `.text('...')` stuurt kale tekst naar een element
 - `.html('...')` stuurt html-geformatteerde tekst naar element
- De opdracht zonder parameters, kent de inhoud van element toe aan variabele
 - `var mijnVar = $('#mijnDiv').html();`

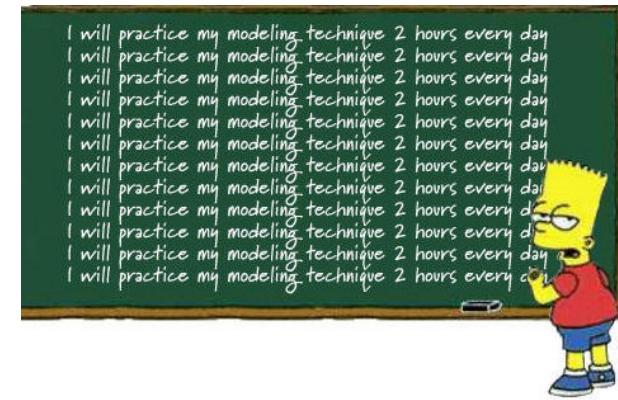
- .append ('...') → inhoud achteraan toevoegen
- .prepend ('...') → inhoud vooraan toevoegen

jQuery: CSS

- `.css (property-name)`
 - retourneert de waarde van de css-eigenschap
- `.css (property, value)`
 - zet de css-eigenschap op een bepaalde waarde
- `.css ({ 'prop1': 'val1', 'prop2': 'value2' })`
 - zet een reeks css-eigenschappen
 - let op objectnotatie!
- `.height () , .width ()`
 - zet hoogte en breedte van de geselecteerde elementen

Oefening

- Maak een formulier waarin de bezoeker via tekstvakken de margin, padding en (achtergrond-)kleur voor een <div> kan opgeven.
- Stel deze waarden vervolgens in via jQuery, na een klik op de knop.
- Gevorderd: verwerk meer CSS-opties. Gebruik de objectnotatie



“huiswerk”

- Boek: hoofdstuk 1 en 2 (t/m p. 53)
- Leer selectors
- Lees de voorbeelden
- Probeer zelf na te maken
Begrip, Begrip, begrip.

Notatie van parameters

- Vaak: *JavaScript Object Notation: JSON*
- Formaat:

```
{  
    naam1 : 'waarde 1',  
    naam2 : 'waarde 2',  
    array1 : [ {naam1: 'waarde1', naamX: 'waardeX'} ] ,  
    naam3: 'waarde3'  
}
```

Let op de positie van accolades,
aanhalingsstekens komma's en dubbele punten.

JSON-notatie

Dit wordt met name veel gebruikt om parameter-objecten door te geven aan functies of plug-ins.

```
12 <script src="scripts/jquery-1.7.1.min.js"></script>
13 <script>
14 $(document).ready(function(e) {
15     var parameterObject = {
16         'background-color' : '#eee',
17         'width' : '400px',
18         'border' : '2px solid green',
19         'height' : '180px'
20     }
21     $('#btnTransform').on('click',function() {
22         $('p').css(parameterObject);
23     });
24 });
25 }
```

Vervolg...

- Formulieren
- Animaties
- HTML5-Storage
- AJAX-opdrachten
- jQuery UI