

Navigér i DOM – vælg søskende, forældre og børn med jquery

Ligesom det er muligt at vælge et vilkårligt element i et html-dokumentets struktur med en jquery-selektor, er der også funktioner til at bevæge sig rundt i DOM-træet ud fra dette valgte element. Det er nyttigt hvis du fx bruger udtrykket `$(this)` og fx vil have fat i det aktuelle elements næste søskende, dets barn nummer 3, eller dets forældrelement etc.

1 Funktioner til det ser sådan ud:

- `.next()` – vælger næste søskende efter elementet
- `.prev()` – vælger forrige søskende efter elementet
- `.parent()` – vælger elementets forældrelement
- `.parent().parent()` – vælger forældrelementets forældrelement

og børn:

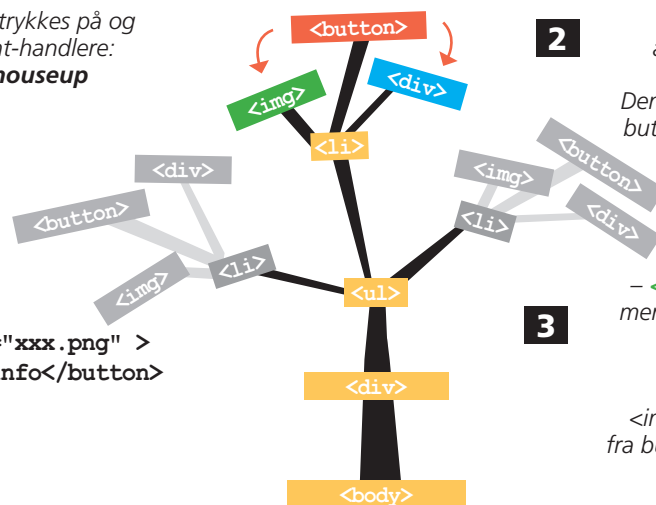
- `.children()` – vælger alle børn
- `.children('img:nth-child(2)')` – vælger det 2. barn (hvis det er et img-tag)

2 Eksempel – det interaktive element, `$(this)`, vælger næste og forrige søskende

I en liste skal der bygges visning af en række billeder, for hvert billede skal en **button** få en **div** til at dukke op, så længe brugeren holder musen trykket ned på knappen, samtidig „dæmpes“ det **aktuelle billede** med opacitet eller blur.

1 Det er **button** der trykkes på og den tildeles to event-handlere: **mousedown** og **mouseup**

```
<ul>
  <li>
    
    <button>info</button>
    <div>
      .....
    </div>
  </li>
</ul>
```



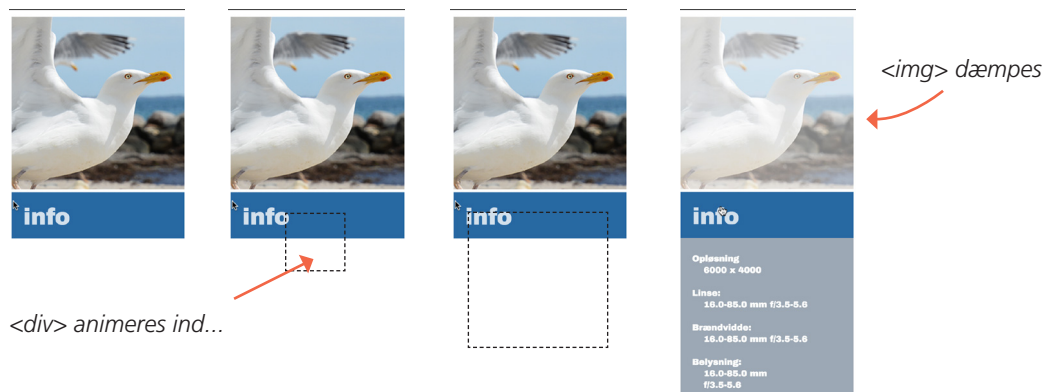
2 – **div**'en, der er usynlig, skal animere ind ved **mousedown** animere væk ved **mouseup**. Den vælges med jquery set ud fra button som den næste søskende, med funktionen:

`$(this).next()`

3 – ****, har fx en opacity på 1, men skal dæmpes lidt fx ved at få opacity .5 ved **mousedown**. Ved **mouseup** skal den have opacity 1 igen. **** vælges med jquery set ud fra button som den forrige søskende, med funktionen:

`$(this).prev()`

Overblik over scenariet:



Den vigtigste css-kode:

```
li img { opacity: 1;
          transition: .4s linear }
```

```
button {
  z-index: 3;
  .... }
```

```
li {
  position: relative }
```

```
li div {
  opacity: 0;
  ....
  position: absolute;
  top: 30px;
  z-index: -1;
  -webkit-transition: .2s .1s cubic-bezier(.58,.85,.98,1.22);
  -webkit-transform: translateY(0) scale(0);
}
```

```
.op_ned {
  -webkit-transform: translateY(340px) scale(1);
  opacity: 1;
}
```

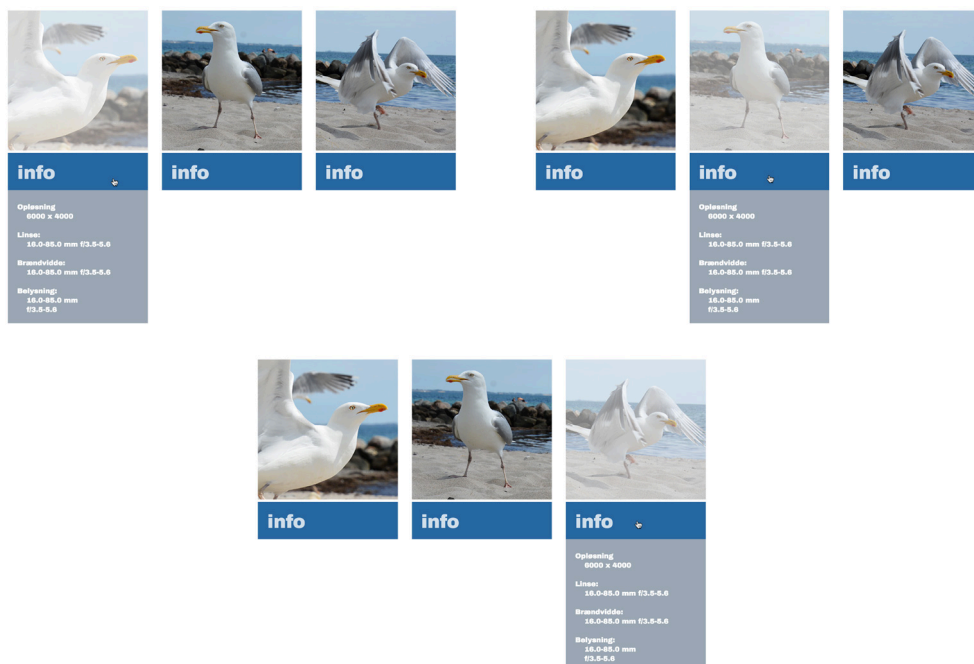
Scriptet ser sådan ud:

```
$( 'button' ).on( 'mousedown', function() {
    $( this ).next().addClass( 'op_ned' );
    $( this ).prev().css( { 'opacity': '.5' } );
});

$( 'button' ).on( 'mouseup', function() {
    $( this ).next().removeClass( 'op_ned' );
    $( this ).prev().css( { 'opacity': '1' } );
});
```

Der er to eventhændlere,
– én for **mousedown**
– og én for **mouseup**

Der navigeres i DOM ud fra <button> der trykkes på:
<div> er det samme som \$(this).next()
og \$(this).prev() er det samme som



Der er nu tre 'er med samme struktur – og hver <button> finder netop sine søskende ud fra next() og prev()