CSS Animationer

CSS Animations

Med CSS kan vi skapa animationer av HTML-element:

- Specificera hur ett element ska se ut under olika perioder i en animation
- Specificera hur lång animationerna ska vara
- Specificera hastighet, delay etc.

Bildlänk

NORTH LINK

Hur kan vi skapa animationer?

För att skapa en animation behöver vi använda oss av två saker

- **@keyframes** - För att skapa vår animation
- **Animation-properties** - För att definiera vad som ska animeras och hur det ska animeras.

@keyframes

Med @keyframes kan vi definera vilka CSS-properties som vi vill animera och hur de ska bete sig under animationens livslängd.

```
@keyframes resize {
  0% {
    height: 0;
    width: 0;
  }
  100% {
    height: 100px;
    width: 100px;
  }
}
```

animation

För att använda en animation behöver vi använda oss av följande animationsregler:

- **animation-name**
- **animation-duration**
- animation-timing-function
- animation-delay
- animation-iteration-count
- animation-direction

animation-name & animation-duration

Vårt minimikrav för att få en animation att fungera är att vi specificerar **animation-name** och **animation-duration.**

```
animation-name: resize;
animation-duration: 1s;
```

Dessa värden sätter vi på det element vi vill ska animeras.

Exempel

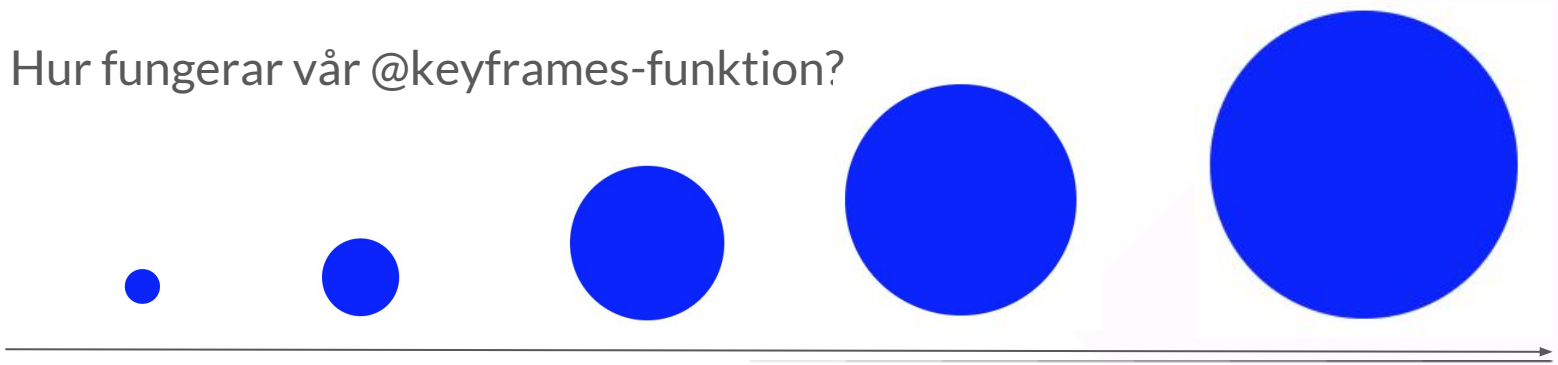
```
.circle {
  animation-name: resize;
  animation-duration: 1s;
  border-radius: 50%;
  background-color: blue;
}

@keyframes resize {
  0% {
    height: 0;
    width: 0;
  }
  100% {
    height: 100px;
    width: 100px;
  }
}
```

Vi använder oss av resize-animationen vi skrev tidigare för att animera en cirkel som växer.

Resultat

Animationens livslängd

Hur fungerar vår @keyframes-funktion?



0%	50%	100%
height: 0	height: 50px	height: 100px
width: 0	width: 50px	width: 100px

animation-timing-function

Hur gör vi om vi vill ändra hastigheten på animationen? Vi vill kanske att en animation ska gå snabbt i början men långsamt i slutet.

Då kan vi använda **`animation-timing-function`** och sätta något av följande värden:

- ease (standard)
- ease-in
- ease-out
- linear
- ease-in-out

NORTH LINK

Exempel

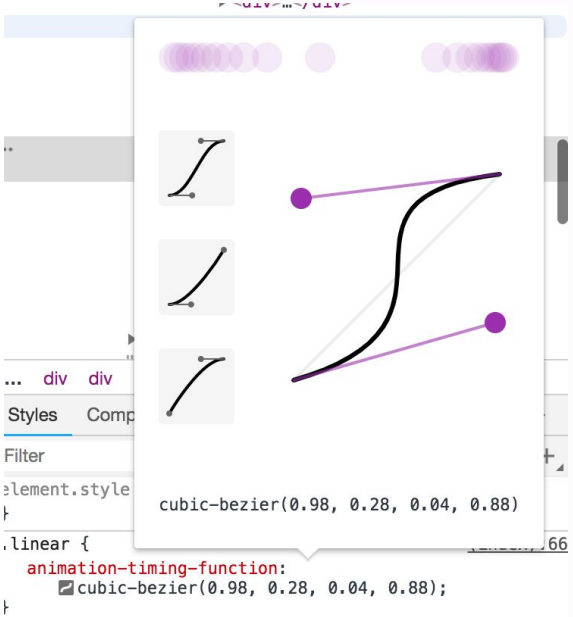
I följande exempel kan vi se ungefär hur de olika funktionerna beter sig:

Exempel

cubic-bezier

Vi kan också finjustera exakta hastigheter med hjälp av funktionen cubic-bezier

Exempel



NORTH LINK

animation-iteration-count

För att få vår animation att upprepa sig flera gånger använder vi oss av **animation-iteration-count**, som tar något av följande:

- animation-iteration-count: <number>
- animation-iteration-count: infinite

Exempel

animation-delay

Vi kan också fördröja våra animationer med hjälp av animation-delay

Exempel

animation-direction

Hittills har vår animation alltid börjat i 0%-läget (där vår cirkel inte syns alls) och slutat i 100%-läget.

Vi kan styra **riktningen** på våra animationer med följande värden:

- normal (standard)
- reverse
- alternate
- alternate-reverse

Exempel

animation-fill-mode

Slutligen så kan vi bestämma om vårt element ska behålla sitt värde från nåt av stegen i animationen när animationen inte körs med hjälp av **animation-fill-mode**

- forwards - behåller det sista tillståndet när animation är slut.
- backwards - Har det första tillståndet innan animation körs (vid t.ex. delay)
- Both - både forwards och backwards gäller

Exempel

animation

När vi applicerar en animation kan vi skriva det på en kortare form med hjälp av propertyn **animation**:

```
animation-name: resize;
animation-duration: 1s;
animation-timing-function: ease;
animation-iteration-count: infinite;
animation-direction: alternate;
```

=

```
animation: resize 1s ease infinite alternate;
```

NORTH LINK

Flera steg

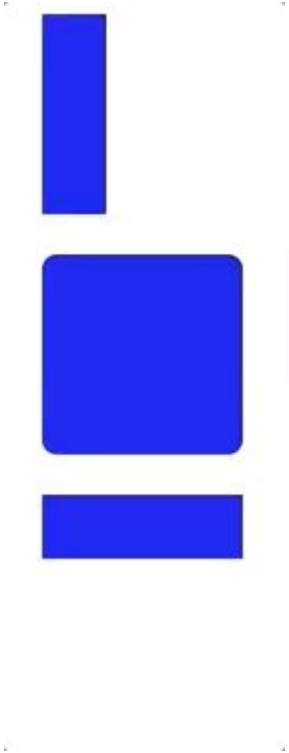
Vi kan också lägga in fler steg i vår animation:

```
@keyframes resize {
  0% {
    height: 0;
    width: 0;
  }
  50% {
    width: 200px;
  }
  100% {
    height: 100px;
    width: 100px;
  }
}
```

Exempel

Exempel

Övning



[Bildlänk]

NORTH LINK

Pulserande animation

Vi ska nu prova att skapa en animation med hjälp av det vi lärt oss.

Vår uppgift är att skapa en animation som får ett element att “pulsera”. Det ska gå att applicera den på olika element **oavsett storlek.**

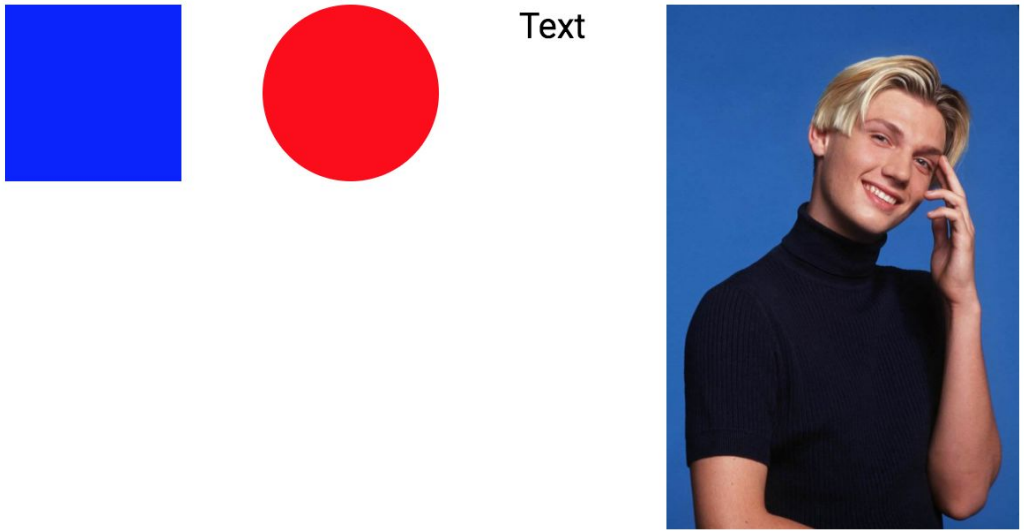
Bildlänk

NORTH LINK

Pulserande animation

Vi börjar med att skapa några element som vi vill få att pulsera:

Exempel



Transform + Animations = Sant

Vi kan använda oss av **transform**-funktionerna som vi lärt oss i tidigare föreläsningar.

I vårt exempel, där vi vill skapa en “pulserande” effekt, passar **scale** väldigt bra.

scale

Vi skriver följande för att få storleken att öka med 50% i slutet av animationen.

```
@keyframes pulse {
  0% {
    transform: scale(1);
  }
  100% {
    transform: scale(1.5);
  }
}
```

Pulse - animation

Nu vill vi applicera vår animation på våra element. Vi börjar med att lägga till animationen på alla elementen med följande kod:

```
animation-name: pulse;
animation-duration: 1s;
```

Exempel

Pulse - animation

För att få till den pulserande effekten vill vi att animationen ska upprepa sig. Vi vill också att riktningen ska alternera.

```
animation-name: pulse;
animation-duration: 0.5s;
animation-direction: alternate;
animation-iteration-count: infinite;
```

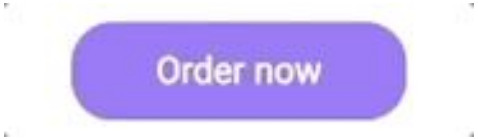
Exempel

NORTH LINK

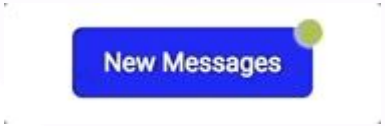
Pulse - animation

Nu har vi lyckats få till den pulserande effekten! Den kan vi använda den på alla möjliga element.

Bildlänk



Bildlänk



Bildlänk

NORTH LINK

Exempel 2

Vi ska nu prova att skapa en animation där solen går upp och ner vid ett berg.

Bildlänk

NORTH LINK

Exempel 2

Vi börjar med att skapa upp våra element utan animationer.

```
<div class="container">
  <div class="sun">
    <img src="https://i.imgur.com/YfjkqF9.png" />
  </div>
  <div class="mountain">
    <img src="https://i.imgur.com/G9t4d5K.png" />
  </div>
</div>
```

Exempel

Exempel 2

Eftersom vi vill att solen ska röra sig i både x- och y-led provar vi att använda transform: translate

```
@keyframes movement {
  0% {
    transform: translate(0, 0);
  }
  25% {
    transform: translate(25%, 100%);
  }
  50% {
    transform: translate(50%, 0);
  }
  75% {
    transform: translate(75%, 100%);
  }
  100% {
    transform: translate(100%, 0);
  }
}
```

Exempel 2

Nu lägger vi på vår animation. Eftersom solen ska röra sig lika snabbt över x-axeln vill vi använda **linear** som vår timing-funktion.

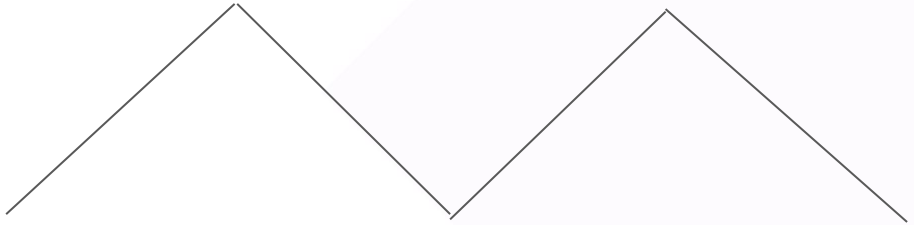
```
.sun {
  animation-name: movement;
  animation-duration: 6s;
  animation-timing-function: linear;
  animation-iteration-count: infinite;
}
```

Exempel

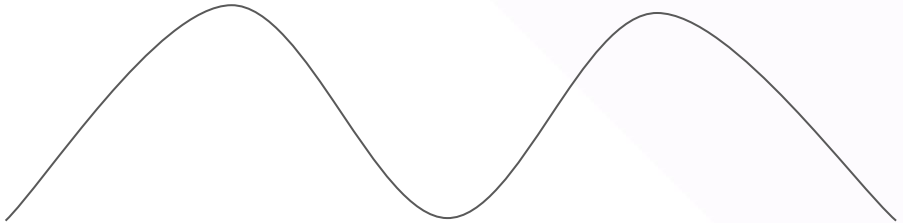
Exempel 2

Solen rör sig upp och ner samtidigt som den rör sig åt höger, men den beter sig inte riktigt som vi vill.

Så här rör den sig nu:



Så här vill vi att den ska röra sig:



NORTH LINK

Exempel 2

Med andra ord så vill vi att den rör sig linjärt i x-led med i y-led vill vi att den rör sig mjukare vid topparna och dalarna, t.ex. med **ease-in-out.**

Tyvärr kan vi inte lägga två animationer på samma element eftersom båda använder transform, men vi kan ge bildens div-container en animation och vår bild en annan.

Exempel 2

Vi skapar följande animationer:

```
@keyframes horizontal {
  0% {
    transform: translateX(0);
  }
  100% {
    transform: translateX(100%);
  }
}
```

```
@keyframes updown {
  0% {
    transform: translateY(0);
  }
  25% {
    transform: translateY(100%);
  }
  50% {
    transform: translateY(0);
  }
  75% {
    transform: translateY(100%);
  }
  100% {
    transform: translateY(0);
  }
}
```

Exempel 2

Och vi applicerar dem på följande sätt:

```
.sun {
  animation-name: horizontal;
  animation-duration: 6s;
  animation-timing-function: linear;
  animation-iteration-count: infinite;
}
```

```
.sun img {
  width: 50px;
  animation-name: updown;
  animation-duration: 6s;
  animation-timing-function: ease-in-out;
  animation-iteration-count: infinite;
}
```

Exempel 2

I vårt slutgiltiga resultat har vi nu en bättre kurva

Resultat

Exempel 3 - Loader

Vi ska nu prova att skapa en loader-animation utifrån en bild som vi vill få att snurra.

NORTH LINK

Exempel 3 - Loader

För att få till att bilden snurrar använder vi **transform: rotateZ**

```
@keyframes spin {
  0% {
    transform: rotateZ(0);
  }

  100% {
    transform: rotateZ(360deg)
  }
}
```

NORTH LINK

Exempel 3 - Loader

Sedan applicerar vi animationen på vår bild

```
img {
  width: 50px;
  animation: spin 1s infinite linear;
}
```

Exempel 3

Resultat

Exempel 4 - Loader

Loaders går också att skapa helt utan bilder med animationer och kan se ut på många olika sätt.

Exempel

Animate.css

Animate.css är ett bibiotek som innehåller färdiga animationer som vi kan använda på våra HTML-element.

Genom att lägga på klasser (likt bootstrap) kan vi använda de animationer som finns i biblioteket.

Animate.css

Några av de animationer som ingår är:

- Flash
- FadeOut
- FadeIn
- Bounce
- SlideIn

[Animate.css på Github]

Animate.css

Exempel

Övningar

https://github.com/LinkNorth/HTMLCSS/tree/master/exercises/week7/animations