jQuery oefeningen:

## Opdracht 1 - “document ready”:

Maak een simpele alert, met daarin de tekst “Hello World”, die geladen wordt als de DOM-ready is. Zoals je weet kan je al je code in een functie steken, die pas wordt uitgevoerd als alle HTML geladen is. (= DOM is ready).

Zorg zelf ervoor dat jQuery geladen wordt; download deze van jQuery.com, plaats hem in de juiste map (/js) en koppel deze onderaan net voor de </body>-tag in het document.

../Schermafbeelding%202018-04-19%20om%2009.09.46.png<--voorbeeld van een 'DOM-ready'-functie.

## Opdracht 2 - “Gebruik een CDN”:

Je hoeft niet altijd jQuery te downloaden en te koppelen als jQuery.js bestand. Een alternatieve manier is om jQuery via het internet te laden. Men noemt dat een 'CDN, of Content Delivery Netwerk'. Gebruik daarvoor deze website: https://cdnjs.com/libraries/jquery/.

Voeg dan in het bestand main.js een alert toe die zich toont als het document geladen is. Verschil met opdracht 1 is dat je een CDN gebruikt.

Uiteraard ook hier is de volgorde belangrijk: Eerst uw jQuery laden, dan uw eigen code oproepen.

Als jQuery juist geladen is, voer dan een alert uit met als boodschap 'hallo wereld' binnen een document-ready statement.

Vul de pagina met een aantal div's, geef deze <div>'s een blauwe achtergrondkleur en vul ze met de tekst 'joepie bijna weekend'.

Plaats een HTML-tabel op de pagina met daarin een Top 3 van favoriete liedjes; eerste kolom 'artiest' en de tweede kolom 'nummer'. Geeft de oneven rijen een zwarte achtergrond met witte tekst, en de even een witte achtergrond met zwarte tekst. Met jQuery uiteraard...

## Vragen:

Zoek het antwoord op de volgende vragen:

- Wat is de kernfunctie van jQuery?

- Waarom is jQuery geen vervanging van JavaScript?

- Wat is het verschil tussen jQuery 1.x, 2.x of 3.x, welke worden afgeraden om nog te gebruiken?

- Wat is het verschil tussen jQuery downloaden en een CDN gebruiken?

- Wat is het verschil tussen: $ en jQuery, zoals in dit voorbeeld, is er een verschil?

- Wat is het nut van een "document.ready"-statement?



## Opdracht 3 - Selectors:

Het is erg belangrijk om de selectors te beheersen in jQuery. Door slim gebruik te maken van de CSS-selectors kan je heel eenvoudig de elementen uit de DOM / HTML halen die je wil manipuleren. In main.js in de begin-map vind je in commentaar 7 selector-oefeningen.

Probeer eens alle selectors te vinden. Tips kan je vinden op de site van jQuery of op W3C-school bij jQuery. Of soms zelfs met Google; documentatie over jQuery is overal… tips:

https://api.jquery.com/category/selectors/

http://lmgtfy.com/?q=jquery+selector+input+of+type+text

## Opdracht 4:

Konijnen die verdwijnen..: schrijf een anonieme functie die als je op een konijn in de hoed klikt, dat konijntje laat verdwijnen. Je zal een event-handler moeten schrijven die een mouse-click kan registeren.

Enkel de foto van het konijn waarop je klikt mag verdwijnen. Probeer daarvoor een selector te vinden die werkt. (hint: https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery\_sel\_this)

Een voorbeeld van zo’n event-handler zie je hier:

$( "a" ).on( "click", function() {

*// do something*

});

## Opdracht 5:

Veel functies binnen javaScript hebben zowel een 'getter' als een 'setter' mogelijkheid. Haar de 'color' op van de H1 (get) en stel deze kleur in op de H2 (set). Gebruik hiervoor de .css() method.

## Opdracht 6:

Koppel aan de knop ‘Hello!’ een functie die de gebruiker groet als erop geklikt wordt.

Ontwikkel een tweede functie die de input-velden controleert of er iets is ingevuld.

Zo niet: zet een rode border rond het input-veld, zo ja: een groene.

De stappen die het script zou moeten doorlopen:

1. je koppelt allereerst een event-handler aan beide inputvelden.
2. Binnen die functie, neem je de waarde van de input-velden met de .val()-method van jQuery.
3. Als de input gelijk is aan niets; voeg de CSS-klasse ‘.red’ toe.
4. Als er wel input is, voeg de CSS-klasse ‘.green’ toe. Als de functie al eens uitgevoerd is, zal je uiteraard eerst de .red-klasse moeten verwijderen. Weet dat je de acties kan ‘chainen’ door ze met een ‘.’ Achterelkaar te schrijven.
5. Tip: als je niet wil dat de pagina steeds refreshed als je op de knop 'submit' duwt, maak dan gebruik van een preventDefault() method. Ik hoef je niet te vertellen dat je dat kan googlen zeker? 😜

## Opdracht 7:

Schrijf een functie die de rij uit de tabel verwijderd. Je detecteert de ‘onClick’-event natuurlijk op een link (a-tag) maar je kan met .parent-method omhoog wandelen in de boom, gebruik dat om de juiste rij te selecteren. Verwijderen uit de DOM kan simpelweg met de .remove-method.

TIP: je kan chainen; ook als je niet direct het juiste element vast hebt.

## Opdracht 8:

Als je op de knop drukt wordt er een functie uitgevoerd. Afhankelijk van het aantal dat standaard op de knop staat ( 5 of 10), laat je bij iedere druk op de knop het getal op die knop met 1 afnemen.

Totdat het '0' bereikt dan schakel je de knop uit. Dat kan door 'disabled' als attribute op de knop toe te voegen.

## Opdracht 9:

Verander de tekst binnen de tag (.text methode), om de tekst te veranderen in #cat naar ' 'miauw 🐈'' En bij '#hond' naar ' woef 🐶'

## Opdracht 10:

Probeer een Boodschappen app te ontwikkelen. Je hebt een input veld waar mensen 'producten' kunnen toevoegen. Deze worden aan een <ul> als <li> toegevoegd.

Als je 1x op de <li> klikt wordt deze doorstreept. Als je een tweede maal klikt verwijder je hem uit de lijst. Je kan hiervoor gebruik maken van .addClass, en hasClass, en een if/else statement.