opatel
opleidingen en consultancy

jQuery Core en jQuery UI
dag - 2

Peter Kassenaar
info@kassenaar.com

opatel
opleidingen en consultancy

Korte herhaling dag 1

“jQuery makes JavaScript suck less…”

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

opleidingen

opatel
opleidingen en consultancy

What is jQuery?

- *Javascript HTML DOM Manipulation library*
- Light weight (~31k compressed)
- Powerful and extremely practical functionality
- Easy to learn and intuitive to use
- *CSS 3 based* selector syntax
  - ID's, Classes, Elements

opatel
opleidingen en consultancy

jQuery statements ontleden

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

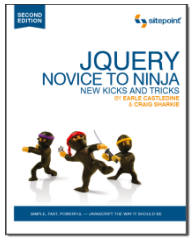
- *selector* is altijd `jQuery`, of kortweg `$`
- *Action* is één van de opdrachten uit de jQuery API
- *parameters* is een lijst van 0, 1, of meer opdrachten
  - kommagescheiden notatie
  - JSON-notatie

opatel
opleidingen en consultancy

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.
});
```



opatel
opleidingen en consultancy

Dag 1: Geschiedenis, Selectors, basisfuncties

Dag 2: LocalStorage, AJAX, meer API-functies, plugins

Dag 3: jQuery UI, meer uitwerkingen

opatel
opleidingen en consultancy

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();`

opatel
opleidingen en consultancy

De DOM-structuur aanpassen

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

opatel
opleidingen en consultancy

Intermezzo – core JavaScript

- De `for()` –lus
- Voert handeling uit zolang een bepaald statement waar (`true`) is
- Notatie
  -
```
for (teller ; voorwaarde ; ophoging){
  // alle opdrachten…
}
```
  -
```
for (var i = 0 ; i < 10; i++) {
  …
}
```

opatel
opleidingen en consultancy

Intermezzo – LocalStorage

Met jQuery gegevens opslaan in de browser

opatel
opleidingen en consultancy

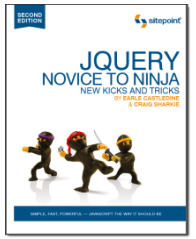
jQuery: CSS

- `.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()`
  - Get en Set hoogte en breedte van selectie

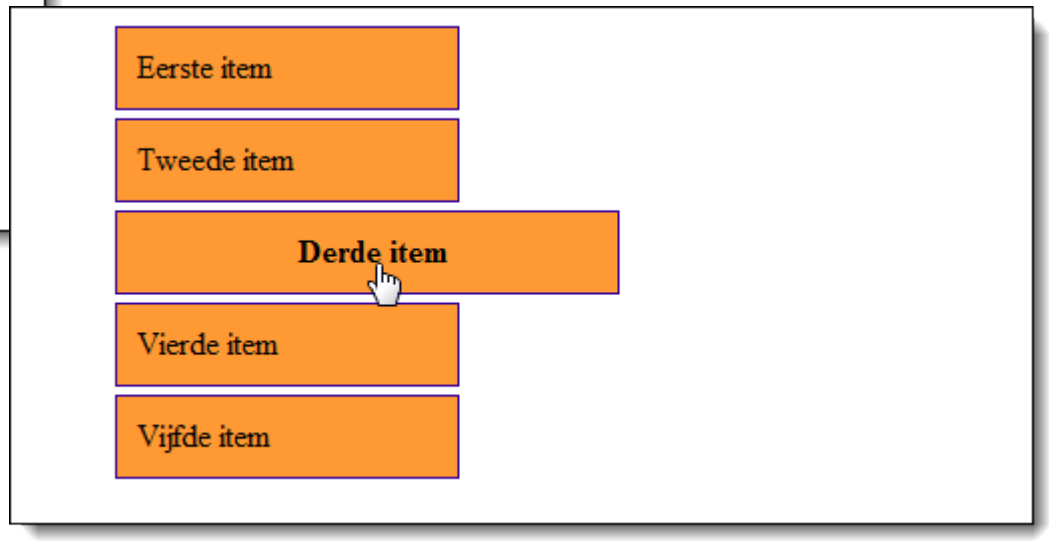
opatel
opleidingen en consultancy

Basis animatie

- jQuery-functie `.animate()`
- Doel: CSS-eigenschappen aanpassen in een bepaalde tijdsduur
- Eigenschappen meegeven in parameter-object
  - JSON-notatie!
  - Tijdsduur aangeven in milliseconden



```
9   -->
10  <script src="scripts/jquery-1.7.1.min.js"></script>
11  <script>
12  $(document).ready(function(e) {
13      var animateOptions = {
14          'padding-left': '400px'
15      };
16      // 1. Eerste paragraaf animeren
17      $('#btnAnimate').on('click', function(){
18          $('#textAnimate').animate(animateOptions, 2000);
19      });
20
```



opatel
opleidingen en consultancy

Typen animatie

- Standaard: lineair en swing
- Meer: via plugins

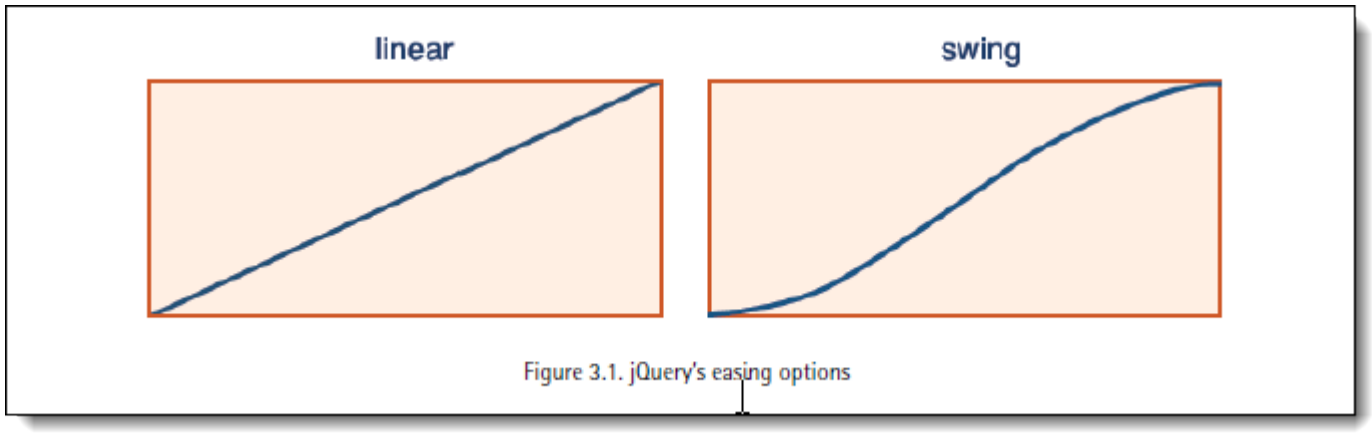


Figure 3.1. jQuery's easing options

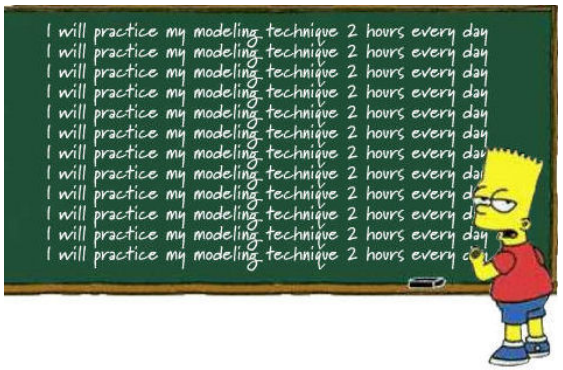
p.56

http://gsgd.co.uk/sandbox/jquery/easing/

opatel
opleidingen en consultancy

Oefening

- Maak zelf een script waarmee je elementen op de pagina animeert
  - test verschillende CSS-eigenschappen
  - test voor diverse events (`click, hover, focus, blur`)
- Gevorderd: de *animation queue*

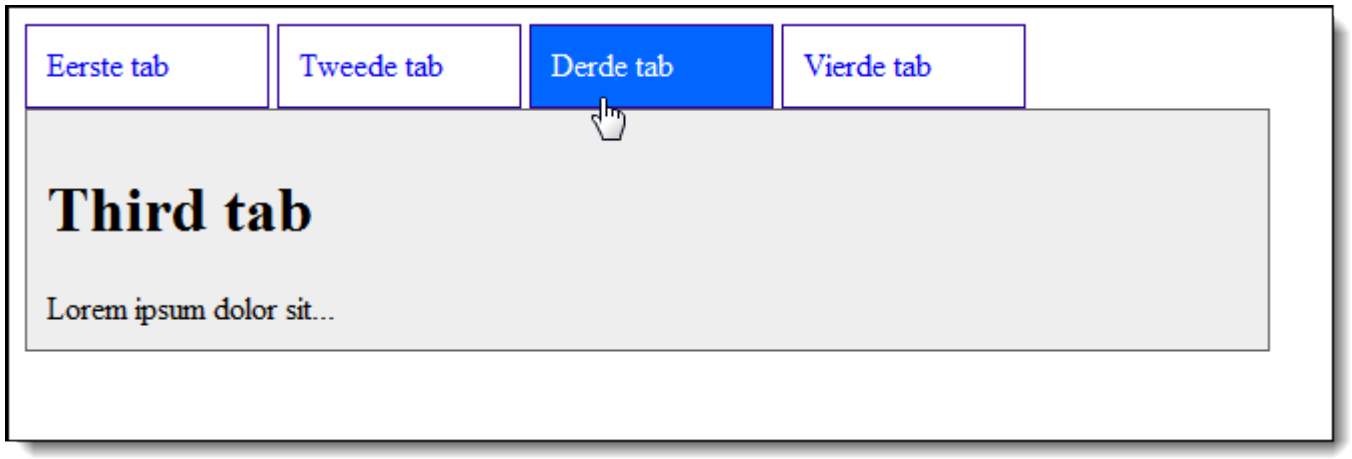


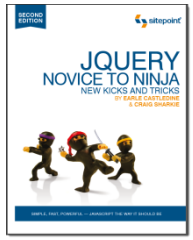
opatel
opleidingen en consultancy

User Interface-elementen: Tabs

- Algemene werkwijze
  - maak een `<ul>` met de menu-items (‘tabs’)
  - maak een `<div>` met de content-items
  - Elke tab komt in een eigen `<div>` te staan
- Zorg voor correcte styling via CSS
  - menu-items: `display: inline;`
  - styling voor de hyperlinks en tab-inhoud

- Script aanhaken
  - Verberg juiste `<div>`
  - Zoek uit op welke link is geklikt en maak deze div zichtbaar.
  - Voeg dynamisch classes toe en verwijder ze



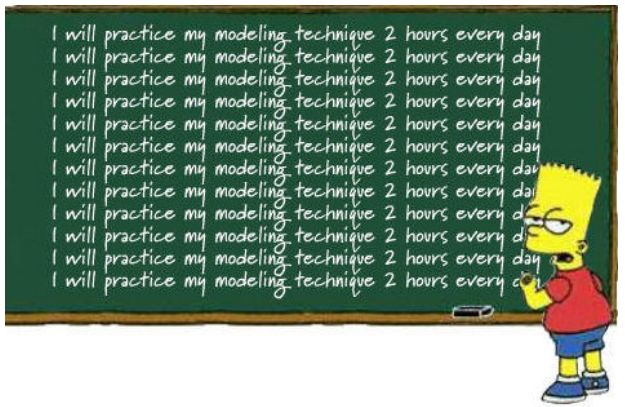


```
Eventueel: invoegen via CDN
<script src="http://code.jquery.com/jquery.min.js"></script>
-->
<script src="scripts/jquery-1.7.1.min.js"></script>
<script>
$(document).ready(function(e) {
    // verberg alle tabbladen, behalve de eerste
    $('#content div:not(:first)').hide();

    // Klikken op hyperlinks aanhaken
    $('#menu li a').on('click', function(evt){
        evt.preventDefault();

        //1. alle tabs verbergen
        $('#content div').hide();

        // 2. huidige class selected verwijderen
        $('#menu li a.selected').removeClass('selected');

        // 3. class selected toevoegen aan aangeklikte link
        $(this).addClass('selected');

        // 4. uitzoeken welke div zichtbaar moet worden
        var clicked = $(this).attr('href');
        $(clicked).fadeIn('fast');
    });
});
```

opatel
opleidingen en consultancy

Oefening

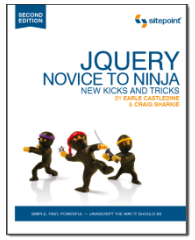
- Maak zelf een script om tabs te tonen en te verbergen
  - Denk eerst aan HTML + CSS
  - Daarna aan scripting
  - Let op je selectors!
- Gevorderd: meer animatie bij selecteren tabs.
  - Gebruik `.hover()` en `.animate()`



opatel
opleidingen en consultancy

Tooltips maken

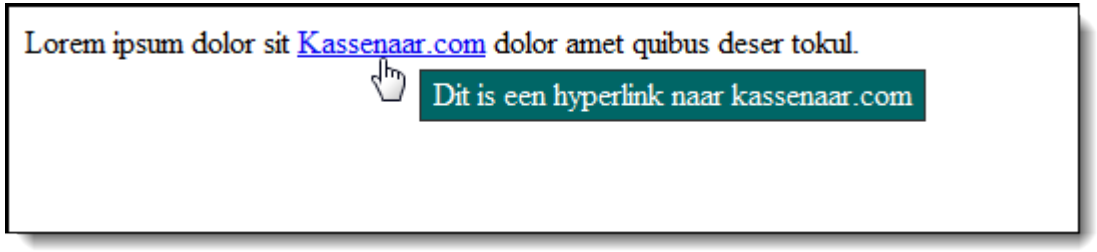
- In feite: *combineren* van bekende technieken
  - 1. Zoek het title-attribuut van de link waarover gehovered wordt.
  - 2. ken dit toe aan een variabele
  - 3. gebruik de functie `.hover()`
  - 4. maak een (tijdelijke) paragraaf met gewenste styling
  - 5. Voeg deze toe aan DOM op gewenste positie.



opatel
opleidingen en consultancy

Nieuw: events meegeven

- mouseevent `evt` meegeven en gebruiken
  - `evt.pageX` en `evt.pageY` gebruiken om locatie te bepalen
  - in mousemove deze coördinaten volgen.



opatel
opleidingen en consultancy

De code

- Voor jezelf:
  - eerst kapstokcode schrijven
  - daarna de invulling schrijven

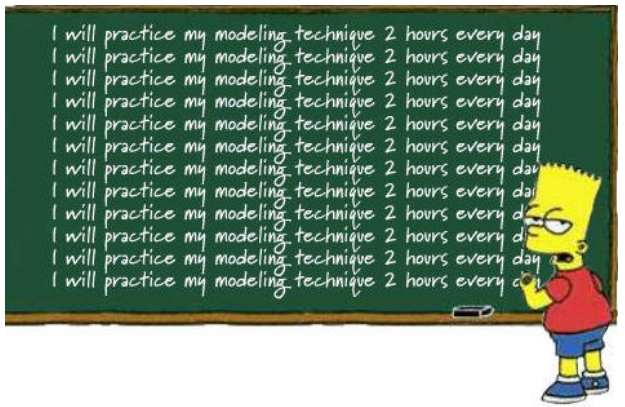
```
$(document).ready(function(e) {
    // stap 1: structuur instellen
    $('.info').hover(
        function(){

        },
        function(){

    }).mousemove(function(){

    });
```

```
34      // stap 2: details invoegen
35      $('.info').hover(
36          function(evt){
37          // 2a. mouseover-functie
38          var titleText = $(this).attr('title');
39          $(this)
40              .data('tipText', titleText)
41              .removeAttr('title');
42          // nieuwe alinea maken, toevoegen aan DOM en infaden
43          $('<p class="tooltip"></p>')
44              .text(titleText)
45              .appendTo('body')
46              .css('top', (evt.pageY - 10) + 'px')
47              .css('left', (evt.pageX + 20) + 'px')
48              .fadeIn('slow');
49          },
50          // 2b. mouseout-functie
51          function(){
52              // title-attribuut herstellen
53              $(this).attr('title', $(this).data('tipText'));
54              // tooltip uit het DOM verwijderen
55              $('.tooltip').remove();
56      }).mousemove(function(evt){
57          // 3. meeverplaatsen met de muis
58          $('.tooltip')
59          .css('top', (evt.pageY - 10) + 'px')
60          .css('left', (evt.pageX + 20) + 'px');
61      });
62  });
63
```

opatel
opleidingen en consultancy

Oefening

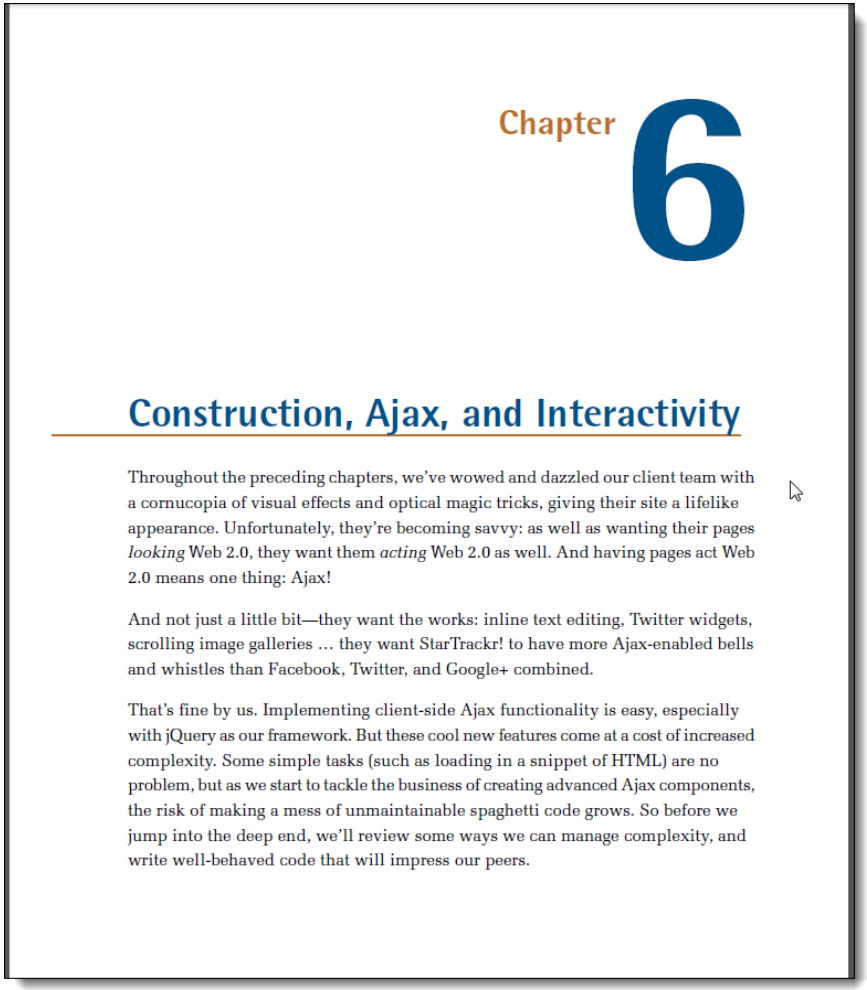
- Bekijk dit voorbeeld en werk het eventueel zelf uit
  - je kunt ook de voorbeeldcode nemen en bestuderen



opatel
opleidingen en consultancy

AJAX-navigation

- Lees zelf: hoofdstuk 6
  - Achtergrondinformatie & best practices



Chapter 6

Construction, Ajax, and Interactivity

Throughout the preceding chapters, we've wowed and dazzled our client team with a cornucopia of visual effects and optical magic tricks, giving their site a lifelike appearance. Unfortunately, they're becoming savvy: as well as wanting their pages *looking* Web 2.0, they want them *acting* Web 2.0 as well. And having pages act Web 2.0 means one thing: Ajax!

And not just a little bit—they want the works: inline text editing, Twitter widgets, scrolling image galleries … they want StarTrackr! to have more Ajax-enabled bells and whistles than Facebook, Twitter, and Google+ combined.

That's fine by us. Implementing client-side Ajax functionality is easy, especially with jQuery as our framework. But these cool new features come at a cost of increased complexity. Some simple tasks (such as loading in a snippet of HTML) are no problem, but as we start to tackle the business of creating advanced Ajax components, the risk of making a mess of unmaintainable spaghetti code grows. So before we jump into the deep end, we'll review some ways we can manage complexity, and write well-behaved code that will impress our peers.

opatel
opleidingen en consultancy

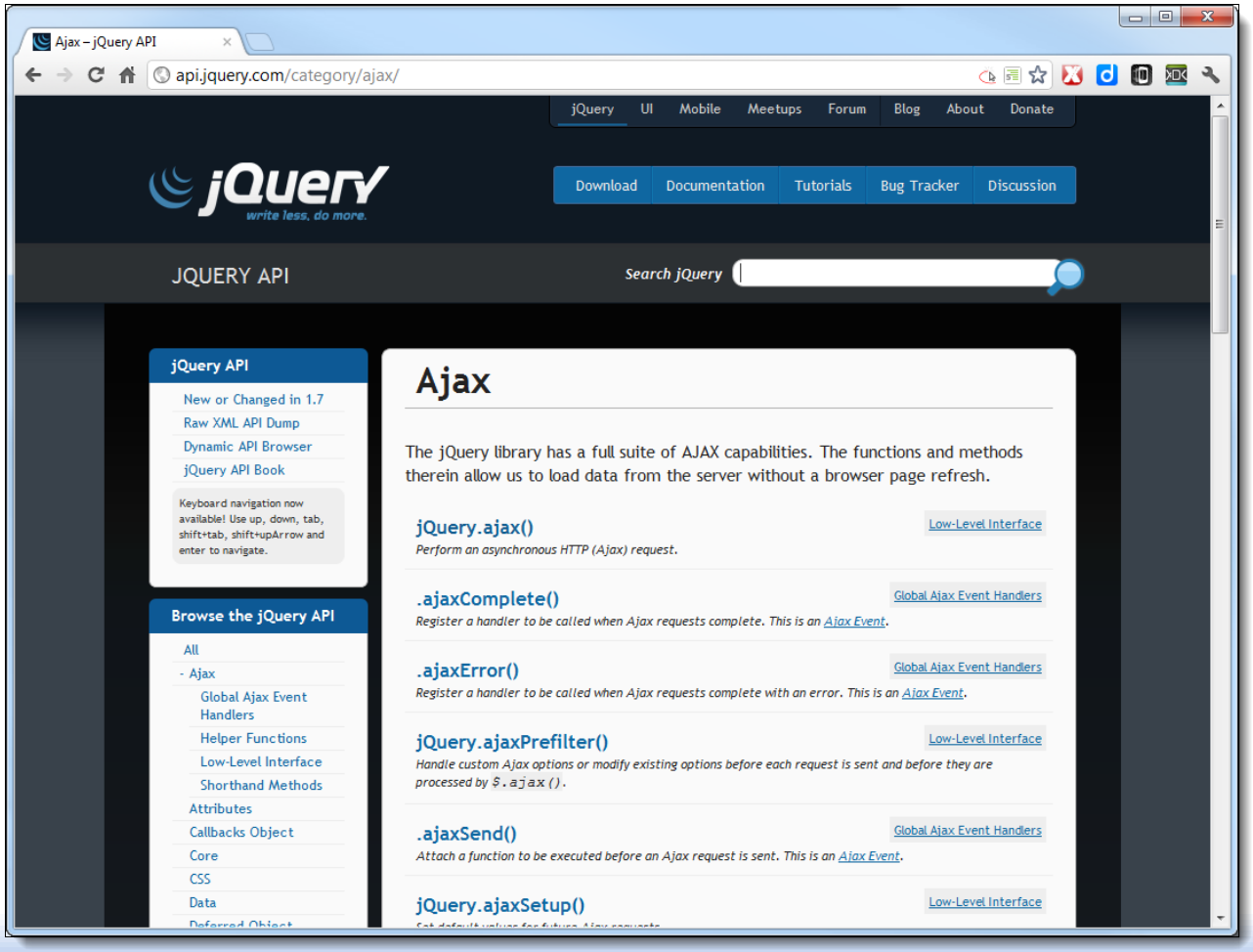
AJAX

- Jesse James Garret – feb. 2005
  - *Asynchronous JavaScript and XML*
- Ajax Calls verlopen via object XMLHttpRequest
  - 'XHR'-object
- Doel: communicatie met de *server*, zonder heen/terug te bladeren door de pagina

opatel
opleidingen en consultancy

AJAX-opties

– http://api.jquery.com/category/ajax/



opatel
opleidingen en consultancy

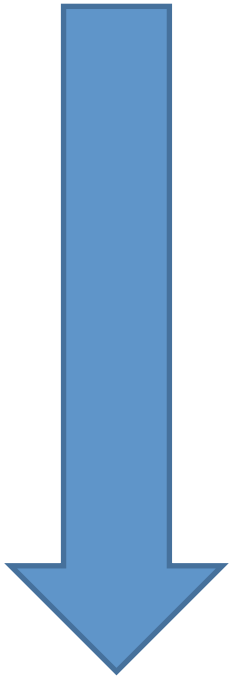
Ajax-methods in jQuery

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

opatel
opleidingen en consultancy

Meer AJAX-functies

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



Eenvoudig

Complex

opatel
opleidingen en consultancy

$.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
```

opatel
opleidingen en consultancy

$.get()

- Voert een standaard GET-request uit naar de server

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

opatel
opleidingen en consultancy

$.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
    $.post('simple.html', data, function (result) {
        $('#myDiv').html(result);
    })
}
```

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

opatel
opleidingen en consultancy

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  }
```

opatel
opleidingen en consultancy

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/

opatel
opleidingen en consultancy

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: meezenden in configuratie-object {..}