

Media, Nyheter i CSS och Repetition

Översikt

- Animationer
- Media
 - video
 - audio
- Nyheter i CSS
 - CSS-variabler
 - display: contents



Animationer - Övning

Vi ska prova att lösa följande uppgift:



Animationer - Övning

Vi börjar med att lägga upp en grundstruktur med en bild och en ruta som kan täcka bilden.

```
<div class="box"></div>
<div class="container">
  
</div>
```

```
.box {
  position: absolute;
  width: 10px;
  height: 300px;
  background-color: black;
}
```

Exempel

Animationer - Övning

Vi kan identifiera 3 olika tillstånd i vår animation:



1



2



3

Animationer - Övning

Först bestämmer vi bredden på rutan i de olika tillstånden.



width: 0%



width: 100%



width: 0%

Animationer - Övning

Vi ser även att rutans position förändras från höger till vänster. Vi kan därför specificera detta med **left**.



width: 0%

left: 0%



width: 100%

left: 0%



width: 0%

left: 100%

Animationer - Övning

Vi skapar därför följande animation:

```
@keyframes slide {  
  0% {  
    width: 0;  
    left: 0;  
  }  
  50% {  
    width: 100%;  
    left: 0;  
  }  
  100% {  
    width: 0;  
    left: 100%;  
  }  
}
```


Animationer - Övning

Sedan applicerar vi den på vårt element.

```
animation: slide 2s infinite;
```

Exempel

Animationer - Övning

Slutligen vill vi att den ska alternera, så vi lägger på **alternate**

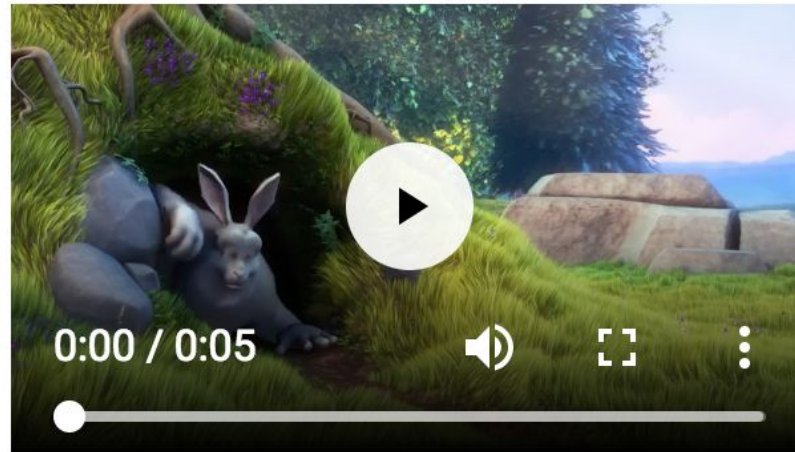
Slutligt resultat

Media

Vi ska nu ta en titt på några av de media-taggar som vi kan använda i HTML, nämligen `<video>` och `<audio>`.

Video

HTML5-elementet **<video>** kan vi använda för att lägga in videoklipp på en sida.



Exempel

Video

Exempel

Video

Några av de attribut vi kan sätta på våra video-element inkluderar följande:

- controls
- autoplay
- height/width
- muted
- loop
- poster

Video - Autoplay

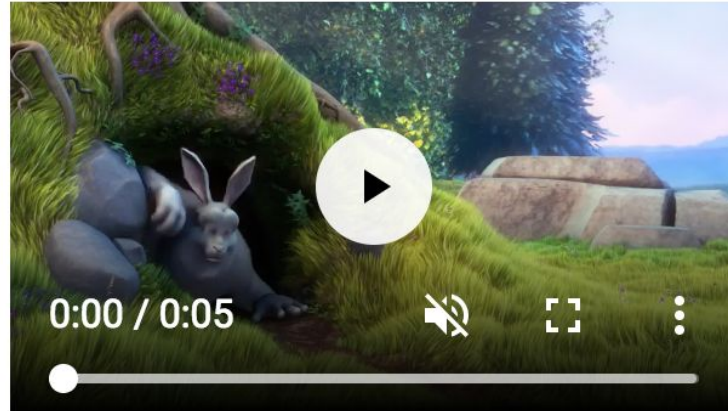
Vi kan sätta autoplay om vi vill att en film ska starta automatiskt när den är laddad:

```
<video
  controls
  autoplay
  src="https://www.sample-videos.com/video/mp4/720/big_buck_bunny_720p_1mb.mp4"
  width="300">
  Your browser doesn't support embedded videos.
</video>
```

Exempel

Video - Muted

Muted kan vi använda om vi vill att ljudet ska vara avstängt på filmen.



Exempel

Video - Poster

Poster kan vi använda för att lägga in en bild på som syns på förhandsgranskningen.

```
<video
  controls
  poster="https://i.ytimg.com/vi/tN1A2mVnr0M/maxresdefault.jpg"
  src="https://www.sample-videos.com/video/mp4/720/big_buck_bunny_720p_1mb.mp4"
  width="300">
  Your browser doesn't support embedded videos.
</video>
```

Exempel

Video - Loop

Vi kan också använda loop om vi vill att vår film ska spelas om och om igen.

[Exempel](#)

Video - Filformat

Alla filformat fungerar inte på alla browsers, det finns en lista här:

https://developer.mozilla.org/en-US/docs/Web/HTML/Supported_media_formats

Video

Vi kan också skriva vilken film vi vill visa med hjälp av <source>-taggar:

```
<video>  
  <source src="big-buck-bunny.mp4" type="video/mp4">  
  <source src="big-buck-bunny.webm" type="video/webm">  
</video>
```

Browsern kommer spela den första filen, men ifall filformatet inte stöds kommer den prova att spela den andra filen.

Audio

Vill vi lägga in ljudfiler kan vi använda oss av <audio>-taggen.



Audio

Med audio kan vi specificera liknande attribut som med video. T.ex. kan vi använda **controls**

```
<audio controls>  
  <source src="https://www.sample-videos.com/audio/mp3/crowd-cheering.mp3" type="audio/mp3">  
</audio>
```

Exempel

Audio

Använder vi inte **controls** kommer användaren inte kunna styra över ljudet (och därmed inte starta spelningen). Däremot kan vi fortfarande lägga på **autoplay** vilket kommer göra att ljudet börjar spela, även om vi inte kan se kontrollerna.

Sidor där ljud börjar spela automatiskt brukar dock sällan uppskattas av användarna.

Exempel

Nytt i CSS

- CSS Grid
- object-fit
- display: contents
- CSS-variabler

display: contents

Med hjälp av `display:contents` så kan vi få ett element som håller ett eller flera andra element att “försvinna”.

När vi sätter `display:contents` på ett element visas inte det elementet längre, men barnen till det elementet kommer istället visas i dess plats.

Exempel: display: contents

Vi utgår från följande kod, där vi har ett antal boxar som är flex-items. Dock är två av boxarna inuti en wrapper, vilket gör att wrappern blir vårt flex-item istället.

```
<div class="container">  
  <div class="box"></div>  
  <div class="box"></div>  
  <div class="specialboxes">  
    <div class="box"></div>  
    <div class="box"></div>  
  </div>  
  <div class="box"></div>  
</div>
```

Exempel

Exempel: display: contents

Om vi vill att boxarna ska bli flex-items men fortfarande ligga kvar i samma container, kan vi använda `display: contents`.

`.specialboxes-diven` kommer inte att påverka strukturen längre, men däremot kommer eventuell styling att fortfarande fungera.

Exempel: display: contents

Vi skriver följande:

```
.specialboxes {  
  display: contents;  
}
```

Vilket kommer få strukturen att bete sig såhär:

```
<div class="container">  
  <div class="box"></div>  
  <div class="box"></div>  
  <div class="specialboxes">  
    <div class="box"></div>  
    <div class="box"></div>  
  </div>  
  <div class="box"></div>  
</div>
```

display:contents



```
<div class="container">  
  <div class="box"></div>  
  <div class="box"></div>  
  
  <div class="box"></div>  
  <div class="box"></div>  
  
  <div class="box"></div>  
</div>
```

Exempel: display: contents

Resultat

CSS-Variabler

Ett relativt nytt koncept i CSS är att vi kan använda oss av variabler istället för att skriva ut våra värden direkt.

En variabel är ett värde som kan ändras och definieras på följande vis:

```
--main-color: #0c0c0c;
```

CSS-Variabler

För att läsa värdet på en variabel använder vi oss av funktionen `var()`.

```
.container .box {  
  background-color: var(--main-color);  
}
```

CSS-Varabler

CSS Variables (Custom Properties) - CR

Usage

Global

Permits the declaration and usage of cascading variables in stylesheets.

Current aligned Usage relative Date relative Apply filters Show all ?								
IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *
	12-14		4-47		10-34			
	2 15	2-30	1 48	3.1-9	1 35	3.2-9.2		
6-10	16	31-61	49-68	9.1-11.1	36-54	9.3-11.2		2.1-4.4.4
11	17	62	69	12	55	11.4	all	67
	18	63-64	70-72	TP		12		

Källa: caniuse.com

CSS-Variabler

Variabler kan vi definiera var som helst i vår CSS-kod men om vi vill kunna återanvända dem överallt kan vi lägga de i vårt översta element.

Det finns en selector som heter **:root**, som alltid väljer det yttersta elementet (som nästan alltid är `<html></html>`)

CSS-Variabler

Vi provar att definiera några variabler och använder dem.

```
:root {  
  --main-color: #f4af4a;  
  --text-color: white;  
  --big-text-size: 30px;  
  --medium-text-size: 20px;  
  --small-text-size: 16px;  
}  
  
.box {  
  width: 100px;  
  height: 100px;  
  background-color: var(--main-color);  
}
```

Exempel

CSS-Variabler

Vi kan också skriva över våra variabler i våra regler. Barn till vårt element kommer då ärva den variabelns värde.

```
.container .box {  
  margin: 40px;  
  width: 100px;  
  min-height: 50px;  
  background-color: var(--main-color);  
  --text-color: red;  
}  
  
.box span {  
  color: var(--text-color);  
}
```

Exempel

CSS-Variabler

De här variablerna kan vi sedan ändra på olika sätt, antingen med JavaScript eller med CSS.

Ett användbart fall är media-queries som vi ofta kan förenkla med hjälp av CSS variabler.

```
.container {  
  display: var(--display-mode);  
  justify-content: center;  
  margin: 100px;  
}  
  
@media (max-width: 500px) {  
  :root {  
    --text-color: blue;  
    --display-mode: block;  
  }  
}
```

Exempel

Övningar

Det finns några övningar på dagens föreläsning:

<https://github.com/LinkNorth/HTMLCSS/tree/master/exercises/week7/media-etc>

Nästa vecka

Måndag, Tisdag, Onsdag: **Repetition**

Tentamen - Fredag 26 okt 09:00-13:00