# △ JS: Partikkel-animasjon

LAST NED PDF

## Introduction

I denne oppgaven skal vi bruke JavaScript til å få figurer vi å bevege seg. Vi skal altså lære å animere ved hjelp av JavaScript og noe som heter `Canvas`. Under ser du animasjonen vi kommer til å lage.

Denne oppgaven er den første i en liten serie av andre `partikkel`-oppgaver, derfor er det viktig å forstå det som skjer i denne oppgaven.

I denne oppgaven vil du få bruk for det du har lært i oppgaven Grunnleggende JavaScript.

## Steg 1: Canvas-elementet

I HTML bruker vi `<canvas>` til å tegne figurer ved hjelp av JavaScript. Selve `<canvas>`-elementet gjør ikke så stor nytte for seg, så derfor bruker vi JavaScript til å fortelle hva slags grafikk `<canvas>`-elementet skal inneholde. La oss skrive det som trengs for å jobbe med `canvas`:

- [ ] Åpne favoritt teksteditoren di
- [ ] Lag en ny HTML-fil som heter `partikler.html
- [ ] Kopier koden under inn i `partikler.html`:

```
<html>
<head>
    <meta charset="UTF-8">
    <title>Partikkel-fest</title>
    <style>
```

```
        body {
            background-color:#666;
        }

        #canvas {
            background-color:#000;
            margin-left:100px;
        }
    </style>

</head>
<body>
    <canvas id="canvas" width="500" height="500"></canvas>
</body>
</html>
```

# Forklaring: Canvas

- `<canvas id="canvas" width="500" height="500"></canvas>` er selve `Canvas`-elementet. Den har en gitt høyde og bredde `500px x 500px`. Vi skal bruke JavaScript til å lage andre elementer inne i `canvas`-elementet.
- I CSSen er det lagt til en `grå` bakgrunnsfarge til `<body>` og sort bakgrunnsfarge til `<canvas>`.

# Steg 2: Tegn et objekt

Nå som vi vet hvordan `canvas` ser ut er det på tide å prøve det ut:

- [ ] Sett inn `<script> </script>` i koden din
- [ ] Lag to tomme variabeler:

```
var canvas;
var ctx;
```

- [ ] Vi skal nå fylle disse variablene når siden vår lastes, da bruker vi noe som heter `window.onload`:

```
window.onload = function() {
    canvas = document.getElementById("canvas");
    ctx = canvas.getContext("2d");
}
```

`canvas`-variabelen holder nå på HTML-elementet vårt.

`ctx`-variabelen vil være det grafiske elementet som blir lagt til i `canvas`, dette elementet kan vi manipulere ved hjelp av stil, som vi skal se på snart.

For å kunne lage grafikk i `canvas` er de to linjene over påkrevd, så nå som vi har det på plass kan vi starte å tegne!

- [ ] Nå skal vi lage objekter, så la oss lære litt om hva et `objekt` er:

# Forklaring: Objekt

La oss nå lage et `objekt` som skal tegnes. I JavaScript er et objekt en variabel som kan holde på flere verdier eller variabler, som vi ofte kaller for `attributter`. La oss se på et raskt eksempel med en bil:

```
var bil = {
    navn: "Volkswagen",
    modell: "Golf"
    antallSeter: 5,
    farge: "Blå",
};
```

Vi kan enkelt hente ut informasjonen vi vil ha fra objektet ved å skive følgende:

```
console.log(bil.navn); // Skriver ut navnet på bilen: Volkswagen
console.log(bil.farge); // Skriver ut fargen på bilen: Blå
```

For å endre på ett av attributtene gjør vi bare følgende:

```
bil.farge = "Rød";
```

Nå vil attributtet `farge` bli endret fra `Blå` til `Rød`.

På denne måten slipper vi å lage mange variabler, som skal høre til samme element, vi bruker bare `objekter`.

- [ ] Lag et objekt som heter `particle` og som inneholder følgende attributter: `x`-posisjon, `y`-posisjon, `størrelse` og `farge`
- [ ] Bestem selv en passende verdi for attributtene. Disse kan være lurt å eksperimentere litt med senere i oppgaven.

HINT

- [ ] Lag en funksjon som heter `draw`. Denne skal tegne elementet for oss.
- [ ] I `draw` skal vi nå legge til hvilke farge vi vil at elementet vårt skal ha, du bestemmer selv hvilken farge:

```
ctx.fillStyle = particle.farge;
```

- [ ] Nå skal vi tegne et kvadrat (firkant hvor alle sidene er like lange) i fargen vi valgte over:

```
ctx.fillRect(particle.x,particle.y,particle.size,particle.size);
```

# Forklaring: ctx.fillRect()

`ctx.fillRect()` tar inn 4 variabler:

```
ctx.fillRect(x-posisjon, y-posisjon, bredde, høyde);
```

Over brukte vi de attributtene vi lagde i objektet `particle`.

I vårt objekt `particle` har vi satt en `x`- og `y`-posisjon, samt en størrelse som vi setter på både `bredde` og `høyde` for å få et kvadrat.

- [ ] Lagre og kjør funksjonen `draw()` når siden lastes.

Forslag til koden så langt:

```
CTYPE html>
<html lang="en">
<head>
```

```
    <meta charset="UTF-8">
    <title></title>
    <style>
        body {
            background-color: #666;
        }

        #canvas {
            background-color: #000;
            margin-left: 100px;
        }
    </style>
    <script>

        var canvas;
        var ctx;

        var particle = {
            x: 0,
            y: 0,
            size: 10,
            farge: "red"
        };

        window.onload = function() {
            canvas = document.getElementById("canvas");
            ctx = canvas.getContext("2d");
            draw();
        };


        //Tegner particle
        function draw() {
            ctx.fillStyle = particle.farge;
            ctx.fillRect(particle.x, particle.y,particle.size,particle.size);
        };

    </script>

</head>
<body>

<canvas id="canvas" width="500" height="500"></canvas>

</body>
</html>
```

## Steg 3: Flytt på partikkelen

Nå som vi har fått frem en rød firkant, som er partikkelen vårt, så skal vi nå se hvordan vi kan få den til å flytte på seg. For å få dette til å skje må vi legge til noen nye attributter i objektet vårt, og endre disse underveis i funksjonen vår. For å gjøre dette må vi lære å bruke `setInterval`, men først må vi endre på objektet vårt.

- [ ] I objektet `particle`, legg til attributtene `xSpeed` og `ySpeed
- [ ] Sett verdiene til `xSpeed` og `ySpeed` til å være `2` foreløpig

I `draw` må vi nå endre `particle` sin `x`-posisjon med `xSpeed`, samme må vi gjøre med `y`-posisjonen. Måten man øker et attributt på er slik:

```
objekt.attributt1 = objekt.attributt1 + objekt.attributt2;
```

- [ ] Legg til det som trengs i `draw` for å få `particle` til å endre `x`- og `y`-posisjonen sin

HINT

For at vi skal få en animasjon så må vi kjører `draw` flere ganger enn bare 1, derfor må vi bruke `setInterval` for å gjenta `draw`.

- [ ] Kjør funksjonen draw hvert 30 millisekund:

```
setInterval(draw, 30);
```

# Forklaring: setInterval

- `setInterval` kjører en funksjon hvert X millisekund.
- Altså betyr `setInterval(draw, 30);` at funksjonen `draw()` kjøres hvert 30 millisekund. NB! 1000 millisekunder er ett sekund.
- [ ] Fjern `draw()`, vi trenger ikke den lenger, ettersom `setInterval` vil kjøre `draw` for oss
- [ ] Lagre og kjør siden vi har laget til nå!

Som du ser så lager den en lang diagonal stripe. Som du kanskje har skjønt må vi finne en måte vi kan fjerne den forrige vi tegnet slik at vi skaper en illusjon om at den flytter på seg og ikke bare lager mange etter hverandre.

- [ ] I starten av `draw` må vi bruke `ctx.clearRect(0,0,500,500);` for å fjerne alt som er innenfor det svarte. Altså fra (`x`,`y`)-posisjonen (0,0) og helt til (500,500).
- [ ] Lagre og kjør på nytt!

**Gratulere du har laget din første animasjon i JavaScript!**

# Utfordring

- [ ] Prøv å få partikkelen til å gå rett frem
- [ ] Få partikkelen til å gå rett ned
- [ ] Få partikkelen til å gå baklengs
- [ ] Får du til at partikkelen bytter til en tilfeldig farge hver gang den bytter posisjon?

Ekssempel på ferdig kode til oppgaven:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title></title>
    <style>
        body {
            background-color:#666;
        }
```

```
        #canvas {
            background-color:#000;
            margin-left:100px;
        }
    </style>
    <script>

        var canvas, ctx;


        var particle = {
            x: 0,
            y: 0,
            xSpeed: 2,
            ySpeed: 2,
            size: 10,
            farge: "red"
        };

        window.onload = function() {
            canvas = document.getElementById("canvas");
            ctx = canvas.getContext("2d");
            setInterval(draw, 30);
        };


        //Tegner og skyter particle opp
        function draw() {

            ctx.clearRect(0,0,500,500);

            ctx.fillStyle = particle.farge;
            ctx.fillRect(particle.x, particle.y,particle.size,particle.size);

            particle.x = particle.x + particle.xSpeed;
            particle.y = particle.y + particle.ySpeed;

        }



    </script>

</head>
<body>

<canvas id="canvas" width="500" height="500"></canvas>

</body>
</html>
```