

Fra i går?

Løsningsforslag på opgaver: github helefred, repository spring2019

Javascript og DOM

Program

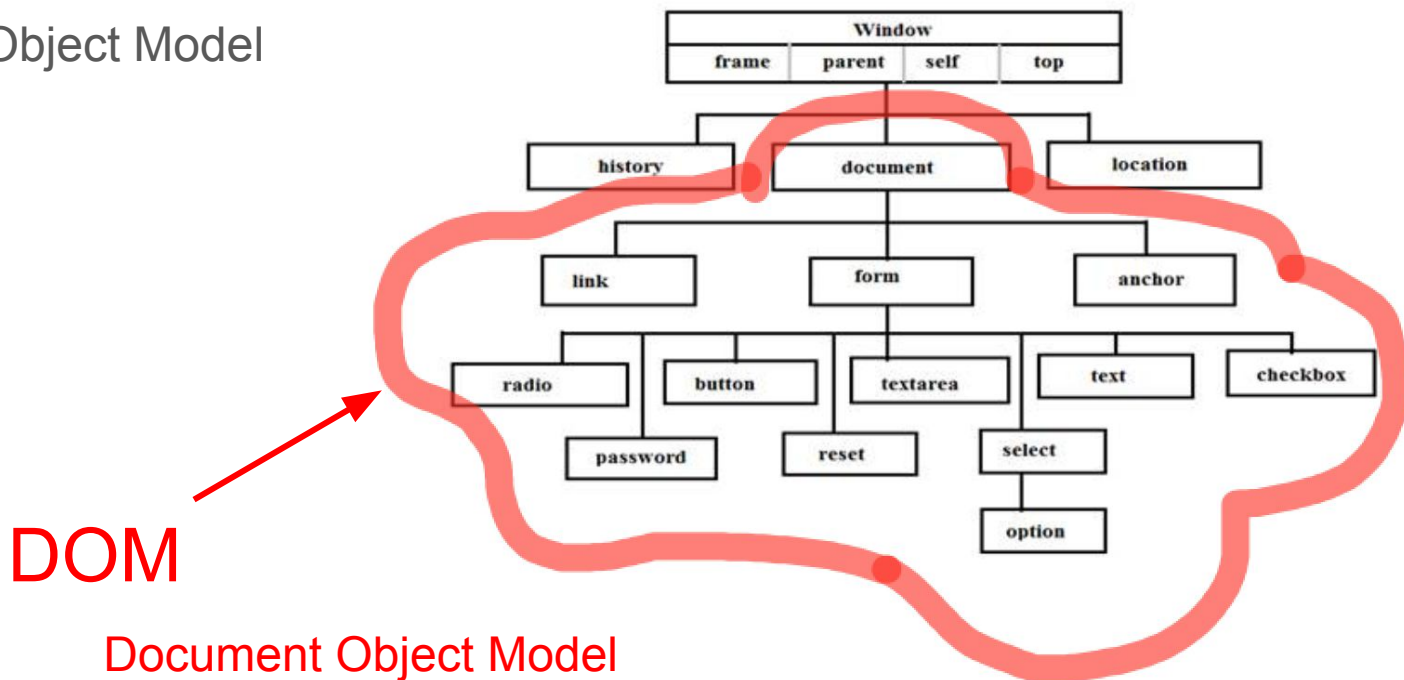
- DOM (Document Object Model)
- Vælg et DOM-element i javascript
- Læg nyt indhold i DOM-element
- eventListener og eventhandler på DOM-element
- Ændringer af css (tilføj/fjern classes og style-property for element)
- Vælg flere DOM-elementer
- Loops
- Ny datatype: array

Slut: 11.45

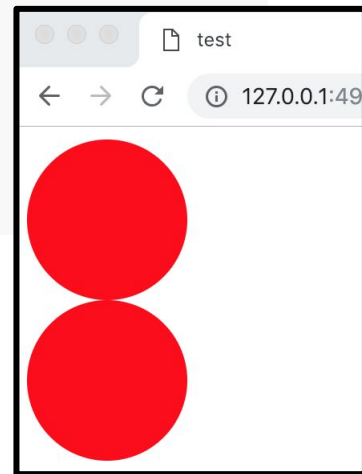
DOM

Window- og document-objektet

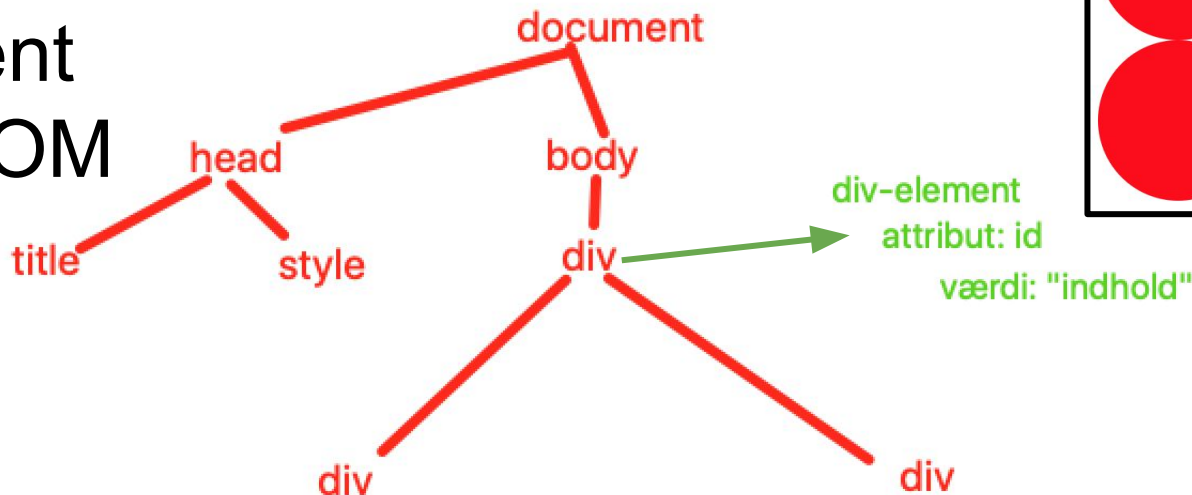
Browser Object Model



```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <title>test</title>
5      <style>.kugle{background-color:red; height:100px; width:100px;border-radius:100px}</style>
6    </head>
7    <body>
8      <div id="indhold">
9        <div class="first kugle"></div>
10       <div class="last kugle"></div>
11     </div>
12   </body>
13 </html>
```



Document ifølge DOM




Vælg DOM-element

document.querySelector - (vælg et DOM-element)

`document.querySelector("#idNavn")`

`document.querySelector(".classNavn")`

`document.querySelector("nav .menupunkt a")`



**Alle CSS-selectorer
kan bruges!**

Er der flere elementer med samme class eller tag, vælges kun det første element!

- Hvis man vil have dem alle? - det vender vi tilbage til!

Gem querySelector i variabel

En selektor gemmes tit i en variabel for ikke at skulle gentage selektionen:

```
let info = document.querySelector("#info");  
console.log(info);
```

Hver gang dette element skal bruges i programmet, kan man nu bruge **info**.

Øvelse 01 - Udvælg DOM-elementer

tema5

undervisningsopgaver

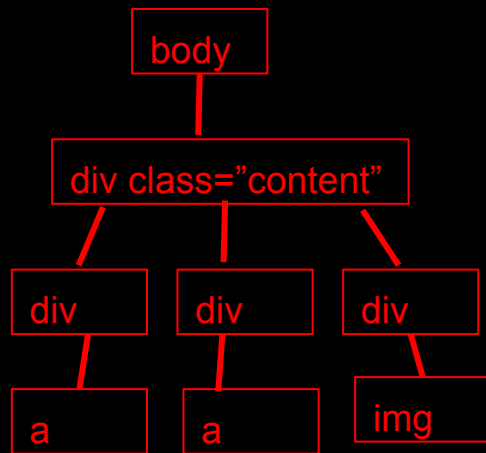
js-DOM:

01-select.html

I mappen **undervisningsopgaver**, laver du en ny undermappe, **js-DOM**. Åbn den i Brackets

Opret en ny html-fil, **01-select.html** (med skelet)

1. I body: læg et div-tag med id="content"
2. I div#content: læg tre div'er uden class eller id
3. I den første: læg et link til keas hjemmeside, i den anden: læg et link til din github-side, i den tredje et billede af en frø (billedfil: <http://helf-kea.dk/test/frog.png>)
4. Lav console.logs på hver af elementerne: #content, de tre div's i #content, de to a-tags og img-tagget



Du får måske brug for en css-selector til et elements nærmeste søskende:

https://developer.mozilla.org/en-US/docs/Web/CSS/Adjacent_sibling_combinator

Opsamling

DOM

Window-objekt

querySelector

attribut

Læg nyt indhold i
DOM-element

Nyt indhold i element (**textContent** og **innerHTML**)

Empty-tags: fx. img, br, hr, input - **containertags**: fx. body, div, h1, p (har slut-tags)

Eksempel på containertag med indhold: `<p id = "info">her stå noget </p>`

Man kan ændre på indholdet af **containertags** med js:

```
let info = document.querySelector("#info");
```

```
info.textContent = "noget helt andet";
```

```
info.innerHTML = "<h1>Noget nyt og anderledes</h1>";
```

W3schools: HTML DOM textContent Property, : https://www.w3schools.com/jsref/prop_node_textcontent.asp

W3schools: HTML DOM innerHTML Property, : https://www.w3schools.com/jsref/prop_html_innerhtml.asp

Øvelse 2 - Læg tekstindhold i elementer

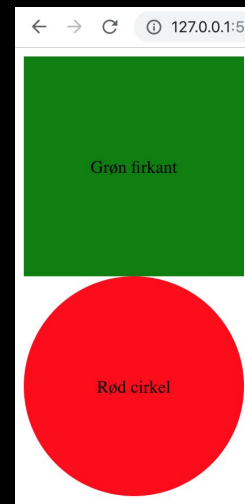
js-DOM:
01-select.html
02-ndhold.html

Opret en ny html-fil, **02-indhold.html**, i mappen **js-DOM**,
- start med html-skelettet.

1. I body skal der være en grøn firkant og en rød cirkel - brug et style-tag i head i stedet for et eksternt stylesheet - du må gerne. De to figurer skal være tomme (teksten skal ind med js i punkt 2.)
2. Læg teksten "Grøn firkant" i firkanten og "Rød cirkel" i cirklen vha javascript

(se næste slide for css-tips til opgaven).

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



Css-tips til øvelse 2

1. Gode css-egenskaber til opgave 0:

height: 100px; width: 100px

line-height: 100px (vertical midt på)

border-radius: 50%; (rund kant)

text-align: center; (horisontalt midt på)



Øvelse 3 - Læg html-indhold i elementer

Gem **02-indhold.html** i en ny kopi, **03-billedIndhold.html**.

Læg nyt javascript i script-tagget, som:

1. Sætter en frø ind i firkanten
2. En gris ind i cirklen.

Her er url'er til billederne:

<http://helf-kea.dk/test/frog.png> og

<http://helf-kea.dk/test/gris.png>

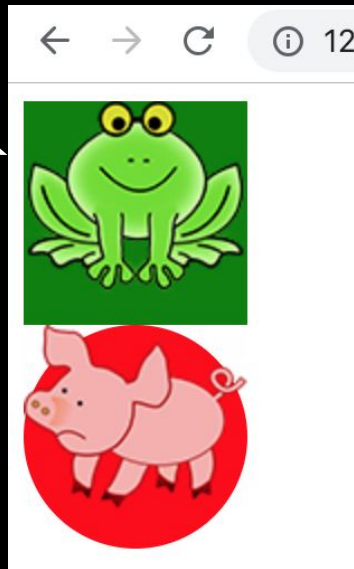
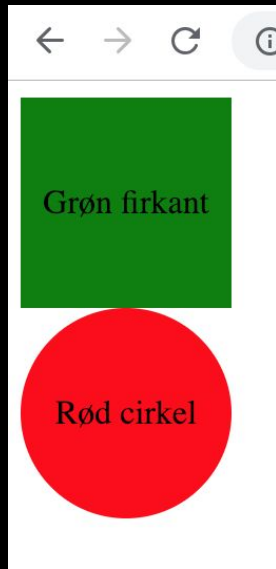
Commit og push til gitHub, når du er tilfreds med opgaven

js-DOM:

01-select.html

02-indhold.html

03-billedIndhold.html



Nyt indhold med **createElement** og **appendChild**

```
let info = document.querySelector("#info");  
info.innerHTML="<h1>Min overskrift til info</h1><img src='http://helf-kea.dk/test/gris.png'>"
```

Alternativ:

```
let h1 = document.createElement("h1");  
let overskrift = document.createTextNode("Min overskrift til info");  
h1.appendChild(overskrift);  
let img = document.createElement("img");  
img.src="http://helf-kea.dk/test/gris.png";  
info.appendChild(h1);  
info.appendChild(img);
```



```
<div id="info">  
  <h1>Min overskrift til info</h1>  
    
</div>
```

textContent vs innerHTML vs createElement/appendChild

- Brug **textContent** til rene tekster
- Brug **innerHTML** til små html-bidder (et eller to tags)
- Brug **createElement** og **appendChild** meget store html-tilføjelser

Øvelse 4 - Tilføj nyt element i element

Gem **03-billedindhold.html** i en ny kopi, som du kalder **04-append.html**.

Tilføj nyt javascript i script-tagget, som:

1. Opretter et nyt img-element.
2. Sætter det nye elements billedfil til "<http://helf-kea.dk/test/fox.png>"
3. Sætter det nye element ind i cirklen. (sæt højde og bredde op til det 200px)

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

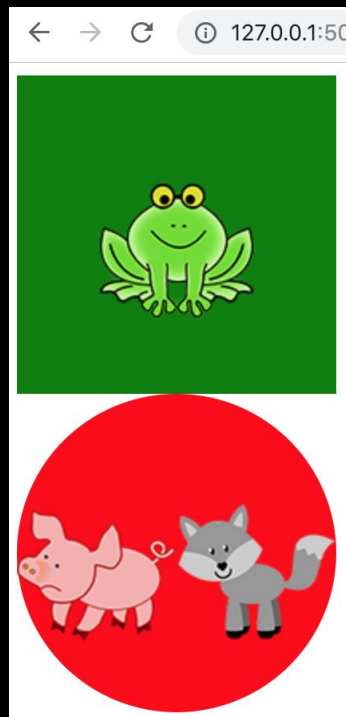
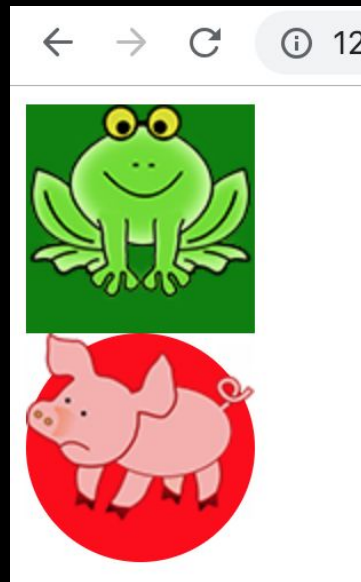
js-DOM:

01-select.html

02-indhold.html

03-billedIndhold.html

04-append.html



DOM-manipulationer - endnu flere

let info = document.querySelector("#info");

```
info.textContent = ... // indsætter tekst i info  
info.textContent += ... // tilføjer tekst til info
```

```
info.innerHTML = // indsæt html-kode i info  
info.innerHTML += ... // tilføjer html til info  
info.insertAdjacentHTML(position, html-kode)
```

```
info.setAttribute('attributNavn', 'værdi') // sætter en attribut-værdi på info  
info.getAttribute('attributNavn') // læser en værdi fra en attribut i info  
info.removeAttribute('attributNavn') // fjerner en attribut
```

```
info.src = ... // sætter src-værdi (img-tag)  
info.alt = ... // sætter img's alt-værdi  
info.href = ... // sætter href-værdi (a-tag)
```

```
info.cloneNode(true) // kopierer et element  
info.appendChild(element) // tilføjer et element  
info.removeChild(element) // fjerner et element
```

Opsamling

textContent

innerHTML

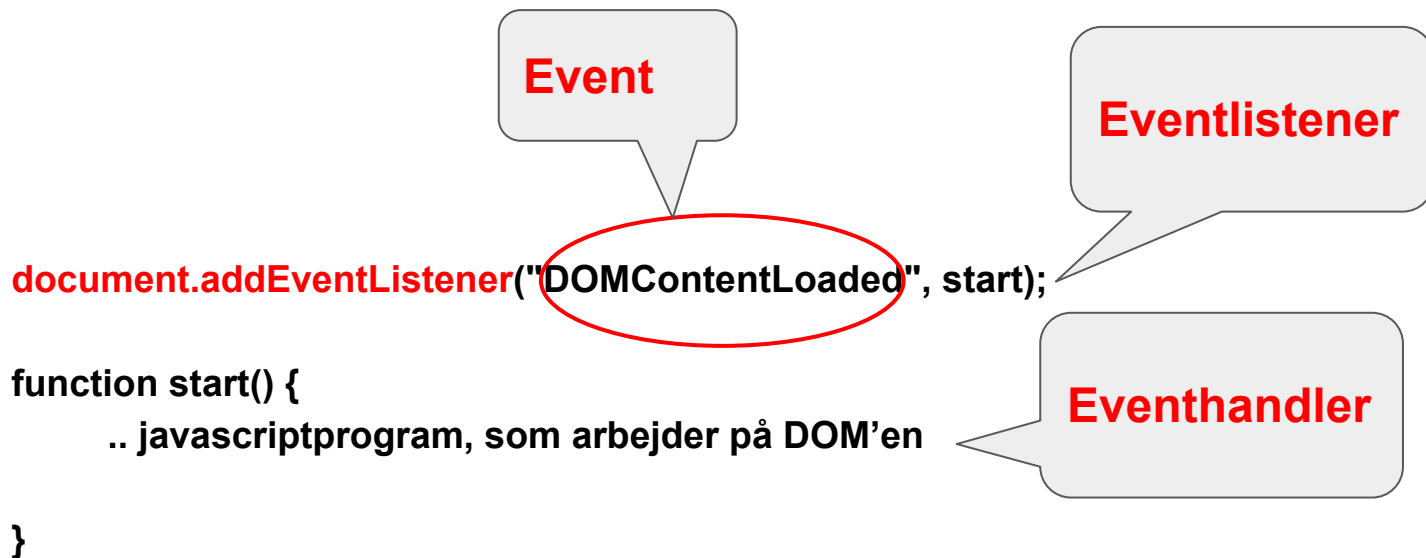
createElement

appendChild

EventListener og eventhandler

ContentLoaded - eventlister og -handler

Javascript bør først udføres, når man er helt sikker på, at DOM'en er loadet



EventListener & EventHandler

```
let element = document.querySelector("#info");
```

```
element.addEventListener("click", doSomething);
```

```
function doSomething() {  
    // det der skal ske  
}
```

Event - hvad skal der ske?

EventListener

Kald til EventHandler-funktion

EventHandler
- Den funktion der bliver
udført når Event indtræffer

For at fjerne eventlistener fra element igen:

```
element.removeEventListener("click", doSomething); // holder op med at virke
```


[video](#)

Øvelse 5 - test om DOM'en er loadet

Gem **03-billedIndhold.html** i en ny kopi med navnet **05-testLoad.html**.

- Arbejd videre på indholdet:
Du skal sørge for at scriptet tester, om DOM'en er loadet, inden det lægger billederne ind.

js-DOM:

01-select.html

02-indhold.html

03-billedIndhold.html

04-append.html

05-testLoad.html

Øvelse 6 - frøhop

Gem **05-testLoad.html** i en ny kopi med navnet **06-klikHop.html**.

Arbejd videre på indholdet:

Når der klikkes på den grønne firkant, skal de to dyr bytte plads

Når der klikkes på den røde cirkel, skal den grønne firkant ikke længere kunne bruges til ombytninger.

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

js-DOM:

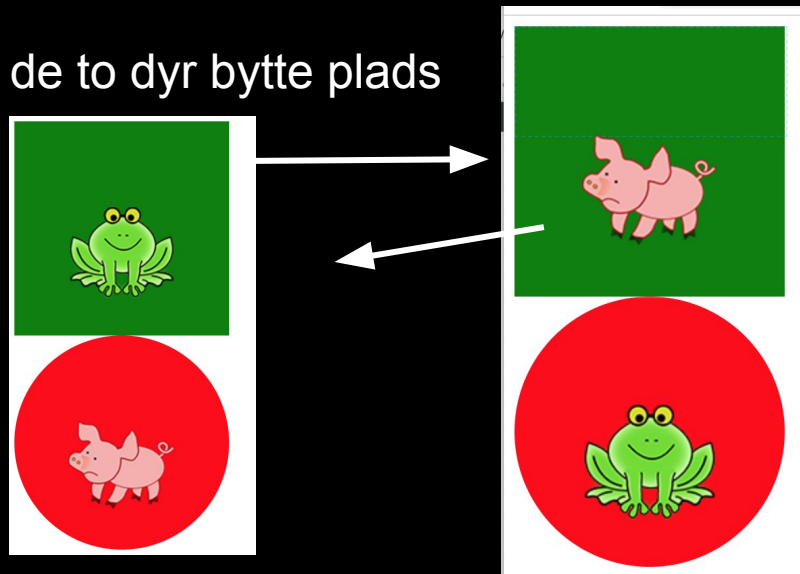
01-select.html

02-indhold.html

03-billedIndhold.html

04-testLoad.html

05-klikHop.html



Flere events

```
let info = document.querySelector("#info");

info.addEventListener("click", doSomething);
info.addEventListener("dblclick", doSomething);
info.addEventListener("mouseover", doSomething);
info.addEventListener("mouseout", doSomething);

function doSomething() {
    // det der skal ske
}
```

<https://developer.mozilla.org/en-US/docs/Web/Events>
https://www.w3schools.com/jsref/dom_obj_event.asp

Endnu flere:

mousedown
mousemove
mouseup
touchdown
touchmove
touchend
animationend
transitionend
keydown
keypress
keyup

...

3 måder at kalde funktionen på i eventListener

```
let element = document.querySelector("#info");
```

1. Almindelig funktion:

```
element.addEventListener("click", doSomething);  
function doSomething() {  
    // det der skal ske  
}
```

2. Anonym funktion:

```
element.addEventListener("click", function() {  
    // det der skal ske  
});
```

3. Arrow-funktion:

```
element.addEventListener("click", () =>{  
    // det der skal ske  
});
```

Opsamling

Event

EventListener

addEventListener

EventHandler

Anonym funktion

Arrow-funktion

*A*Endringer af css

Tilføj eller fjern klasser fra element

```
let info = document.querySelector("#info");
```

```
info.classList.add("classNavn") // indsætter eller tilføjer class
```

```
info.classList.remove("classNavn") // fjerner class
```

```
info.classList.toggle("classNavn") // skiftevis fjerner og tilføjer class
```

Direkte manipulation af elementet

```
let info = document.querySelector("#info");
```

```
info.style.border = "none";  
info.style.backgroundColor = "paleyellow";  
info.style.fontFamily = "Helvetica, Sans-serif";  
info.style.height = "2rem";  
info.style.display = "none";  
info.style.display = "block";
```

Navne på style-properties:

fontFamily (css: font-family)

backgroundColor (css: background-color)

- bindestreg erstattes af camelCase

Øvelse 7 - styling med javascript

Gem **06-klikHop.html** under navnet **07-styling.html**, og arbejd videre på programmet:

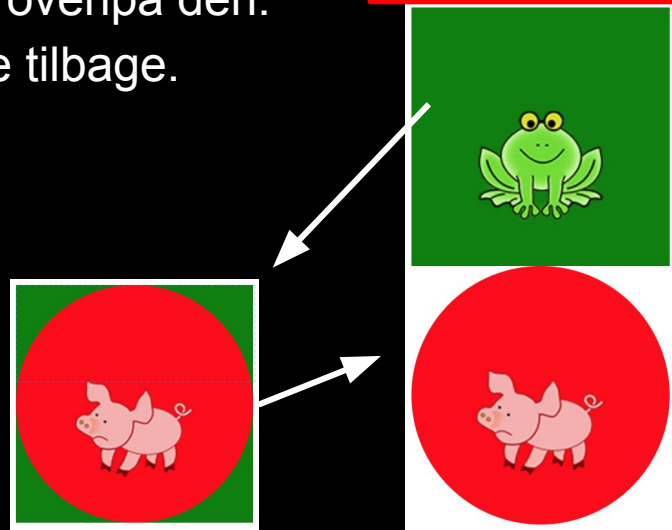
Ved klik på firkanten, skal cirklen hoppe op ovenpå den. Når man klikker på cirklen, skal den springe tilbage.

Nyttige css-regler:

position: absolute;
top:0;
position:static;

js-DOM:

01-select.html
02-indhold.html
03-billedIndhold.html
04-append.html
05-testLoad.html
06-klikHop.html
07-styling.html



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

Opsamling

style-property

backgroundColor-property

classList-property

classList.add()

classList.remove()

Vælg flere DOM-elementer

querySelectorAll

Eksempel: Alle elementer på siden med class="red", skal forsvinde:

```
document.querySelector(".red").style.display="none"; //kun den første :(
```

Problem: kun det første element forsvinder! :(

For at få fat i dem alle:

```
let alle = document.querySelectorAll(".red");
```

For at “gøre noget ved alle elementer”, eksempel: få dem til at forsvinde:

```
alle.forEach (element => {  
    element.style.display="none";  
});
```

Øvelse 8: Eventlisteners på alle p-tags

1. Tag udgangspunkt i denne klargjorte html-fil, og gem den som **08-alle.html**
2. I scriptet - lav en eventlistener på, om DOM'en er Indlæst. Sæt herefter en eventlistener på hvert p-tag: når der klikkes på en sætning, skal den styles med class `typo`.
3. Lav en reset-knap øverst på siden.
4. Når der klikkes på knappen, skal class `typo` fjernes fra alle p-tags igen
5. Udvid programmet med en klasse, `typo2`, som styles alle overskrifter på samme måde - men med dobbelt størrelse.

js-DOM:
select.html
indhold.html
billedIndhold.html
testLoad.html
styling.html
08-alle.html



Arrays og loops

Ny datatype: Array []

Datatyper: tekster, tal, boolean (simple datatyper) og nu også **array** (ikke helt så simpel)

Et **array** er en liste/tabel med flere elementer

Variabel med array:

```
let lærerTeam = ["Helle", "Martin", "Klaus", "Kamilla", "Louise", "Birgitte"]; // array
```

```
console.log(lærerTeam[0]); // returnerer "Helle"
```

```
console.log(lærerTeam[3]); // returnerer "Kamilla"
```

```
lærerTeam.length, giver antallet af elementer i array'et lærerTeam // 6
```

Tomt array:

```
let lærerTeam=[];
```

OBS!

Første
index er **0**.

Læs mere om Arrays:

All about JavaScript Arrays in 1 article, [Rajesh Pillai](https://codeburst.io/all-about-javascript-arrays-in-1-article-39da12170b1c) Follow, 2018: <https://codeburst.io/all-about-javascript-arrays-in-1-article-39da12170b1c>

MDN: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array

Loop: forEach

En “loop”(løkke) kalder man et statement, som gentager den samme kode-sekvens

forEach er én af de loop-syntaxer javascript byder på

```
let lærerTeam = ["Helle", "Martin", "Kamilla", "Louise", "Birgitte"]; //et array erklæres
lærerTeam.forEach( lærer => {
    console.log(lærer);
});
```

lærer:

et variabelnavn, du selv finder på. Hver gang loopet behandler et nyt element, kaldes de lærer

Øvelse 9 - array

Tag udgangspunkt i denne klargjorte version, og kald
filen **09-array.html**

js-DOM:

01-select.html

02-indhold.html

03-billedIndhold.html

04-testLoad.html

05-klikHop.html

06-styling.html

07-array

Lav et script:

Erklær et array “playliste” med navne
på dine 10 yndlingstracks.

Loop arrayet igennem og få hvert tracknavn
til at stå i div-boxen “display” , på hver sin linie

Sound of Music
Dancing in the Rain
What a wonderful world
Lille Peter Edderkop
Yesterday
Den knaldrøde gummiåb
Krøller eller hvad
Ved landsbyens gadekær
Smilende Sussi
Så længe jeg lever

Opsamling

querySelectorAll()

forEach

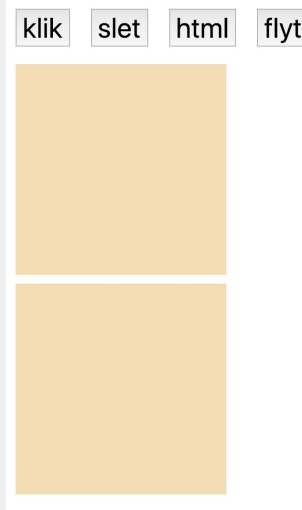
Array

Array-index

Eftermiddagsøvelser

Øvelse 6: klikpaaknap.html

1. Hent denne klargjorte html-side.
2. Der er en knap, klik og en div id, "box"
Når man klikker på knappen "klik", skal "box" vise teksten: "KLIK".
3. Der er også en knap, slet
Når man klikker på "slet", skal "box" resettes
(viskes ren) = vise teksten: " ".
4. Knappen "html" skal vise en `<h2>` og en `<p>` i boksen.
5. Lav så en knap "flyt", der kan flytte "box"'s html over i den anden boks, "bax" !



Hvis du hidtil har brugt navngivne funktioner, så prøv med anonyme eller arrow

Øvelse 7: styleBoxes.html

Du skal bruge 4 div'er med class "box" og med hver sin id (box1, box2, box3, box4)
Giv box-class lidt css, så boxene ser ens ud.

[Klargjort version](#)

Når man klikker på dem hver især skal de skifte udseende.

- gerne med en transition --

Så alle 4 ser forskellige ud.

Du kan gøre det ved at tilføje hver sin class, på 2 af boksene.

På de resterende 2, prøv at style direkte med js, Prøv at bruge andre properties end end dem i eksemplerne ..

Næste trin - til denne vintereftermiddag

Øvelser 1,2,4 på Github <https://github.com/kvikea/JS-2-exercises>

Efter idag, vil de sikkert forekomme dig temmelig nemme !!

(3'eren springer vi over).