CSS: Animasjon

Skrevet av: Lars Klingenberg

Oversatt av: Stein Olav Romslo

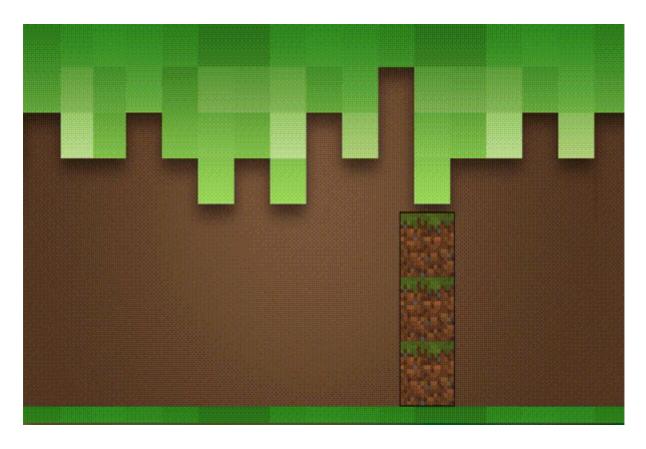
Kurs: Web

Tema: Tekstbasert, Nettside, Animasjon

Fag: Matematikk, Programmering, Kunst og håndverk Klassetrinn: 5.-7. klasse, 8.-10. klasse, Videregående skole

Introduksjon

I denne oppgåva skal du lære å animere HTML-objekt ved hjelp av CSS. Under ser du korleis resultatet vil bli til slutt:



Før me startar å lage animasjonen over må me lære om korleis animasjon fungerer ved hjelp av CSS. Så la oss starte med det grunnleggjande!

For å lære mest mogleg bør du åpne ei tom .html -fil og skrive koden for hand når du les oppgåva, då kjem du til å bli ein racer i CSS-animasjon!

Steg 1: Animasjons-attributten

Animasjon i CSS er ganske enkelt. I utgangspunktet har animasjonen to stadium: start og slutt. Mellom start og slutt kan du leggje inn ulike faser, det skal me sjå på seinare. Animasjonen vil heller ikkje få i loop (altså gjenta seg sjølv) med mindre du fortel at den skal gjere det.

Før me skal sjå på eit enkelt døme skal me sjå på animation -attributtar. Me skal bruke desse:

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

- name: Namnet på animasjonen.
- duration: Kor lenge (i sekund) skal animasjonen vare.
- timing-function : Korleis mellom-fasane er berekna.
- delay: Kor lang forseinking det skal vere f\u00far animasjonan startar. Standard er 0 sekund.
- iteration-count : Kor mange gonger animasjonen skal bli gjenteke.
- direction : Bestemmer om animasjonen skal gå baklengs eller ikkje.
- fill-mode: Kva stilar som er lagt til før og etter start av animasjonen.

Her er eit enkelt døme på ein boks som går frå venstre til høgre:

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

La oss sjå nærare på koden over:

Me har ein <div> med ID boks, den er 50x50px med blå bakgrunnsfarge. Posisjonen er relative, som vil seie at me har moglegheita til å flytte på den.

animation -attributtane:

- name: fram-og-tilbake
- duration: 2s (sekund)
- timing-function: Ikkje gjeve, er ease som standard.
- delay: Ikkje gjeve, sidan me vil at animasjonen skal starte med ein gong og standard er 0s.

iteration-count : infinite (uendeleg, så den vil ikkje stoppe).					
direction : alternate (for at den skal gå fram og tilbake)					
fill-mode: Ikkje gjeve, sidan animasjonen startar med ein gong og aldri sluttar treng me ikkje ein fill-mode før eller etter animasjonen.					
@keyframes fram-og-tilbake er det me brukar for å spesifisere kva som skal skje under animasjonen. I dette tilfellet har me sett namnet til animasjonen med animatior name: fram-og-tilbake, så me brukar @keyframes fram-og-tilbake for å beskrianimasjonen.					
No kan me spesifisere kva me vil at animasjonen skal gjere. Det gjer me innanfor @keyframes . Me har to fasar, ein start og ein slutt. 0% er starten på animasjonen og 100% er slutten. Difor vil boksen vår starte til venstre (left: 0px) og slutte lengre til høgre (left: 100px).					
NB! Verdiane i animation -attributtane kan òg skrivast som ei eiga linje, men då er det litt vanskelegare å finne ut kva som er kva:					
<pre>#boks { animation: fram-og-tilbake 2s; }</pre>					
Utfordring					
Skriv koden inn i favoritt-teksteditoren din, lagre den som ei .html -fil og gjer oppgåvene under.					
Få animasjonen til å byte farge frå blå til raud undervegs.					
Klarar du å få boksen til å flytte seg nedover og oppover?					
Prøv å få boksen til å bevege seg i ein firkant.					

Steg 2: @keyframes

La oss sjå nærare på @keyframes . @keyframes er CSS som fortel kva steg ein animasjon består av.

Her kjem nokre døme:

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

Dette dømet får eit objekt til å gå diagonalt sidan det startar på top: 0px; left: 0px; og ender på top: 100px; left: 100px;.

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

Her går HTML-objektet nedover ved hjelp av top -attributten.

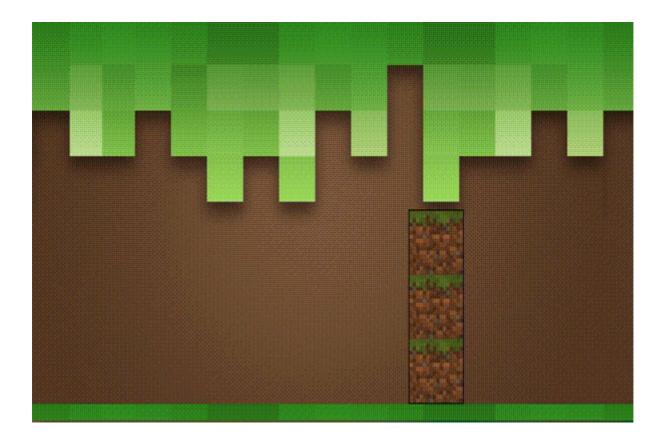
```
@keyframes skifte-farge {
    0% {
      background-color: blue;
    }
    50% {
      background-color: yellow;
    }
    100% {
      background-color: red;
    }
}
```

Merk at i dette dømet har me lagt inn 50%. Dette er eit døme på at du kan dele inn animasjonen i fasar mellom 0% og 100%. Du kan leggje til så mange fasar du vil ved å bruke %.

Merk at du ikkje kan endre på kor lenge animasjonen varar med @keyframes og %, då må du endre på animation-duration.

Steg 3: Pakke ut filene

No skal me animere øksa og Minecraft-logoen:



- Last ned og pakk ut minecraft_animasjon.zip (minecraft_animasjon.zip).
- Apne index.html i favoritt-teksteditoren din og i ein nettlesar.

No vil du ha ei nettside som ser slik ut:



I koden til index.html har me eit bakgrunnsbilete og 3 div-ar med følgjande ID: pickaxe, minecraft og block. Alle desse ID-ane er eit bilete på nettsida, bakgrunnsbiletet ligg i CSS-en under body.

Dette skal me programmere:

(pickaxe -a ska	al kome flygande	inn og treffe	blokkene.
7	p = 0.110.110 01 01.11			

Når øksa har treft blokkene skal logoen kome inn.

Steg 4: Flygande øks

No skal me få pickaxe -a til å fly. Me startar med å beskrive animasjonen med keyframes .



Lag ein @keyframes med animasjonsnamnet move-pickaxe.
La figuren starte utanfor skjermen. Hint: bruk ein negativ verdi av left.
La pickaxe-biletet bevege seg bort til blokkene. Klarar du å finne ut kor langt det er? Hint: positiv verdi av left.
Legg til rotasjon med transform: rotate(antall grader).
Kan du tenke deg kor transform: rotate() bør vere? I 0% eller 100%?
Prøv deg fram med kor mange gradar du treng for at den skal bli riktig. Hint: 360 gradar er ein gong, og 720 gradar er to gonger, rundt seg sjølv.

Så legger me til animasjonen til øksa.



Finn #pickaxe i CSS-en.			
Legg til animasjonsnamnet frå keyframes med animation-name	<u>.</u>		
Legg til animation-duration på 2s.			
Legg til eit animation-delay på 1s.			
Set animation—timing—function til linear.			
Set animation-fill-mode til forwards.			
Forslag til kode så langt			

Steg 5: Flygande logo

No som du har klart å få pickaxe til å flyge inn med rotasjon er den neste oppgåva di å få #minecraft til kome flygande inn etter at pickaxe har stoppa.

Bruk det du har lært i oppgåva til no, og prøv å få logoen til å kome inn når pickaxe er ferdig med animasjonen sin. Forslag til kode for Minecraft-logo

Gratulerer! Du har laga din fyrste animasjon!

Lisens: CC BY-SA 4.0 (http://creativecommons.org/licenses/by-sa/4.0/deed)