

WAT IS JQUERY

JAVASCRIPT

1

“**jQuery** is a multi-browser JavaScript library designed to simplify the client-side scripting of HTML.”

Bron: Wikipedia

jQuery

is een JavaScript bibliotheek
versimpelt op die manier het schrijven van JavaScript
is snel te leren en gemakkelijk te gebruiken

alle voorbeelden uit de slides:

<https://bitbucket.org/PetervandenHeuvel/jquery-voorbeelden-slides/downloads/>

VOORDELEN BETER DAN JAVASCRIPT ?

2

jQuery **is** JavaScript (een beknopte en dus snelle bibliotheek) maar versimpelt event handling, animatie en AJAX interactie.

+

- cross-browser compatibel
- klein en compact (zelfs via embedding)
- steile leercurve en veel online documentatie
- plugins en jQuery UI
- CCS3 selectors
- open source

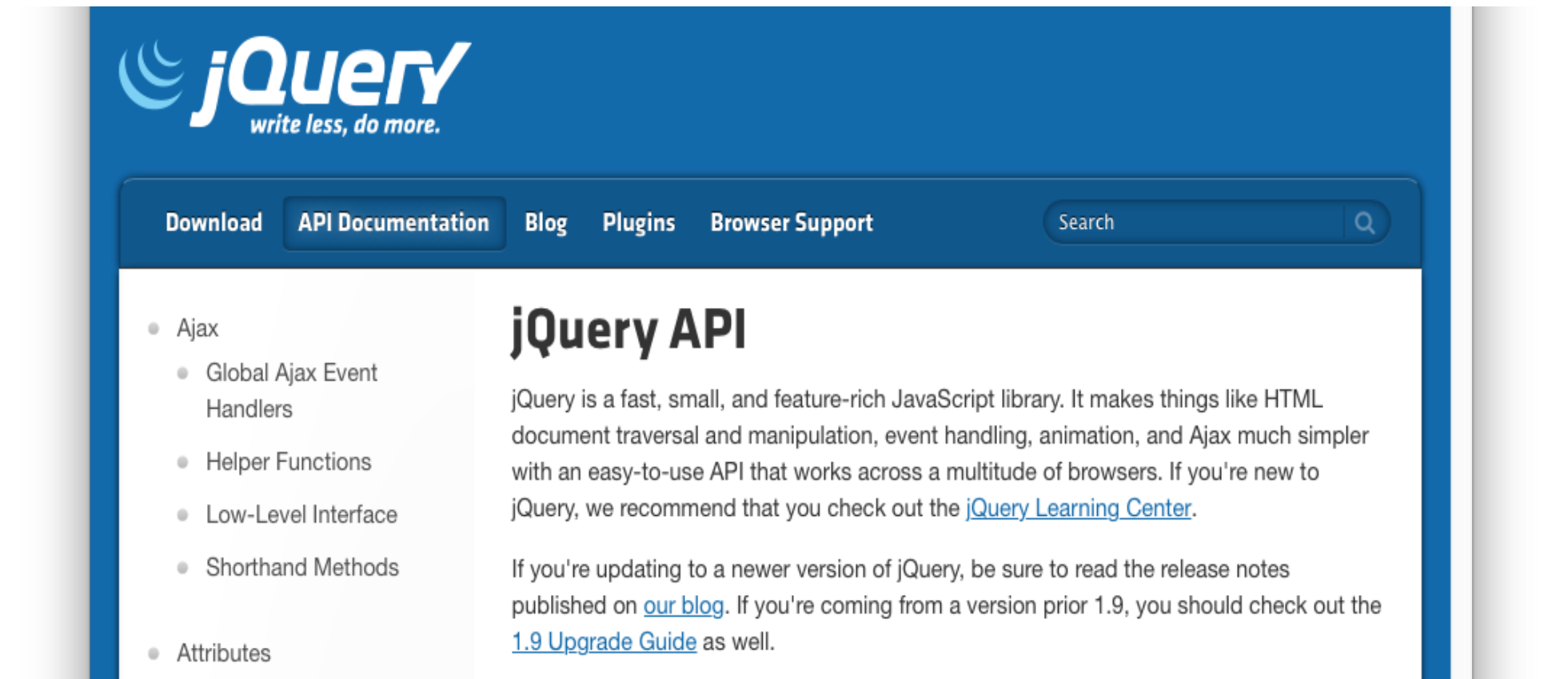
DOCUMENTATIE

WAAR KAN JE MEER INFORMATIE VINDEN?

3

Het is niet mogelijk om alles over **jQuery** in deze slides te verzamelen...

De meest relevante informatie vind je op: <http://api.jquery.com>



The screenshot shows the jQuery API documentation website. The header is blue with the jQuery logo and the tagline "write less, do more.". Below the header is a navigation bar with links: "Download", "API Documentation" (which is highlighted), "Blog", "Plugins", and "Browser Support". There is also a search bar on the right. The main content area has a sidebar on the left with a list of topics: "Ajax", "Global Ajax Event Handlers", "Helper Functions", "Low-Level Interface", "Shorthand Methods", and "Attributes". The main content area is titled "jQuery API" and contains text describing the library as a fast, small, and feature-rich JavaScript library. It mentions that it makes HTML document traversal and manipulation, event handling, animation, and Ajax much simpler. It also recommends checking out the "jQuery Learning Center" for new users and the "1.9 Upgrade Guide" for users updating from previous versions.

jQuery
write less, do more.

Download API Documentation Blog Plugins Browser Support Search

- Ajax
 - Global Ajax Event Handlers
 - Helper Functions
 - Low-Level Interface
 - Shorthand Methods
- Attributes

jQuery API

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 a multitude of browsers. If you're new to jQuery, we recommend that you check out the [jQuery Learning Center](#).

If you're updating to a newer version of jQuery, be sure to read the release notes published on [our blog](#). If you're coming from a version prior 1.9, you should check out the [1.9 Upgrade Guide](#) as well.

STARTEN MET JQUERY

LINK EN DOWNLOAD



Stap1: <http://jquery.com/>
Stap2: download / kopiëren
Stap3: linken voor de `</body>`

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8" />
<title>Hello</title>
</head>

<body>
  <h1>End</h1>
  <p>Ga naar jQuery.com en download daar de jquery.js file, plaats deze in de /js map.</p>
  <p>Vervolgens koppel je voor je eigen ./main.js bestand een script-tag.</p>

  <script src="./js/jquery-3.4.1.js"></script>
  <script src="js/main.js"></script>
</body>
</html>
```

STARTEN MET JQUERY

CDN

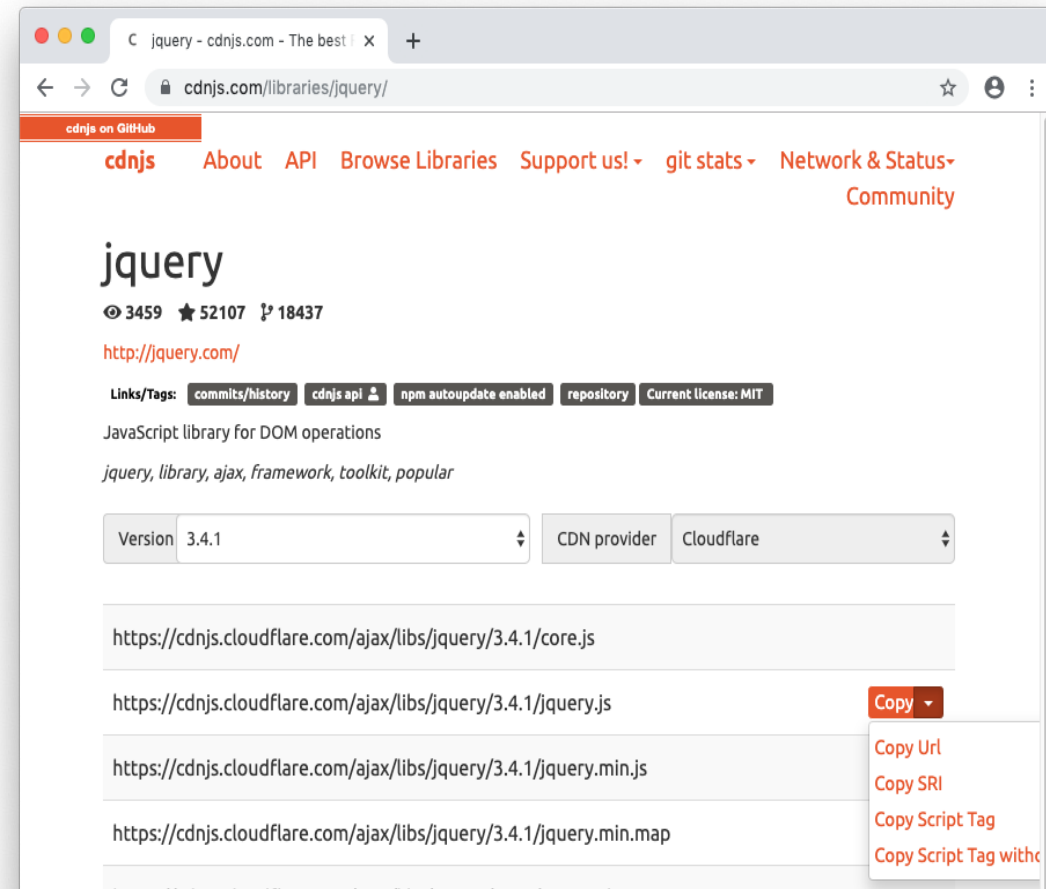
5

Stap1: <https://cdnjs.com>

Stap2: Selecteer jQuery-script tag

Stap3: Koppelen voor `</body>`

of gebruik cdnjs-plugin in Visual Studio Code



```
<script src='https://cdnjs.cloudflare.com/ajax/libs/jquery/3.4.1/jquery.min.js'></script>
```

STARTEN MET JQUERY

LOKAAL & CDN



Het is ook mogelijk om beide methodes te gebruiken:

- Eerst controleren of **jQuery** geladen is via de CDN.
- Als dat niet werkt een lokaal gedownload versie gebruiken.
- Werkend voorbeeld in map: / **02_CDN & Lokaal**

```
<!-- First try to load jQuery from the internet using a CDN -->
```

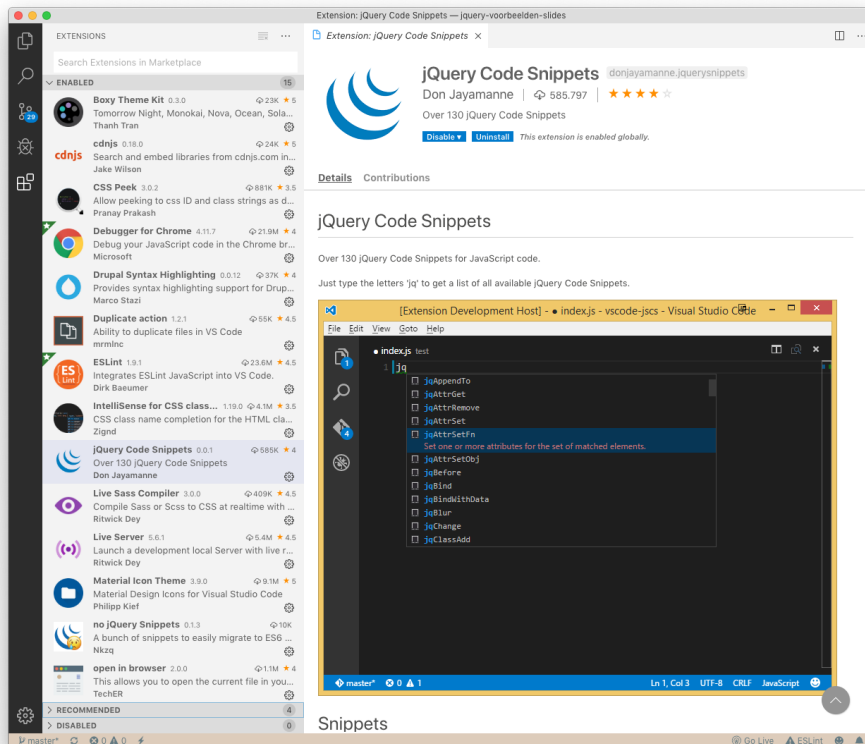
```
<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
```

```
<!-- Second: Check if jQuery is chained to window-object, if it isn't use javascript to print <script> tag. -->
```

```
<script>window.jQuery || document.write('<script src="js/jquery-3.4.1.js"></script>')</script>
```

STARTEN MET JQUERY PLUGIN VOOR VSCODE

7



Voorbeelden:

Start met het typen van 'jq' in een .js-bestand en vul dan aan met de functie die je zoekt:

bijv. 'jqdoc' voor document-ready, of 'jqfa' voor alle 'fade-functies'. Het werkt met slim aanvullen.

slim aanvullen

jqdo

- ☐ jqDocReady
- ☐ jqDocReadyShort
- ☐ jqKeyDown
- ☐ jqDataSetObj
- ☐ jqMouseDown
- ☐ jqSlideDown
- ☐ jqSlideDownFull
- ☐ jqOneWithData
- ☐ jqDataRemove
- ☐ jqFadeOut
- ☐ jqFadeOutFull

Function to execute w
fully loaded. (jQuery
s)
\$(document).ready(fun
});
uitleg

OPDRACHT # 1

MAAK EEN SIMPELE HTML-PAGINA

8

Oefening 1 - Koppelen jQuery met een CDN

Stap1: Maak een leeg HTML5 document

Stap2: Download jQuery

Stap3: Koppel een CDN

Stap4: test in je script: `$("p").css('border', '1px solid red');`

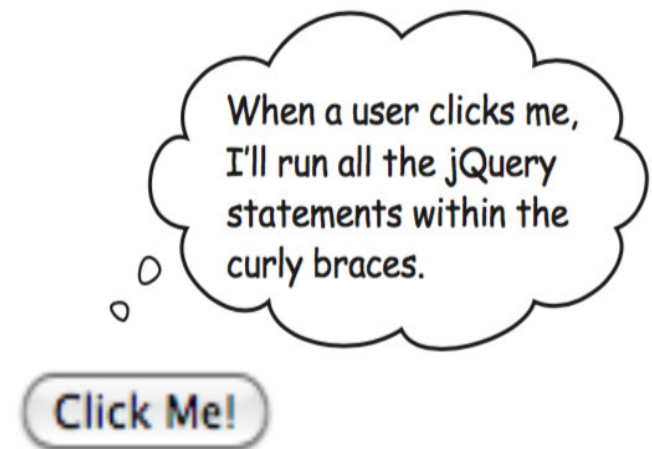
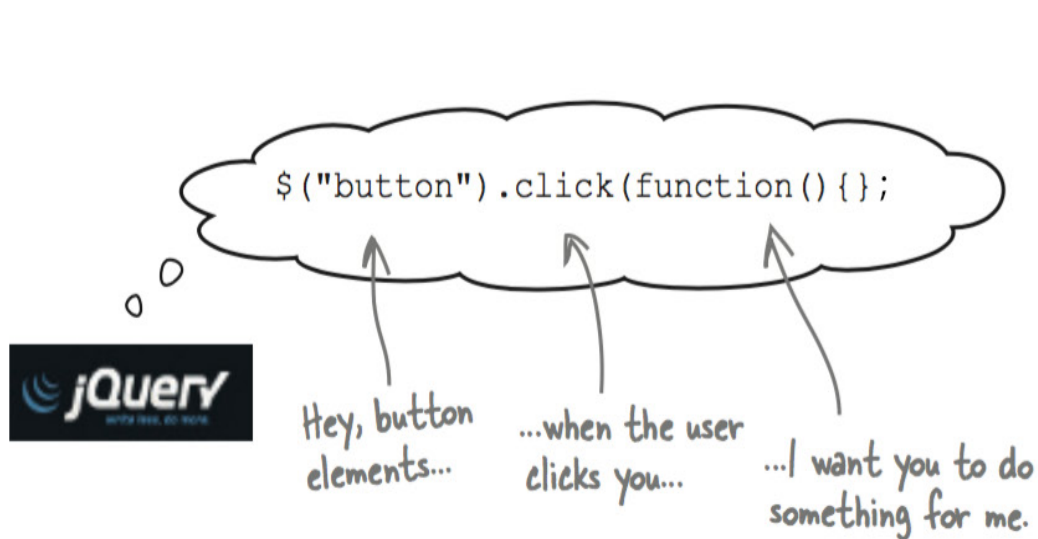
Oefening 2 - Lokale variant

Maak een tweede variant met enkel een lokale versie van jQuery, die je download van de website van jquery.com

VOORBEELD 1

SYNTAX

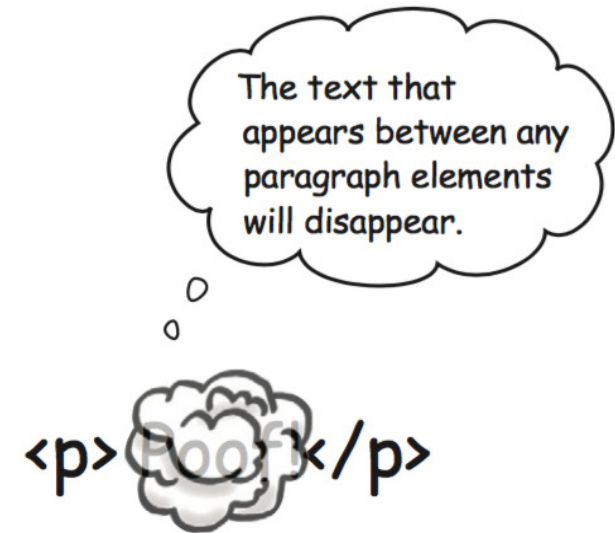
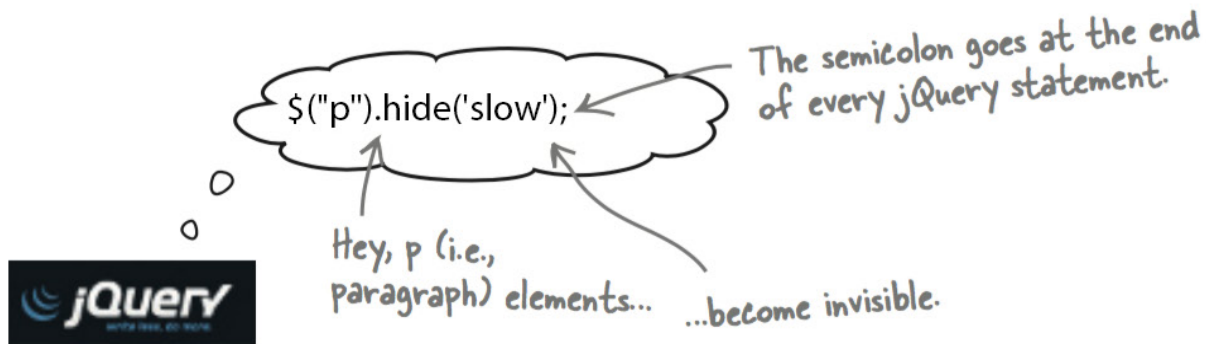
a



VOORBEELD 2

SYNTAX

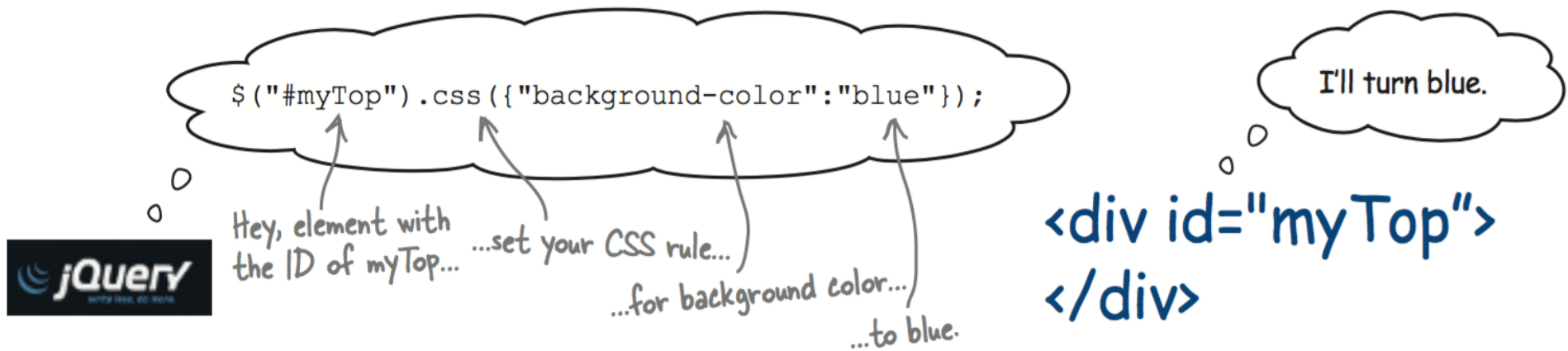
10



VOORBEELD 3

SYNTAX

11



\$ (selector) . action ()

↓ ↓ ↓

1 2 3

“**Syntax** is the study of the principles and processes by which sentences are constructed in particular languages.”

Bron: Wikipedia

1. Dit teken geeft weer dat we **jQuery** gebruiken
2. Een **selectie** van een HTML element adhv CSS selectors
3. Een **jQuery actie** die op dit element wordt toegepast

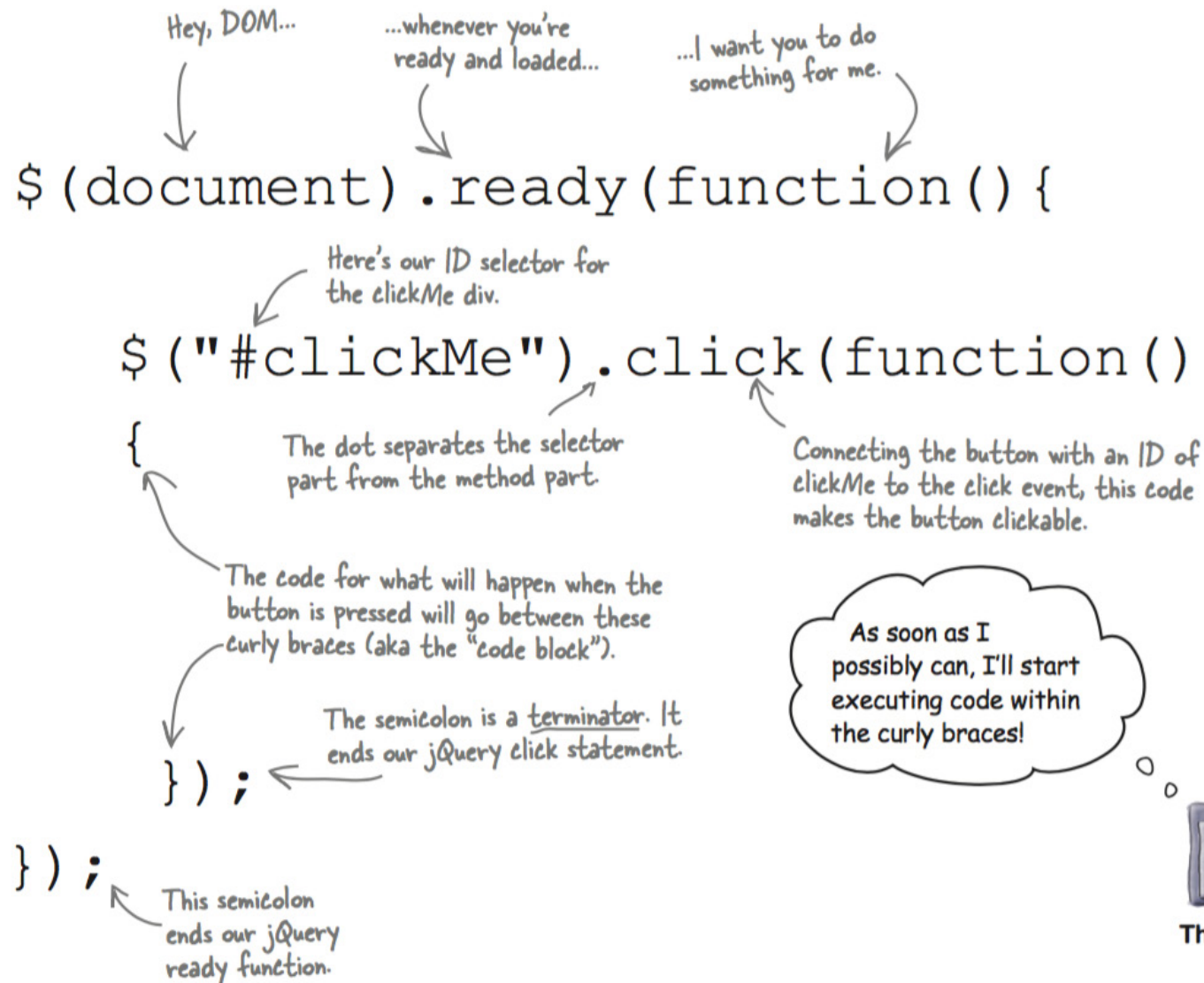
```
$(document).ready(function(){  
    // hier komen jQuery functies te staan...  
});
```

Het uitvoeren van **jQuery** code zou pas morgen gebeuren als het document daar volledig klaar voor is. Mogelijke problemen:

→ elementen manipuleren die nog niet bestaan
informatie opvragen van nog niet bestaande elementen
waarom nodig?

VISUEEL: DOCUMENT.READY SYNTAX

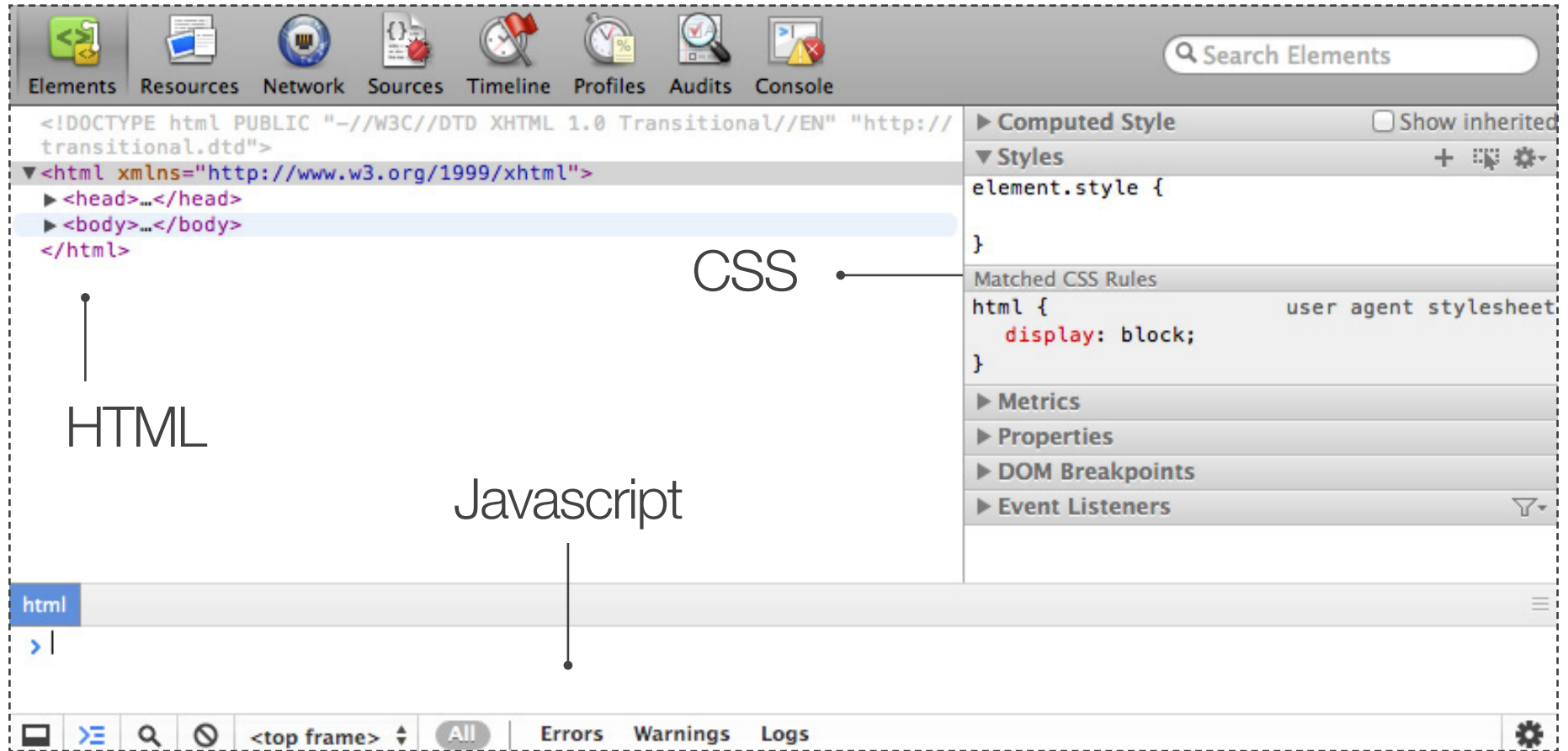
11



JQUERY IN DE CONSOLE

EERSTE STAP ALS IETS NIET WERKT

15



console

Werkt je jQuery code niet? Stap 1: controleer je console.

VOORBEELD VAN ERROR CONSOLE

16



```
</head>
<body>
  <!-- partial:index.partial.html -->
  <h1>jQuery doesn't work</h1>
  <!-- partial -->
  <script src="./script.js"></script>
</body>
```

stom: jquery wordt niet geladen, dus: error!

<https://codepen.io/petervandenheuvel/pen/oNveXMz>

JQUERY SELECTORS 1/7

CSS SELECTORS

17

`$ (selector) . action ()`
 ↑

1. Selectie van het te manipuleren element
2. Ondersteunt CSS selectors en eigen selectors

`$('*')`

`$('h1')`

`$('h3.class')`

`$('h5#id')`

`$('button')`

`$(':even')`

`$(':odd')`

`$('parent child')`

`$('[href]')` with attribute href

`$('[href=#]')` with attribute value #

`$('[src$='.jpg]')` ending .jpg

`$('div#box.doos')`

both this # and classname

bron: http://www.w3schools.com/jquery/jquery_ref_selectors.asp

JQUERY SELECTORS 2/7

SELECTEREN VIA EEN TAG (ELEMENT)

18

Javascript

```
document.getElementsByTagName("naamtag")
```

jQuery

```
$("naamtag")
```

Bijv.: `$("div").css("border", "3px solid red");`

JQUERY SELECTORS 3/7

SELECTEREN VIA EEN CLASS

19

JavaScript

```
document.getElementsByClassName('naam-van-de-class')
```

jQuery

```
$('.naam-van-de-class')
```

voorbeeld

```
$('.doos').css('width', '250px');
```

JQUERY SELECTORS 4/7

SELECTEREN VIA EEN ID

20

JavaScript

```
document.getElementById("uniekeNaam")
```

jQuery

```
$("#uniekeNaam")
```

voorbeeld

```
$("#menu").css({"width": "250px", "height":  
"100px", "color": "#00FF00"});
```

JQUERY SELECTORS 5/7

GEAVANCEERDE SELECTORS

21

1. parent-child

`$("#doos p")`

2. direct decendent

`$("#doos > p")`

3. meerdere elementen

`$("h1, h3, h3, .bold, #footer")`

4. filters

`$("ul li:first") of $(".inhoud:has(p)")`

5. attribute

`$('div[id="uniekeNaam"]')`

JQUERY SELECTORS 6/7

FILTER SELECTORS

22

:even en :odd

:header

:eq(x), :gt(x), :lt(x)

:first-child, :last, :only

:has(element)

:contains('deze inhoud')

:empty

:parent

:hidden, :visible

alle even of oneven elementen

alle h1, h2, h3, h4, h5 en h6 tags

elementen gelijk aan of < of > getal

eerste, laatste of enig kind

bevat specifiek element

bevat deze exacte inhoud

is leeg

is een 'parent' element (ouder)

is verborgen of zichtbaar

Tip: Raadpleeg jQuery.com of W3C-schools voor exacte werking en voorbeelden.

JQUERY SELECTORS 7/7

ATTRIBUTE SELECTORS

23

`$("[attribute*='waarde']")`

bevat een deel van de waarde

`$("[attribute = 'waarde']")`

start met of bevat de waarde

`$("[attribute~='waarde']")`

bevat een deel van de waarde (spaties)

`$("[href$='.jpg']")`

eindigt met waarde

`$("[href='default.htm']")`

gelijk aan de waarde 'default.html'

`$("[title^='Tom']")`

start met de waarde 'Tom'

`$("[href!='.png']")`

bevat de waarde niet

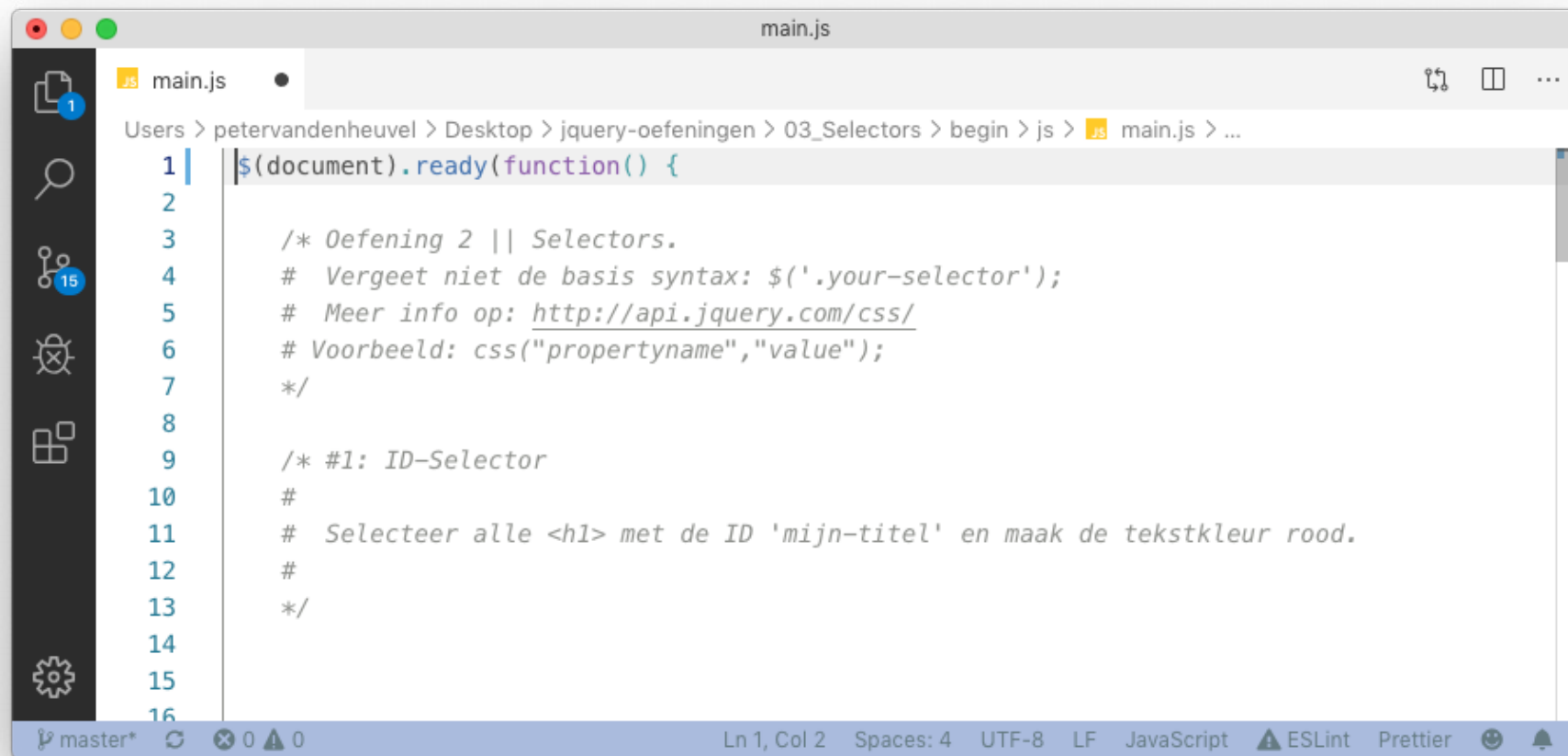
Tip: http://www.w3schools.com/jquery/jquery_ref_selectors.asp

Aanvullend thema: DOM Traversing. (Zie laatste slides)

OEFENING 3

AAN DE SLAG MET SELECTORS

21

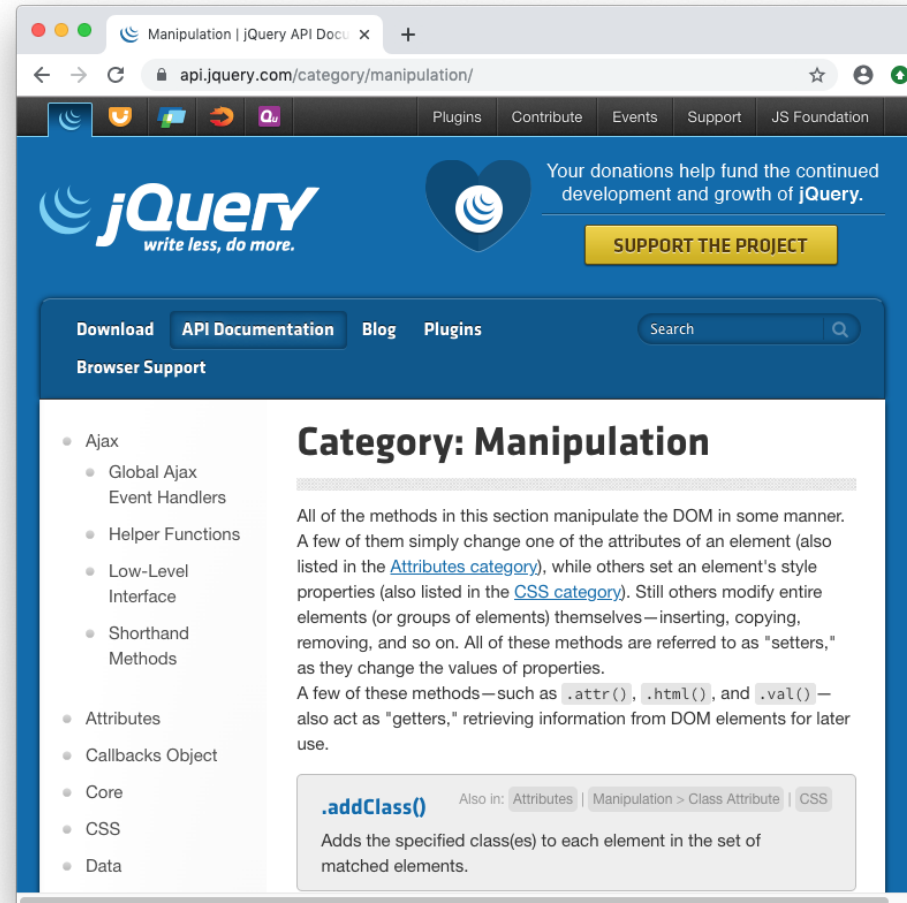


```
main.js
Users > petervandenheuvel > Desktop > jquery-oefeningen > 03_Selectors > begin > js > main.js > ...
1 | $(document).ready(function() {
2 |
3 |     /* Defening 2 || Selectors.
4 |     # Vergeet niet de basis syntax: $('your-selector');
5 |     # Meer info op: http://api.jquery.com/css/
6 |     # Voorbeeld: css("propertyname","value");
7 |     */
8 |
9 |     /* #1: ID-Selector
10 |    #
11 |    # Selecteer alle <h1> met de ID 'mijn-titel' en maak de tekstkleur rood.
12 |    #
13 |    */
14 |
15 |
16 |
```


DOM MANIPULATION

API.JQUERY.COM/CATEGORY/MANIPULATION/

25



JQUERY EN INHOUD 1/3

.(INNER) HTML

26

<https://codepen.io/petervandenheuvel/pen/OJLjzmP?editors=1010>

.html behandelt inhoud als HTML

.text behandelt inhoud als tekst

Beiden gebruik je ze om inhoud in de DOM te plaatsen, maar het is op een iets andere manier.

```
<!-- .html() -->
<div id="inhoud"><div class="tekst">error</div></div>

<!-- .text() -->
<div id="inhoud2">&lt;div class="tekst"&gt;error&lt;/div&gt;</div>

<p id="inhoud3">inhoud3</p>
```

Wat is het verschil?

Met .html kan je ook de tags veranderen waarin het oorspronkelijke element in zat. Je kan een <p> selecteren en veranderen naar een <h2> bijvoorbeeld.

html()

```
$("#inhoud").html('<div class="error">error</div>');
```

text()

```
$("#inhoud").text('<div class="error">error</div>');
```

JQUERY EN INHOUD 2/3

.(INNER) HTML

27

<https://codepen.io/petervandenheuvel/pen/vYBJpdq>

Onthouden: binnen het geselecteerde element



```
<div id="inhoud"> text ervoor inhoud text erachter </div>
```

prepend()

```
$("#inhoud").prepend('<p>Text erbij</p>');
```

append()

```
$("#inhoud").append('<p>Text erbij</p>');
```

JQUERY EN INHOUD 3/3

.(INNER) HTML

28

<https://codepen.io/petervandenheuvel/pen/aboyEYj>

Onthouden: buiten het geselecteerde element

```
text ervoor
  <div id="inhoud">inhoud</div>
text achter

<div id="inhoud2">inhoud2</div><h1>text erna</h1>
```

before()

```
$("#inhoud").before('</p>Text ervoor</p>');
```

after()

```
$("#inhoud").after('</p>Text achter</p>');
```

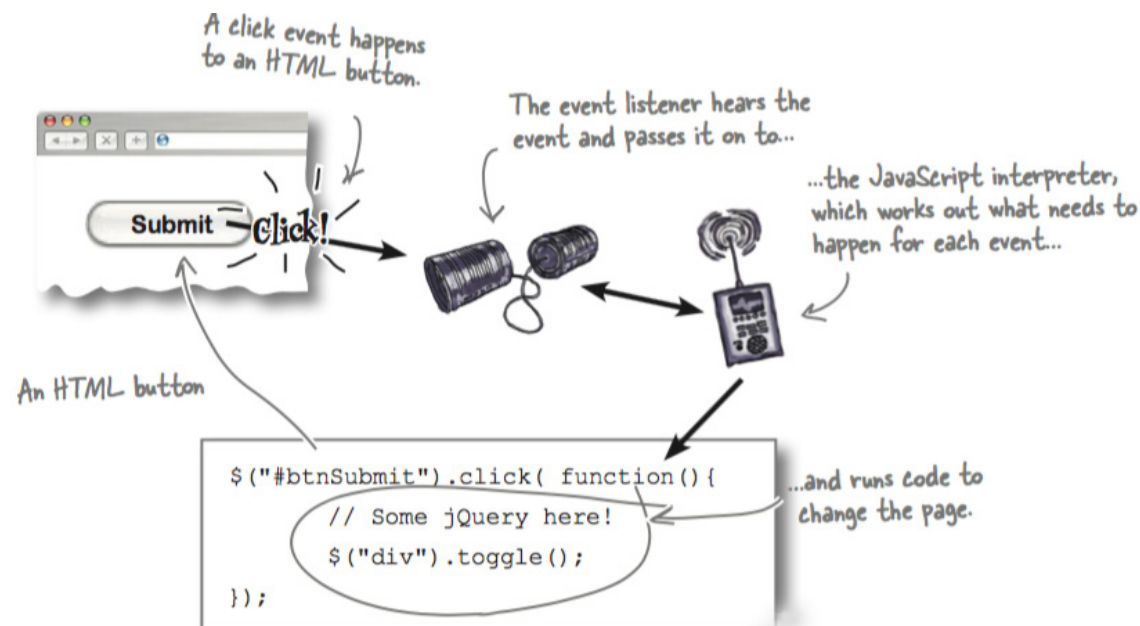
JQUERY EVENTS

REAGEREN OP ACTIES

29

We hebben nu enkel traditionele scripts gezien: top-to-bottom.

In de praktijk zullen we **acties** vaak pas willen uitvoeren als er iets gebeurt op de pagina: de gebruiker **drukt op een knop**, de **input van een text-veld wordt ingevuld** of de **gebruiker scroll door de pagina**.



JQUERY SELECTORS

EVENTS

30

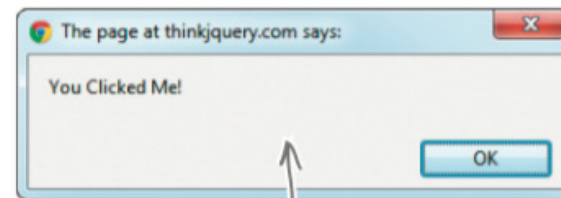
Event Listeners zijn 'telefoontjes' waarmee de browser probeert te luisteren naar wat een gebruiker doet op een pagina, en als ze '**gebeld**' worden om javascript een actie te laten uitvoeren.



jQuery maakt het veel eenvoudiger om voor alle browsers event-listeners te maken, op wat voor element dan ook. Op deze manier kan alles klikbaar worden; niet alleen links en buttons!

An element with the ID of showMessage. → `$("#showMessage").click(function() {
alert('You Clicked Me!'); ← Run this code.
});`

An element is clicked.



The visitor sees something happen on the page.

OPDRACHT

SCROLL EVENT

31

```
$(document).scroll(function(){  
    console.log('scroll works');  
})
```

aan het document koppelen we een event-listener 'scroll' met een functie

```
let distanceScrolled = $(document).scrollTop();  
console.log(distanceScrolled);
```

een variabele met de afstand via de scrollTop-method van jquery

```
if(distanceScrolled > 50){  
    console.log('more then 50 scrolled');  
}
```

dan een simpele if/else om te kijken hoe ver er gescrolled is.

<https://codepen.io/petervandenheuvel/pen/VwZzVBQ>

JQUERY EN EVENTS 1/6

EVENTS EN FUNCTIONS

32

<code>.ready()</code>	wanneer alle HTML geladen is (document.ready..)
<code>.load()</code>	wanneer alle bestanden geladen zijn
<code>.unload()</code>	wanneer een gebruiker het tabblad verlaat
<code>.resize()</code>	wanneer een gebruiker het venster herschaalt
<code>.scroll()</code>	wanneer een gebruiker scrolt
<code>.error()</code>	wanneer de gebruiker een error ontvangt

JQUERY EN EVENTS 2/6

EVENT BINDING

33

Via een **event bind** kan je een functie (of meerdere functies) linken aan een bepaald element

```
$(selector).bind(event), function)
```

voorbeeld:

```
$('.eenKnop').bind(click, naamFunctie);
```

```
function naamFunctie(){  
    alert('Hello World!');  
}
```

JQUERY EN EVENTS 3/6

MOUSE EVENTS

34

Wil je up-to-date informatie? Bekijk dan: <http://api.jquery.com/category/events/>

click	klikken en loslaten v.d. linkermuisknop
dblclick	dubbelklik op de linkermuisknop
mousedown	klikken op de linkermuisknop
mouseup	loslaten van de linkermuisknop
mouseenter	muisknop over bepaald element
mouseleave	muisknop uit bepaald element
mousemove	muisknop bewegen via bepaald element
mouseout	muisknop verlaat een element
mouseover	muisknop beweegt over een element



```
$(document).ready(function() {  
  
    // second event:  
    $(".knop").click(function() {  
        //$("#inhoud").show();  
        //$("#inhoud").hide();  
        $("#inhoud").toggle();  
    });  
  
}); // End of document.ready functie.
```

JQUERY EN EVENTS 5/6

FORMULIER EVENTS

36

.change()

.focus()

.focusin()

.focusout()

.blur()

.select()

.submit()

.reset()

aanpassen van een veld

aanklikken van een veld

aanklikken van een veld in een ander element

wegklikken van een veld in een ander element

wegklikken van een veld

selecteren van tekst in een element

verzenden van een formulier

resetten van een formulier

TIP: Raadpleeg jQuery.com voor voorbeelden en aanvullende informatie.

JQUERY EN EVENTS 6/6

KEYBOARD EVENTS

37

`.keydown()`

gebruiker drukt voor de eerste keer op knop

`.keypress()`

wanneer gebruiker toets op toetsenboard indrukt

`.keyup()`

wanneer de gebruiker de knop loslaat

TIP: Raadpleeg [jQuery.com](https://jquery.com) voor voorbeelden en aanvullende informatie.

VOORBEELD

KEY DOWN EVENT

38

aan het document koppelen we een event-listener 'scroll' met een functie, die reageert op 'keydown'-events

```
$(window).on('keydown', (e)=>{
  let keyPressed = e.keyCode;
  alert(keyPressed);

  // move right
  if(keyPressed == 39){
    $('.player').animate({left: "+=50px"});
  }
});
```

binnen de anonieme functie, leggen de waarde van de gedrukte toets vast in een var.

als de toets 'pijl rechts' is, animeren we de left-waarde, en tellen we er steeds 50px bij.

<https://codepen.io/petervandenheuvel/pen/wwwqRrK>

De `.val()` method wordt primair gebruikt om waarden van formulieren op te vragen, inputs, selectboxes en text-area.

MEER INFO: [HTTP://API.JQUERY.COM/VAL/](http://api.jquery.com/val/)



```
$( 'a' ).on( 'click', function() {  
    var inputValue = $( 'input' ).val();  
    alert( inputValue );  
    $( '#result' ).text( 'Waarde ingegeven is: ' + inputValue );  
})
```

<https://codepen.io/petervandenheuvel/pen/OJLjdWp>

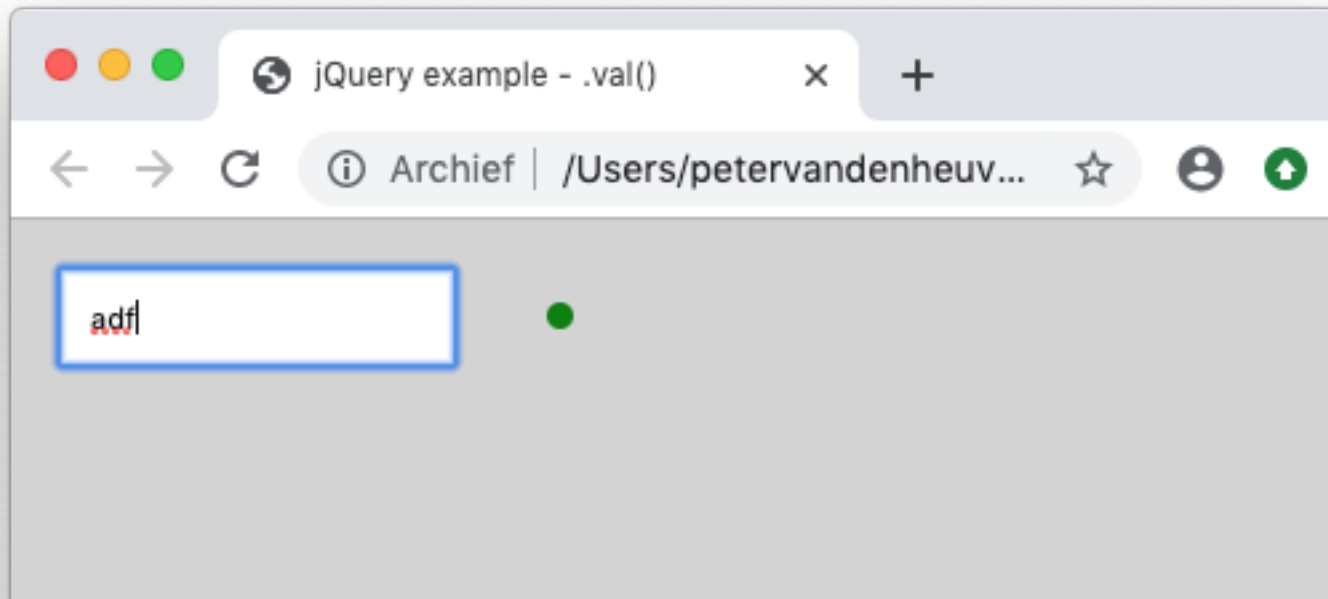
OEFENING

INPUT CONTROLE

10

Oefening:

Schrijf een functie die een groen of rood signaal na input als de input groter is dan 10 tekens.



OEFENING

DISTANCE TO MARS

A1

Oefening:

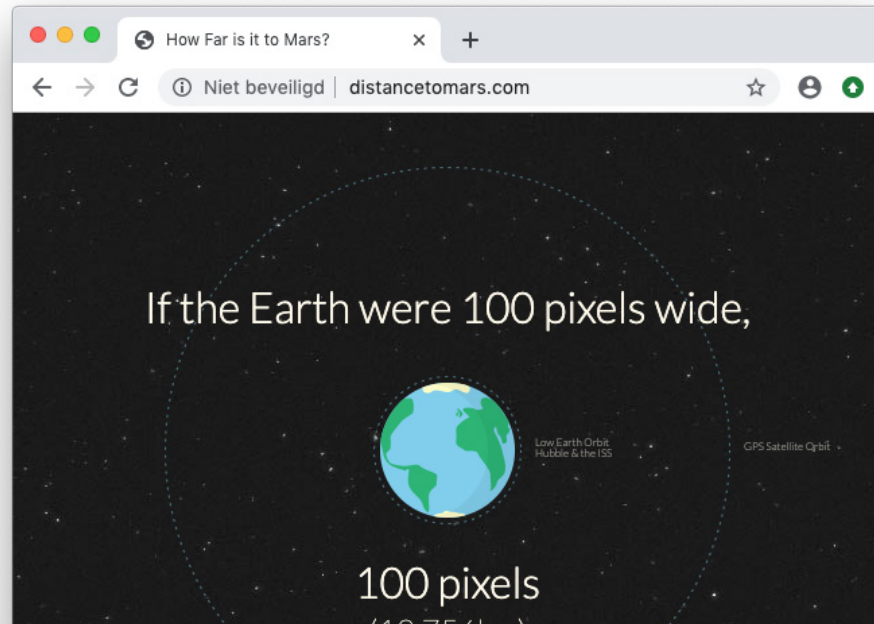
Ga naar de website 'distance to mars.com'.

Maak een vereenvoudigde versie van deze website.

Het helpt om eerst een PSD te maken van de stappen die je nodig hebt.

Voorbeeld code:

<https://codepen.io/petervandenheuvel/pen/mgyENj>



EXTRA

.ATTR()

12

Verkrijg de value van een attribute van het eerste element dat overeenkomt met de selector.
<https://codepen.io/petervandenheuvel/pen/xxKLeGp>



```
7      <meta name="description" content="">
8      <meta name="viewport" content="width=device-width, initial-scale=1">
9      <script type="text/javascript" src="http://code.jquery.com/jquery.min.js"></script>
10     </head>
11     <body>
12       <a id="link" href="www.tweakers.net">Tweakers</a> <br />
13       <a href="www.nu.nl">nu.nl</a>
14       <script>
15         $( document ).ready(function() {
16           var $attr = $('a').attr('href');
17           console.log($attr);
18         });
19       </script>
20     </body>
21 </html>
```

Line 8, Column 77 Tab Size: 4 HTML

Vooralsnog hebben we alle jQuery statements achter elkaar geschreven, en steeds de selector herhaald.

Echter in jQuery bestaat de techniek 'chaining', waarmee je meerdere commando's in 1x kan uitvoeren.

Om te 'chainen' koppel simpelweg de acties aanelkaar met een . (punt)

.....

voorbeeld

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

EXTRA

.QUERYS HERGEBRUIKEN



Een andere optimalisatie is om Query's op te slaan in een variabele. In plaats van steeds opnieuw de DOM te doorlopen is het verstandiger om deze op te slaan in een variabele.

```
1 $('#item').css ('color', '#123456');
2 $('#item').html ('hello');
3 $('#item').css ('background-color', '#ffffff');
4
5 // you could use this instead
6 $('#item').css ('color', '#123456').html ('hello').css ('background-color', '#ffffff');
7
8 // or even
9 var item = $('#item');
10 item.css ('color', '#123456');
11 item.html ('hello');
12 item.css ('background-color', '#ffffff');
```

Line 12, Column 42 Tab Size: 4 JavaScript

EXTRA

VARIABELE IN SELECTORS

15

<https://codepen.io/petervandenheuvel/pen/jONLRqW>

```
12 <script type="text/javascript">
13 $(function(){
14     $("button").click(function(){
15         var name = $(this).attr("name");
16         console.log(name);
17         $("#" + this.name).slideUp();
18     });
19 });
20 </script>
21
22 <input type="text" id="x"/>
23 <button type="button" name="x">x</button> <br>
24
25 <input type="text" id="y"/>
26 <button type="button" name="y">y</button>
27 </body>
28 </html>
```

EXTRA

CALLBACK FUNCTION

16

```
8      <meta name="viewport" content="width=device-width, initial-scale=1">
9      <script type="text/javascript" src="http://code.jquery.com/jquery.min.js"></script>
10     </head>
11     <body>
12     </body>
13     <script type="text/javascript">
14
15     function getSize(width, height, depth){
16         var volume = width * height * depth;
17         return volume;
18     }
19
20     var Object1 = getSize(3, 2, 3);
21
22     console.log(Object1);
23     </script>
24 </html>
```

Line 1, Column 1

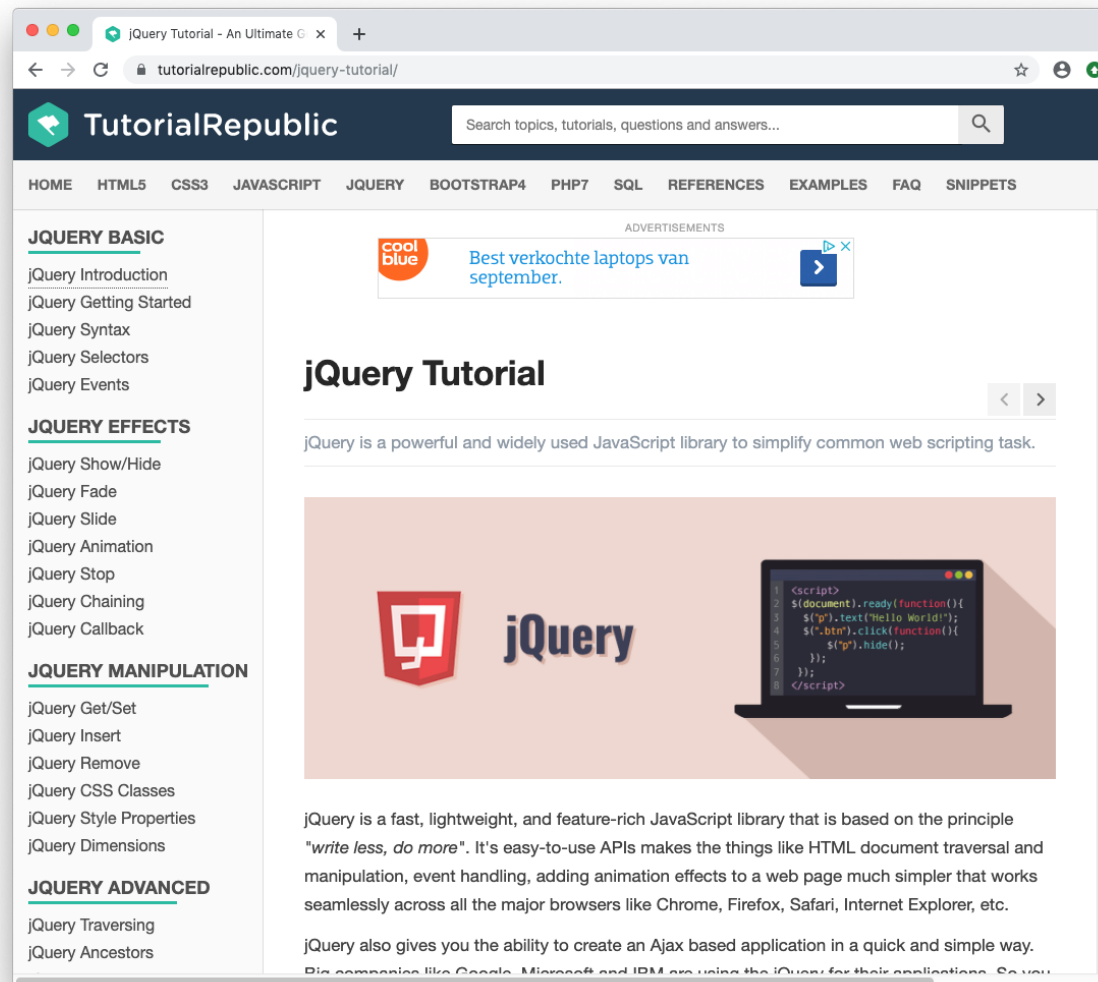
Tab Size: 4

HTML

VERDER OEFENEN?

TUTORIALS:

A7



VERDER OEFENEN?

DOCUMENTATIE:

18

