**Opgavesæt til skriftlig eksamen i Distribueret Programmering 21. juni 2016**

Vægtning af spørgsmål:

Opgave 1.1 3 points

Opgave 1.2 3 point

Opgave 1.3 5 points

Opgave 1.4 4 points

Opgave 2.1 4 points

Opgave 2.2 5 points

Opgave 2.3 6 points

Opgave 3.1 3 points

Opgave 3.2 5 points

Opgave 3.3 12 points

Opgave 3.4 3 points

Opgave 3.5 2 points

Opgave 4.1 10 points

Opgave 5.1 3 points

Opgave 5.2 7 points

Opgave 5.3 2 points

Opgave 5.4 4 points

Opgave 5.5 4 points

Opgave 6.1 3 points

Opgave 6.2 4 points

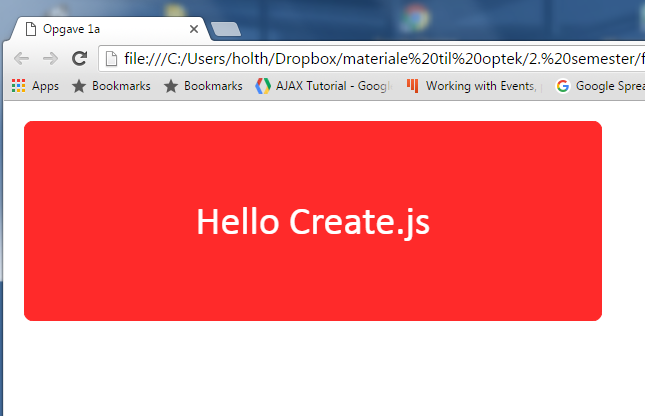
Opgave 6.3 3 points

Opgave 6.4 5 points

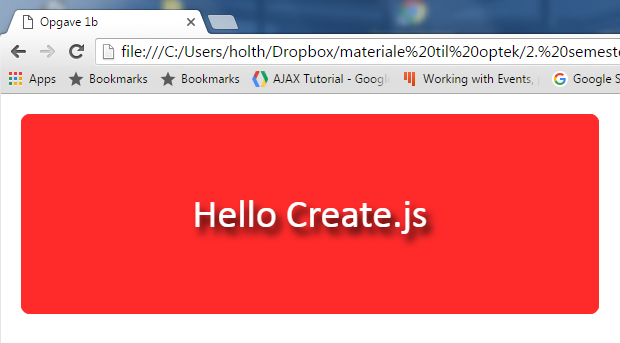
Alle hjælpemidler er tilladt. De nødvendige biblioteker ligger i directoriet lib. Nogle billeder, som skal bruges i opgave 3, ligger i biblioteket images. I gemmer jeres løsninger i den directory-struktur, som er defineret i den udleverede zip-fil. Husk at skrive navn i hvert enkelt dokument, I afleverer, fx i title-taggen.

**Opgave 1**

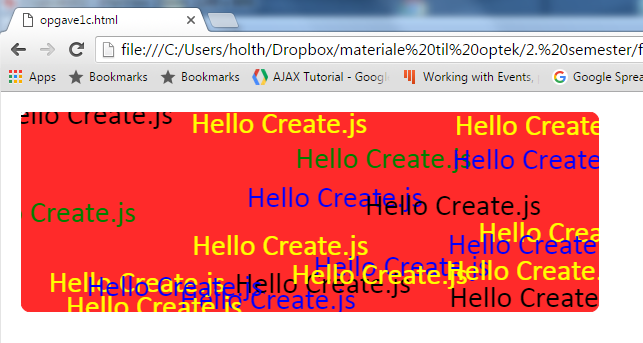
1. Tag udgangspunkt i det vedlagte html-dokument opgave1Template.html. Brug CreateJS til at programmere nedenstående billede (eller noget, der ligner nogenlunde). Teksten har font size 40px og font family Calibri. Teksten har farven hvid. Baggrundsfarven har værdien '#FF2A2A'. Bemærk at der ikke er markeret nogen border omkring firkanten. Gem din besvarelse i opgave1a.html.



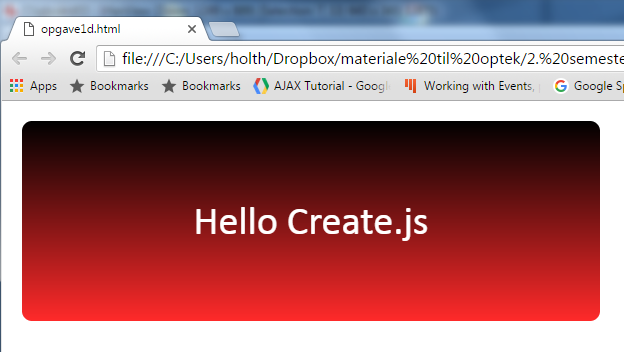
1. Tag udgangspunkt i opgave1a.html og tilføj en sort skygge til teksten, som vist på nedenstående screendump. Du skal ikke bestræbe dig på at ramme det grafiske udtryk på screendumpet præcist. Vær opmærksom på, at CreateJS ikke har nogen shadow-metode rettet specifikt mod tekst, men derimod arbejder med en generel shadow-klasse, som også kan anvendes på fx images (se fx side 80 i ”Beginning HTML5 Games with CreateJS”). Gem din besvarelse i opgave1b.html.



1. Tag igen udgangspunkt i opgave1a.html. Opret et array, hvis elementer er farverne ’black’, ’yellow’, ’blue’ og ’green’. Opret JavaScript-kode, som skriver teksten 20 gange i rektanglet. Teksten må gerne rage ud over rektanglet, og de enkelte eksemplarer må gerne overskrive hinanden. De enkelte tekster skal have tilfældige begyndelseskoordinater indenfor rektanglet. Tekstfarven skal i hvert eksemplar af teksten være en tilfældigt valgt farve i det oprettede farvearray. Fontstørrelsen er 30px. Nedenfor er vist et eksempel på, hvordan output kan være. Gem din besvarelse i opgave1c.html.

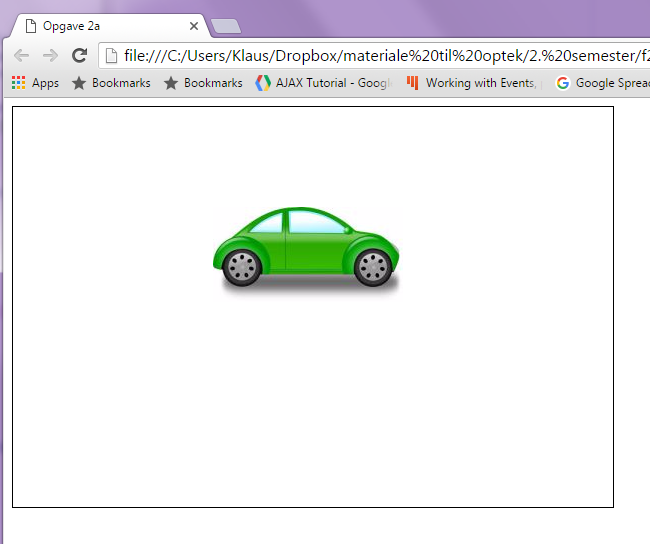


1. Tag igen udgangspunkt i opgave1a.html. Find fx ved en Google-søgning ud af, hvordan man laver en lineær farvegradient i CreateJS. Lav en lineær farvegradient, som løber lodret fra rektanglets øverste til dets nederste kant. Den øverste farve er ’#000000’, den nederste '#FF2A2A'. Nedenfor er output vist. Gem din besvarelse i opgave1d.html.

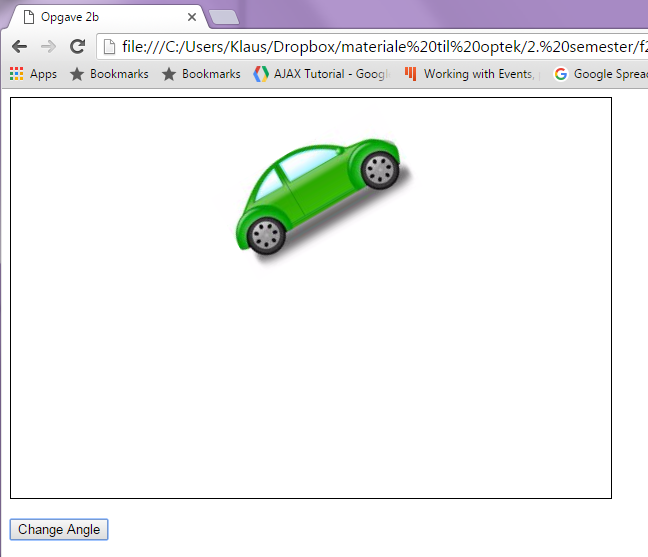


**Opgave 2**

1. I dokumentet opgave2Template.html er der en canvas-tag med id = ”canvas”. Tag udgangspunkt i dette dokument og brug CreateJS til at tegne billedet af bilen med source img/car.png i dette canvas. Billed-objektet skal have attributterne x sat til 200, y til 100 og rotation til 0, se nedenstående figur. Gem din besvarelse i opgave2a.html.

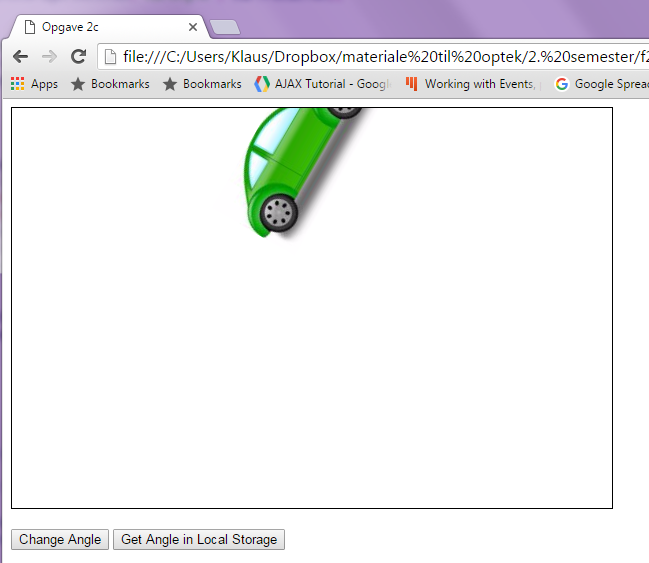


1. Tilføj en button til dokumentet og skriv ved hjælp af jQuery en kode, som bevirker, at bilens vinkel (rotation) vokser med 10 grader mod urets retning for hvert tryk på denne button, se nedenstående figur, hvor der er klikket tre gange på knappen. Gem din besvarelse i opgave2b.html.



1. Tag udgangspunkt i opgave2b.html og gem nedenstående ændringer til koden under navnet opgave2c.html. Skriv videre på koden til ”Change Angle”-knappen, så rotationsvinklen gemmes i local storage.

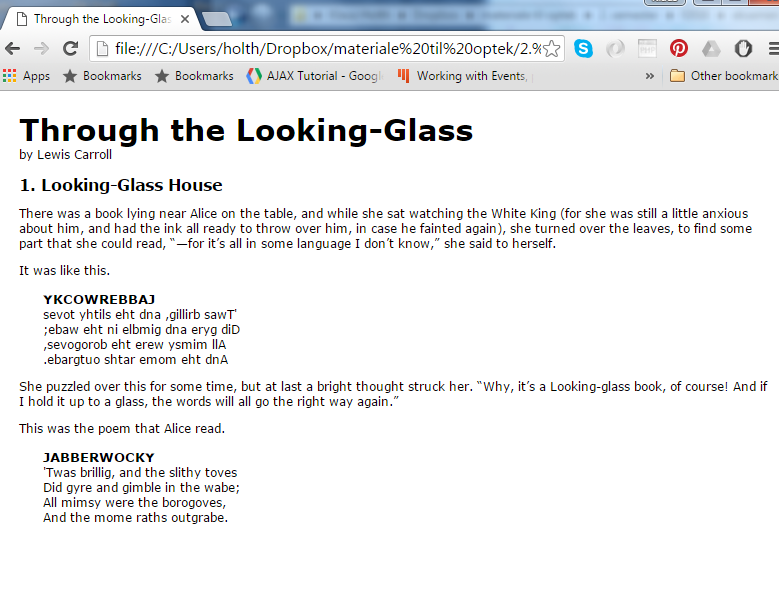
Tilføj nu en ny button, og skriv ved hjælp af jQuery en kode, som bevirker, at værdien for billedets vinkel hentes fra local storage. Vær opmærksom på, at local storage gemmer alle data i string format, så når vi skal bruge denne gemte værdi, skal den konverteres til integer igen. Find evt. ved hjælp af en Google-søgning ud af, hvilken JavaScript-metode der konverterer string til integer.



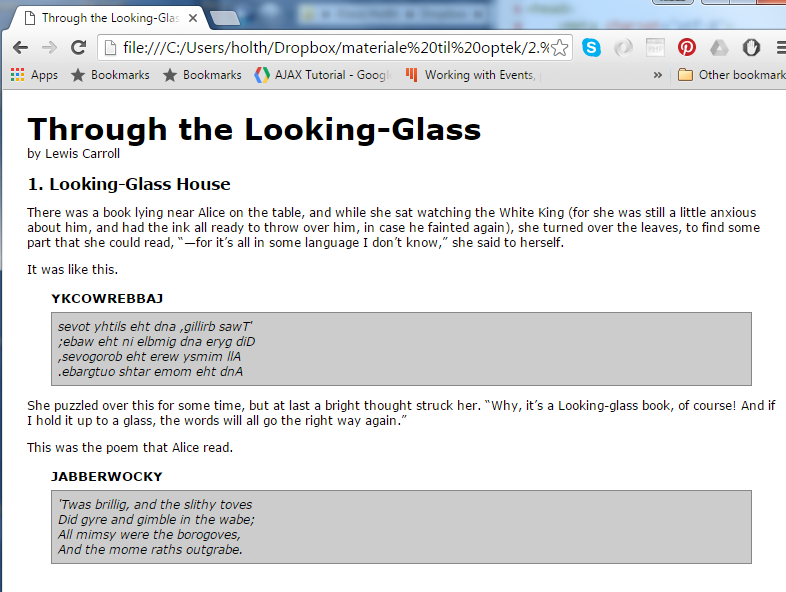
**Opgave 3**

Denne opgave drejer sig især om brugen af selectors. Til løsningen af de fleste punkter skal man bruge jQuerys css-metode.

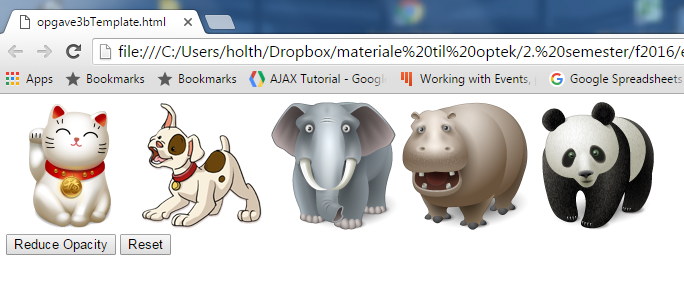
1. I dokumentet opgave3aTemplate.html er der et uddrag af ”Through the Looking-Glass” :



Brug jQuery til at aktivere stylingen af de to vers, så det kommer til at se således ud. Gem resultatet i opgave3a.html.

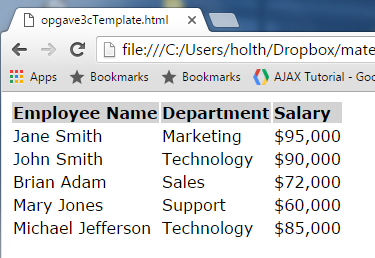


1. I dokumentet opgave3bTemplate.html er der vist fem billeder af nogle dyr. Der er også to buttons:

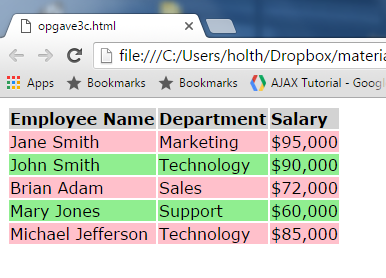


Brug jQuery til at knytte flg. funktionalitet til knapperne. Når man klikker på ”Reduce Opacity” skal opaciteten for alle billederne reduceres til 0,3. Når man klikker på ”Reset” skal den sættes til 1 igen. Gem din besvarelse i opgave3b.html.

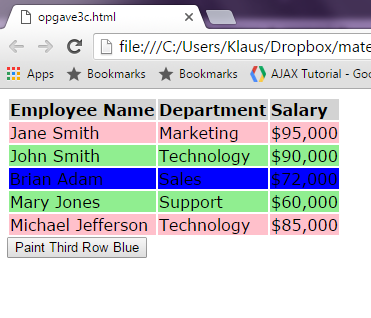
1. I dokumentet opgave3cTemplate.html er vist en tabel:



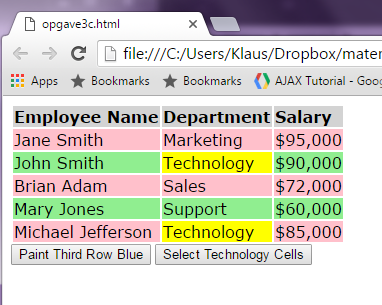
Brug jQuery til at ændre dette dokument, således at data-rækkerne skiftevis har farverne ”pink” og ”lightgreen”.



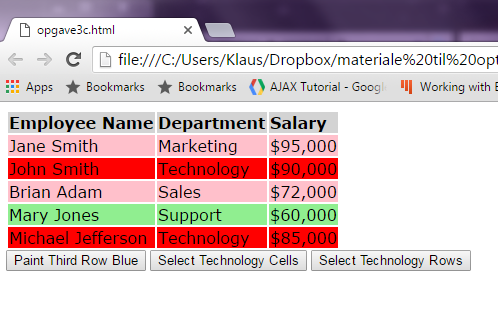
Tilføj en button med teksten Paint Third Row Blue. Når man klikker på denne knap skal tredie række farves blå:



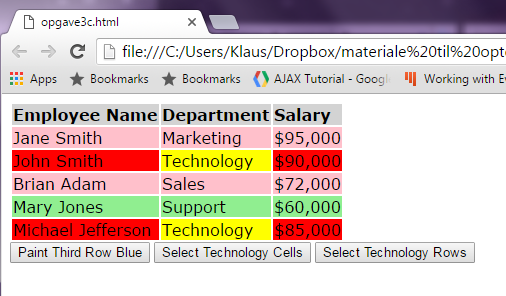
Undersøg, hvad jQuery-selectoren contains() gør. Opret dernæst en button, som farver *cellerne*, der indeholder teksten ”Technology” gule, når man klikker på den:



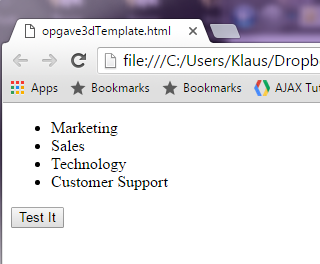
Opret dernæst en button, som farver *rækkerne*, der indeholder teksten ”Technology” røde, når man klikker på den:



Hvis man nu klikker på ”Select Technology Cells”, bliver de berørte celler som forventet farvet gule. Når man så atter klikker på ”Select Technology Rows”, forbliver de berørte celler gule. Hvordan kan dét være? Gem din besvarelse i opgave 3c.html.

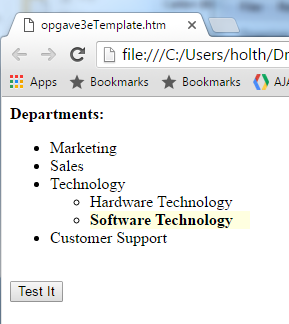


1. I opgave3dTemplate.html er vist en unordered list samt en button med teksten ”Test it”:

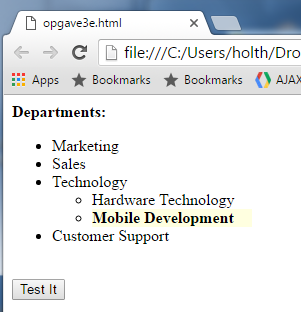


Skriv jQuery-koden, som bevirker, at der føjes punktet ”Implemenation” til listen. Skriv endvidere kode, så man kun kan føje ”Implementation” til listen én gang. Gem din besvarelse i opgave3d.html.

1. Dokumentet opgave6eTemplate.html giver flg. output:



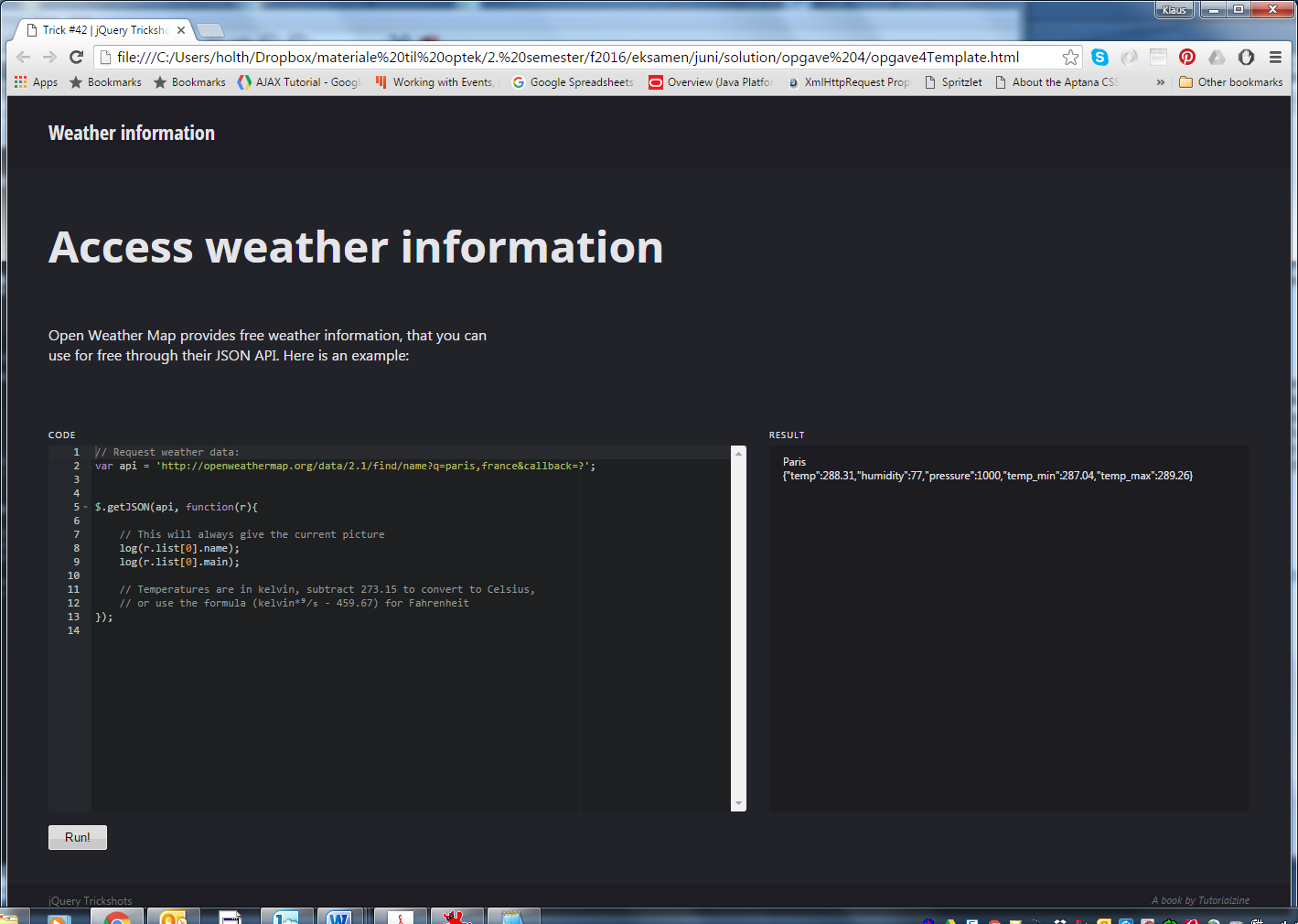
Skriv jQuery-koden, som bevirker, at punktet ”Software Technology” ændres til ”Mobile Development”, når man klikker på knappen med teksten ”Test it”. Gem din besvarelse i opgave3e.html.



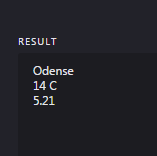
**Opgave 4**

1. Tag udgangspunkt i det vedlagte html-dokument opgave4Template.html.

Man kan frit hente oplysninger om vejret fra tjenesten openweathermap.org via dens api, på samme måde som man frit kan hente geografiske oplysninger fra Google Maps. Åbn opgave4Template.html og klik på knappen med teksten ”Run”. Du vil se en række oplysninger om vejret i Paris:

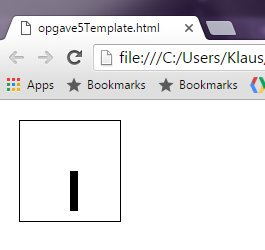


Gå ind på openweathermap.org og brug dokumentationen for api’et for ”Current weather data” på denne hjemmeside til at udskrive temperatur (i Celcius) og vindhastighed i Odense. Temperaturen skal angives afrundet til nærmeste hele tal. Gem din besvarelse i opgave4a.html. Bemærk, at de angivne værdier er øjebliksværdier, så dit resultat vil sikkert afvige fra det nedenstående.

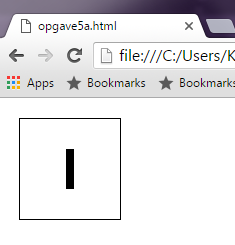


**Opgave 5**

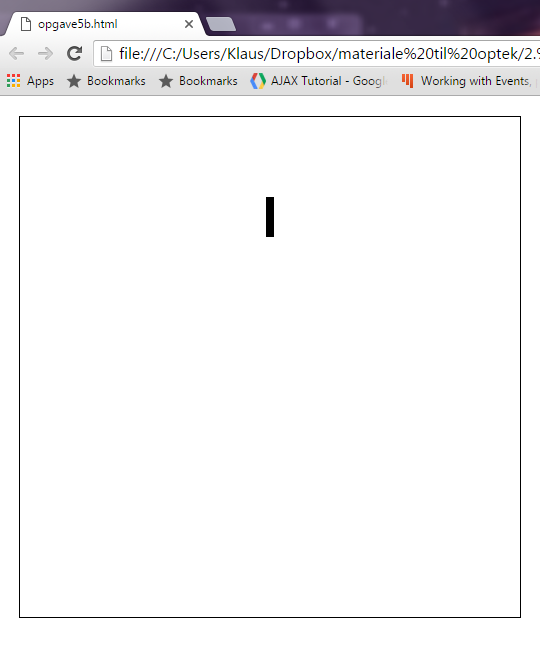
1. I dokumentet opgave5Template.html er der en canvas-tag med både width og height sat til 100 pixels. Ved hjælp af CreateJS er der tegnet et sort rektangel med width sat til 8 pixels og height til 40 pixels. Rektanglet har x-koordinaten sat til halvdelen af canvas-elementes bredde, og y-koordinaten sat til halvdelen af canvas-elementets højde:



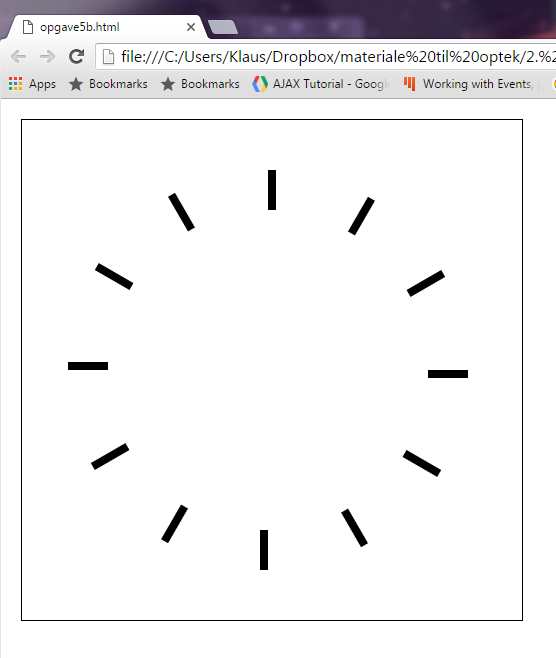
Forklar, hvorfor rektanglet ikke er centreret i canvas-elementet. Skriv dernæst den nødvendige kode til at centrere rektanglet. For en ordens skyld skal det anføres, at der besyndeligt nok ikke findes metoder i CreateJS til at finde en shapes width eller height! Gem løsningen i opgave5a.html:



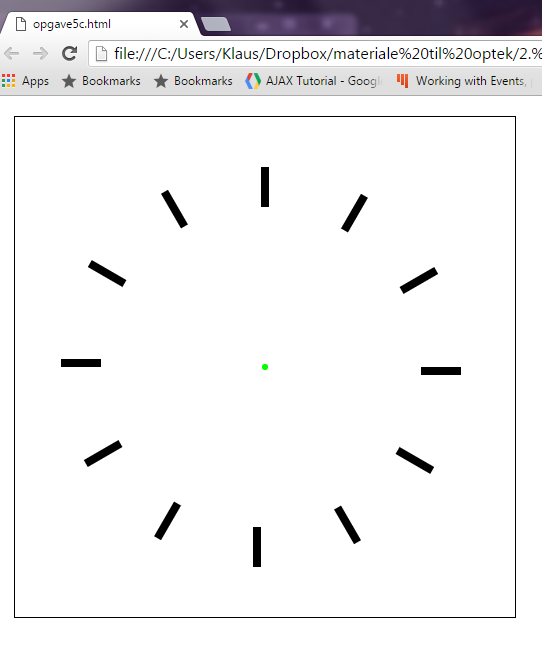
1. Kopiér opgave5a.html over i opgave5b.html og arbejd videre med denne. Sæt både canvas-elementets width og height til 500. Sæt y-koordinaten til rektangels registreringspunkt til 150:



Du skal nu konstruere en figur, som ligner en urskive. Hjælp: Brug clone()-metoden til at konstruere 12 eksemplarer at rektanglet, og rotér hvert eksemplar et passende antal grader. Det er også en fordel at lægge de 12 eksemplarer ind i et array.

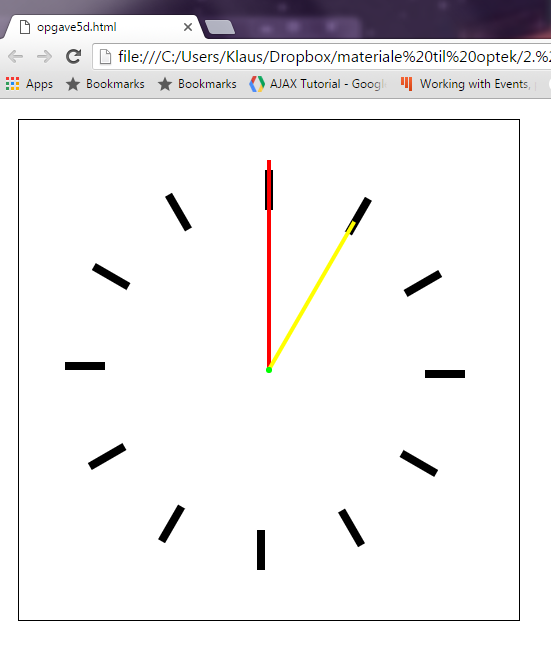


1. Anbring en grøn cirkel med radius 5 i midten af canvas-elementet og gem resultatet i opgave5c.html. Du kan godt løse opgave 5.3, 5.4 og 5.5 uafhængigt af opgave 5.2.

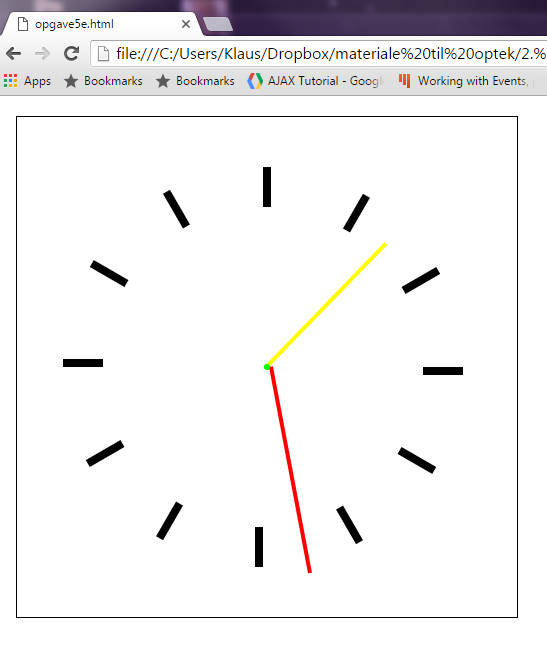


1. Kopiér opgave5c.html over i opgave5d.html. Konstruér en lang viser. Den skal være rød og have width 4 samt height 210. Registsreringspunktets x-koordinat er viserens width (4), det y-koordinat er viserens height (210).

Tegn dernæst en kort viser. Den skal være gul og have width 4 samt height 170. Registreringspunktets x-koordinat er viserens width (4), det y-koordinat er viserens height (170). Den er drejet til ”klokken 1”. Gem din besvarelse i opgave5d.html.



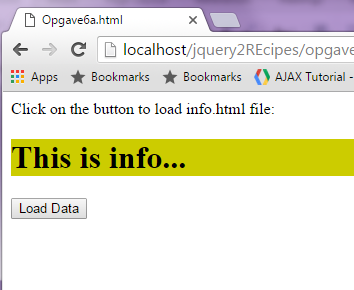
1. Lav to tweens. I den første skal den store viser dreje cirklen rundt på 5 sekunder. På den anden skal den lille viser dreje fra klokken 1 til klokken to også på 5 sekunder. Gem resultatet i opgave5e.html. Du skal ikke bestræbe dig på at få viserne til at rotere helt præcist om den grønne cirkel.



**Opgave 6**

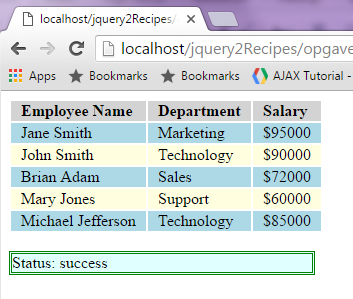
Disse opgaver drejer sig om ajax-metoden. Løsningerne skal derfor køres fra localhost. Kopiér hele directoriet opgave6 ind under localhost (fx AppServ/www) og løs opgaverne dér. Løsningerne skal selvsagt afleveres i den vedlagte directory-struktur under opgave6. Husk også at kopiere jquery.js med over til localhost.

1. I dokumentet opgave6aTemplate.html er der en knap med teksen ”Load Info”. Når man klikker på denne knap, skal indholdet af filen info.html skrives ved div-taggen med id = ”stage”:

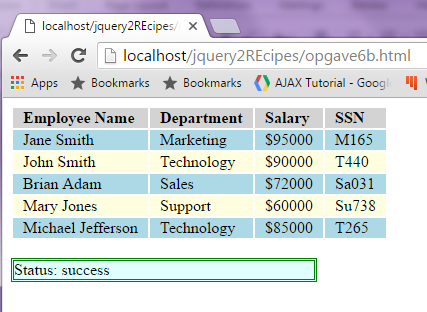


Brug jQuerys ajax load-metode til at implementere dette. Gem din besvarelse i opgave6a.html.

1. I dokumentet opgave6bTemplate.html bruges jQuerys $.ajax-metode til at læse nogle data fra dokumentet employees.json ind i en html-tabel:

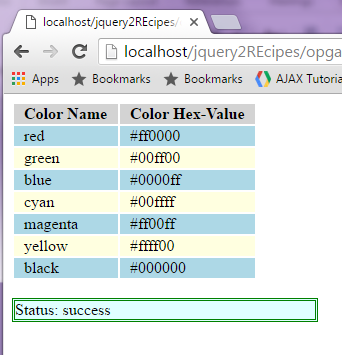


Tag udgangspunkt i disse filer til at modficere putput således:



Gem resultatet i opgave6b.html.

1. Modificér opgave6b.html så kun de rækker som indeholder oplysninger om ansatte i ”Technololgy” bliver udskrevet. Gem resultatet i opgave6c.html.
2. Skriv et html/js-dokument, som læser dokumentet colors.json med ajax-get med flg. output som resultat:



Gem resultatet i opgave6d.html.