

Variables, Conditions, Scope

Cadiz, Spanien



Vi skal bygge en ny version, kun med 'frem' og 'tilbage' pile.

Det giver mulighed for at se nærmere på **variabler** og deres **scope**, og på **conditions**.

Download *galleri_pile_start.zip* fra ItsLearning.

1. Højrepilen skal have en **event handler** som **kaller** en ny **function**: **onclick="next()"**

Billederne hedder cadiz1.jpg, cadiz2.jpg ... osv.

Derfor skal funktionen blot sætte tallet op (først fra 1 til 2, næste gang til 3, osv..)

.. og derefter indsæt billedet med det nye tal.

2. Vi har brug for en **variabel** som kan indeholde det aktuelle tal.

Denne type variabel kalder man typisk 'index'.

I Javascript, opret en ny variabel: **var index = 1;**

(Værdien 1 svarer til tallet i det aktuelle foto når siden åbnes.)

3. Opret funktionen:

```
function next(){  
  
}
```

4. For hver klik, kan vi sætte værdien af index op, med **index++**;
Indsæt i funktionen - det virker formentligt - men vi kan ikke se det.

5. For at se en variabel, brug: **console.log(index)**;
Åbn browser DevTools og vælg Console. Klik flere gang for at se tallet stige for hver klik.

6. Nu kan vi bruge tallet til at ændre billedet: vi kan kombinere (**concatenate**) den første del af stivejen (som er fast) med det aktuelle tal - og med slutningen:

```
var nyFoto = "images/cadiz"+index+".jpg";
```

Brug console.log til at checke.

7. Så kan vi indsætte billedet med:

```
document.querySelector("#galleri").setAttribute("src", nyFoto);
```

Test.

Nu har vi kun ét problem: hvad sker der efter det sidste billede? ...

Conditions! (Betingelser)

8. Brug:

```
if(index < 6){  
    index++;  
}
```

.. og man stopper på sidste billede.

9. For at skifte tilbage til det første foto efter det sidste:

```
if(index < 6){  
    index++;  
}else{  
    index = 1;  
}
```

.. og man starter forfra.

10. Skriv kode for tilbage-pilen på tilsvarende måde.

Scope

index er en **global** variabel, fordi vi erklærer den udenfor funktionerne. Den kan nu bruges både indenfor og udenfor funktionerne.

Variabler som er erklæret (eng: **declared**) i en funktion er lokale (**local**) variabler, som dør når funktionen slutter.

Det kan have både fordele og ulemper - man skal bare være klar over hvor man erklærer variabler, og hvilken **scope** de derfor får.

Prøv at erklære **index** inden i en funktion og test - mens du har Console åben i Dev Tools.

```
<script>
function begynd(){
    var index=1;           [lokal variabel i en funktion]
    console.log(index);    [tester: variabelen bliver oprettet... men kun lokalt]
}

begynd();                 [kalder funktionen når siden åbnes]

function frem(){
    if(index<6){...        [fortsætter som før..]
```

Omvendt kunne man erklære *nyFoto* i starten (uden værdi) som global, i stedet for at oprette 2 lokale variabler i de 2 funktioner. Så kan man blot henvise til den:

```
var index=1;
var nyFoto;
function frem(){
    if(index<7){
        index++;
    }else{
        index=1;
    }
    nyFoto="fotos/cadiz"+index+".jpg";
    document.querySelector("#galleri").setAttribute("src", nyFoto);
}
```