**Opgavesæt til skriftlig eksamen i Distribueret Programmering 22. august 2016**

Vægtning af spørgsmål:

Opgave 1.1 4 points

Opgave 1.2 4 points

Opgave 1.3 4 points

Opgave 1.4 4 points

Opgave 1.5 4 points

Opgave 2.1 3 points

Opgave 2.2 3 points

Opgave 2.3 6 points

Opgave 2.4 4 points

Opgave 2.5 4 points

Opgave 3.1 3 points

Opgave 3.2 3 points

Opgave 3.3 2 points

Opgave 3.4 3 points

Opgave 3.5 3 points

Opgave 3.6 3 points

Opgave 3.7 3 points

Opgave 4.1 4 points

Opgave 4.2 5 points

Opgave 4.3 6 points

Opgave 4.4 5 points

Opgave 5.1 5 points

Opgave 5.2 5 points

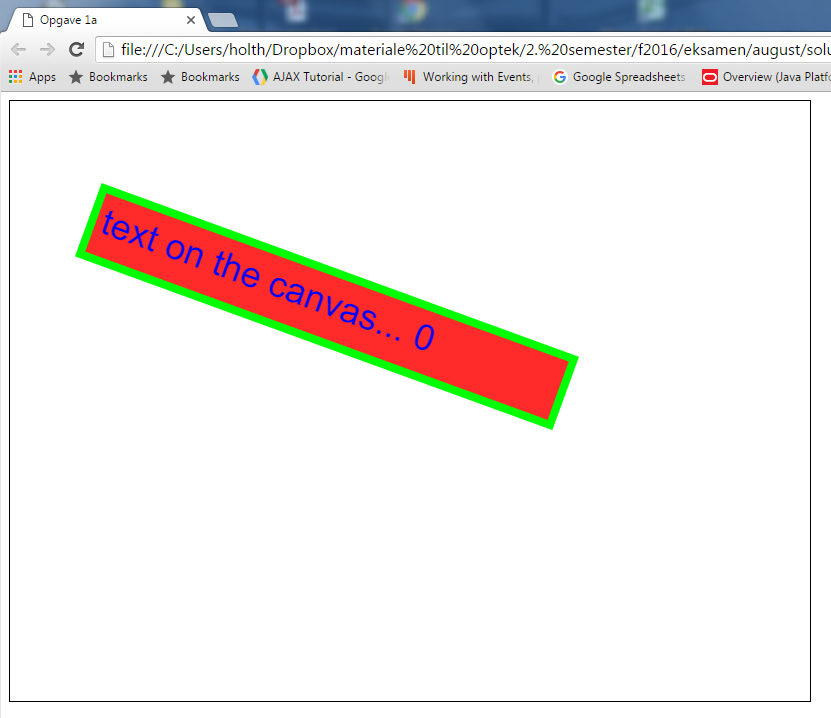
Opgave 5.3 5 points

Opgave 5.4 5 points

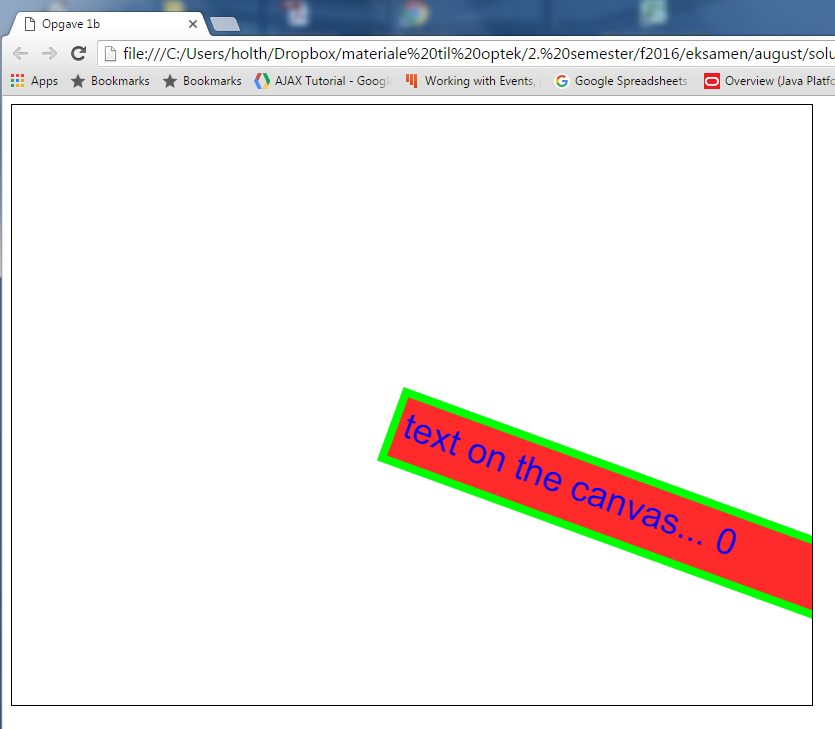
Alle hjælpemidler er tilladt. De nødvendige biblioteker ligger i directoriet lib. Nogle billeder, som skal bruges i opgave 3, ligger i biblioteket img. I gemmer jeres løsninger i den directory-struktur, som er defineret i den udleverede zip-fil. Husk at skrive navn i hvert enkelt dokument, I afleverer,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

**Opgave 1**

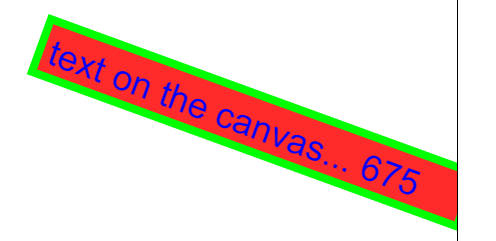
1. Tag udgangspunkt i det vedlagte html-dokument opgave1Template.html. Brug CreateJS til at programmere nedenstående billede (eller noget, der ligner nogenlunde). Teksten har font size 36px og font family Arial. Teksten har farven blå. Teksten har x-koordinaten 100 og y-koordinaten 100. Teksten er drejet 20 grader. Baggrunden har samme koordinater som teksten og er ligeledes drejet 20 grader. Baggrundsfarven har værdien '#FF2A2A'. Baggrunden har width 500 og height 70. Den grønne kant (stroke) omkring den røde firkant har width 8. Gem din besvarelse i opgave1a.html.



1. Tag udgangspunkt i opgave1a.html. Flyt nu både tekst og baggrund til midten af canvas. Du skal løse denne opgave således, at figuren automatisk flytter sig, hvis man ændrer på width og height for canvas-elementet. Gem din besvarelse i opgave1b.html.



1. Tag udgangspunkt i opgave1b.html. Easel.js har to metoder, som man kan bruge til at vurdere bredden og højden af tekst. De hedder getMeasuredWidth() og getMeasuredLineHeight(). Find dokumentationen (se fx i createjs.com) for disse to metoder og brug dem til at erstatte bredden på 500 og højden på 70 for den røde firkant med et udtryk, som dynamisk retter sig ind efter tekstens længde og skriftstørrelse. Gem din besvarelse i opgave1c.html.
2. Tag udgangspunkt i opgave1c.html. Hvis du ikke har løst opgave 1.3, kan du tage udgangspunkt i opgave1b.html. Indsæt en tæller i teksten, så den bliver opdateret én gang for hver frame, dvs efter fx 675 frames skal der stå ”text on the canvas… 675”. Gem din besvarelse i opgave1d.html.

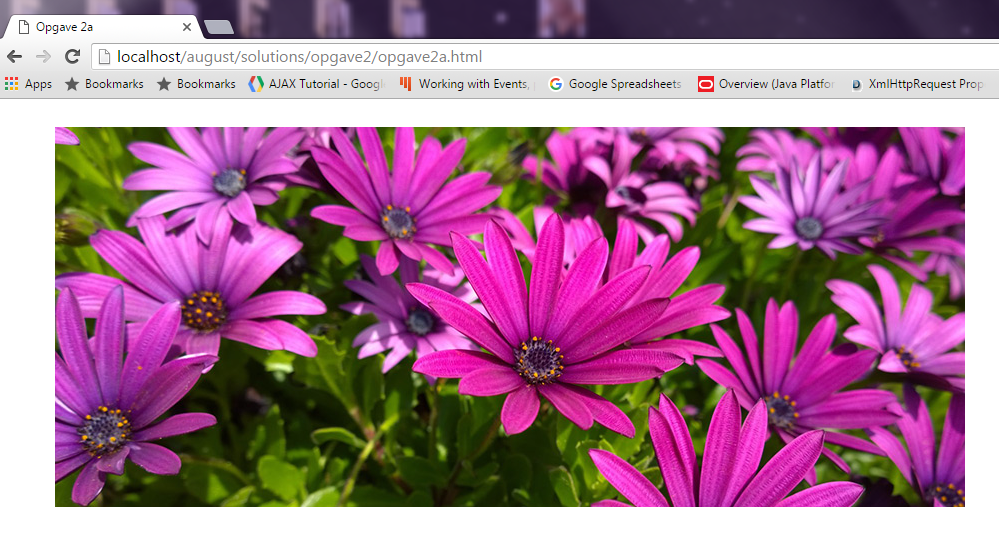


1. Tag udgangspunkt i opgave1d.html (eller eventuelt opgave1b.html). Opret en Container, som indeholder både teksten og baggrunden. Placér containeren midt i canvas. Lav en tween på containeren, så den drejer 10 grader med uret på 4000 millisekunder. Gem din besvarelse i opgave1e.html.

**Opgave 2**

1. Opgave 2.2 - 2.5 skal køres fra localhost.

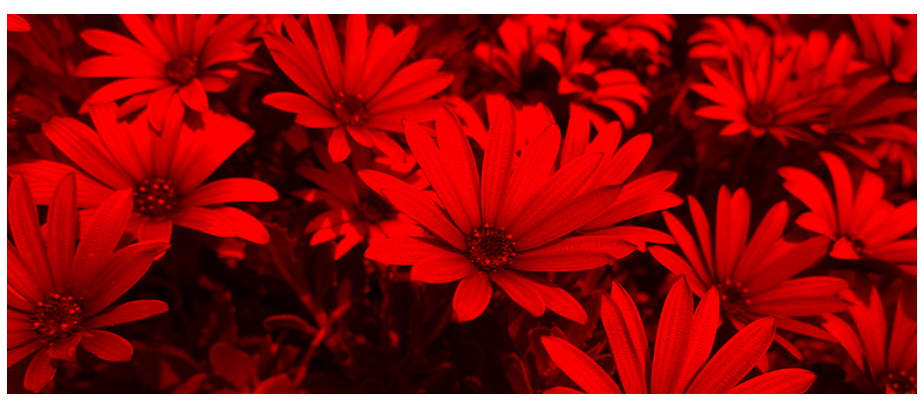
I dokumentet opgave2Template.html er der en canvas-tag med id = ”testCanvas”. Tag udgangspunkt i dette dokument og brug CreateJS til at tegne billedet af blomsterne med source img/flowers.jpg i dette canvas. Billed-objektet skal have attributterne x sat til 50 og y til 20, se nedenstående figur. Gem din besvarelse i opgave2a.html.



1. Tilføj et ColorFilter til billedet. Du kan se dokumentationen for dette filter på flg. link:

<http://www.createjs.com/docs/easeljs/classes/ColorFilter.html>

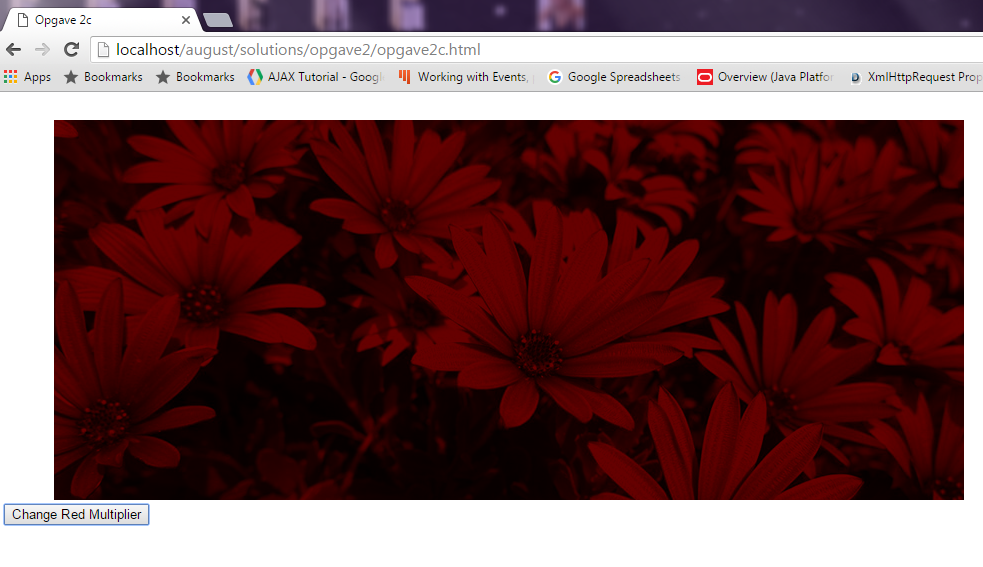
Sæt redMultiplier og alphaMultiplier til 1 og alle andre parametre til 0. Gem din besvarelse i opgave2b.html. Billedet skal nu se således ud:



1. Tag udgangspunkt i opgave2b.html og gem nedenstående ændringer til koden under navnet opgave2c.html.

Tilføj en button med teksten ”Change Red Multiplier” til dokumentet og lad programmet starte op med redMultiplier sat til 0.0. Herved bliver billedet helt sort. Når man klikker på knappen, skal redMultiplier øges med værdien 0.10, indtil redMultiplier får en værdi større end eller lig med 1. Herefter skal redMultiplier formindskes med værdien 0.10 for hvert klik på knappen. Dette forsætter indtil dens værdi bliver mindre end eller lig med 0, hvorefter der atter skal lægges 0.10 til osv.

For hvert klik skal billedet gentegnes. På nedenstående screendump er der klikket 4 gange.

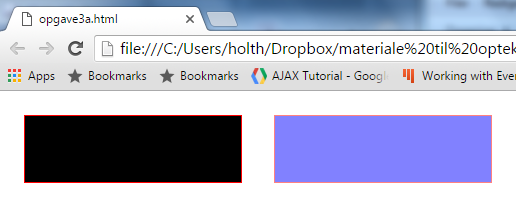


1. Tag udgangspunkt i opgave2c.html. Gem værdien af redMultiplier i local storage. Gem endvidere en oplysning, som fortæller, om redMultiplier er på vej op eller ned. Kald denne variabel for increaseRed. Gem din løsning i opgave2d.html.
2. Tag udgangspunkt i opgave2d.html. Skriv videre på programmet, således at det starter op med de værdier for redMultiplier og increaseRed, som det havde, da det sidst blev lukket. Vær opmærksom på, at local storage gemmer alle data i string format, så når vi skal bruge de gemte værdier, skal den konverteres til integer eller float (decimaltal) igen. Find evt. ved hjælp af en Google-søgning ud af, hvilken JavaScript-metoder der konverterer string til integer/float.

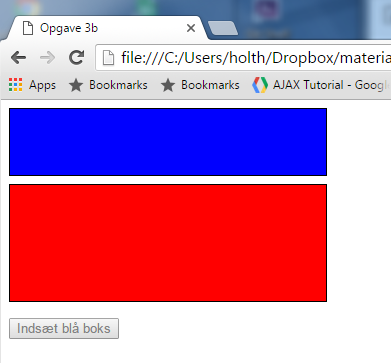
**Opgave 3**

Denne opgave drejer sig især om brugen af selectors. Til løsningen af nogle af punkterne skal man bruge jQuerys css-metode. Til hver opgave skal der tages udgangspunkt i et template-dokument, fx opgave3aTemplate.html.

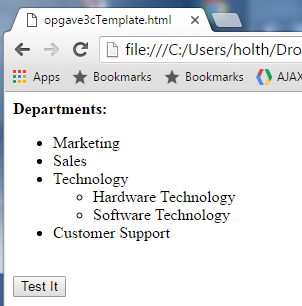
1. I dokumentet opgave3aTemplate.html er der to firkanter, en sort og en blå. Kopier templaten over i opgave3a.html og tilføj flg. funktionalitet. Når man klikker på den sorte firkant, skal den blå firkant i løbet af 3 sekunder fade til værdien 0.30. Når denne animation er afsluttet, skal den sorte firkant fjernes fra dokumentet, så den blå firkant rykker til venstre.



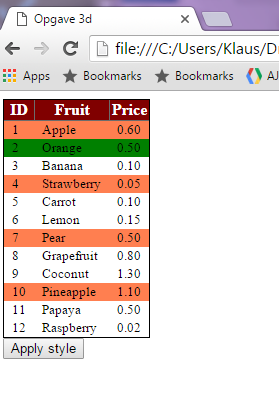
1. I dokumentet opgave3bTemplate.html er der vist en rød firkant samt en knap med teksten ”Indsæt blå boks”. Kopier templaten over i opgave3b.html og tilføj flg. funktionalitet. Når man klikker på knappen skal der indsættes en blå firkant oven over den røde. Stylingen af den blå firkant fremgår af templaten. Når der er klikket én gang på knappen, skal den disables.



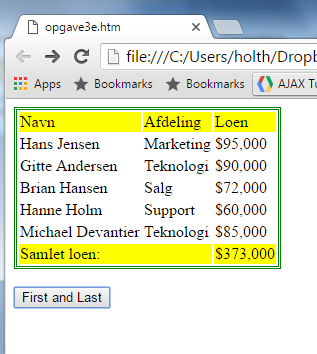
1. I dokumentet opgave3cTemplate.html er der vist en uordnet liste samt en knap med teksten ”Test it”. Kopier templaten over i opgave3c.html og tilføj flg. funktionalitet. Når man klikker på knappen, skal teksten ”Software Technology” skiftes ud med teksten ”Mobile Technology”.



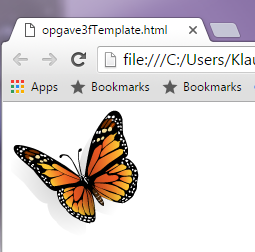
1. I dokumentet opgave3dTemplate.html er der vist en simpel tabel samt en knap med teksten ”Apply style”. Kopier templaten over i opgave3d.html og tilføj flg. funktionalitet. Når programmes startes, skal rækken med info om appelsiner (orange) tilddeles en grøn baggrundsfarve. Når man dernæst klikker på knappen skal hver tredie række i tabellen have en rødbrun baggrundsfarve (coral). Den nødvendige styling står allerede i templaten.



1. I dokumentet opgave3eTemplate.html er der vist en kort tabel samt en knap med teksten ”First and Last”. Kopier templaten over i opgave3e.html og tilføj flg. funktionalitet. Når man klikker på knappen skal første og sidste linie (incl. overskriften) farves gul.



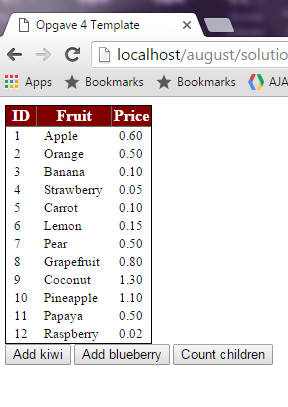
1. I dokumentet opgave3fTemplate.html er der vist et billede af en sommerfugl, ../img/butterfly3.png. Kopier templaten over i opgave3f.html og tilføj flg. funktionalitet. Lav en rollover på billedet, der skiftes ud med ../img/butterfly2.png, når musen føres ind over ../img/butterfly3.png. Når musen igen fjernes fra billedet, skal det oprindelige billede atter dukke op.



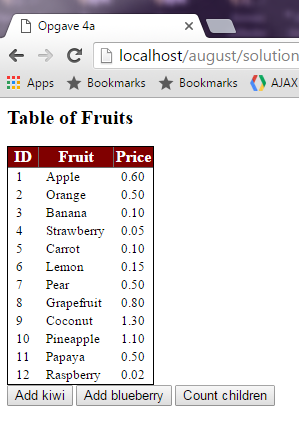
1. I dokumentet opgave3gTemplate.html er der vist en gul firkant med grøn kant samt en knap med teksten ”Animate”. Kopier templaten over i opgave3g.html og tilføj flg. funktionalitet. Når man klikker på knappen, skal firkanten bevæge sig 300 pixels til højre på 2000 millisekunder. Dernæst skal den bevæge sig 100 pixels ned, mens den samtidig fader til en opacity med værdien 0.40.

**Opgave 4**

1. Tag udgangspunkt i det vedlagte html-dokument opgave4Template.html, som viser en tabel med information om priserne på nogle frugter. Endvidere er der 3 buttons i dokumentet. Gem opgave4Template.html under navnet opgave4a.html og arbejd videre med denne. Temaet i denne opgave er Ajax og jQuery, og løsningerne skal derfor afvikles fra localhost.



Brug load()-metoden til at hente dokumentet headline.html ind i opgave4a.html (ved opstart af dokumentet), så vi får flg. resultat:



1. Gem opgave4a.html i opgave4b.html og arbejd videre med denne. Når man klikker på knappen med teksten ”Add kiwi”, skal man hente oplysningerne fra dokumentet kiwi.html ind i opgave4b.html, således at der tilføjes en ny række til tabellen. Vær opmærksom på, at man, inden man læser kiwi.html ind i dokumentet, skal oprette en passende tr-tag til at ”modtage” kiwi.html.



1. Gem opgave4b.html i opgave4c.html og arbejd videre med denne. Hvis du ikke kunne løse opgave 4.2, kan du i stedet arbejde videre med opgave4a.html. Når man klikker på knappen med teksten ”Add blueberry”, skal man hente oplysningerne fra dokumentet blueberry.json ind i opgave4c.html, således at der tilføjes en ny række til tabellen. blueberry.json er i JSON-format. Vær opmærksom på, at man, inden man læser blueberry.json ind i dokumentet, skal oprette en passende tr-tag til at ”modtage” blueberry.json.



1. Tag udgangspunkt i opgave4c.html eller opgave4a.html. Kopiér over i opgave4d.html og arbejd videre med denne. Det er uhensigtsmæssigt, at de forskellige tr-tags har id’er, hvis værdier er de samme som navnene på de relevante frugter. Skriv en metode, som giver ”systematiske” id’er til rækkerne, nemlig ”fruit0”, ”fruit1” osv. Metoden skal køres, når man klikker på knappen med teksten ”Count children”.

Hjælp: Udfyld de udkommenterede linier i nedstående.

$('#countChildren').click(function(){

$('#fruitTable').find('tr').each(function(rowIndex, row) {

//

//

});

});

**Opgave 5**

1. Tag udgangspunkt i det vedlagte html-dokument opgave5Template.html, som viser et canvas-element med width 800 og height 400. Gem opgave5Template.html under navnet opgave5a.html og arbejd videre med denne.

Undersøg dokumentationen for createjs’ metode til at tegne en radial gradient:

<http://www.createjs.com/docs/easeljs/classes/Graphics.html#method_beginRadialGradientFill>

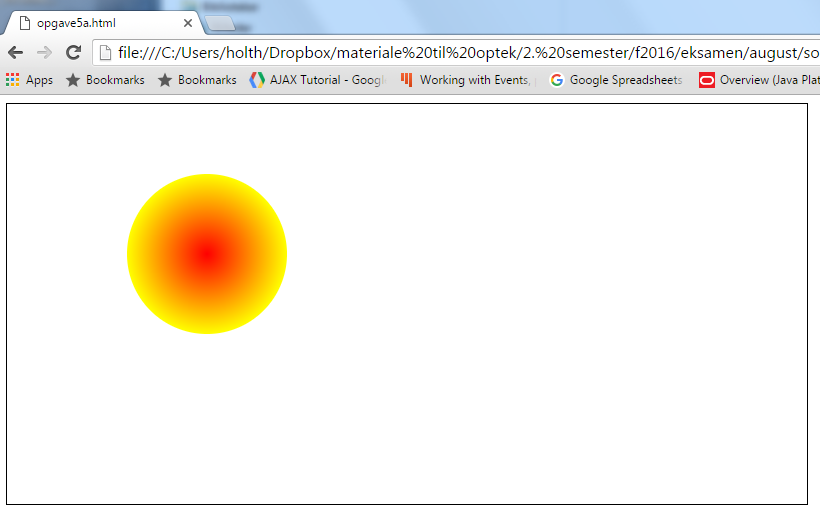
og tegn nedenstående cirkel (centrum: (200, 150), radius: 80) med flg. parametre:

farver: #FF0000 og #FFFF00

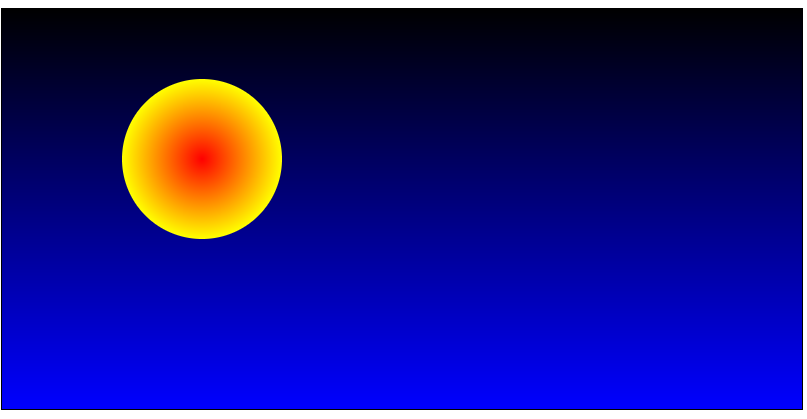
gradientpositioner: 0 og 1

x0, y0 r0: 200, 150, 0

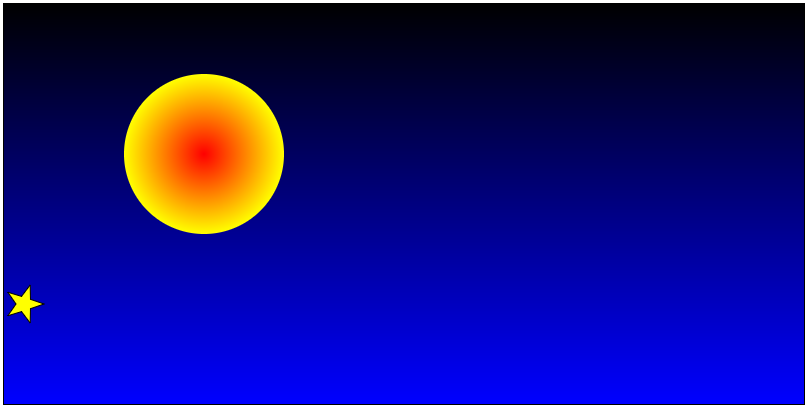
x1, y1, r1: 200, 150, 80



1. Kopiér opgave5a.html over i opgave5b.html. Hvis du ikke har løst opgave 5.1, kan du tage udgangspunkt i templaten. Arbejd videre med opgave5b.html. Lav en lineær gradient til et rektangel, som fylder hele canvas-elementet ud. Metoden, du skal bruge, hedder beginLinearGradientFill(). De to farver, som skal bruges, er #000000 og #0000ff. Resultatet skal se således ud:



1. Kopiér opgave5b.html over i opgave5c.html. Hvis du ikke har løst opgave 5.2, kan du tage udgangspunkt i templaten. Arbejd videre med opgave5c.html. Lav en femtakket stjerne og placér den på canvas med koordinaterne (20, 300). Stjernen behøver ikke at ligne den her viste præcist.



1. Kopiér opgave5c.html over i opgave5d.html og arbejd videre med denne. Lav en Tween på stjernen, således at den roterer 360 grader om sig selv, samtidigt med at den bevæger sig lineært fra (20, 300) til (780,100).

