

OEFENINGEN JQUERY

01 – EENVOUDIG BEGINNEN

- a) Maak een template/kapstokpagina waarin jQuery wordt geladen.
 - Schrijf een `document.ready`-function en toon een alert (of schrijf in de console) zodra het DOM gereed is.
 - Er zijn verschillende notatiewijzen voor `document.ready`. Gebruik ze alle drie.
 - Plaats drie div's in de pagina, elk met een eigen ID. Geef de div's een rode, blauwe en groene achtergrond bij het laden van de pagina.

02 – SELECTIES EN ACTIES

- a) Maak een script waarbij na een klik op een knop alle links in de pagina een andere achtergrondkleur krijgen.
- b) Maak een tabel. Geef de rijen van de tabel om-en-om een andere achtergrondkleur.
 - Geef de koptekst van de tabel een andere achtergrondkleur dan de even/oneven rijen.
- c) Maak een event handler voor een knop met `.on()`. Verwijder deze daarna via een klik op een (andere) knop met `.off()`. Test dit.
 - Lees de documentatie voor `.one()`. Verzin zelf een nuttige toepassing hiervan.
- d) Maak een pagina met een afbeelding. Bij een mouseover wordt de afbeelding vervangen door een andere afbeelding. Bij een mouseout wordt de oorspronkelijke afbeelding weer getoond.

03 – HTML- EN CSS-FUNCTIES

- a) Schrijf een script waarbij een div (of een afbeelding, of ander element) zichtbaar wordt na een klik op een knop.
 - Experimenteer met `.show()` en `.hide()`, maar ook met `.slideUp()`, `.slideDown()` en `.toggle()`.
 - Gebruik callbackfuncties om met één klik op een knop een serie div's zichtbaar te maken: Zodra de eerste div geheel getoond wordt, wordt de tweede div zichtbaar, enzovoort.
- b) Maak een pagina met een tekstvak. Plaats hierbij knoppen met de volgende functies.
 - Voeg de inhoud van het tekstvak als tekst toe aan (een div in) het document.
 - Voeg de inhoud van het tekstvak als HTML toe aan het document
 - De inhoud van een tekstvak kun je uitlezen met `$('#tekstvak').val()`.
 - Plaats ook knop waarmee een bestaande div wordt gekopieerd/gekloond
 - Voeg (dynamisch) aan elke nieuw toegevoegde div een koppeling of knop Verwijderen toe, waarmee de node uit het DOM wordt verwijderd (vergelijk met een 'prullenbakje' achter e-mailberichten in een lijst).
- c) Bekijk zelf de werking van de DOM-functies `.wrap()`, `.unwrap()`, `.wrapAll()` en `.wrapInner()`.
 - Verzin een eenvoudig voorbeeld waarin je deze functies toepast.
 - Leg dit uit/bespreek dit met een collega.
- d) Maak een pagina met vier CSS-klassen die elk een andere achtergrondkleur hebben. Plaats in de HTML een selectielijst waarmee de user een kleur kan kiezen. Zorg er voor dat een resultaat-div de gekozen kleur krijgt:
 - Na een klik op een knop;
 - Direct na het kiezen van de kleur uit de lijst (gebruik de event `change`)
- e) Maak een formulier waarin de user in tekstvakken de margin, padding en achtergrondkleur van een div kan opgeven.

- Stel deze waarden in voor `div`, via een klik op een knop.
- f) Maak een pagina met vijf alinea's of `div`'s. Zorg ervoor dat via een klik op een knop de oneven alinea's (of `div`'s) een andere achtergrondkleur en lettertype krijgen.
 - De waarden hiervoor zijn aanwezig in de CSS-klasse `bgHighlight` (schrijf deze uiteraard zelf).
 - Als op een alinea wordt geklikt, wordt in een alert getoond of de klasse `bgHighlight` op dit moment van toepassing is op de alinea.
 - Nog een klik op de knop verwijdert de klasse `bgHighlight` weer van de oneven alinea's.
- g) Maak een pagina met één image. In de site zijn drie (of meer) images beschikbaar.
 - Een klik op een knop laat de volgende afbeelding zien.
 - In een `div` onder de image wordt het pad + bestandsnaam van de afbeelding getoond.
 - Zorg er voor dat de attributen `title` en `alt` voor de image ook worden aangepast als de volgende afbeelding wordt geladen.
 - Verwijder de knop en zorg er voor dat een klik op de afbeelding zelf de volgende afbeelding ophaalt en toont.

04 – FORMULIEREN

- a) Maak een formulier met een checkbox 'Ik ga akkoord met de algemene voorwaarden'. Alleen als het vakje is aangevinkt, mag het formulier verzonden worden. Geef anders een alert-melding.
- b) Maak een lijst met vier of meer keuzerondjes. Een klik op de knop laat in een alert-melding zien welk van de keuzerondjes is geselecteerd.
 - Verwijder de knop bij de lijst. Een klik op een keuzerondje laat nu direct (in een `div` op de pagina) zien welke optie is gekozen.
- c) Vervang de keuzerondjes door checkboxes. Een klik op de knop laat zien welke checkboxes (nul, één of meer) zijn geselecteerd.
 - Verwijder de knop bij de lijst. Zodra een vakje aan- of uit wordt gezet, wordt dit in een `div` op de pagina getoond. De `div` toont altijd de actuele status van de lijst.
 - Let er op dat er bij checkboxes meerdere vakjes tegelijk geselecteerd kunnen zijn. Tip: maak een hulp-array waarin je bijhoud welke opties zijn aangevinkt. Toon de status van de array in de pagina.
- d) Maak een formulier met een 'zoekvak'. Zodra het vak focus krijgt, wordt het twee keer zo breed weergegeven en krijgt het een gekleurde achtergrond. Verwijder deze eigenschappen als het vak focus verliest.

05 – ANIMATIE

- a) Maak een script waarmee elementen op de pagina worden geanimeerd. Gebruik hiervoor de functie `.animate()`.
 - Test dit voor verschillende eigenschappen (`margin`, `padding`, `positie`, `lettergrootte`).
 - Test dit met diverse events (`click`, `hover`, `focus`, `blur`)
 - Zoek zelf uit hoe je complexe waarden (kleurwaarden) kunt animeren. Pas dit toe.
- b) Onderzoek zelf de werking van de *animation queue*. Maak een voorbeeld waarin je dit toepast. Demo dit aan een collega en licht toe.
 - Zie voor informatie bijvoorbeeld <http://api.jquery.com/animate/>, <http://learn.jquery.com/effects/queue-and-dequeue-explained/> en <http://learn.jquery.com/effects/uses-of-queue-and-dequeue/>

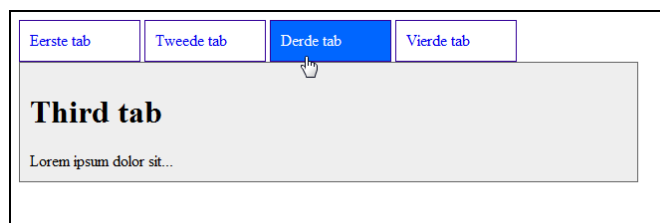
06 – AJAX

- a) Schrijf zelf een jQuery `.load()`-opdracht.

- Maak bijvoorbeeld diverse HTML-bestanden die via een 'menu' worden ingeladen in een result-div op de pagina.
- b) Gebruik de opdrachten `$.get()` en `$.post()`
 - Retourneer eenvoudige tekst, of haal 'echte' data op, die op een server hebt geplaatst.
 - Doe hetzelfde met de opdracht `.getJSON()`.
 - Verwerk de resultaten in de pagina; naar wens door zelf een for-lus te schrijven, of door een data-binding framework te gebruiken als knockout.js of jQuery templates (verouderd).
- c) Maak dezelfde pagina, nu met de opdracht `.ajax()`
 - Stel een configuratiescript samen en geef dit mee als parameter.
 - Gebruik diverse parameters: `url`, `dataType`, `contentType`, `success`, `error` en meer. Zie voor een overzicht <http://api.jquery.com/jquery.ajax/>.
 - Schrijf ook een cross-domain Ajax-call waarbij JSONP wordt gebruikt. Zie voor mogelijke adressen bijvoorbeeld de JavaScript-oefeningen.
- d) Plaats bovenin het script de functie `$.ajaxSetup()` en bekijk vervolgens hoe je de (diverse) Ajax-calls kunt vereenvoudigen.
- e) Maak een script waarbij gebruik wordt gemaakt van een deferred object.
 - Schrijf een chain ajax-calls en resolve deze met `.then()`.
 - Denk aan foutafhandeling met `.fail()`.
 - Optioneel: bestudeer de werking van `.done()` en `.always()`.
 - Demonstreer de werking aan een collega en licht deze toe.
 - Documentatie is te vinden bijvoorbeeld op <http://api.jquery.com/category/deferred-object/>, <http://net.tutsplus.com/tutorials/javascript-ajax/wrangle-async-tasks-with-jquery-promises/> of <http://sebastianfastner.de/javascript-promises-explained.html> (meer algemeen).

07 – USER INTERFACE ELEMENTEN ZELF MAKEN

- a) Maak zelf een eenvoudig systeem met tabbladen. De actieve/geselecteerde tab moet onderscheidend worden getoond. Gebruik HTML, CSS en jQuery. Aanwijzingen:
 - Maak een `` met de menu-items (dit worden de 'tabs')
 - Maak een `<div>` met content-items. Elke tab krijgt een eigen `<div>`
 - Zorg voor correcte styling via CSS; gebruik bijvoorbeeld voor de menu-items `display:inline`.
 - Schrijf script dat aanhaakt op het klikken op de tabs/menu-items. Zoek uit op welke link is geklikt en maak deze tab zichtbaar. Het systeem kan er bijvoorbeeld als volgt uitzien.



- b) Maak een systeem met custom tooltips.
 - De inhoud van het title-attribuut wordt in een custom HTML-tooltip getoond wanneer over een afbeelding of hyperlink gehoverd wordt.
 - Tip: gebruik de jQuery-method `.hover()`.
 - Gebruik de mouse events `e.pageX` en `e.pageY` om de tooltip te laten meebewegen met de muis.



- c) Maak een eenvoudig dropdown menu
- Bij een mouseover wordt het menu getoond (via een slide, of custom animatie)
 - Bij een mouseout wordt het menu weer verborgen.
 - Zorg er daarna voor dat pas bij een muisklik het menu geopend/gesloten wordt.
 - Indien je werkt met een mouseover: zoek op wat `hoverIntent` doet, en pas dit nuttig toe in je menu. Gebruik <http://cherne.net/brian/resources/jquery.hoverIntent.html> als uitgangspunt.

08 – JQUERY UI EN ANDERE PLUG-INS

- a) Download en gebruik jQuery UI in een pagina. Plaats op de pagina minimaal vier widgets, denk bijvoorbeeld aan
- `Datepicker`, `Dialog`, `Progressbar`, `Slider` of anderen.
 - Test op welke wijze deze widgets geconfigureerd kunnen worden. Plaats één 'standaardwidget' op de pagina, en één geconfigureerde versie.
- b) Maak een Drag & Drop interface met jQuery UI. Bedenk bijvoorbeeld een eenvoudig spelletje of een winkelwagentje dat je met producten kunt vullen.
- c) Bekijk en test enkele mogelijkheden uit de groep Effects van jQuery UI.
- Maak bijvoorbeeld een kleuranimatie voor een knop: als je op een knop klikt, animeert een div van groen naar rood.
 - Of maak een highlight-functie: bij een klik op een knop animeert/highlight de achtergrondkleur van een stuk tekst van wit naar lichtgeel en weer terug.
- d) Download en gebruik minimaal twee plug-ins uit de jQuery Plug-in Registry. Denk bijvoorbeeld aan plug-ins voor formuliervalidatie, lightboxes, paginering, Google Analytics en meer.