

# jQuery

## Inleiding

### jQuery linken:

```
<script src="//code.jquery.com/jquery-2.1.3.min.js"></script>
```

### \$(document).ready()

```
$(document).ready(function () {
    console.log('ready!');
});
```

### Shorthand

```
$(function() {
    console.log('ready!');
});
```

## Basismethodes

.text()	Krijg de (gecombineerde) tekstinhoud van de geselecteerde element(en)
.html()	Krijg de inhoud van het eerste geselecteerde element Bepaal de inhoud van de geselecteerde elementen
.attr()	Krijg de waarde van een attribuut van het eerste geselecteerde element Bepaal de waarde van een attribuut van de geselecteerde elementen
.val()	Krijg de waarde van form-elementen

## Selectors

→ volgens standaard CSS-selectors

### Syntax

```
$('.selector');
```

### Testen of een selector elementen bevat

```
$('.selector').length !== 0
```

### Selectors verfijnen

\$('.sel1').has('.sel2');	Elementen van selector1 met afstammelingen die matchen met selector2
\$('.sel1').not('.sel2');	Alle elementen van selector1 die niet matchen met selector2
\$('.sel1').filter('.sel2');	Alle elementen van selector1 die matchen met selector2
\$('.sel1').find('.sel2');	Alle child-elementen van selector1 die matchen met selector2
\$('.selector').first();	Het eerste element van alle elementen die matchen met de selector
\$('.selector').eq(n);	Het <i>n</i> -de element van alle elementen die matchen met de selector

## Pseudo-selectors

:animated	Alle elementen waarop een animatie bezig is
:contains( <i>tekst</i> )	Alle elementen die bepaalde <i>tekst</i> bevatten
:empty	Alle elementen zonder children of text nodes
:even	Alle elementen uit de selectie met een even index
:gt( <i>n</i> )	Alle elementen met een index groter dan <i>n</i>
:header	Alle elementen die headers zijn (h1, h2, h3, etc.)
:hidden	Alle elementen die niet zichtbaar zijn
:lt( <i>n</i> )	Alle elementen met een index kleiner dan <i>n</i>
:odd	Alle elementen uit de selectie met een oneven index
:parent	Alle elementen met minstens één child node/element
:visible	Alle elementen die zichtbaar zijn

## Formulieren

:button	Alle <button> en <input type="button"> elementen
:checkbox	Alle <input type="checkbox"> elementen
:checked	Alle geselecteerde checkboxes of radio buttons
:disabled	Alle uitgeschakelde formulierelementen
:enabled	Alle ingeschakelde formulierelementen
:file	Alle <input type="file"> elementen
:image	Alle <input type="image"> elementen
:input	Alle <input>, <textarea> en <select> elementen
:password	Alle <input type="password"> elementen
:radio	Alle <input type="radio"> elementen
:reset	Alle <input type="reset"> elementen
:selected	Alle geselecteerde <option> elementen
:submit	Alle <input type="submit"> elementen
:text	Alle <input type="text"> elementen

## CSS, stijlen en dimensies

```
.css()
$('selector').css('css property');           // Getter
$('selector').css('css property', 'css value'); // Setter
```

→ werkt zowel met properties met camelCase-notatie als gewone CSS-notatie

```
.animate()
$('sel').animate({property: 'value', property: 'value'}, duur, easing, function() {
    // functie die uitgevoerd wordt als de animatie voltooid is
});
```

→ zelfde als .css(), maar past een custom animatie toe

→ meer easing-opties met de jQuery Easing-plugin (<http://easings.net/nl>)

### Klassen

```
$('selector').addClass('big');
$('selector').removeClass('big');
$('selector').toggleClass('big');
```

### Dimensies

```
$('sel').width('50px');  Setter van de breedte
$('sel').width();        Getter van de breedte
```

```
$('sel').height('50px'); Setter van de hoogte
$('sel').height();       Getter van de hoogte
```

```
$('sel').position();      Geeft de positie relatief tot de offset parent
$('sel').position().top;  Geeft de top-positie relatief tot de offset parent
$('sel').position().left; Geeft de left-positie relatief tot de offset parent
```

```
$('sel').offset();        Geeft de positie relatief tot het document
$('sel').offset().top;    Geeft de top-positie relatief tot het document
$('sel').offset().left;   Geeft de left-positie relatief tot het document
```

## Traverseren

→ methodes waarmee vanuit een jQuery-selector door het document gegaan kan worden

<code>\$('sel1').next('sel2');</code>	Eerstvolgende sibling van een element
<code>\$('sel').parent();</code>	Parent van een element
<code>\$('sel1').closest();</code>	Traverseert omhoog de DOM en geeft het eerste element die sel2 matcht
<code>\$('sel1').parents();</code>	Traverseert omhoog de DOM en geeft alle parents (die ev. sel2 matchen)
<code>\$('sel1').children();</code>	Traverseert omlaag de DOM en geeft alle children (die ev. sel2 matchen)
<code>\$('sel1').siblings();</code>	Geeft alle siblings (die ev. sel2 matchen)

`$('sel').each(function(index, element) {});`

Itereert over alle elementen die een selector matchen.

- het eerste argument van de callbackfunctie houdt de index van de iteratie bij
- het tweede argument houdt een referentie naar het gematchte element bij

## Elementen manipuleren

### Elementen creëren

```

bv. $('<p>This is a new paragraph</p>');

$('<li class="new">new list item</li>');

$('<a/>', {
  html : 'This is a <strong>new</strong> link',
  'class' : 'new',
  href : 'foo.html'
});

```

### Elementen invoegen

<code>\$('target').after('content');</code>	<i>Content</i> invoegen direct na <i>target</i>
<code>\$('content').insertAfter('target');</code>	
<code>\$('target').before('content');</code>	<i>Content</i> invoegen direct voor <i>target</i>
<code>\$('content').insertBefore('target');</code>	
<code>\$('target').append('content');</code>	<i>Content</i> invoegen als laatste child van <i>target</i>
<code>\$('content').appendTo('target');</code>	
<code>\$('target').prepend('content');</code>	<i>Content</i> invoegen als eerste child van <i>target</i>
<code>\$('content').prependTo('target');</code>	

### Elementen kopiëren

<code>\$('content').clone();</code>	Een kopie maken van een element
-------------------------------------	---------------------------------

### Elementen verwijderen

<code>\$('content').remove();</code>	Het element verwijderen + bijhorende data/events
<code>\$('content').detach();</code>	Het element verwijderen - bijhorende data/events
<code>\$('content').empty();</code>	innerHTML van een element leegmaken

## jQuery Core vs. jQuery Prototype

- ✓ **jQuery Prototype** (\$.fn namespace): methodes die worden opgeroepen op jQuery-selecties
  - ↳ bv. `$('.h1').remove();`
- ✓ **jQuery Core** (\$) namespace): methodes die worden opgeroepen zonder jQuery-selecties
  - ↳ a.k.a. 'utility methods'
  - ↳ bv. `$.trim(' blablabla ');`  
`jQuery.trim(' blablabla ');`

### Utility Methods

<code>\$.trim('string');</code>	Spaties in het begin en op het einde van de string verwijderen
<code>\$.each([], function());</code> <code>\$.each({}, function());</code>	Itereren over arrays of objecten
<code>\$.inArray(value, array)</code>	Returnt de index van <i>value</i> in <i>array</i> , of -1 indien de <i>value</i> niet bestaat
<code>\$.extend(obj1, obj2);</code>	Kopieert de attributen van <i>obj2</i> naar <i>obj1</i>
<code>\$.isFunction(value)</code>	Checkt of <i>value</i> een functie is en return een boolean
<code>\$.isArray(value)</code>	Checkt of <i>value</i> een array is en return een boolean
<code>\$.isNumeric(value)</code>	Checkt of <i>value</i> een nummer is en return een boolean
<code>\$.isPlainObject(value)</code>	Checkt of <i>value</i> een object is en return een boolean
<code>\$.isFunction(value)</code>	Checkt of <i>value</i> een functie is en return een boolean

## Events

<code>\$('.selector').on(event, function())</code>	Maakt een event handler voor alle geselecteerde elementen
<code>\$('.selector').one(event, function())</code>	Maakt een event handler die slechts 1x wordt uitgevoerd
<code>\$('.selector').off(event)</code>	Verwijdert een eerder gemaakte event handler
<code>event.preventDefault</code>	Wordt geplaatst in een handler, zorgt ervoor dat de standaardfunctionaliteit niet wordt uitgevoerd (bv. submit)

## Enkele events

<code>\$(document).ready(function());</code>	De pagina is volledig ingeladen
<code>\$('.sel').on(click, function());</code>	Er wordt geklikt op het geselecteerde element
<code>\$('.sel').on(focus, function());</code>	Een element/invoerveld krijgt focus
<code>\$('.sel').on(blur, function());</code>	Een element/invoerveld verliest focus
<code>\$('.sel').on(change, function());</code>	De waarde van een <code>&lt;input&gt;</code> , <code>&lt;textarea&gt;</code> of <code>&lt;select&gt;</code> -element verandert
<code>\$('.sel').on(submit, function());</code>	Een formulier wordt verzonden
<code>\$('.sel').on(mouseenter, function());</code>	De cursor gaat over het geselecteerde element
<code>\$('.sel').on(mouseleave, function());</code>	De cursor verlaat het geselecteerde element
<code>\$('.sel').on(hover, function());</code>	Shorthand voor mouseenter/leave ↳ 1 functie meegeven = voor beide events ↳ 2 functies meegeven = 1 per event
<code>\$('.sel').on(toggle, function());</code>	Wisselt af van functie elke keer een click-event voorkomt ↳ minimum 2 functies moeten meegegeven worden
<code>\$('.sel').on(keydown, function(e) {     // De keycode van de toets     e.which });</code>	Een toets wordt ingedruwd

## Event handler-object

<code>\$('.sel').on(event, data, function(e) {</code>	
<code>e.pageX</code>	De x-positie van de muis op het moment van het event
<code>e.pageY</code>	De y-positie van de muis op het moment van het event
<code>e.type</code>	Het type van het event (bv. click)
<code>e.which</code>	De (muis)knop die werd ingedrukt
<code>e.data</code>	Data die werd meegestuurd met het event
<code>e.target</code>	Het DOM-element die het event heeft veroorzaakt
<code>e.preventDefault</code>	Voorkomt dat de standaardfunctionaliteit wordt uitgevoerd
<code>}</code>	

## Effects

<code>\$('#sel').show();</code>	Een verborgen element weergeven
<code>\$('#sel').hide();</code>	Een element verbergen
<code>\$('#sel').fadeIn();</code>	De opacity van een selectie animeren naar 100%
<code>\$('#sel').fadeOut();</code>	De opacity van een selectie animeren naar 0%
<code>\$('#sel').slideDown();</code>	Een verborgen selectie weergeven met een slide-beweging
<code>\$('#sel').slideUp();</code>	Een verborgen element weergeven
<code>\$('#sel').slideToggle();</code>	Een verborgen element weergeven

## De snelheid van een effect aanpassen

Als argument van een functie

bv. `$('#sel').fadeIn('slow'|300);`

## Standardsnelheden aanpassen

```
jQuery.fx.speeds: {
  // Standardsnelheid en -keywords aanpassen
  slow: 600,
  fast: 200,
  _default: 400

  // Zelf keywords definiëren
  supersnel: 100
}
```

## Functie uitvoeren als de animatie voltooid is

Als argument van een functie

bv. `$('#sel').fadeIn('slow', function());`

## Zelf animaties maken met `$.fn.animate`:

→ zie hoofdstuk CSS

## Animaties beheren

<code>\$('#sel').stop();</code>	Actieve animaties stoppen
<code>\$('#sel').delay();</code>	Een aantal milliseconden wachten tot de volgende animatie w uitgevoerd
<code>jQuery.fx.off = true;</code>	Globale optie om animaties aan/uit te zetten