OEFENINGEN ECMASCRIPT 2015

01 – KORTE KENNISMAKING - COMPATIBLITY TABLE

a) Bekijk de ECMAScript 2015-compatilibity table op https://kangax.github.io/compat-table/es6/. Leer deze lezen en begrijpen voor de verschillende browsers. Bekijk:
   - De nieuwe keywords aan de linkerkant
   - De generatie Compilers/transpilers en polyfilss (Traceur, Babel, TypeScript).
   - De generatie browsers. Hoe scoort jouw browser op dit moment?

02 – MAPS EN SETS

a) Een Map zou je kunnen zien als een 'super array'. Schrijf een script waarin een `Map` wordt gedefinieerd. Voeg hierin toe de leden van je gezin, collega's of anders.
   - Probeer een dubbele vermelding toe te voegen. Wat gebeurt er?
   - Schrijf de inhoud van de map naar de console (gebruik `console.log()`).
   - Test de werking van `.keys()`, `.values()` en `entries()`.
   - Zie voor uitgebreider gebruik van `Map()` eventueel http://exploringjs.com/es6/ch_maps-sets.html#_maps

b) Maak een Set met items. Gebruik eventueel dezelfde data als in de vorige oefening. Gebruik nu de methode `new Set()`.
   - Voeg items toe aan de set en toon ze in de console.
   - Sets zijn *geen* key/value-pairs. Bekijk wat je wél in een set kunt opslaan. Arrays?, objecten?
   - Toon het aantal items in de set.
   - Test of een item aanwezig is en verwijder het daarna.
   - Lees documentatie over Set op http://exploringjs.com/es6/ch_maps-sets.html#_set.

03 ARROW FUNCTIONS

a) Schrijf een functiedeclaratie op de 'ouderwetse' manier (ES5), met parameters, een functiebody en een return-resultaat. Maak bijvoorbeeld een function `optellen(a, b)` die de som van `a` en `b` retourneert.
   - Herschrijf deze functie zodanig dat nu gebruik wordt gemaakt van een arrow-function.
   - Doe hetzelfde voor een andere functie die je schrijft.
   - Compileer en test de functie in de browser.

04 CLASSES

a) Maak een class `Familie` (of `Collega's`, of `Vrienden`), met diverse CRUD-operations:
   - Zorg er voor dat je familieleden kunt toevoegen, opvragen of verwijderen.
   - Maak waar mogelijk gebruik van arrow functions en/of Maps & Sets om de familieleden te beheren.
   - Instantieer de klasse en schrijf het resultaat van toevoegen/verwijderen etc naar de console.

b) Maak een base class (bijvoorbeeld `Animal`) en verschillende subklasses die daarvan zijn afgeleid (met bijvoorbeeld zoogdieren, amfibieën etc) .
   - Instantieer de subklasses en maak verschillende dieren. Schrijf ze naar de console.

c) Lees http://exploringjs.com/es6/ch_classes.html voor verdere informatie over classes.

d) Gebruik waar mogelijk template literals (voorheen: *template strings*) en string interpolation om output mooier en leesbaarder te formatteren.

05 - MODULES

a) Maak een module waaruit een aantal properties worden geëxporteerd.
   - Zie bijvoorbeeld modules-01.html en modules-02.html voor een voorbeeld.
   - Definieer één module per file en één file per module. Let er op dat je nu niet meer een `<script>`-tag per module hoeft te gebruiken, maar met Traceur de module laadt.
   - Zie http://exploringjs.com/es6/ch_modules.html#ch_modules voor meer voorbeelden.

b) Maak een class die wordt geëxporteerd.
   - Gebruik in je HTML-file `<script type="module">` om de module te importeren. Denk aan het keyword `import`. Test daarna of de module werkt.

c) Optioneel: installeer en gebruik browserify om modules-binnen-modules te importeren, te bundelen en te gebruiken in de browser. Zie de folder `\browserify` voor een voorbeeld.

06 - BLOCK SCOPE EN PARAMETERS

a) Bekijk de voorbeelden met `let` en `const` en gebruik deze in een eigen script.
   - Gebruik een bestaand script van jezelf en vervang hierin `var` door `let` en `const`. Compileer het script en onderzoek of het nog werkt.

b) Schrijf een functie met een aantal default parameters, bijvoorbeeld `function optellen( a=10, b=20){…}`.
   - Roep de functie op verschillende manieren aan, mét en zonder parameters.
   - Schrijf een functie met named parameters. Let op de objectnotatie van de named parameters. Gebruik deze functie in een script. Bekijk eventueel `namedParameters.es6` voor een voorbeeld.

c) Bekijk het voorbeeld `restParameters.es6`.
   - Schrijf een functie die een willekeurig aantal getallen bij elkaar optelt en het resultaat retourneert.

07 - OVERIG

a) Neem een bestaande JavaScript-applicatie van jezelf en ga deze upgraden naar ECMAScript 2015/ES6. Onderstaande volgorde kan hierbij nuttig zijn:
   - Stel een transpiler in voor je project (suggestie: babel 5.8.x)
   - Vervang var door `let` en `const`.
   - Vervang functies-body's en callbacks door arrow functions.
   - Bekijk waar je template literals `${ … }` kunt inzetten.
   - Herschrijf functions naar classes.
   - Plaats classes in een eigen module en exporteer deze voor gebruik in andere modules.
   - Bekijk of je array's of andere collections kunt vervangen door `Map()` en `Set()`.
   - Onderzoek of je browserify (of webpack) kunt inzetten voor bundling. Bekijk ook de opties voor gulp-minify om je code te minimaliseren.

b) Ga naar http://exploringjs.com/es6/ en bekijk de inhoudsopgave van het boek van Axel Rauschmayer. Lees enkele hoofdstukken en bekijk voorbeelden van onderdelen waarover je meer wilt weten.
   - Kopieer codevoorbeelden naar je eigen editor en probeer ze uit.