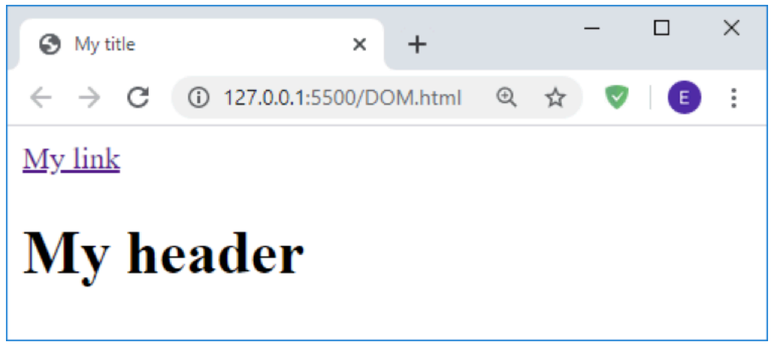
Distribueret Programmering

Lektion 04: DOM

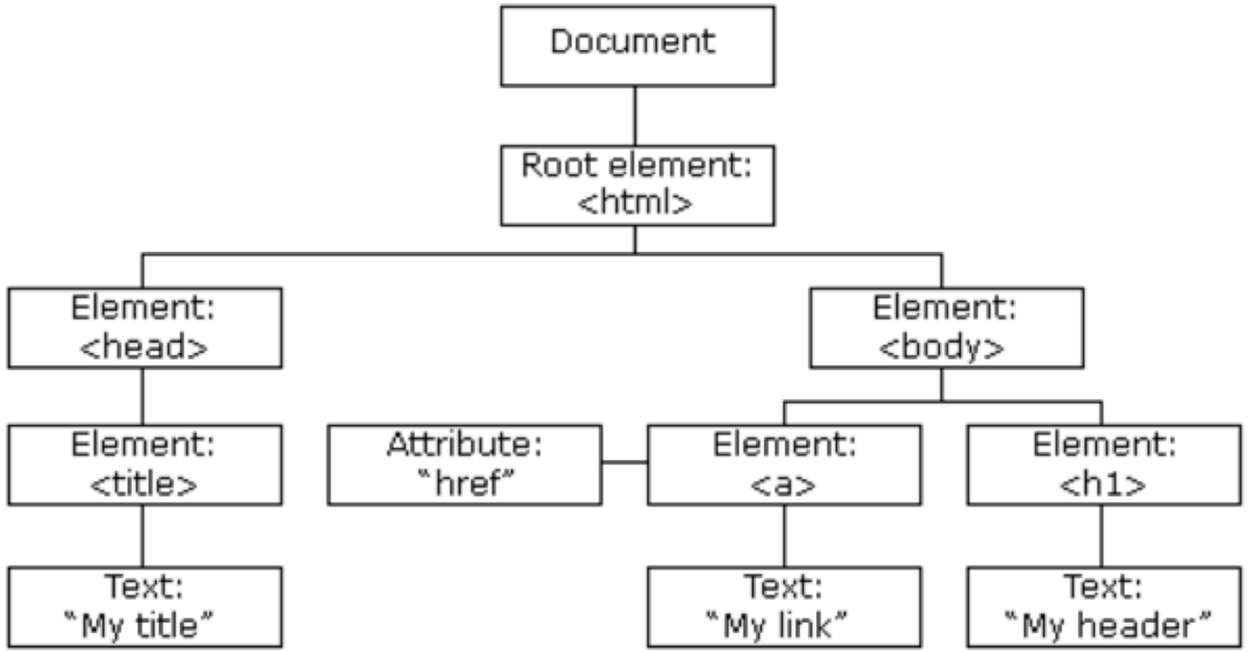
DOM er forkortelse for *Document Object Model* og den beskriver/repræsenterer websiden i en træstruktur som kan læses og manipuleres i javascript via et DOM API. Denne API ser vi på i dag

```
<!-- DOM.html -->
<!DOCTYPE html>
<html>
  <head>
    <title>My title</title>
  </head>
  <body>
    <a href="http://www.w3schools.com/js/js_htmldom.asp">My link</a>
    <h1>My header</h1>
  </body>
</html>
```



Nodes

DOM'en har forskellige slags knuder:



<script>

- En webside kan have et eller flere <script> elementer, der hver især refererer til et JavaScript program
- <script> elementer placeres i <head>
- Når et <script … defer> element mødes under indlæsning af HTML dokumentet, startes der en asynkron indlæsning og parsning af det refererede program – men programmet udføres først, når DOM'en er opbygget
- Til forskel fra defer vil <script … async> udføre programmet, straks det er indlæst
- Endelig blokerer <script … > indlæsningen af HTML dokumentet, indtil programmet er indlæst, parset og udført

Referere elementer

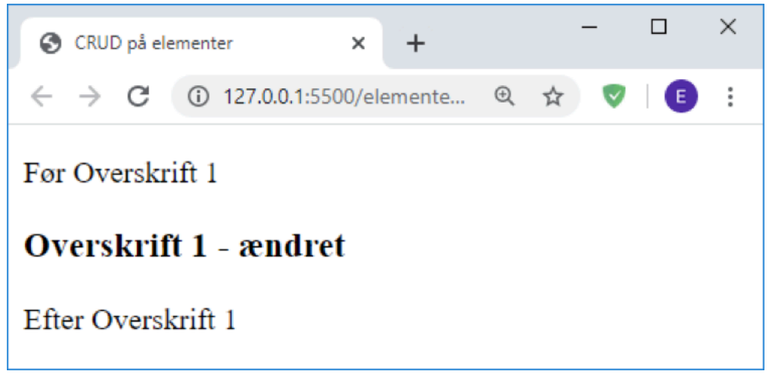
- DOM'en refereres med variablen document
- Elementer på websiden refereres primært vha. CSS selektorer:
  - `document.querySelector(*selektor*)` – returnerer kun det første element
  - `document.querySelectorAll(*selektor*)` – returnerer en array-like NodeList
  - `document.body`
  - `document.head`

```html
<!-- querySelector.html -->
<!DOCTYPE html>
<html>
  <head>
    <title>My title</title>
    <script src="querySelector.js" defer></script>
  </head>
  <body>
    <a href="http://www.w3schools.com/js/js_htmldom.asp">My link</a>
    <h1>My header</h1>
  </body>
</html>
```

```javascript
// querySelector.js
let h1 = document.querySelector('h1');
h1.style.color = 'red';
```

CRUD på elementer

```html
<!-- elementer.html -->
<!DOCTYPE html>
<html>
<head>
  <title>CRUD på elementer</title>
  <script src="elementer.js" defer></script>
</head>
<body>
</body>
</html>
```




```javascript
// elementer.js
document.body.innerHTML = '<h1>Overskrift 1</h1><h2>Overskrift 2</h2>';
let h1 = document.querySelector('h1');
h1.outerHTML += '<p>Efter Overskrift 1</p>';
h1 = document.querySelector('h1');
h1.outerHTML = '<p>Før Overskrift 1</p>' + h1.outerHTML;
h1 = document.querySelector('h1');
h1.outerHTML = '<h3>' + h1.innerHTML + ' - ændret</h3>';
let h2 = document.querySelector('h2');
h2.outerHTML = '';
```

Eksempel: ur

```html
<!-- ur.html -->
<!DOCTYPE html>
<html>
    <head>
        <title>Ur</title>
        <script src="ur.js" defer></script>
    </head>
    <body>
        <div></div>
    </body>
</html>
```

```js
// ur.js
function opdaterUr() {
    let ur = document.querySelector("div");
    console.log(new Date().toLocaleTimeString());
    ur.innerHTML = new Date().toLocaleTimeString();
    setTimeout(opdaterUr, 1000);
}
opdaterUr();
```

Properties til navigation

- `element.parentElement`
- `element.firstElementChild`
- `element.lastElementChild`
- `element.previousElementSibling`
- `element.nextElementSibling`
- `element.children`

CRUD på attributter

Operationer på HTML attributter:

- `element.attribute = value;`
- `value = element.attribute;`
- `value = element.attribute;`
- `element.hasAttribute(attribute);`
- `element.removeAttribute(attribute);`

CRUD på attributter

`class` attributen:

- skrives som `element.className`

`style` attributten har properties:

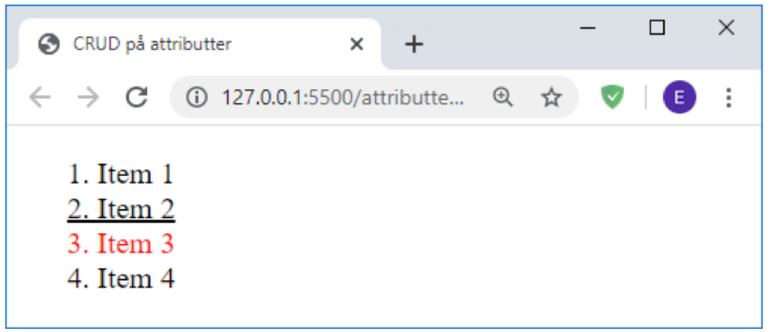
- `element.style.property = value`
- `value = element.style.propert`

properties med bindestreg i navnet skrives med kamel-notation (camelCase)
fx skal 'text-align' skrives som textAlign

CRUD på attributter

```html
<!-- attributter.html -->
<!DOCTYPE html>
<html>
<head>
  <title>CRUD på attributter</title>
  <script src="attributter.js" defer></script>
</head>
<body>
  <ol>
    <li>Item 1</li>
    <li>Item 2</li>
    <li>Item 3</li>
    <li>Item 4</li>
  </ol>
</body>
</html>
```

```js
// attributter.js
let lis = document.querySelectorAll('li');
let id = 1;
for (let li of lis) {
    li.className = id % 2 === 0 ? 'lige' : 'ulige';
    li.id = 'id' + id++;
}
document.querySelector('#id3').style.color = 'red';
let li = document.querySelector('.lige');
li.style.textDecorationLine = 'underline';
```


CRUD på attributter
127.0.0.1:5500/attributte...
1. Item 1
2. Item 2
3. Item 3
4. Item 4

Window

- window er det såkaldte globale scope objekt, som er JavaScripts programmeringsomgivelse i browseren
- window har en række generelle JavaScript properties, som i Node.js:
  undefined, NaN, Math, …
  isNaN(), parseInt(), setTimeout(), …
- window har desuden en række browser specifikke properties
  document, screen, location, history, navigator, cookie, …
- globale funktioner samt variable erklæret med eller uden var bliver også properties på window

```html
<!-- window.html -->
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>window</title>
  <script src="window.js" defer></script>
</head>
<body>
  <h3>Tryk på F12 og vælg Console fanebladet</h3>
  Åben dernæst window objektet og find x, y og f
</body>
```

```javascript
// window.js
x = 1;
var y = 2;
let z = 3;
const u = 4;
function f() {}
console.log(window);
console.log(x, y, z, u);
```