

Man kan jobba fri-tva om man vill!

## Obligatorisk uppgift 2 - Virtuellt skrivmaskin

### Syfte

Uppgiften syftar till att ge en introduktion till Javascript (funktionalitet), och hur det samverkar med HTML (semantik) och CSS (design) som behandlades i obligatorisk uppgift 1.

Det som särskilt skall tränas är följande:

- Skapa och använda variabler,
- skapa funktioner,
- anropa funktioner från HTML, och
- manipulera DOM-trädet.

För Javascript finns många externa bibliotek som underlättar mycket. Exempel på sådana är JQuery och Dojo. I den här uppgiften skall dock sådana inte användas, då fokus för uppgiften ligger på att träna på grunderna i Javascript.

### Uppgift

Den här uppgiften går ut på att skapa en liten virtuell skrivmaskin med hjälp av Javascript. En HTML-sida med tillhörande stilmall (CSS) som implementerar utseendet för en skrivmaskin är given, och den innehåller alla taggar och Javascript-bindningar (onclick) för att uppgiften ska kunna lösas. *Förklara för tillgång till, vkt ins!*

#### HTML-sidan (*index.html*)

På sidan finns ett antal grafisk komponenter i form av knappar. Till varje komponent kan man ange vad som ska ske när knappen klickas. Det vanligaste är ett anrop till en Javascript-funktion, eventuellt med argument.

I denna uppgift finns knapparna A-Z, . (punkt), (komma) och SPACE (mellanslag) som alla anropar samma funktion, *addLetter*, men med olika argument (värde som skickas till funktionen). Argumentet är det tecken som motsvarar knappen, alltså knappen märkt med A, anropar funktionen med argumentet 'a' (en sträng).

Det finns ytterligare en knapp som är märkt ENTER. När denna knapp klickas, anropas i stället funktionen *addParagraph*. Det finns också en länk som heter "Räkna bokstäver". När den klickas anropas funktionen *countLetters* och svaret från denna funktion redovisas i en dialogruta.

På HTML-sidan finns också en sektion med taggen `<div>` som är märkt med ett id-attribut med värdet 'output'. Detta id kan användas för att hitta denna specifika nod i DOM-trädet. Till denna nod, ska vi sedan lägga till nya paragrafer.

*Du ska inte förändra något i HTML-filen.*

#### Javascript-sidan (*script.js*)

I Javascript-filen finns början till de tre funktionerna som anropas när knapparna/länken klickas. I varje funktion finns från början en rad där `alert()` anropas. Funktionen öppnar en dialogruta med en meddelandetext.

I Javascript-filen ska följande nu göras.

1. Skapa två variabler utanför funktionerna. Variablerna ska hålla reda på den paragraf där tecken ska läggas till, respektive om paragrafen är jämn eller udda (se punkt 2).
2. I funktionen `addParagraph()` ska en ny paragraf-nod skapas. Till denna nod ska en ny textnod länkas. Första variabeln i 1) ska hålla reda på denna textnod. Paragraf-noden ska läggas till som sista barnnod till noden med id-attributet 'output'. Vidare ska varje ny paragraf-nod få ett attribut "class", vars värde för varannan är "even" och för varannan är "odd" (låt den första paragrafen få värdet "odd", nästa "even", osv). För att hålla reda på när värdet ska vara "odd" resp. "even" använder du den andra variabeln du skapade tidigare i 1).
3. I funktionen `addLetter()` ska två saker göras. Först, om variabeln i 1) inte håller reda på en textnod (dvs. att en paragraf inte ännu är skapad), ska funktionen `addParagraph()` anropas. Sedan, ska argumentets värde (tecknet) läggas till i slutet på texten i noden som variabeln håller reda på.
4. I funktionen `countLetters` ska antalet bokstäver i den pågående paragrafen (den man skriver till) räknas och returneras. Som bokstäver räknas **enbart tecknen a-z** (enbart små bokstäver finns), **inte punkt, komma eller mellanslag**.

## Övriga Krav

- HTML- och CSS-filerna får inte modifieras på något sätt, du ska endast skriva Javascript-kod och kommentarer i Javascript-filen.
- Om ni vill skriva ut någon sorts spårtext i er kod ska `console.log()` användas, `document.write()` och `alert()` ska inte användas. Er Javascript-kod ska inte ge ut några varningar/fel, den ska vara väl strukturerad med indentering, och eventuella 'svåra' avsnitt ska översiktligt kommenteras.
- Javascript ska skrivas i "strict" läge, genom att det längst upp i filen står "*use strict*"; (med citattecknen). Detta läge tar bort bakåtkompatibilitet mot tidigare versioner av Javascript, så en del saker som bedömts vara dåliga/olämpliga och därmed tagits bort i de senare versionerna av Javascript (och som därmed inte borde användas) kommer inte heller gå att använda.  
(<http://ejohn.org/blog/ecmascript-5-strict-mode-json-and-more/>)
- *addLetter* ska inte mellanlagra texten till en sträng eller skriva till paragrafens *innerHTML* (eller liknande-attribut), utan den ska skriva direkt till en textnod i paragrafen.
- Inlämningen måste fungera i Firefox (v10.0+).

## Givna filer

Tillsammans med uppgiften finns tre filer; `index.html` (skrivmaskinens HTML), `style.css` (skrivmaskinens CSS) och `script.js` (Javascript-filen ni ska skriva er kod i).

Observera att Javascript-filen innehåller en del `alert()` i funktionerna, dessa är endast plattshållare för att ni lättare ska hitta åt rätt funktion och börja skriva er kod, alla `alert()` i `script.js` ska tas bort innan inlämning.

## Inlämning

Lämnar in din `script.js` fil via kurssidan på Cambro.

Observera att det är viktigt att filen har exakt det namn som specificerats i det här dokumentet och att filnamnet bara ska anges med små bokstäver. Funktionernas namn får heller inte ändras. Observera också att filen inte ska packas i en zip-fil.

## Lämpliga avsnitt i kurslitteraturen

*Eloquent Javascript: Kapitel 1-4 samt 12.*

*På Internet det finns i klass också boken finns att hitta på pdf format*

 Javascript - Intro, Datatyper, Variabler, Arrayer, Funktioner, Parametrar, och Händelser

## Tips på funktioner

Nedan finns en lista på de funktioner som behövs för att lösa uppgiften. Börja med att sätta dig in i den givna koden. Se till att du förstår alla dess delar. Läs på om DOM-trädet. Trädet är centralt i den här uppgiften. Få sedan funktionen `addParagraph()` att fungera. Testa att skapa nya noder när knappen ENTER klickas. Texten du lägger till kan vara godtycklig. Som sista steg, få `addLetter()` att fungera.

## Användning

För att hitta en sektion med ett visst ID.

För att skapa en nya paragraf-nod

För att skapa en textnod till en paragraf-nod.

För att lägga till en textnod till en paragraf-nod, eller att lägga till en nod i DOM-trädet.

För att lägga till mer text till en textnod.

För att sätta ett attributs värde på en nod.

## Funktionens namn

`document.getElementById(id)`

`document.createElement(name)`

`document.createTextNode(text)`

`node.appendChild(node)`

`textnode.appendData(text)`

`node.setAttribute(name,value)`

När det står `node`, eller `textnode` framför funktionen ska man tänka på att detta är en variabel skapad i 1) eller en lokal variabel i funktionen av den datatyp som avses.

En textsträngs antal tecken finns i attributet `length`.

*Lycken till*

*Jay*