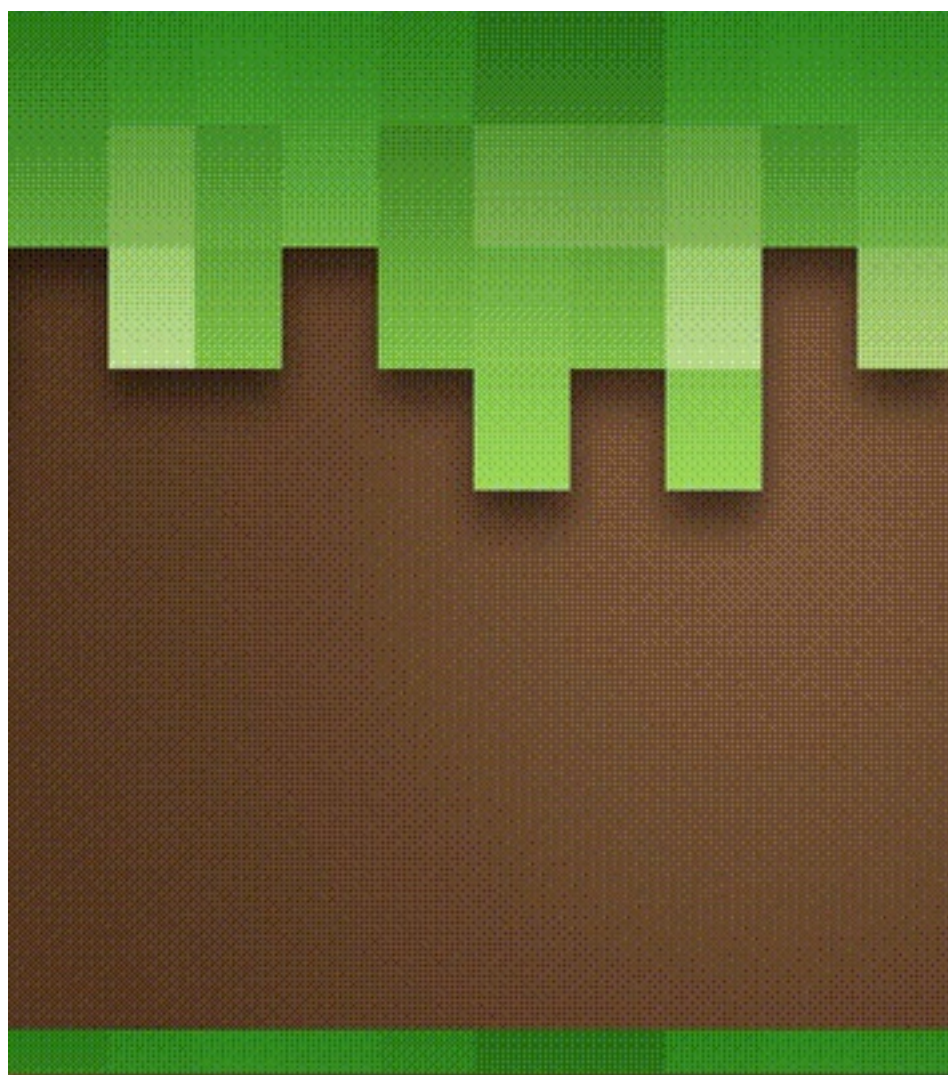


# </> CSS: Animasjon

## Introduksjon

I denne oppgaven skal du lære å animere HTML-objekter ved hjelp av



Men før vi starter å lage animasjonen over må vi lære om hvordan an med det grunnleggende!

**For å lære mest mulig bør du åpne en tom `.html`-fil og skrive kommer du til å bli en racer i CSS-animasjon!**

# Steg 1: Animasjons-attribu

Animasjon i CSS er ganske enkelt, i utgangspunktet har animasjonen kan du legge inn forskjellige faser, som vi skal se på litt senere. Animasjon med mindre du forteller at den skal gjøre det.

Før vi skal se på et enkelt eksempel skal vi se på `animation`-attributtet

```
#id {  
  animation-name: et-navn;  
  animation-duration: 1s;  
  animation-timing-function: linear|ease|ease-in|ease-out|ease-in-out;  
  animation-delay: 1s;  
  animation-iteration-count: nummer|infinite;  
  animation-direction: normal|reverse|alternate|alternate-reverse;  
  animation-fill-mode: none|forwards|backwards|both;  
}
```

- ☐ `name` : Navnet på animasjonen.
- ☐ `duration` : Hvor lenge (i sekunder) skal animasjonen vare.
- ☐ `timing-function` : Hvordan mellom-fasene er kalkulert.
- ☐ `delay` : Hvor mye forsinkelse det skal være før animasjonen starter.
- ☐ `iteration-count` : Hvor mange ganger skal animasjonen gjenta.
- ☐ `direction` : Bestemmer om animasjonen skal gå baklengs eller forlengs.
- ☐ `fill-mode` : Hvilke stiler som er lagt til før og etter start av animasjonen.

Her er et enkelt eksempel på en boks som går fra venstre til høyre:



```
<!DOCTYPE html>
<html>
<head>
<style>
  #boks {
    height: 50px;
    width: 50px;
    background-color: blue;
    position: relative;
    animation-name: frem-og-tilbake;
    animation-duration: 2s;
    animation-iteration-count: infinite;
    animation-direction: alternate;
  }
  @keyframes frem-og-tilbake {
    0% {
      left: 0px;
    }
    100% {
      left: 100px;
    }
  }
</style>
</head>
<body>
  <div id="boks"></div>
</body>
</html>
```

## La oss se nærmere på koden over:

Vi har en `<div>` med ID `boks`, den er 50x50px med blå bakgrunnsfar muligheter for å flytte på den.

`animation`-attributtene:



`name`: frem-og-tilbake

- ☐ `duration` : 2s (sekunder)
- ☐ `timing-function` : Ikke oppgitt, er `ease` som standard.
- ☐ `delay` : Ikke oppgitt, siden vi ønsker at animasjonen skal starte
- ☐ `iteration-count` : infinite (uendelig, så den vil ikke stoppe).
- ☐ `direction` : alternate (for at den skal gå frem og tilbake)
- ☐ `fill-mode` : Ikke oppgitt, ettersom animasjonen starter med en eller etter animasjonen.

`@keyframes frem-og-tilbake` er det vi bruker for å spesifisere hva som er animasjonens start og `100%` er animasjonens slutt. Derfor vil boksen flytte seg til høyre ( `left: 100px` ).

Innen for `@keyframes` kan vi nå spesifisere hva vi vil at animasjonen skal gjøre. `0%` er animasjonens start og `100%` er animasjonens slutt. Derfor vil boksen flytte seg til høyre ( `left: 100px` ).

**NB!** Verdiene i `animation`-attributtene kan også skrives som en egen linje, som er hva:

```
#boks {  
  animation: frem-og-tilbake 2s ...;  
}
```

## Utfordring

- ☐ Skriv koden inn i din favoritt teksteditor, lagre det som en `.css` fil.
- ☐ Få animasjonen til å bytte farge fra blå til rød underveis.
- ☐ Klarer du å få boksen til å flytte seg nedover og oppover?



Prøv å få boksen til å bevege seg i en firkant.

## Steg 2: @keyframes

La oss nå se nærmere på `@keyframes`. `@keyframes` er CSS som fortel

Her kommer noen eksempler:



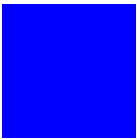
```
@keyframes diagonal {  
  0% {  
    top: 0px;  
    left: 0px;  
  }  
  100% {  
    top: 100px;  
    left: 100px;  
  }  
}
```

Dette eksempelet får et objekt til å gå diagonalt siden det starter på `left: 100px;`.



```
@keyframes ned {
  0% {
    top: 0px;
  }
  100% {
    top: 100px;
  }
}
```

Her går HTML-objektet nedover ved hjelp av `top`-attributtet.



```
@keyframes skift-farge {
  0% {
    background-color: blue;
  }
  50% {
    background-color: yellow;
  }
  100% {
    background-color: red;
  }
}
```

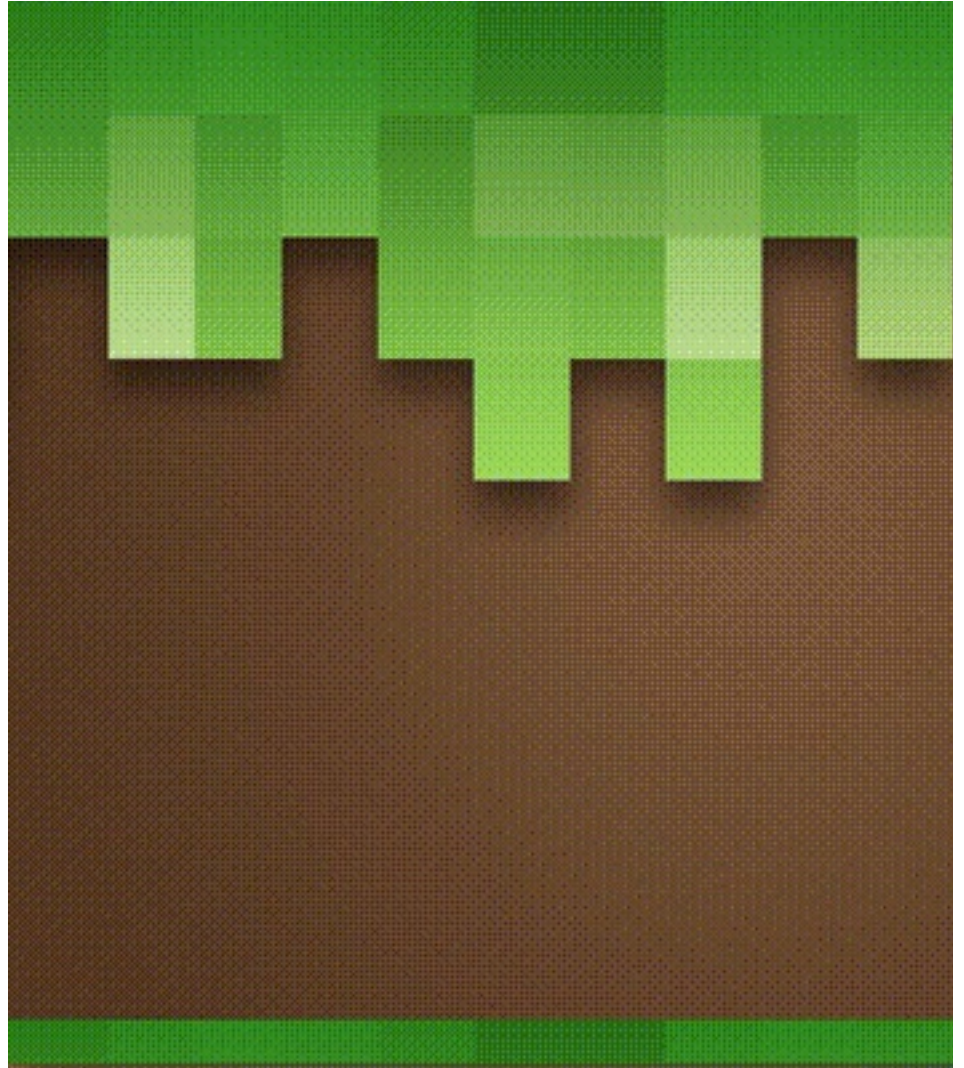
Merk at i dette eksempelet har vi lagt inn `50%`. Dette er et eksempel på å bruke `100%`. Du kan legge til så mange faser du vil ved å bruke `%`.

**Merk** at du ikke kan endre animasjonens varighet med `@keyframes` o

## Steg 3: Pakke ut filene

Nå skal vi animere øksen og Minecraft-logoen:





- ☐ Last ned og pakk ut [minecraft\\_animasjon.zip](#).
- ☐ Åpne `index.html` i din favoritt teksteditor og i en nettleser.

Du vil nå ha en nettside som ser noe sånt ut:



I koden til `index.html` har vi et bakgrunnsbilde og 3 div-er med følgende: ene er et bilde på nettsiden, bakgrunnsbildet ligger i CSS-en under `background-image`.

**Dette skal vi programmere:**

- ☐ `pickaxe`-en skal komme flyvende inn og treffe blokkene.
- ☐ Når øksen har truffet blokkene skal logoen skal komme inn.

## Steg 4: Flyvende øks



Nå skal vi få `pickaxe` -en til å fly. Vi begynner med å beskrive animasj

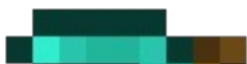
## ✓ Sjekkliste

- ☐ Lag en `@keyframes` med animasjonsnavnet `move-pickaxe`.
- ☐ La figuren starte utenfor skjermen. **Hint:** bruk en negativ verdi
- ☐ La `pickaxe` -bildet bevege seg bort til blokkene. Klarer du å finn
- ☐ Legg til rotasjon med `transform: rotate(antall grader)`.
- ☐ Kan du tenke deg hvor `transform: rotate()` bør være? I `0%` el
- ☐ Prøv deg frem med hvor mange grader du trenger for at den ska  
er to ganger rundt seg selv.

Så legger vi animasjonen til øksen.

## ✓ Sjekkliste

- ☐ Finn `#pickaxe` i CSS-en.
- ☐ Legg til animasjonsnavnet fra `keyframes` med `animation-name`
- ☐ Legg til `animation-duration` på `2s`.
- ☐ Legg til et `animation-delay` på `1s`.
- ☐ Sett `animation-timing-function` til `linear`.
- ☐ Sett `animation-fill-mode` til `forwards`.





Forslag til kode så langt

## Steg 5: Flyvende logo

Nå som du har klart å få `pickaxe` til å fly inn med rotasjon er oppgaven etter at `pickaxe` har stoppet.



Bruk det du har lært i oppgaven til å og prøv få logoen til å kom

Forslag til kode for Minecraft-logo

**Gratulerer!** Du har nå laget din første animasjon!

**Lisens:** [CC BY-SA 4.0](#)