

OEFENINGEN JQUERY ADVANCED

01 – EENVOUDIGE SELECTIES, EVENTS EN ACTIES

- a) Maak een pagina met dummytekst en een aantal dummy hyperlinks.
 - Schrijf een script waarbij na een klik op een knop alle links in de pagina een andere achtergrondkleur krijgen. Gebruik eventueel *jQuery_adv_oefening01a.html* als uitgangspunt.
 - Vervolg: maak een array met achtergrondkleuren. Elke klik gebruikt de volgende kleur uit de array.
 - Vervolg: maak een tweede array met voorgrondkleuren. Elke klik neemt een random voorgrondkleur en random achtergrondkleur uit de array's en combineert deze voor de hyperlinks.
- 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.
 - Enkele voorbeeld-afbeeldingen vind je in de map `\img`
- e) Voeg enkele jpg-afbeeldingen toe aan de pagina. Maak een knop waarmee de jpg-afbeeldingen worden vervangen door png-afbeeldingen. Schrijf de selector zo efficiënt mogelijk.
- f) Maak een pagina met enkele selectievakjes. Schrijf een script dat in de console (of in een alert) toont welke vakjes geselecteerd zijn. Gebruik eventueel *jQuery_adv_oefening01f.html* als uitgangspunt.
 - Doe dit eerst waarbij na een klik op de knop het resultaat wordt getoond in de console of in een alert.
 - Doe dit daarna real-time, waarbij de selectie in de pagina direct zichtbaar is, zodra op een vakje wordt geklikt.
- g) 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.

02 – WERKEN MET `.EACH()`, `.DATA()` EN `.EXTEND()`

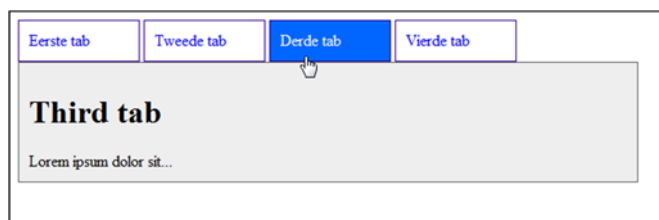
- a) Schrijf een eenvoudig formulier waarvoor alle invoervelden verplicht zijn. Check dit met `$.each()` als het formulier verzonden wordt.
 - Gebruik eventueel *jQuery_adv_oefening02a.html* als uitgangspunt, of maak zelf een formulier.
- b) Maak een pagina waarin diverse HTML-elementen een data-attribuut gebruiken. Lees de waarde van de attributen uit met `.data()` en toon ze in de pagina, in de console of in een alert.
 - Gebruik eventueel *jQuery_adv_oefening02b.html* als startpunt.
 - Breidt de code uit met een voorbeeld waarbij je `.data()` gebruikt als setter. Lees verderop in het script de toegekende waarde uit en toon hem in de pagina.
- c) Maak een script waarbij twee (of meer) objecten worden samengevoegd in één hoofdobject.
 - Gebruik eventueel *jQuery_adv_oefening02c.html* als startpunt.
 - Wat gebeurt er als de diverse objecten dezelfde keys/values aanwezig zijn? Test dit.
 - Voer de samenvoeging uit naar een bestaand object, of naar een nieuw object.

03 – AJAX

- a) Schrijf zelf een `jQuery .load()`-opdracht.
 - o Laadt de inhoud van een tekstbestand in een div.
 - o 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()`
 - o Retourneer eenvoudige tekst, of haal 'echte' data op, die op een server hebt geplaatst.
 - o Als voorbeeldbestanden kun je `ajaxData.txt` of `ajaxData.json` gebruiken.
 - o Doe hetzelfde met de opdracht `.getJSON()`.
 - o 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()`
 - o Stel een configuratiescript samen en geef dit mee als parameter.
 - o Gebruik diverse parameters: `url`, `dataType`, `contentType`, `success`, `error` en meer. Zie voor een overzicht <http://api.jquery.com/jquery.ajax/>.
 - o 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.
 - o Schrijf een chain ajax-calls en resolve deze met `.then()`.
 - o Denk aan foutafhandeling met `.fail()`.
 - o Optioneel: bestudeer de werking van `.done()` en `.always()`.
 - o Demonstreer de werking aan een collega en licht deze toe.
 - o 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).

04 – 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:
 - o Uitgangspunt: *jQuery_adv_oefening04a.html*, of een eigen pagina.
 - o Maak een `` met de menu-items (dit worden de 'tabs')
 - o Maak een `<div>` met content-items. Elke tab krijgt een eigen `<div>`
 - o Zorg voor correcte styling via CSS; gebruik bijvoorbeeld voor de menu-items `display:inline`.
 - o 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.
 - o De inhoud van het title-attribuut wordt in een custom HTML-tooltip getoond wanneer over een afbeelding of hyperlink gehoverd wordt.
 - o Uitgangspunt: *jQuery_adv_oefening04b.html*, of een eigen pagina.
 - o 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.

05 – PLUG-INS

- Download en gebruik de plug-in Cycle. Lees de documentatie en bekijk hoe je de plug zowel kunt gebruiken via HTML-attributen als via jQuery. Dit is *niet* standaard! De meeste plug-ins kun je alleen via jQuery-script bereiken.
 - <http://jquery.malsup.com/cycle2/>
 - Maak een eigen pagina met deze plug-in, op basis van `jQuery_adv_oefening05a.html`.
- Download en gebruik minimaal andere twee plug-ins uit de jQuery Plug-in Registry. Denk bijvoorbeeld aan plug-ins voor formulervalidatie, lightboxes, paginering, Google Analytics en meer.
- Schrijf een eigen, eenvoudige plug-in die de achtergrondkleur van de paragrafen op een pagina aanpast, op basis van het voorbeeld in de les.
 - Gebruik eventueel `jQuery_adv_oefening05c.html` als uitgangspunt.
 - Meer lezen: <http://www.queness.com/post/112/a-really-simple-jquery-plugin-tutorial> en/of <http://learn.jquery.com/plugins/basic-plugin-creation/>
- Maak op basis van je code uit oefening 1b (om-en-om gekleurde tabelrijen) een plug-in die hetzelfde doet.
 - Zorg er voor dat de plug-in op deze manier kan worden aangeroepen
`$('#tableName').zebra();`
 - Breidt de plug-in uit, zodanig dat de eerste (kop-)rij van de tabel een afwijkende opmaak krijgt. Breidt de plug-in daarna uit zodat elke rij opvallend gekleurd wordt bij een mouseover.
 - Zorg er voor dat je de plug-in zodanig kunt configureren dat je zelf de achtergrondkleur voor de tabelrijen kunt opgeven, het lettertype voor de tabelrijen en eventuele andere configuratieopties die je zelf verzint.