

## 16. JQUERY

jQuery is een populaire JavaScriptbibliotheek (*library*). De bedoeling ervan is, om de JavaScriptcode op je webpagina's te vereenvoudigen, o.a. door vele regels JavaScript voor bv. een animatie samen te vatten tot één regel.

jQuery wordt voornamelijk gebruikt om

- elementen in de DOM-boomstructuur eenvoudig te selecteren;
- CSS-manipulaties via JavaScript te vereenvoudigen;
- effecten en animaties snel en efficiënt toe te passen;
- snel en eenvoudig met AJAX te werken;
- de logge cross-browsers scripts uit de wereld te helpen.

### 16.1 JQUERY OP JE WEBSITE

Om te beginnen download je de *development version* van [jquery.com/download](http://jquery.com/download).

Zet ze in een handige (sub)map, pas de naam aan tot `jquery.js` en dan kan je ernaar refereren vanuit de head:

```
<script src="[submap]/jquery.js"></script>
```

Is je website klaar, dan heb je twee mogelijkheden:

- vervang de development version door de *production version*, die is gecomprimeerd.
- verwijst naar een online library, bv. die van Google:

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>
```

En dan kan je een eerste script uitproberen.

```
<!DOCTYPE html>
<html>
  <head>
    <title>jQuery voorbeeld 1</title>
    <meta charset="UTF-8" />
    <script src="scripts/jquery.js"></script>
    <script type="text/javascript">
      $(document).ready(function() {
        $('#output').html("<h1>Hello world</h1>");
      });
    </script>
  </head>
  <body>
    <div id="output"> </div>
  </body>
</html>
```

In plaats van een anonieme functie kan je ook een benoemde gebruiken:

```
$(document).ready(toonHelloWorld);
function toonHelloWorld() {
  $('#output').html("<h1>Hello world</h1>");
}
```

Zet best altijd de code binnen of na een `$(document).ready()`, dan ben je zeker dat de pagina volledig geladen is vooraleer de JavaScript-code geladen en uitgevoerd wordt.

## 16.2 SELECTORS

In plaats van `document.getElementById("ID")` biedt jQuery je verscheidene handige methoden om gegevens te selecteren.

De basis syntax is: `$(selector).action();`

Waarbij selector kan zijn:

Selector	Voorbeeld	Betekenis
HTML-element	<code>\$("a")</code>	alle links, d.w.z. elementen <code>&lt;a&gt;</code>
#ID	<code>\$("#test")</code>	het element met ID "test"
.class	<code>\$(".donker")</code>	alle elementen met class "donker"
*	<code>\$("*")</code>	alle HTML-elementen
this	<code>\$(this)</code>	het geselecteerde HTML-element
combinatie	<code>\$("section.links")</code>	alle <code>&lt;section&gt;</code> -elementen met class "links"
	<code>\$("tr:odd")</code>	alle oneven rijen in een tabel (even = <code>tr:even</code> )
afstammeling	<code>\$("div a")</code>	alle links (elementen <code>&lt;a&gt;</code> ) die in een <code>&lt;div&gt;</code> staan
	<code>\$("ul li:first")</code>	het eerste <code>&lt;li&gt;</code> -element van de eerste <code>&lt;ul&gt;</code>
attribuut	<code>\$("[href]")</code>	alle elementen met een attribuut <code>href=</code>
	<code>\$("input[name]")</code>	alle input-elementen met attribuut <code>name=</code>
	<code>\$("input[name='post']")</code>	het input-element met <code>name="post"</code>

Komt het je bekend voor? Dit is inderdaad grotendeels dezelfde manier als de CSS-selectors.

## 16.3 EVENT HANDLING

De events in jQuery komen zowat overeen met die in JavaScript zelf.

Hieronder een overzicht van veelgebruikte events:

<code>\$(document).ready()</code>	als de pagina geladen is
<code>\$(window).resize()</code>	als het browservenster van formaat veranderd wordt
<code>\$(window).scroll()</code> of <code>\$(element).scroll()</code>	de gebruiker scrollt in het venster of in een HTML-element
<code>\$(element).click()</code>	klik op het HTML-element
<code>\$(element).dblclick()</code>	dubbeltklik op het HTML-element
<code>\$(element).focusin()</code>	de gebruiker klikt in/op het element of activeert het met Tab
<code>\$(element).focusout()</code>	het element verliest de focus
<code>\$(element).hover()</code>	de muiscursor beweegt over het element; combinatie van <code>mouseenter()</code> en <code>mouseleave()</code>
<code>\$(element).mousedown()</code>	een muisknop wordt ingedrukt op een element
<code>\$(element).mouseup()</code>	een muisknop wordt losgelaten boven een element
<code>\$(element).mouseenter()</code>	de muiscursor komt op het element, begint erover te bewegen
<code>\$(element).mouseleave()</code>	de muiscursor verlaat het element
<code>\$(element).mouseover()</code>	de muiscursor komt op het element of een kindelement
<code>\$(element).mouseout()</code>	de muiscursor verlaat het element of een kindelement
<code>\$(element).mousemove()</code>	de muiscursor beweegt boven het element

Specifiek voor formulierelementen:

<code>\$(input).focus()</code>	een inputveld krijgt de focus
<code>\$(input).blur()</code>	het inputveld verliest de focus
<code>\$(select).change()</code>	de waarde van het element werd aangepast; werkt met <code>&lt;input&gt;</code> , <code>&lt;select&gt;</code> en <code>&lt;textarea&gt;</code>
<code>\$(input).keydown()</code>	een toets wordt ingedrukt in een inputveld
<code>\$(input).keypress()</code>	gebeurt net na <code>keydown()</code>
<code>\$(input).keyup()</code>	een toets wordt losgelaten in een inputveld
<code>\$(input).select()</code>	de gebruiker selecteert tekst in een tekstveld of een textarea
<code>\$("form").submit()</code>	het formulier wordt verstuurd, er wordt op de submit-knop geklikt

Tussen de haakjes komt dan ofwel een functienaam ofwel een anonieme functie:

```
$(selector).action( function() {
    //actie(s);
});
```

Bovendien kan je met **on()** één of meer acties aan een element binden.

één actie, bv. verberg het `<p>`-element als erop geklikt wordt:

```
$("p").on("click", function(){
    $(this).hide();
});
```

Meer acties, bv. verschillende opmaak:

```
$("p").on({
    mouseenter: function(){
        $(this).css("background-color", "lightgray");
    },
    mouseleave: function(){
        $(this).css("background-color", "lightblue");
    },
    click: function(){
        $(this).css("background-color", "yellow");
    }
});
```

## 16.4 INTERACTIE MET DOM

jQuery bevat verscheidene methoden om elementen uit de HTML-DOM aan te spreken en aan te passen. We bekijken weer de meest gebruikte.

Allereerst gegevens opvragen of aanpassen:

<code>\$(element).text()</code>	geef de waarde van een HTML-element
<code>\$(element).text("waarde")</code>	geef het HTML-element een nieuwe waarde
<code>\$(element).html()</code>	geef de waarde van een HTML-element, inclusief kindelementen
<code>\$(element).html("tekst")</code>	geef het HTML-element een nieuwe waarde, inclusief HTML-tags
<code>\$(formelement).val()</code>	geef de waarde uit een formulierelement
<code>\$(formelement).val("waarde")</code>	geef het formulierelement een nieuwe waarde
<code>\$(element).attr("attrNaam")</code>	geef de waarde uit het gevraagde attribuut
<code>\$(element).attr("attr-Naam", "waarde")</code>	geef het attribuut de opgegeven waarde

Ook is het mogelijk om HTML-elementen toe te voegen:

<code>\$(element).append("tekst")</code>	voegt tekst toe aan het element of (met tags erbij) voegt een element toe achter het bestaande
<code>\$(element).prepend("tekst")</code>	voegt tekst toe vóór het element of (met tags erbij) voegt een element toe vóór het bestaande

In plaats van `append()` kan je ook `after()` gebruiken en `before()` in plaats van `prepend()`.

Of wil je elementen verwijderen?

<code>\$(element).remove()</code>	verwijdert het element en alle kindelementen
<code>\$(element).remove(".test, .demo")</code>	verwijder het element als het behoort tot class test of demo
<code>\$(element).empty()</code>	maak het element leeg, d.w.z. verwijder alle kindelementen

Uiteraard kan je ook de stijl aanpassen:

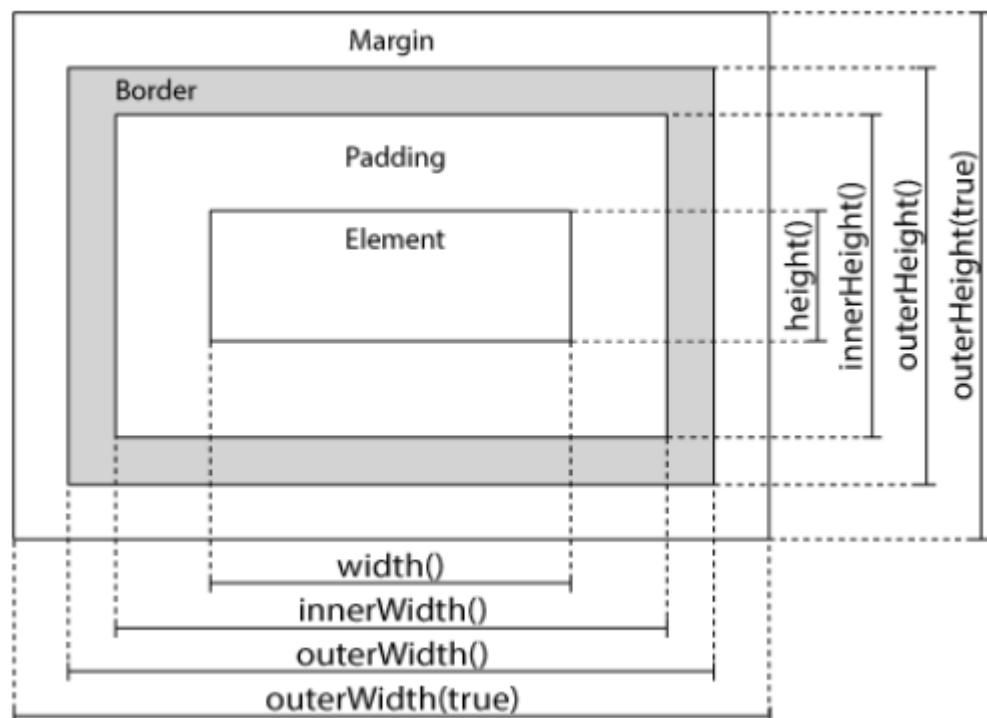
<code>\$(element).css("eigenschap")</code>	geef de waarde van "eigenschap"
<code>\$(element).css("eigenschap", "waarde")</code>	geef eigenschap de nieuwe waarde
<code>\$(element).addClass("classNaam")</code>	geef het element de (bestaande) class classNaam
<code>\$(element).removeClass("classNaam")</code>	verwijder class classNaam van het element
<code>\$(element).toggleClass("classNaam")</code>	voeg ofwel class classNaam toe aan element of verwijder ze

Je kan ook meer CSS-eigenschappen tegelijk instellen, bv. voor alle elementen `<p>`:

```
$("p").css({"background-color": "yellow", "font-size": "200%"});
```

Verder is het belangrijk om de instellingen voor hoogte en breedte goed te begrijpen:

- `width()` en `height()` van het element zelf;
- `innerWidth()` en `innerHeight()` = element + padding;
- `outerWidth` en `outerHeight` = element + padding + border;
- `outerWidth(true)` en `outerHeight(true)` = element + padding + border + margin



## 16.5 JQUERY EFFECTEN

Met jQuery is het erg eenvoudig om verscheidene animaties op elementen toe te passen.

Elementen laten **verdwijnen en verschijnen** bijvoorbeeld:

```
<!DOCTYPE html>
<html>
  <head>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/
jquery.min.js"></script>
    <script>
      $(document).ready(function() {
        $("#verdwijn").click(function() {
          $("p").hide();
        });
        $("#toon").click(function() {
          $("p").show();
        });
      });
    </script>
  </head>
  <body>
    <p>Als je op de verdwijnpknop klikt, eh ... verdwijnt ik.</p>
    <button type="button" id="verdwijn">Verdwijn</button>
    <button type="button" id="toon">Verschijn</button>
  </body>
</html>
```

In plaats van `hide()` en `show()` kan je ook functie `toggle()` gebruiken, om te wisselen tussen verdwijnen en verschijnen.

Bovendien kan je een vertraging toevoegen: "slow", "fast" of aantal milliseconden.

Een variant op verschijnen of verdwijnen met vertraging is het effect **fade in/out**.

Een voorbeeld van langzaam laten verschijnen:

```
<style>
  #div1 {width:80px;height:80px;display:none;background-color:red;}
  #div2 {width:80px;height:80px;display:none;background-color:green;}
  #div3 {width:80px;height:80px;display:none;background-color:blue;}
</style>
<script>
  $(document).ready(function() {
    $("button").click(function() {
      $("#div1").fadeIn();
      $("#div2").fadeIn("slow");
      $("#div3").fadeIn(3000);
    });
  });
</script>
</head>
<body>
  <p>Demonstratie van fadeIn() met verschillende parameters</p>
  <button type="button">Laat de vierkanten verschijnen</button><br /><br />
  <div id="div1"></div><br />
  <div id="div2"></div><br />
  <div id="div3"></div>
```

De mogelijkheden zijn:

<code>fadeIn()</code>	langzaam laten verschijnen
<code>fadeOut()</code>	langzaam laten verdwijnen

fadeToggle()	wissel tussen fadeIn en fadeOut
fadeTo(snelheid, doorzichtigheid)	vervaag tot een opgegeven doorzichtigheid tussen 0 en 1

Je kan telkens een snelheid toevoegen, nl. "slow", "fast" of een waarde in milliseconden.

Nog een manier van verschijnen is het element naar beneden laten openvallen - en dan weer naar boven toe dichtvouwen: **slideDown()** en **slideUp()**.

slideDown()	laat openvallen
slideUp()	vouw naar boven toe
slideToggle()	wissel tussen openvallen en toevouwen

Ook hier kan je telkens een snelheid toevoegen: "slow", "fast" of een waarde in milliseconden.

En dan is er nog method **animate()** om CSS-eigenschappen te animeren:

```
$(element).animate({CSSeigenschap: "waarde"}, snelheid);
```

De snelheid is optioneel.

Je kan zowat alle CSS-eigenschappen animeren, behalve color.

Bij eigenschappen die uit twee woorden bestaan, gebruik je camelCase, bv: paddingLeft

Je bent ook niet beperkt tot één effect, je kan er verscheidene na elkaar laten uitvoeren

- met **callback**: voeg als laatste parameter in een functie een nieuwe functie toe
- met **chaining**: door verscheidene functies na elkaar te zetten, telkens met een punt tussen.

Een voorbeeld van chaining: maak de paragraaf eerst rood, vouw hem dan toe en vervolgens open.

```
$("#p1").css("color", "red").slideUp(2000).slideDown(2000);
```

Nuttige links:

- tutorial op W3Schools: <http://www.w3schools.com/jquery/default.asp>
- uitgebreide jQuery API documentatie: <http://api.jquery.com/>
- meer info over selectors: <http://code.tutsplus.com/tutorials/the-30-css-selectors-you-must-memorize--net-16048>
- Oefenen van jQuery selectors: <http://codylindley.com/jqueryselectors/>

Op Pluralsight:

- [jQuery: getting started](#)
- [jQuery Fundamentals](#)
- [JavaScript: Advanced Fundamentals to jQuery & Pure DOM Scripting](#)

Op Codeschool: [Try jQuery](#)