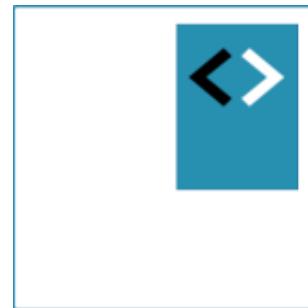




Ministerie van Defensie

HTML, CSS en JavaScript

Module : jQuery



Peter Kassenaar – info@kassenaar.com



Kennismaken met jQuery

Een abstractie-bibliotheek voor JavaScript. Eenvoudig cross-browser compatible script schrijven

De plaats van JavaScript. En dus ook jQuery!



Javascript – the ugly Stepchild

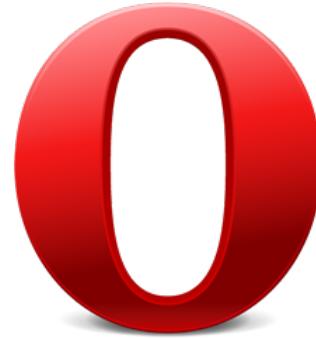
- Javascript's been around forever but...
 - Minimally used for its first 10 years
 - Few app developers were *really* proficient with Javascript
- Browser Incompatibilities
 - Varying Dom/CSS implementations
 - Varying event management interfaces
 - Varying feature support

Conclusie

JavaScript is OK...



...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

Javascript voorkennis bekend verondersteld

de browser is de *container* waarbinnen het
jQuery/JavaScript wordt uitgevoerd



2006

The screenshot shows the official jQuery website homepage from June 2006. The header features 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 navigation bar includes links for Download, API Documentation, Blog, Plugins, and Browser Support. A search bar is also present. The page highlights three key features: "Lightweight Footprint" (represented by a cube icon), "CSS3 Compliant" (represented by a stylized '3' icon), and "Cross-Browser" (represented by a globe icon). A prominent orange button on the right says "Download jQuery v1.10.0 or v2.0.1". Below the main content, there's a section titled "Resources" with links to the Core API Documentation, Learning Center, Blog, Contribute, About the Foundation, and Bug browser. A "TRY jQuery" banner with a plane icon is at the bottom.

jQuery write less, do more.

jQuery conference PORTLAND | JUNE 13 - 14
Oregon Convention Center
Members - \$50 Off

Download API Documentation Blog Plugins Browser Support Search

Lightweight Footprint
Only 32kB minified and zipped.
Can also be included as an AMD module

CSS3 Compliant
Supports CSS3 selectors to find elements as well as in style property manipulation

Cross-Browser
IE, Firefox, Safari, Opera, Chrome, and more

Download jQuery
v1.10.0 or v2.0.1

[View Source on GitHub →](#)
[How jQuery Works →](#)

Resources

- [jQuery Core API Documentation](#)
- [jQuery Learning Center](#)
- [jQuery Blog](#)
- [Contribute to jQuery](#)
- [About the jQuery Foundation](#)
- [Browse or Submit jQuery Bugs](#)

TRY jQuery

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 - Minder gebruikt momenteel (Bootstrap)
- jQuery Plugins, <http://plugins.jquery.com/>
 - Extending the power of jQuery



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 !

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="js/jquery.js"></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');

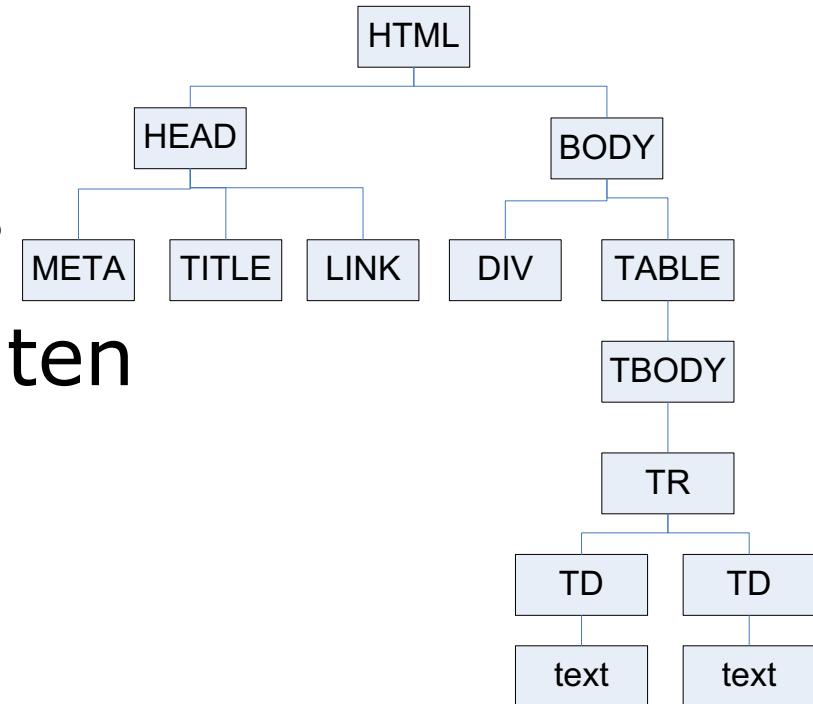
- *selector* is altijd jQuery, of kortweg \$
- *Action* is één van de opdrachten uit de jQuery API
- *parameters* is een lijst van 0, 1, of meer opdrachten
 - kommagescheiden notatie
 - JSON-notatie

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:

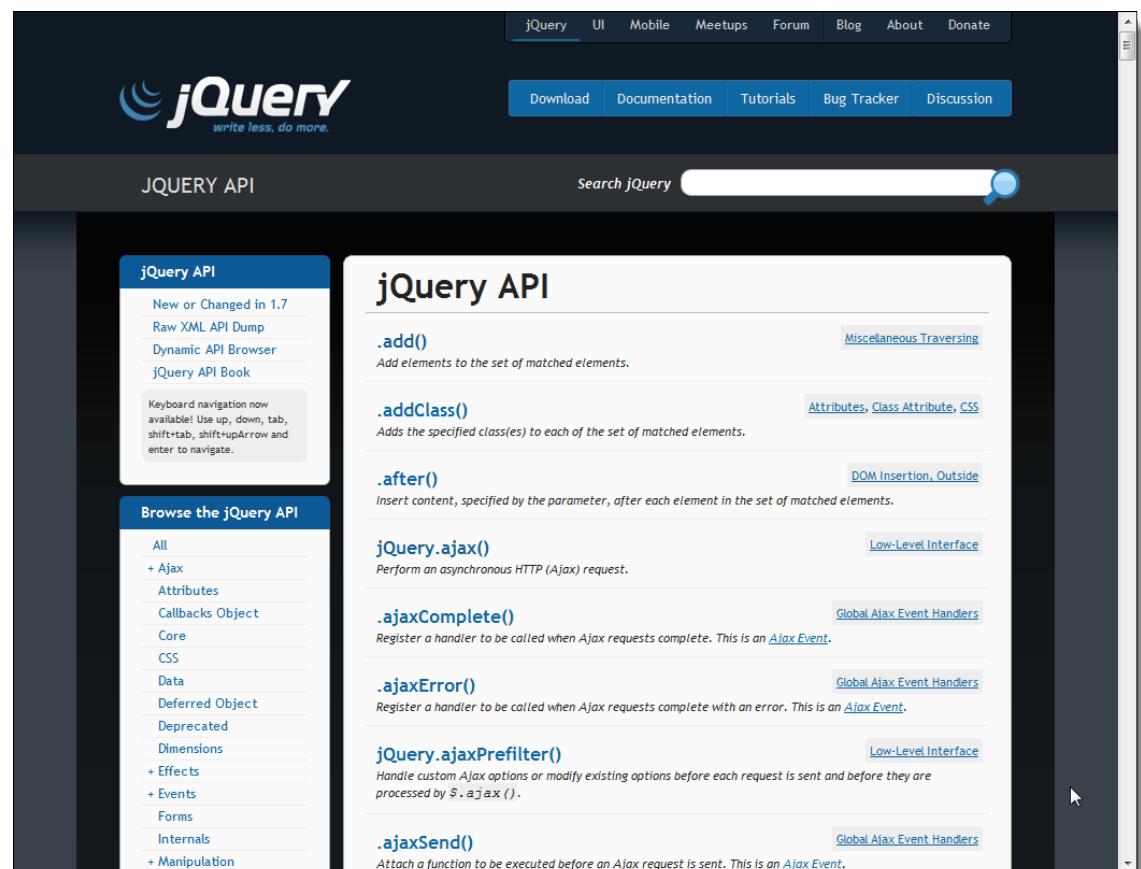
```
23 <body>  
24 <h2>Dit is een koptekst</h2>  
25 <p>Dit is de eerste paragraaf</p>  
26 <p>Dit is de tweede paragraaf</p>  
27 <p>Dit is de derde paragraaf</p>  
28 <button id="btnHide">Laat de paragrafen verdwijnen!</button>  
29 </body>
```

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 navigation bar with 'Download', 'Documentation', 'Tutorials', 'Bug Tracker', and 'Discussion'. The main content area has a dark header with the 'jQuery' logo and the tagline 'write less, do more.' To the right of the header is a search bar with the placeholder 'Search jQuery' and a magnifying glass icon. The left side features a sidebar with a 'jQuery API' section containing links for 'New or Changed in 1.7', 'Raw XML API Dump', 'Dynamic API Browser', and 'jQuery API Book'. Below this is a 'Keyboard navigation now available! Use up, down, tab, shift+tab, shift+upArrow and enter to navigate.' message. The main content area is titled 'jQuery API' and lists several methods:

- .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

Verdere selectie

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

Optioneel: filters:

:even

:odd

:checked

:input

...

Selectie van elementen

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

Complexe selectie

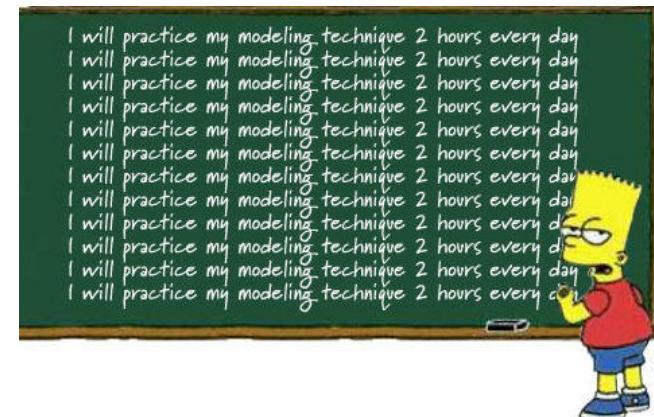
- Selectie op attributen:
 - `$(".[href='#']")` → alle tags met een href-attribuut dat gelijk is aan #
 - `$(".[src$='.jpg']")` → alle tags met een src-attribuut dat eindigt op .jpg
- Of op positie in een lijst:
 - `$(".ul li:first")` → alle eerste li-tags in elke lijst
 - `$(".tr td:last")` → elke laatste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

Oefening

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



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/>

The screenshot shows the jQuery API documentation for the `.on()` method. The left sidebar contains links for the API, a raw XML dump, a dynamic browser, and a book. The main content area has a title `.on()` and a subtitle `.on(events [, selector] [, data], handler(eventObject))`. It indicates that this version was added in 1.7 and returns a `jQuery` object. The **Description** section states: "Attach an event handler function for one or more events to the selected elements." Below the description are three parameters: `events`, `selector`, and `data`.

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.

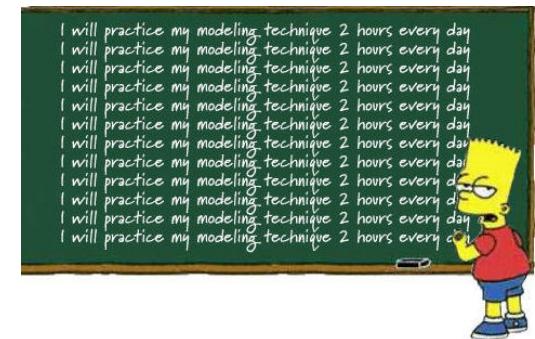
`data` Data to be passed to the handler in `event.data` when an event is triggered.

Ook beschikbaar:

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

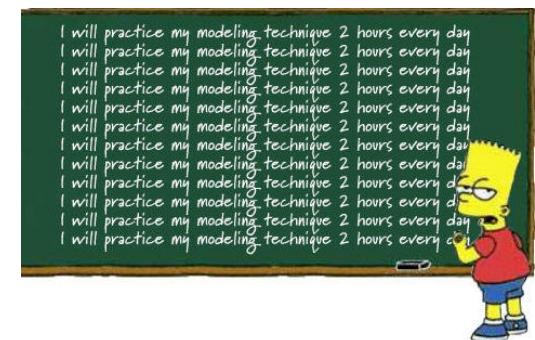
Oefening

- 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.



Optioneel: callback-functie

- wordt uitgevoerd na afloop van de animatie/hoofdfunctie

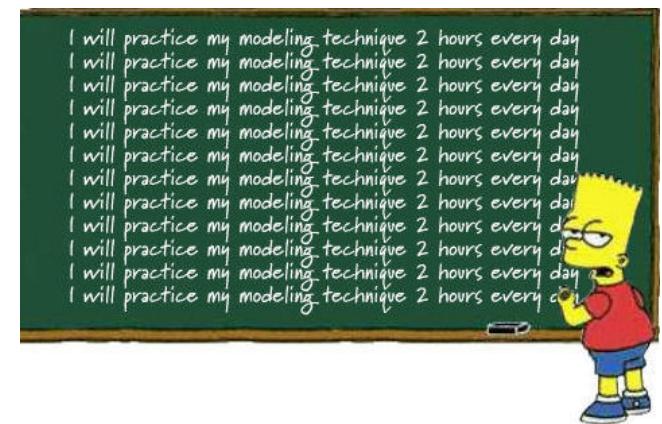
```
18 $(document).ready(function(e) {
19     // 'Algemene voorwaarden' tonen
20     $('#btnVoorwaarden').on('click', function(){
21         $('.voorwaarden').show(
22             'fast',           // parameter met de snelheid
23             function() { // callback-functie die wordt uitgevoerd na afloop 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();`

HTML verder bewerken

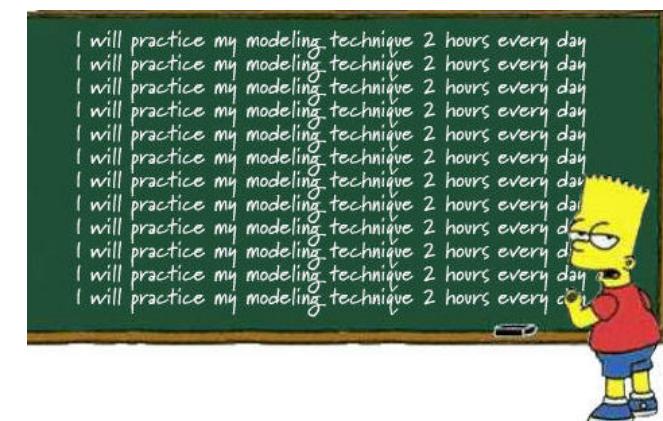
- .append('...') → inhoud achteraan toevoegen
- .prepend ('...') → inhoud vooraan toevoegen

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





JQuery Ajax

Communiceren met externe webservices

AJAX

- Jesse James Garret – feb. 2005
 - *Asynchronous JavaScript and XML*
- Ajax Calls verlopen via object XMLHttpRequest
 - 'XHR'-object
- Doel: communicatie met de *server en database*, zonder heen/terug te bladeren door de pagina
- Je moet een *API* beschikbaar hebben
 - API = Application Programming Interface, gemaakt op de server met PHP, Java, C#, enzovoort!

AJAX-opties

- <http://api.jquery.com/category/ajax/>

The screenshot shows a web browser window displaying the jQuery API documentation for the 'Ajax' category. The URL in the address bar is <http://api.jquery.com/category/ajax/>. The page title is 'Ajax – jQuery API'. The main content area features a large heading 'Ajax' with a subtext: 'The jQuery library has a full suite of AJAX capabilities. The functions and methods therein allow us to load data from the server without a browser page refresh.' Below this, there is a list of Ajax-related methods and events:

- 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
- jQuery.ajaxSetup()** - Low-Level Interface

On the left side, there is a sidebar with the 'jQuery API' logo and several navigation links:

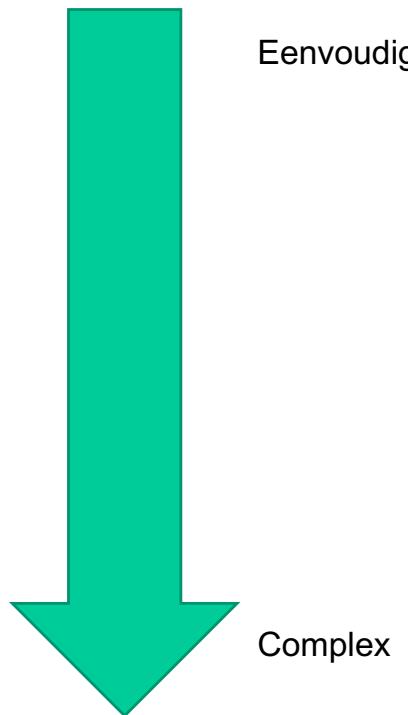
- New or Changed in 1.7
- Raw XML API Dump
- Dynamic API Browser
- jQuery API Book
- Keyboard navigation now available! Use up, down, tab, shift+tab, shift+upArrow and enter to navigate.
- Browse the jQuery API
 - All
 - Ajax
 - Global Ajax Event Handlers
 - Helper Functions
 - Low-Level Interface
 - Shorthand Methods
 - Attributes
 - Callbacks Object
 - Core
 - CSS
 - Data
 - Deferred Object

Ajax-methods in jQuery

- `.load()`
- Denk aan standaard jQuery-werkwijze:
 - Selecteer element met `$(...)`
 - roep method `.load()` aan
 - gebruik success / fail callback-functie

Meer AJAX-functies

- `$.getJSON()`
- `$.get()`
- `$.post()`
- `$.ajax()`



Eenvoudig

\$.getJSON()

Shorthand method voor ophalen van JSON-gegevens

Parameters: url, data, callbackfunction

```
// de klik op de knop omzetten in een Ajax-request
function handleClick(event) {
    // 1. variabelen instellen
    $.getJSON('ajaxData.json', // dit is de URL waar de call naar toe gaat
        {}, // Het object dat wordt meegestuurd (in dit geval leeg)
        function (data) {
            // Callback functie bij succes.
            // Over het object loopen en tonen in de UI
            console.log(data)
            for (var i in data) {
                $('#myDiv').append('<p>' + i + ': <strong>' + data[i] + '</strong></p>');
            }
        });
} // einde function handleClick
```

\$.get()

- Voert een standaard GET-request uit naar de server

```
// de klik op de knop omzetten in een Ajax-request
function handleClick(event){ ⑥
    // 1. $.get() aanroepen. URL als parameter meegeven, callbackfunctie voor resultaat
    $.get('simple.html', function (result) {
        $('#myDiv').html(result);
    })
}
```

\$.post()

- Voert een standaard POST-request uit
- Vaak gebruikt voor meesturen gegevens

```
// de klik op de knop omzetten in een Ajax-request
function handleClick(event) {
    // 1. data-object maken van tekstveld
    var data = {};
    data.name = $('#txtName').val();
    // 2. $.post() aanroepen en data meegeven. Op de server moet iets wachten op deze request
    $.post('simple.html', data, function (result) {
        $('#myDiv').html(result);
    })
}
```

Kijk in de Developer tools voor details Network Request
(zoals request-header en Form Data)

Generic workhorse: `$.ajax()`

- Alle parameters zelf instellen
- Meeste flexibiliteit
- Configuratieobject meegeven

```
13  function handleClick(event){ 10
14      // 1. variabelen instellen
15      $.ajax({
16          dataType: 'json',
17          contentType: 'application/json',
18          url : 'ajaxData.json',
19          success: handleData,    // success callbackfunctie
20          error : handleError   // error callbackfunctie
21      });
22 }
```

Ajax-instellingen instellen

- Standaard Ajax-instellingen:

- `$.ajaxSetup();`

- Basisinstellingen voor

- `type (GET, POST)`
 - `contentType`
 - `error-callback`
 - ...

```
1 // Generieke instellingen voor jQuery- AJAX-calls
2 $.ajaxSetup({
3     type: 'POST',
4     contentType: 'application/json',
5     error: function (xhr, status) {
6         alert('Er is een jQuery/AJAX-fout opgetreden : ' + xhr.responseText);
7     }
8 });
9
```

<http://api.jquery.com/jQuery.ajaxSetup/>

Parameters voor `$.ajax()`

- url : Adres waar call naar toe gaat
- type: GET- of POST-request
- contentType : Gegevenstype voor verzenden
- succes: Callback bij succes
- error: Callback bij optreden van fout
- beforeSend : Functie voorafgaand aan verzenden
- data: De gegevens die worden meegezonden

Gebruikelijk: meezendende configuratie-object {..}