

DOM SCRIPTING med JAVASCRIPT

Event Listeners

Indtil nu har vi skrevet event handlers inde i html:

```
<img onclick="next()"...
```

Det fungerer, men det er bedre hvis vi kan skrive javascript kun i js filer.

Event Listeners giver os mulighed for at tildele event handlers til html elementer fra js filen.



Øvelse 1 :

Lav en ny kopi af den sidste "Tim Walker" øvelse (eller hent den fra ItsLearning).

Slet de to **onclick** event handlers fra html, og giv hver pil en ID: 'next' og 'back'.

Lav en ny .js fil, og link til den fra bunden af siden.

Flyt al javascript koden over i den nye fil.

Der skal ikke være `<script></script>` tags i .js filen.

Tilføj til .js filen:

```
document.querySelector("#next").addEventListener("click", frem);
```

.. og tilsvarende for tilbage pilen.

Test.

Bemærk at man her *ikke* skal skrive de runde parenteser efter funktionskaldet.

Fra nu af, anbefaler vi at man så vidt muligt altid bruger Event Listeners.

Når man ikke skriver de runde parenteser, kan man ikke sende **parameter arguments** på samme måde som vi plejer.

Men man kan gøre det på 2 andre måder. Her ser vi især på én metode, som i praksis dækker de fleste behov.

A event objektet

Øvelse 2 :

Lav nye kopier af både html og js filerne, og link dem sammen.

Giv begge knapper event listeners, som kalder den samme funktion:

```
document.querySelector("#next").addEventListener("click", skift);
document.querySelector("#back").addEventListener("click", skift);
```

Ændr navnet på den første funktion til **skift(e)**, og slet den anden.

e er event'en, som bliver sendt med fra en event listener.

Comment ud koden i funktionen, og tilføj:

```
console.log(e);
```

Så kan man se enormt mange informationer om denne event.

Prøv nu:

```
console.log(e.target);
```

Her har vi det element som er blevet klikket på.

Nu kan vi sige:

"Hvis man har klikket på frem-pilen, tæl op, og ellers tæl ned – og derefter skift billedet."

```
if (e.target.getAttribute("id")==="next"){...
```

Skriv selv funktionen færdigt.

Øvelse 3 :

Foroven havde vi kun 2 knapper der skulle have eventListeners, så det hurtigste er at skrive 2 linjer kode.

Men når man har flere knapper – fx thumbnails – kan det gøres meget effektivt ved at kombinere eventListeners med arrays og løkker.

Download det nye fil sæt 'Tim Walker med Thumbnails – start'

Her har vi eventListeners for de 2 pile, men vi skal også aktivere alle thumbnails.

1. Vi skal bruge en liste med alle thumbnails:

```
var thumbs = document.querySelectorAll(".thumbs img");
```

2. Så kan vi loop igennem dem, og give dem alle eventListeners:

```
for(var i=0; i<thumbs.length; i++){
  thumbs[i].addEventListener("click", show);
}
```

3. Start funktionen, og få fat i src for den aktuelle thumbnail. Test med console.log:

```
function show(e){
  var foto = e.target.getAttribute("src");
  console.log(foto);
}
```

4. Giv denne src til det store billede:

```
document.querySelector("#photo").setAttribute("src", foto);
```

Test i browser.

5. Nu er vi *næsten* færdig . . men når man først vælger et foto fra en thumbnail, og derefter klikker på en pil, kommer man til det forkerte foto – fordi de 2 metoder ikke er koblet sammen.

Vi har brug for at opdatere index (som bruges af pilene) til det aktuelle foto-nummer.

Når alle thumbs ligger i en liste (variablen 'thumbs'), burde vi kunne få at vide hvilken nummer i listen den aktuelle foto har.

Det kan gøres med `indexOf()` – men her møder vi en af de små forskelle mellem ægte Arrays og de *htmlCollections* eller *NodeLists* som vi får fra `querySelectorAll` .

De er ligesom Arrays på rigtige mange måder – men kan ikke bruge `indexOf()` .

Du kan evt. teste det med `console.log(thumbs);` - det giver 'NodeList'.

Heldigvis er det nemt at lave en kopi af en NodeList, som er en ægte Array:

```
var thumbArray = Array.from(thumbs);
```

Nu kan vi synkronisere index med det aktuelle thumbnail nummer:

```
index = thumbArray.indexOf(e.target);
```

Test i browser.

B anonymous functions

Hvis man har brug for at sende flere parametre end e (event'en), kan man også indsæt en anonymous function som det andet parameter for `addEventListener()` :

```
document.querySelector("#next").addEventListener("click", function(){
    console.log("hello");
    frem(index);                      //herfra kan man sende argumenter..
});
function frem(nummer){               //..som modtages af funktionen
    if(nummer<fotoListe.length-1){
        nummer++;
    }else{
        nummer=0;
    }
    index=nummer;
    nyFoto="fotos/"+fotoListe[index]+".jpg";
    document.querySelector("#photo").setAttribute("src", nyFoto);
}
```