



jQuery Advanced



Complexe webapps met jQuery

Peter Kassenaar – info@kassenaar.com



Introductie

Peter Kassenaar;

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

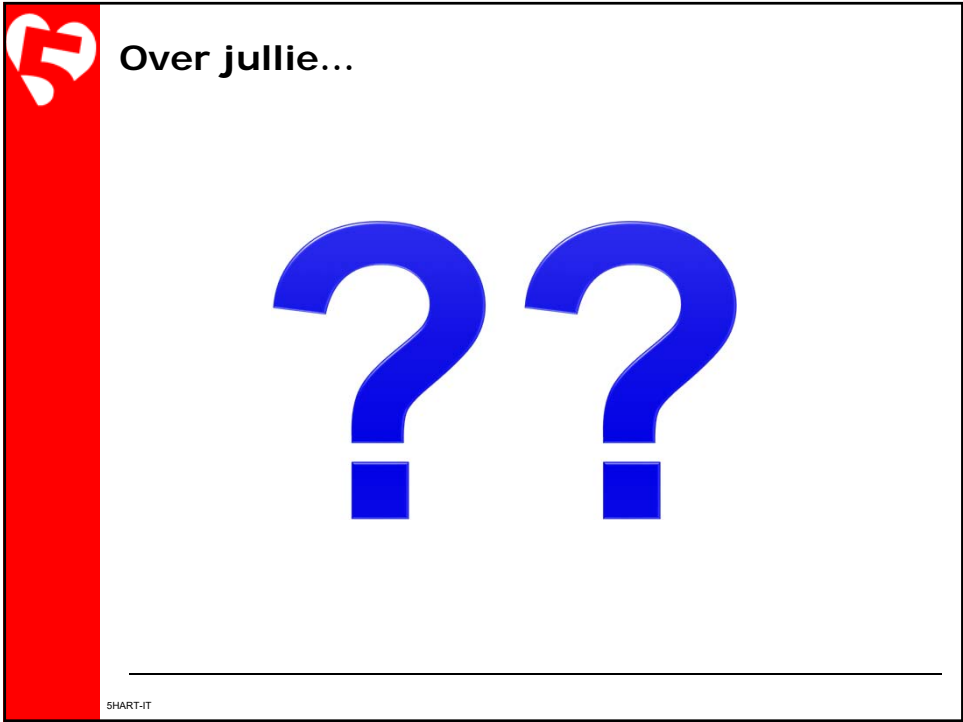
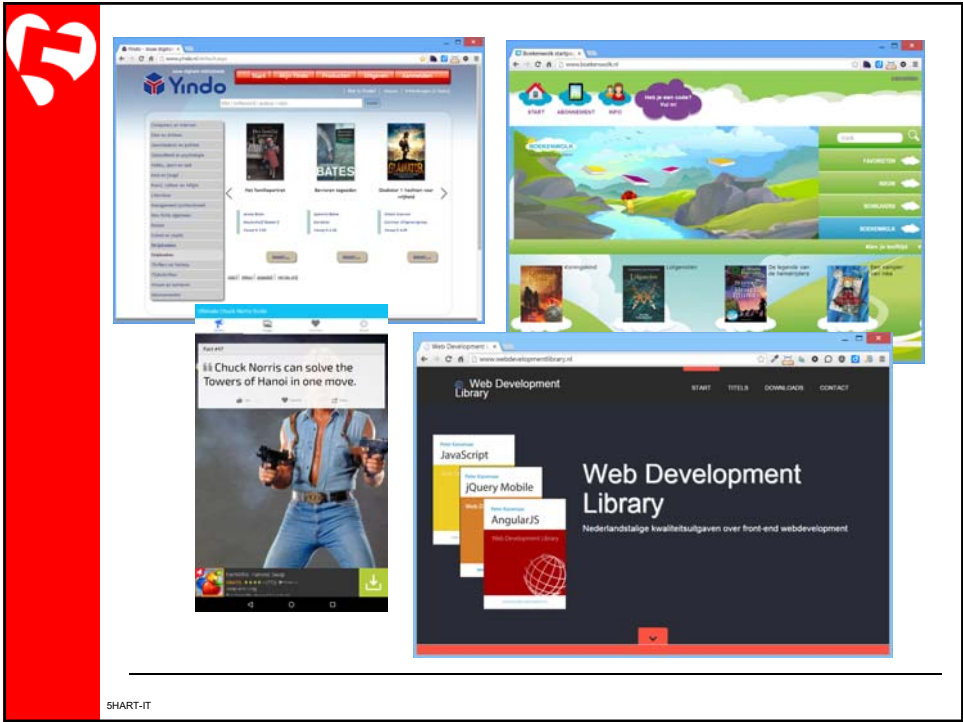
www.kassenaar.com/blog

info@kassenaar.com

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



SHART-IT





Voorkennis webdesign, mobile apps?

Voorkennis andere (web)talen?

Verwachtingen van de cursus?

Concrete projecten?

SHART-IT



Materialen

Software

Handouts

Oefeningen

Websites



SHART-IT



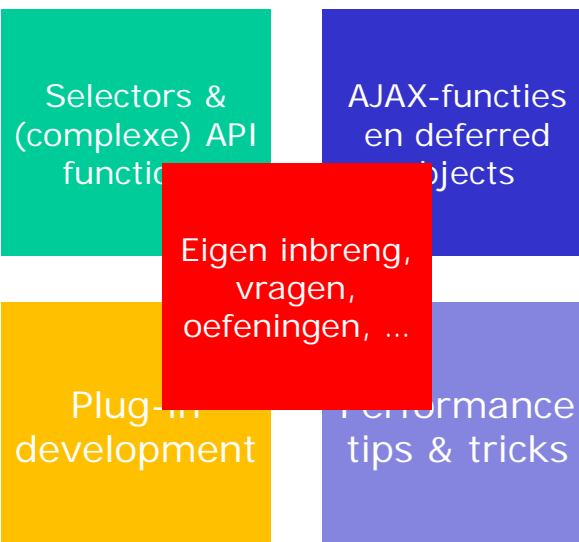
Indeling

- Geschiedenis - plaatsbepaling – context
- Webapps met jQuery
- In detail:
 - Selectors, .each(), .data(), .extend() en meer.
 - Werken met Ajax
 - Deferred objects
 - Plug-ins ontwikkelen
 - ...


SHART-IT




In een notendop



SHART-IT



Werkwijze



...

SHART-IT



Vragen?

SHART-IT



"jQuery makes JavaScript suck less..."

"jQuery is what the HTML DOM should have been!"

SHART-IT



2006

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

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

Cross-Browser
Browsers: Firefox, Chrome, Safari, Opera, IE.

Download jQuery
v1.10.2 or v2.0.2

View Source on GitHub
How jQuery Works

What is jQuery?
jQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across all major browsers.

Resources

- jQuery Core API Documentation
- jQuery Learning Center

UI Mobile API

SHART-IT



What is jQuery?

- *Javascript HTML DOM Manipulation library*
- Light weight (~80k compressed, all modules)
- Powerful and extremely practical functionality
- Easy to learn and intuitive to use
- *CSS 3 based* selector syntax
 - ID's, Classes, Elements
- Custom Event handling
- Easily to extend through plug-in API

SHART-IT



Decompose jQuery statements

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

- *selector* is always jQuery, or it's alias \$
- *Action* is one of the jQuery API commands
- *parameters* a list of 0, 1, or more arguments
 - Comma-separated
 - JSON-notation

SHART-IT



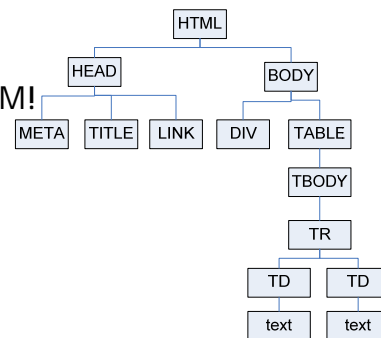
The Document Object Model

What to select?

- Elements in the DOM!

Use CSS-syntax
to select elements

- elements/tags
- classes
- id's



SHART-IT



Always make sure the Page is ready

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

```

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

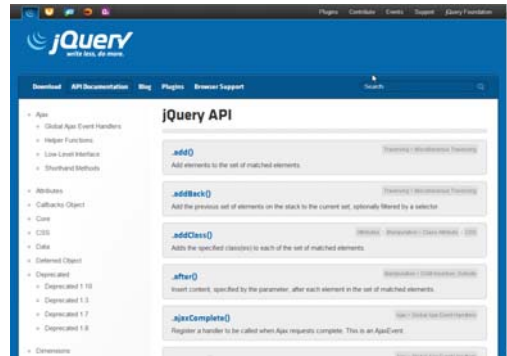
In plain JavaScript: `window.onload=function(){...}`

SHART-IT



The jQuery API

- All options for `.action()` are described internet the *jQuery API*
- Learn and use!



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

SHART-IT



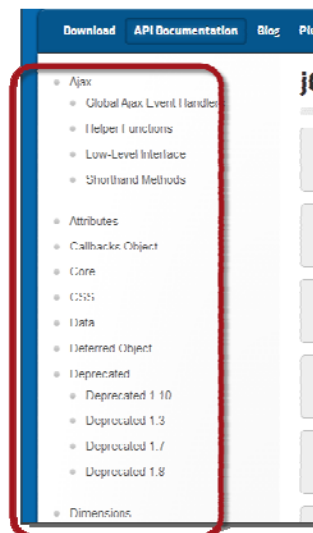
API categories

Most used:

- Ajax
- Attributes
- CSS
- Data
- DOM
- Effects
- Events
- Forms

Supportive

- Utilities
- Dimensions
- Offset
- ...



SHART-IT



Enkele complexe jQuery methods

Complexe selectie, `.data()`, `.each()` en `.extend()`

SHART-IT



Complex selection

Select on attributes:

- `$("[href='#']")` → alle tags met een href-attribuut dat gelijk is aan #
- `$("[src$='.jpg']")` → alle tags met een src-attribuut eindigt op .jpg

SHART-IT



Select on position within an element:

- `$("ul li:first")` → alle eerste li-tags in elke lijst
- `$("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

SHART-IT



Select form elements

- `$(":text")` → alle input elementen van het type tekst
- `$(":password")` → alle input elementen van het type password
- `$(":selected")` → alle elementen die geselecteerd zijn
- `$(":checked")` → alle selectievakjes die geselecteerd zijn
- `$(":visible")` → alle zichtbare elementen

SHART-IT



Selecting specific elements

- Child elements only
 - `$('#nav li > a');`
 - `$('#nav li').children();`
 - `$('#nav li').children('a');` // filter specific type of child
- Terminologie:
 - Descendant: een compleet element, inclusief child elements van een parent element
 - Child : alleen het eerste child-element

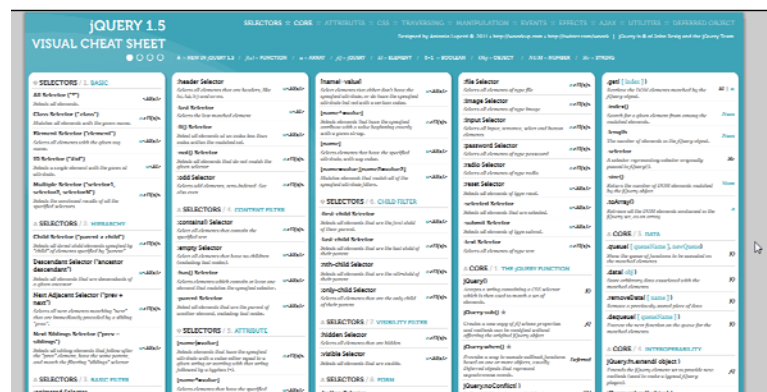
```
<ul id="nav">
  <li><a href="#">Item 1</a></li>
  <li><a href="#"><span>Item 2</span></a></li>
  <li><a href="#">Item 3</a></li>
  <li><a href="#">Item 4</a></li>
  <li><a href="#">Item 5</a></li>
</ul>
```

SHART-IT



Reference: cheat sheets

- zoek op trefwoord 'jQuery Cheat Sheet'



SHART-IT



De functie `$().each()`

- Een functie uitvoeren voor elk van de elementen in de selectie
- Nodig: als je *meerdere* opdrachten wilt gebruiken voor elementen binnen een set
- Notatie: `$('#selectie').each(function(i, el){...});`
- Parameters
 - `i`: index (nummer) van het element
 - `el`, het betreffende element waarop de `each` van toepassing is.
 - Dus: `el === this`, binnen de functie.

SHART-IT



Bijvoorbeeld: hyperlinks selecteren → tooltip instellen
→ `click()`-event instellen → en speciale opmaak toekennen

```
// state van de accordion met categorieën herstellen
var categoryID = localStorage.getItem('categoryID');
if (categoryID) {
  $('#accordion h2').each(function () {
    if ($(this).data('categoryid') == categoryID) {
      $(this).click();
    }
  });
}
```

<http://api.jquery.com/each/>

SHART-IT



De functie .data()

- Werkt samen met HTML5-attribuut data="..."
- Zowel getter als setter
- HTML:

```
<div data-customid="5">Data-div met ID 5</div>
```

Script

```
$( 'div' ).data( 'customid' )
```

.data(key)

Returns: *Object*

Description: Return the value at the named data store for the first element in the jQuery collection, as set by data(name, value) or by an HTML5 data-* attribute.

.data(key)

version added: 1.2.3

key
Type: <i>String</i>
Name of the data stored.

<http://api.jquery.com/data/>

SHART-IT



De functie .extend()

- Inhoud van twee of meer objecten samenvoegen in het eerste object
- In de praktijk: vooral bij plug-ins, meegegeven configuratieobject verwerken

```
var fruit = {
  appel: 5,
  banaan: 10,
  peer: 3
}
var meerFruit = {
  kiwi: 13,
  mango: 8
}

$.extend(fruit, meerFruit);
```


SHART-IT



jQuery AJAX

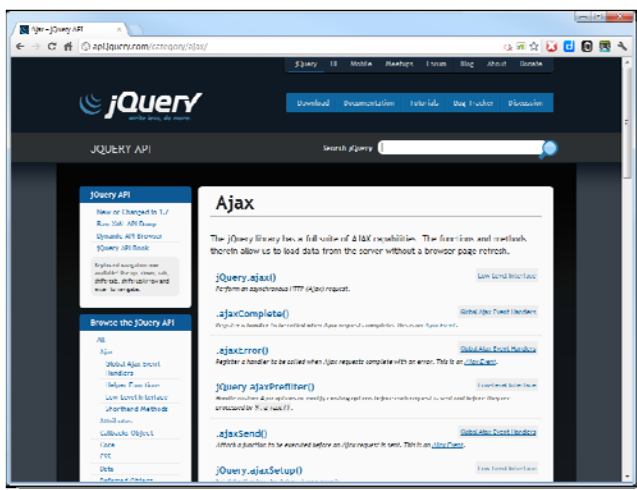
AJAX-functies en deferred objects

SHART-IT



AJAX-opties

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



SHART-IT



Ajax-methods in jQuery

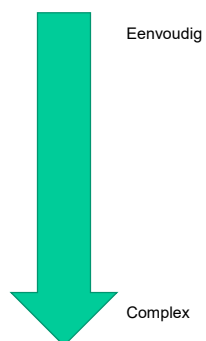
- `.load()`
- Denk aan standaard jQuery-werkwijze:
 - Selecteer element met `$(...)`
 - roep method `.load()` aan
 - gebruik success callback-functie

SHART-IT



Meer AJAX-functies

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



SHART-IT



\$.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 lopen 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
```

SHART-IT



\$.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);
    });
}
```

SHART-IT



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

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

SHART-IT



Oefening

- Schrijf zelf jQuery \$.get() en \$.post() - opdrachten
- Retourneer eenvoudige tekst, of haal 'echte' data op.
- Kijk ook naar \$.getJSON()
- Documentatie op <http://api.jquery.com>



SHART-IT



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 }

```

SHART-IT



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 {..}

SHART-IT

5

jQuery.ajax()

Categories: [Ajax](#) • [Low-Level Interface](#)

jQuery.ajax(url [, settings]) Returns: [jqXHR](#)

Description: Perform an asynchronous HTTP (Ajax) request.

jQuery.ajax(url [, settings]) version added: 1.5

url
Type: [String](#)
A string containing the URL to which the request is sent.

settings
Type: [PlainObject](#)
A set of key/value pairs that configure the Ajax request. All settings are optional. A default can be set for any option with [\\$.ajaxSetup\(\)](#). See [jQuery.ajax\(settings\)](#) below for a complete list of all settings.

jQuery.ajax(settings) version added: 1.0

settings
Type: [PlainObject](#)
A set of key/value pairs that configure the Ajax request. All settings are optional. A default can be set for any option with [\\$.ajaxSetup\(\)](#).

accepts (default: depends on DataType)
Type: [PlainObject](#)
The content type sent in the request header that tells the server what kind of response it will accept in return. If the `accepts` setting needs modification, it is recommended to do so once in the `$.ajaxSetup()` method.

async (default: `true`)
Type: [Boolean](#)
By default, all requests are sent asynchronously (i.e. this is set to `true` by default). If you need synchronous requests, set this option to `false`. Cross-domain requests and `dataType: "jsonp"`

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

SHART-IT

5

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/>

SHART-IT



Jsonp via \$.ajax()

- Voor Cross-domain calls
- Parameter: `dataType : 'jsonp'`
- Achter de schermen regelt jQuery juiste callback, etc.

```

12 // de klik op de knop omzetten in een Ajax-request
13 function handleClick(event){
14     // 1. variabelen instellen
15     $.ajax({
16         dataType: 'jsonp',
17         contentType: 'application/json',
18         url: 'http://api.yindo.com/api/214f032c-de61-4b86-87d2-8b1f7f5a7ca9/book/new/10/1/10',
19         success: handleData, // success callbackfunctie
20         error: handleError // error callbackfunctie
21     });
22 }

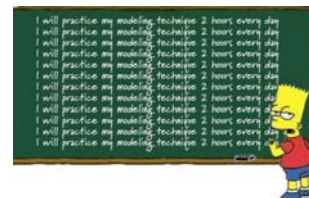
```

SHART-IT



Oefening

- Schrijf zelf jQuery `$.ajax()`-opdrachten
- Retourneer eenvoudige tekst, of haal 'echte' data op.
- Gebruik het configuratie-object `url, dataType, success, etc.`



SHART-IT



Deferred object

`$.when()` en `$.then()`

SHART-IT

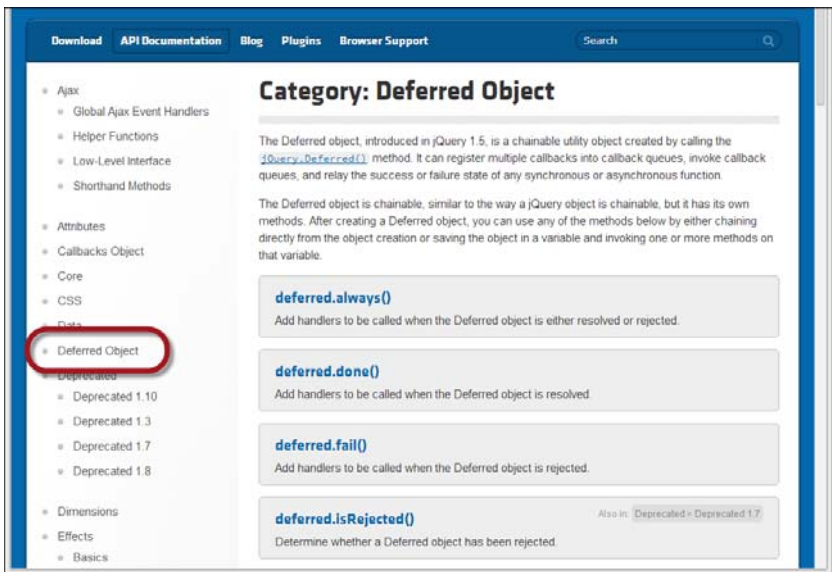


Deferred object

- Als een resultaat van meerdere ajax-calls afhankelijk is.
 - Of: als meerdere ajax-calls in chain moeten worden uitgevoerd
 - Doel: betere leesbaarheid, meer modulaire opbouw, betere foutafhandeling.
 - Gebruik jQuery `$.when()` en `$.then()`
 - jQuery 1.8+
-

SHART-IT

5



<http://api.jquery.com/category/deferred-object/>

SHART-IT


5

Case – 3 ajax calls

jQuery deferred object - 01

\$.get().tekst

Klaar!



SHART-IT

23


5

```

// Event handler
function handleClick(event) {
  // 1. $.get() aanroepen. URL als parameter meegeven, callbackfunctie voor resultaat
  $.get('simpleText.txt', function (result) {
    $('#myDiv1').html(result);
  });
  // 2. maar wat nu als een van de calls heel lang gaat duren... (simuleer met setTimeout())
  setTimeout(function () {
    $.get('simpleText.txt', function (result) {
      $('#myDiv2').html(result);
    });
  }, 2500);
  // 3. derde call + updaten UI
  $.get('simpleText.txt', function (result) {
    $('#myDiv3').html(result);
  });
}

// Event handler
function handleClick(event) {
  // 1. $.get() aanroepen. URL als parameter meegeven, callbackfunctie voor resultaat
  $.get('simpleText.txt', function (result) {
    $('#myDiv1').html(result);
    // 2. tweede call (traagheid gesimuleerd met setTimeout())
    setTimeout(function () {
      $.get('simpleText.txt', function (result) {
        $('#myDiv2').html(result);
        // 3. derde call + updaten UI
        $.get('simpleText.txt', function (result) {
          $('#myDiv3').html(result);
          $('.result').addClass('klaar');
        });
      }, 2500);
    });
  });
}

```



SHART-IT

5

Deferred Methods

- Some methods available on deferred objects
 - Return deferred object for a process flow
 - when
 - promise
 - Attach handler functions to deferred object
 - then
 - done
 - fail
 - progress
 - always
 - Change state of deferred object
 - resolve
 - reject

SHART-IT



Ander voorbeeld van deferreds

Case – afhankelijkheid van meerdere events

<http://stackoverflow.com/questions/10945643/correct-way-of-using-jquery-mobile-phonegap-together>

115 You can use deferred feature of JQuery.

```

var deviceReadyDeferred = $.Deferred();
var jqmReadyDeferred = $.Deferred();

document.addEventListener("deviceReady", deviceReady, false);

function deviceReady() {
    deviceReadyDeferred.resolve();
}

$(document).one("mobileinit", function () {
    jqmReadyDeferred.resolve();
});

$.when(deviceReadyDeferred, jqmReadyDeferred).then(dowhenBothFrameworksLoaded);

function dowhenBothFrameworksLoaded() {
    // TBD
}
  
```

SHART-IT



Oefening

- Leer werken met het deferred object
- Schrijf een chain ajax-calls en resolve deze met `.then()`
- Maak foutafhandeling met `.fail()`
- Documentatie:

<http://api.jquery.com/category/deferred-object/>



SHART-IT