



# JS & CSS



# JS kan manipulere CSS

Vi kan manipulere vores CSS med JavaScript.

Vi kan fx tilføje eller fjerne klasser fra et element:

```
const htmlElement = document.querySelector("#html_element");
```

```
htmlElement.classList.add("class_navn");// Tilføjer class
```

```
htmlElement.classList.remove("class_navn");// Fjerner class
```

```
htmlElement.classList.toggle("class_navn");// Skiftevis fjerner og  
tilføjer class
```



# Manipulér CSS direkte

Man kan også sætte CSS-properties direkte fra JavaScript:

```
const htmlElement = document.querySelector("#html_element");  
  
    htmlElement.style.backgroundColor = "lightblue";  
    htmlElement.style.fontSize = "2rem";
```



# CSS properties i CSS vs. CSS properties i JavaScript

De to CSS-properties vi ændrede på i forrige slide ser sådan ud i JS

```
const htmlElement = document.querySelector("#html_element");  
htmlElement.style.backgroundColor = "lightblue";  
htmlElement.style.fontSize = "2rem";
```

Hvis vi skulle skrive en CSS-regel der gør det samme ville den se sådan ud:

```
#htmlElement {  
    background-color: lightblue;  
    font-size: 2rem;  
}
```

Så i JS er CSS-properties **camelCasede** hvor de i CSS er med **binde-streg**.

Læg også mærke til at values (værdierne) er tekststreng i JS: **"de står i anførselstegn også kendt som gåseøjne eller gnyffer"** hvor de i CSS er uden anførselstegn.

# Øvelse 7 - styling med javascript

Gem **06-skiftBillede.html** i en ny kopi med navnet **07-boxShadow.html**, og arbejd videre på programmet:

De tre articles skal pålægges en klasse der indeholder en box-shadow (lav en flot en [her](#))  
Lav herefter to eventhandlers ([mouseover](#), [mouseout](#)) der toggler shadow classen således at når man ruller over en article, er box-shadow væk, og når man ruller ud er den på igen (oldschool 🤖)

Ekstra:

Tilføj en [transition](#), så skyggen kommer gradvist på.

Prøv med andre properties.

Commit og push til gitHub, når du er tilfreds med opgaven