

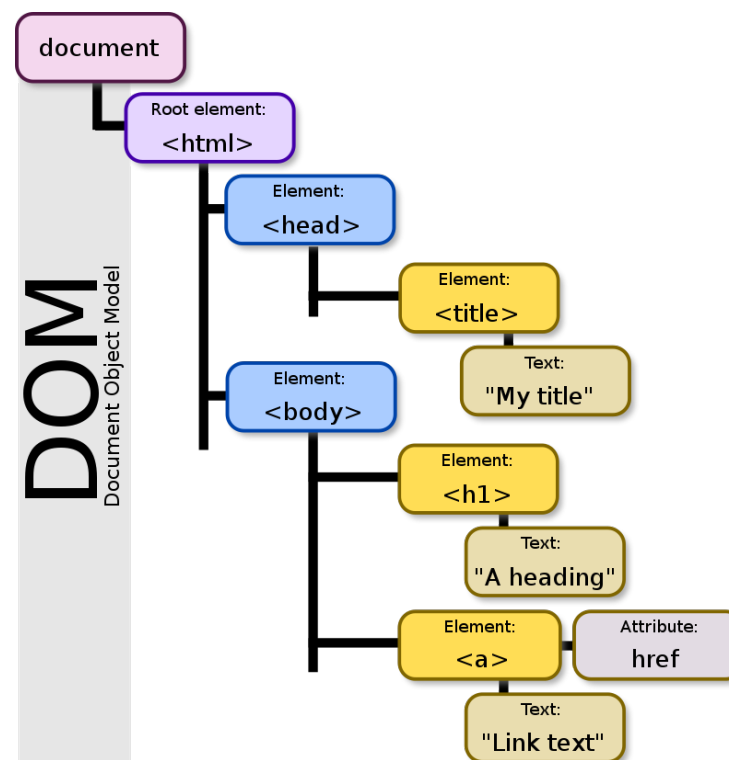
HTML: DOM – document object model

TRIN 1: Forstå et markup-sprog til opsætning af en grafisk brugergrænseflade, en GUI, her HTML. Forstå først DOM-modellen og herunder diverse basale HTML-tags som fx div, head, body osv., og videre selektioner som ID og CLASS. Og forstå nestede elementer, hvor fx en div lægges i en div.

Via **et markup-language** har man en standard til opsætning & visning af elementer i et dokument, mens et rigtigt **programmeringssprog** kan kontrollere programmets gang, lave udregninger, ændre data dynamisk osv.

Via C# kan bruge XAML til desktop-programmert (wpf-frameworket)

Via Java kan man bruge FXML også til desktop-programmer (javaFX-frameworket)



HTML: DOM – document object model

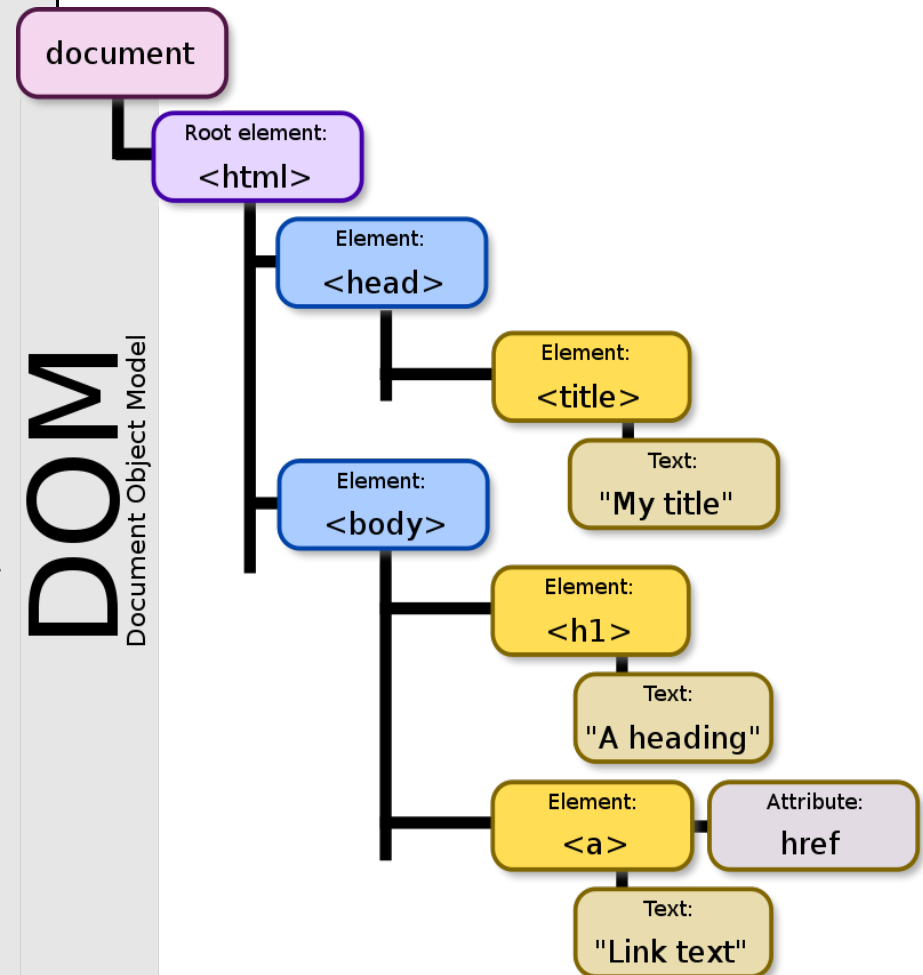
DOM repræsenterer HTML-dokumentet som en **hierarkisk træstruktur**.

Denne struktur gør det muligt via kode at tilgå og ændre i HTML-elementerne i træet – fx afhængigt af brugerinput eller andre begivenheder.

Dokumentets rod-objekt er hele html-siden, som igen indeholder to børn, head og body, og disse indeholder atter diverse børn.

Body indeholder i dette tilfælde en **<h1>**, heading med værdien “A heading” samt et link-tag **<a>** med en “href”, som refererer til en adresse på www samt en værdi “Link text”, som ender med at udgøre det klikbare link.

Head indeholder elementet **<title>** med værdien “My title”, og dette udgør sidens titel i browseren.



HTML: DOM – document object model

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title>Page Title</title>
```

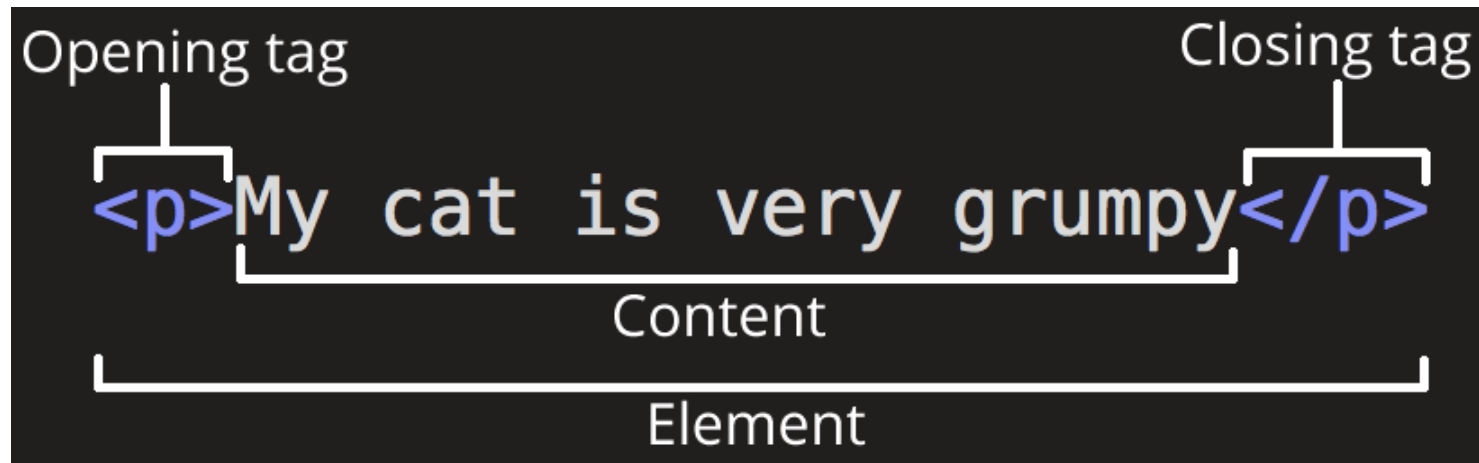
```
  </head>
```

```
  <body>
```

```
    <p>My cat is very grumpy</p>
```

```
  </body>
```

```
</html>
```



HTML: DOM – document object model

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title>Page Title</title>
```

```
  </head>
```

```
  <body>
```

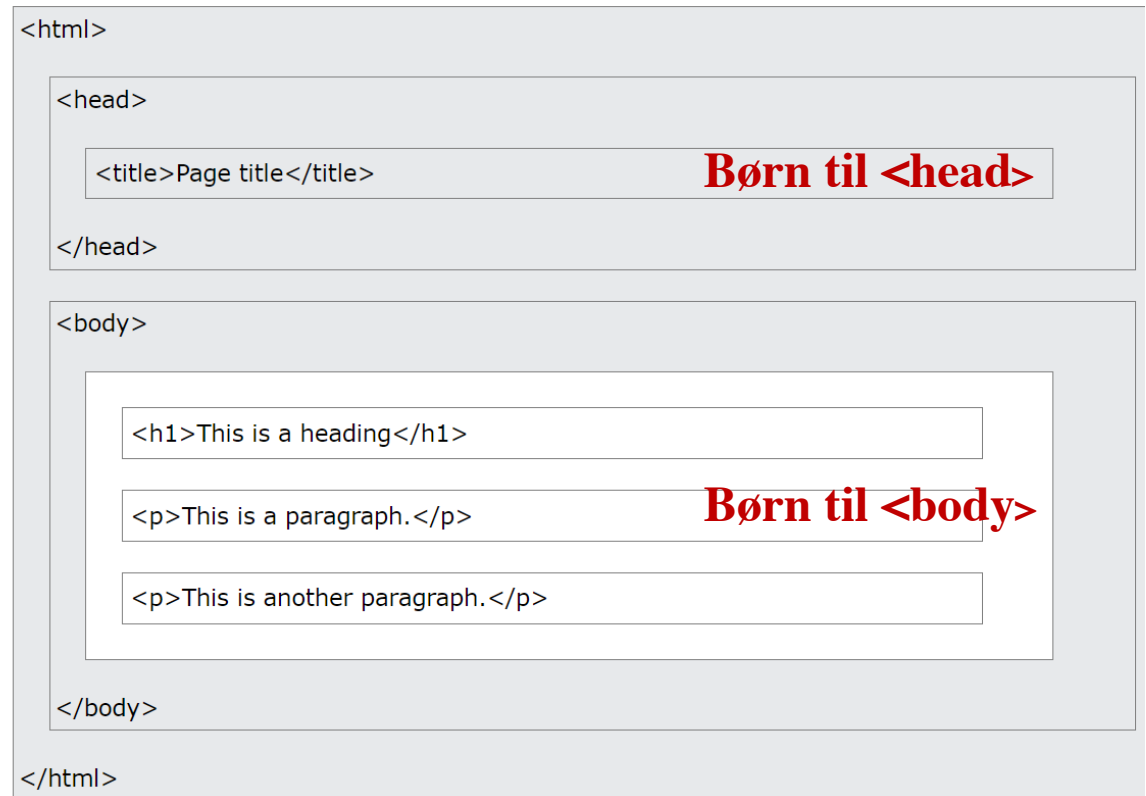
```
    <h1>This is a heading</h1>
```

```
    <p>This is a paragraph</p>
```

```
    <p>This is another paragraph</p>
```

```
  </body>
```

```
</html>
```



// **html-tagget er rod-objektet med to hovedbørn, `<head>` og `<body>`, med egne børn**
// **Øverst fortæller `<!DOCTYPE html>` at det er HTML5-dokument til browseren**

HTML: DOM – document object model

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title></title>
```

```
</head>
```

```
<body>
```


```
</body>
```

```
</html>
```

**Giv dokumentet en
titel!**



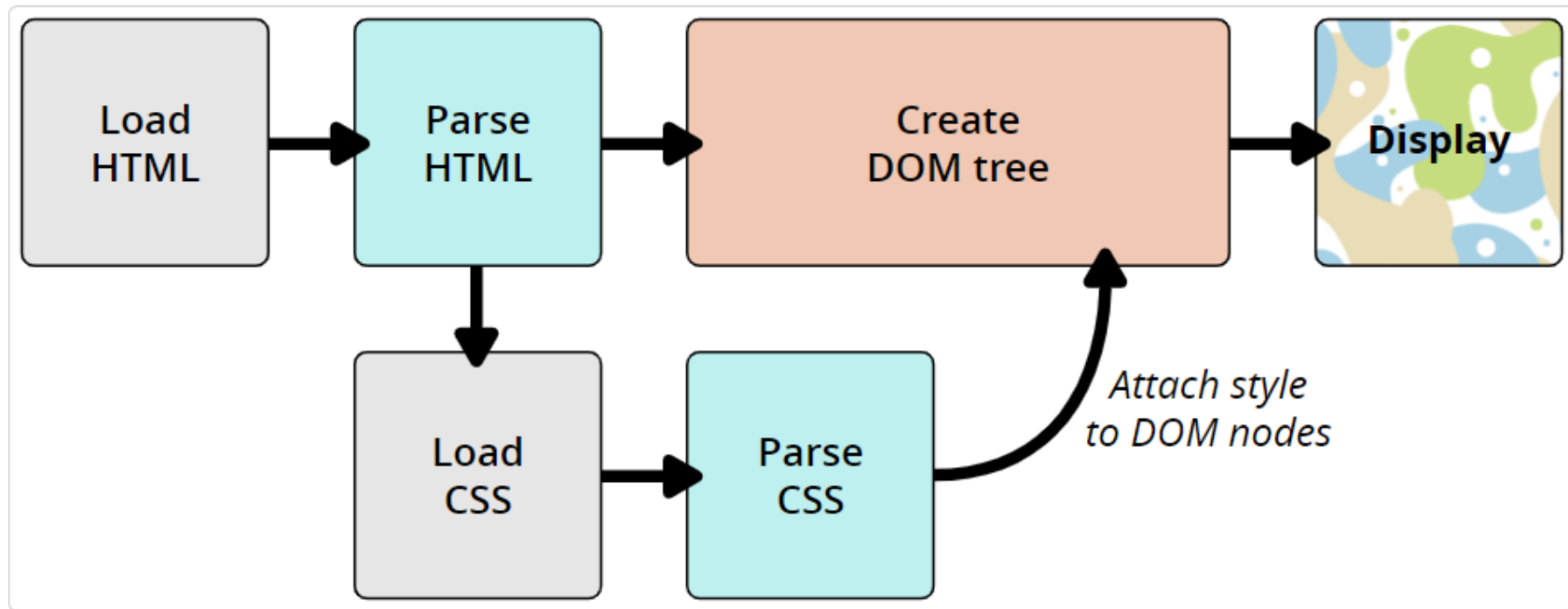
**Indsæt html-elementer i body-
tagget:**
Fx h1, h2, p osv.



https://www.w3schools.com/html/html_elements.asp

HTML: DOM – document object model

Træstrukturen bliver dannet af browseren, og elementerne i den bliver stiliseret via CSS – og endelig vist på skærm



HTML: <head> og <body>

<head> tagget indeholder info om html-dokumentet, men ikke selve indholdet
Heri angiver man dokumentets titel via <titel>, link til diverse filer og ikoner via <link> samt evt. javascript-kode samt fx beskrivelser og nøgleord til søgemaskiner.

<body> tagget indeholder den synlige info til brugeren, via forskellige elementer, fx:

**** - billeder

<a> - link

<p> - tekst-paragraf

**** - en ordnet liste med **** list-items

<table> - en tabel

<form> - fx brugerinput i tekstboks via <input type= "text">

<canvas> til fx animationer eller spil programmeret via javascript

<button> til knapper (øvrige elementer kan også tildeles knap-funktion)

<div> en inddeling, division, af siden som igen kan indeholde øvrige elementer

<!--...--> man kan lægge kommentarer ind i koden

HTML: html-elementernes attributter & block-elementer vs inline-elementer

Elementer inden for <body> tagget kan tildeles attributter/værdier, fx:

En paragraf gøres gul via style:

<p style="color: yellow;"> Denne tekst er nu gul via style </p>

Et canvas gives en størrelse, på 500 x 200 pixel og via style en sort kant på 5 pixel:

<canvas width="500" height="200" style="border:5px solid #000000;"></canvas>

Et image får angivet sin specifikke billedfil via src samt bredde og højde:

Visse elementer lægger sig i udgangspunktet på en ny linje, det er **block-elementer**:

<p> paragraf tager fx ny linje, det samme gør en <div>, en <table> og et <canvas> osv.

Andre elementer fortsætter så vidt muligt på samme linje, det er **inline-elementer**:

 billede fortsætter på samme linje, ligeledes med fx <button>, <input> osv.

HTML: html-elementernes attributter & block-elementer vs inline-elementer
Elementer inden for <body> tagget kan tildeles attributter/værdier, fx:

Tilføj et canvas i jeres html-dokument:

Et canvas gives størrelse, på **500 x 200 pixel** og via style en sort kant på **5 pixel**:
<canvas width="500" height="200" style="border:5px solid #000000;"></canvas>

Tilføj også et billede i html-dokumentet:

Et image får angivet sin specifikke billedfil via src samt bredde og højde:

Visse elementer lægger sig i udgangspunktet på en **ny linje**, det er **block-elementer**:
<p> paragraf tager fx ny linje, det samme gør en <div>, en <table> og et <canvas> osv.
Andre elementer fortsætter så vidt muligt på **samme linje**, det er **inline-elementer**:
 billede fortsætter på samme linje, ligeledes med fx <button>, <input> osv.

HTML: div-element og class & id

<div> benyttes til at lave en division, en inddeling, og den kan fungere som beholder for øvrige elementer – andre **<div>** eller fx bare **<p>** paragraf eller **** billede

```
<div class="beholder">
```

```
  <p>En paragraf lagt ind i en div</p>
```

```
  </img>
```

```
</div>
```

class giver html-element et navn, og så kan man tilgå den nemt via Javascript-kode eller stilisere den via CSS. Flere elementer kan høre til samme **class**.

Den samme gælder **id**, fx **id="beholder"**, mens **class**-tagget kan deles af flere html-elementer, er **id** individuelt på det specifikke element.