Indhold

```
<!DOCTYPE html>
<html>
<body>

<h2>Hvad kan man bruge JavaScript til?</h2>

<p id="demo">Med JavaScript kan du ændre indholdet af html-elementer.</p>

<button type="button" onclick='document.getElementById("demo").innerHTML = "Og det fede er, at det er on-the-fly!"'>klik her!</button>

</body>
</html>
```



```
<!DOCTYPE html>
<html>
<body>
 <div style="text-align:center">
   <h2>Hvad kan man bruge JavaScript til?</h2>
   <p>JavaScript kan ændre HTML attributværdier.</p>
   <p>I dette eksempel JavaScript ændrer værdien af src (source) attributtem af et image-tag.</p>
   <button onclick="document.getElementById('myImage').src='pic_bulbon.gif'">Tænd lyset</button>
   <img id="myImage" src="pic_bulboff.gif" style="width:100px">
   <button onclick="document.getElementById('myImage').src='pic_bulboff.gif'">Sluk lyset</button>
 </div>
</body>
</html>
```

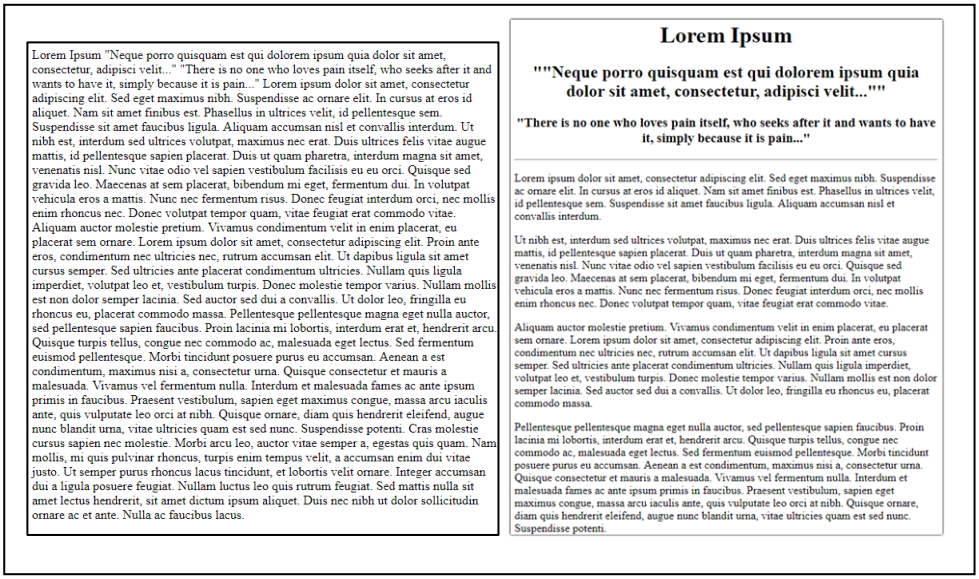
Intro

HTML står for Hypertext Markup Language. Hypertext, refererer til muligheden for at ”linke” til andre sider, og Markup Language, betyder at det ikke er et programmeringssprog, men er et markeringssprog. HTML er den generelle struktur bag alle hjemmesider. At HTML er markeringssprog betyder, at vi skal markere hver del af teksten med den kategori, som den bedst tilhører. En overskrift kunne være én kategori, et afsnit kunne være en anden.

Lektion 1

HTML

HTML er oprindeligt tænkt som en måde, hvorpå man kan skrive og dele information. Derfor er mange af elementerne inspireret af den måde man skriver artikler på. Når vi markerer hvert tekstelement, kan browseren hjælpe os med at formatere teksten. Det kender vi også fra tekstbehandling, hvor vi definerer tekst som overskrift, citat eller afsnit. Når tekst er formateret, ser det pænere ud og det øger læsbarheden af teksten. Men det hjælper også søgemaskinerne med at finde ud af, hvad siden handler om, og hvor den vigtige information står. Tekster som er udgivet i HTML er direkte søgbare for søgemaskinerne.



Figur 1 Til venstre: ren tekst. Til højre: samme tekst HTML formateret.

I HTML benytter vi to typer markeringstags. Containertags og seperatortags. Et containertag indeholder tekst og separatortags deler artiklen ind i afsnit og kapitler eller angiver linieskift, links eller billeder.

I Figur 1, kan du til venstre se en tekst som ikke er formateret. Feltet til højre viser en tekst som er formateret i HTML. Der er brugt 6 forskellige HTML tags.

De øverste 3 linjer er centreret. Hver linje er markeret som overskrift. Den første som overskrift 1, den næste som overskrift 2 og den sidste som overskrift 3. Herefter har jeg lavet en linjeskift som er repræsenteret ved den vandrette streg. Og til sidst er hvert afsnit markeret separat som afsnit.

Editor

For at kunne skrive HTML filer har du brug for en editor. En editor er et program som kan skrive ren tekst. Word er en tekst editor, men kan ikke bruges da den gemmer oplysninger om formateringen i filen (.docx eller .doc). Derfor kan du kun bruge et program som kan gemme ren tekst. Det kunne være Notesblok, som er inkluderet i Windows eller TextEdit, som er inkluderet i MacOs. Hvis du ønsker at bruge TextEdit, skal du ændre tekst formatet fra RTF (Richt text format) til Plain Text. Dette gøres under: (Format > Make Plain Text) eller hvis du ønsker at benytte det som standart: (Preferences > Format). Man skal også huske at ændre at HTML skal åbnes som RTF.

Du kan teste om du gemmer i ren tekst, ved at lave en fil hvor du gemmer f.eks. to tegn. Det kan være to bogstaver f.eks. ”AA”. Når du gemmer filen skal du se hvor meget filen fylder. Fordi computeren bruger 1 byte til at angive et tegn, fylder filen 2 bytes. Fylder den mere er det ikke ren tekst!

Alternativt kan man hente en editor på nettet. For eksempel Brackets eller Atom.

Tags

Et containertag er et tagnavn er omgivet af to tegn ”<tagnavn>” og udgør tilsammen et tag. Containertags optræder par vis. Det første tag er et start-tag det sidste er et slut-tag. Et slut-tag er skiller sig ud ved at have et /-tegn foran tagnavnet ”</tagnavn>”. Containertags kan indehold andre separator og container – tags.

Separatortags består kun af et element fx:
. Der hører altså ikke noget sluttag til.

Tags kan have attributter, som angiver en egenskab. Det kan være størrelsen på et billede eller højden på en font. Vi vender tilbage til tags-attributter senere.

Tags, skrives med små bogstaver.

På engelsk kan begreberne for tags variere. Man kan tale om *Start tag* eller O*pening tag* og mene det samme. Ligeledes kan *Closing tag* og *End tag* være det samme. Containertags og seperatortags omtales også som elementer.

Her er alle de elementer som der er brugt til at formatere teksten i eksemplet.

- …

- <h1>…</h1>
- <h2>…</h2>
- <h3>…</h3>
-
- <p>…<p>

Center-tag centrerer teksten på siden. H1 (Heading), er overskrift 1. H2 er overskrift 2 og H3 er overskrift 3. br betyder ”break” og angiver et linje skift. Brug ikke
 til at adskille afsnit, men angiv hvert afsnit som et separat afsnit. P er paragraph og angiver et tekstafsnit efterfulgt af ekstra linjeafstand.

Opgave 1:

Opg A. Kan du se hvilke tags som er containertags og hvilke som er seperatortags?

Opg B.

- Hent eksempelteksten https:eastcape.dk/hhx/tekst.txt.
- Gem teksten på din computer og omdøb filen til tekst.html.
- Åben filen i en editor. Indsæt de 6 forskellige tags i teksten så den kommer til at ligne Figur 1.
- Brug din browser for at se resultatet.

Tags og attributter

At lave et hyperlink <a>

Tags kan indeholde attributter. Attributter angiver en egenskab. Ikke alle tags kræver attributter. For eksempel definerer containertagget ”<a>” et hyperlink. ”<a>”-taggets vigtigste attribut er ”href” som angiver url’en som hyperlinket peger på, href betyder hyperlink reference. Man kan lave hyperlink til næsten alt indhold som kan findes på internettet.

```
<a href="https://www.IBC.dk">Besøg IBC’s internetside</a>
```

Figur 2, eksempel på et <a> tag

Eksemplet i Figur 2, vil vise en blå understreget tekst ”Besøg IBC’s internetside” indtil brugeren klikker på hyperlinket. Et link som har været aktiveret, vil være purple.

Lektion 2

At indsætte et billede <img>

Imagetagget indsætter et billede på siden. Den kræver to attributter: ”src” og ”alt”. ”scr” er referencen til billedet. Det kan både være lokalt på computeren eller en url. Vi bruger også begreberne absolut og relativ reference. Den absolutte er en komplet url, f.eks. scr=”https:IBC.dk/logo.png” og den relative kun peger på filen f.eks.: scr=”logo.png”.

Attributten ”alt”, giver brugeren en alternativ information om billedet. Internettet bliver brugt af mange mennesker, men ikke alle er i stand til at læse eller se. Internet organisationen w3c (https://www.w3.org/) har derfor bestemt, at alle billeder beskrives med en tekst, således at svagtseende kan få fortalt hvad billedet forestiller. Hvis du gerne vil teste, om din side overholder reglerne kan du tjekke det her: https://validator.w3.org/

I eksemplet, er det yderligere to attributter. ”Height” og ”width”, som angiver højden og bredden for billedet i pixels. Som hovedregel, skal billedet passe i størrelsen, men disse to attributter, tillader at tilpasse billedet.

```
<img src="smiley.gif" alt="Smiley face" height="42" width="42">
```

Figur 3, eksempel på et <img> tag

HTML filstruktur

Der findes flere versioner af HTML, og for at brugerens browser ved hvilken version vores dokument er skrevet i, skal det første tag angive versionen ”<!doctype html>”.

Dette er faktisk ikke et tag, men en instruktion, udelukkende for at fortælle browseren hvilken version af html dokumentet er skrevet i. I html 5 finders der kun denne ene version.

Hele dokumentet skal være skrevet i <html> tagget. Dette tag er rod elementet i hele dokumentet. Html-elementet består af to yderligere elementer. <head> og <body>

```
<!doctype html>
<html lang=”da”>
  <head>
    <meta charset=”utf-8”>
    <title>Dokument titel</title>
    <style>
    </style>
  </head>
 <body>

 </body>
</html>
```

Figur 4, html fil struktur

Header-sektion indeholder metadata omkring siden, sidens titel og style sektion.

Body-sektionen er sidens indhold og består af tekst, hyperlinks, billeder, tabeller, lister m.m.

Bemærk hvordan jeg har indrykket tags i fig.4. Det hjælper med strukturen at åbne og lukke tags står over hinanden og, hvis det er et container tag, at de er indrykket.

Opgave 2.

Du skal lave et nyt html-dokument hvor du efterlyser din kat.

MIN KAT JULIUS

-er løbet væk!

Julius stak af fra min have i går den 1. oktober.
Han er orangerød og er en Main Coone.
Jeg er **MEGET** glad for den.

Hvis du ser min kat, ring på 123456789

TAK

Før du går i gang, skal du have nogle ting på plads. Du skal oprette en folder på din harddisk. F.eks. ”c:\efterlysning”. I denne mappe skal du gemme det billede af katten du vil bruge, og der er her, du skal oprette html-filen. Du skal altså bruge en relativ reference til billedet. Billedet finder du https:eastcape.dk/hhx/julius/julius.png.

Nu er du klar til at oprette HTML-filen som du skal navngive ”index.html”. Index.html er den fil i roden af dit websted, som bliver automatisk åbnet af din browser. Filen kan også hedde home eller default og have andre extensions som fx .php, .asp eller .htm.

1. Opret en ny fil og kald den index.html
2. Start med at skrive den samme struktur som du ser i Figur 4.
3. Lav en titel og en overskrift. Husk at titel skal være i head og overskriften i body sektionen.
4. Den første overskrift skal være H1
5. Hent billedet af julius https:eastcape.dk/hhx/julius/julius.png og gem det i samme folder som din HTML fil.
6. Nu kan du i body-sektionen oprette et <img> tag, hvor du bruger en relativ reference til billedet. Glem ikke <alt> attributten.

7. Gem HTML filen og åben den i din browser.

Ser billedet rigtigt ud i størrelsen? Hvis ikke kan du rette det til med width og height attributterne. Hvis ikke du kan se billedet, kan du altid i Chrome trykke F12 eller højre klik og vælge vis kilde. Her kan du se det browseren ser.

```
<img src="julius2.png" alt="min kat Julius" height="300" width="300">
```

Underbilledet skal der være en overskrift-2 (h2) og herefter en kort beskrivelse. Til dette kan du bruge <p> tags. <p> bruges til afsnit og betyder paragraph.

```
<h2>-er løbet væk.</h2>
<p>Hvis du har set Julius, så send en email til mig på cbru@ibc.dk</p>
```

Det er muligt at få browseren til at åbne et mail-program. Dette kan vi gøre ved at definere e-mailadressen som et hyperlink. Bemærk, det virker kun, hvis du har et email program installeret på computeren.

```
<p>Hvis du har set Julius, så send en email til mig på <a
href=”mailto:cbru@ibc.dk”> cbru@ibc.dk<a/></p>
```

Gem HTML filen og åben den i din browser. Hvis du allerede har den åben, kan du trykke F5 for at opdatere op Windows og ctrl+R på Mac.

For at understrege, hvor gerne jeg vil have katten tilbage, kan vi lægge lidt tryk på, ved at bruge tagget. -tagget giver en kursiv effekt.

```
<p>Jeg er <em>meget<em> glad for den!</p>
```

En anden mulighed for at lægge vægt på vigtigheden, er ved at bruge .

```
<p><strong>Tak</strong><p>
```

Ofte er det vil det være en anden person som skal ændre i filen næste gang. Derfor kan det være hensigtsmæssig at efterlade kommentarer. Det kan vi i HTML gøre ved at bruge <!— kommentar -->. Du kan kun se kommentarer hvis du kigger i koden og vil altså ikke kunne se det på siden.

```
<!-- Katten er slet ikke væk – det er bare en opgave -->
```

Hvad kan du gøre for at gøre siden bedre? Tilføje flere billeder? Et kort over hvor katten blev væk?

Når katten bliver fundet kan du overstrege teksten med <del> taggen og med store bogstaver skrive ”JULIUS ER FUNDET!”.

Du ved nu hvad en editor er, og du ved, hvordan man gemmer en fil som ren tekst.

Du ved hvordan du laver et HTML element. At der findes container tags og seperator tags, at det kaldes en attribut når man tilføjer en egenskab til elementet.

Du ved at man skal bruge en bestemt skabelon, når man laver et HTML dokument.

Fordi du kender skabelonen ved du, hvad man bruger <body> </body>til.

Du kan med HTML tags lave overskrift, underoverskrift, afsnit, indsætte billeder, lave links, linje og emne -skift.

Lektion 3

Landesprog

Hvis du følger strukturen som vist i fig. 4, kan du også angive hvilket sprog siden er på. Det gør du i HTML tagget som skal være root elementet på din side. Husk at lukke </html>-tagget som det sidste tag.

```
<html lang=”DA”>…</html>
```

Det er også god praksis at angive hvilket tegnsæt som siden understøtter. For at få dansk understøttelse bruger vi normalt UTF-8 tegnsættet. <meta>-tags hører til i <head>-sektionen

```
<meta charset=”UTF-8”>
```

Metadata

Metadata er information om siden. Der er en lang række ting man kan skrive som metadata i et HTML dokument. Meta tags skrives i <head>-sektionen (se fig. 4). Når du åbnet dit dokument i en browser vil man ikke kunne se hvad man skriver i <meta>-tags. Men google og andre søgemaskiner kan og bruger informationen til deres søge algoritmer.

F.eks. kan man bruge <meta>-tags til at skrive hvem forfatteren er.

```
<meta name=”author” content=”skriv dig navn her”>
```

For at søgemaskinerne bedre kan kategorisere hvad siden handler om er det godt at angive den beskrivelse og nogle nøgleord.

```
<meta name=”description” content=”siden er lavet for at finde min kat”>
<meta name=”keyword” content=”Julius, kat, væk”>
```

Style-attributten

Et HTML element kan have attributter. Vi har allerede lært scr, alt, href, name, charset som er de attributter vi brugte i Julius opgaven. I dette afsnit skal vi arbejde med style-atributten.

Style attributten kan bruges til at ændre på udseende af elementet. Vi kan bestemme tekst størrelse, farve, kursiv og mange andre ting. Har du lavet opgaven med Julius, vil jeg foreslå at du bruger den. Ellers kan du finde min version her:

```
<!DOCTYPE html>
<html lang="DK">
 <head>
   <meta charset="UTF-8">
   <title>Julius er væk</title>
   <style>
   </style>
 </head>
 <body>
   <!-- katten er ikke løbet væk. Den har det godt og er hjemme !-->
  <div style=”text-align:center”>
       <h1>Min kat Julius</h1>
      <img src="julius.png" alt="min kat Julius">
      <br>
   <h2><del>-er løbet væk!</del></h2>
   <h1>Julius er fundet!</h1>
   <p>Julius stak af fra min have i går den 1. oktober.
   <br>Han er orangerød og er en Main Coone.<br>Jeg er <em>MEGET</em> glad for den.
   <h3>Hvis du ser min kat, ring på 123456789</h3>
   <h3>eller send en email på: <a href="emailto:cbru@ibc.dk">cbru@ibc.dk</a>
   <p><strong>TAK</strong></p>
  </div>
 </body>
</html>
```

Nu kunne jeg godt tænke mig, at vi arbejder videre med siden og gør den mere til en hjemmeside for Julius. Først kunne jeg godt tænke mig at gøre noget ved overskriften. Måske vil det passe med en rød-overskrift, så det passer til kattens farve.

Hvis vi vil ændre på farven af et element til rød, kan v tilføje attributten: style=”color:red”. I HTML kan farver angives på forskellige måder. De kan angives som RGB, HEX HSL, RGBA eller HSLA værdier. I kan se en liste over de foruddefinerede farver her: https://htmlcolor-codes.com/color-names. Men i første omgang kan i bare bruge blue, black, red, yellow etc.

Lad os ændre farven på overskriften til rød. Tilføj attributten: style=”color:red” til dit <h1>-tag:

```
<h1 style="color:red">Min kat Julius</h1>
```

Jeg synes heller ikke t overskriften er stor nok. Jeg ønsker at gøre den stører. Det kan du ændre den med værdien ”font-size”. Font-size hører også til style attributten og den kan derfor bare tilføjes. Husk at bruge ”; ” til at separerer værdierne.

```
<h1 style="color:red; font-size:100px">
```

Størrelser

Nu er billedet bare blevet for lille til overskriften. For at gøre billedet stører, kan vi tilføje to attributter til img attributten, height og width. Prøv at angive height til 450 og width til 600. De to tal angiver bredden og højden af billedet i pixels.

```
<img src="julius.png" alt="min kat Julius" height="450" width="600">
```

Jeres skræm har måske en opløsning på 1920x1080 pixels og 1 pixel er én lille prik på skærmen. Størrelser kan angives på to måder. Absolutte og relative størrelser. Den absolutte angivelse er f.eks. angivelsen i pixels. Den relative følger den størrelse som skærmen og dermed også browservinduet har. Den dynamiske størrelse kan angives i procent af bredden og højden. Hvis ikke i kan klare jer med pixels eller procent kan i se alle størrelses angivelser her: https://www.w3schools.com/CSSref/css_units.asp

Style i head

Som en sidste ting kunne jeg godt tænke mig at få en rød kasse rundt om overskriften. Kassen skal have tykkelsen 10px. Vi kan tilføje en ny egenskab til vores attribut, ”border”. Border, tegner en kant rundt om hele elementet som det er angivet i. Det kan være rundt om et billede eller en tekst. Hvis man har mange container elementer, kan det være en fordel at give den en border, så man kan se hvor de er.

Border kan angives som værende solid, dotted, dashed, double og mange flere. Se dem her: https://www.w3schools.com/css/css_border.asp

```
<h1 style="color:red; font-size:100px; border:solid 10px">
```

Som du kan se, vokser vores <h1>-element, og der skal ikke mange attributter til, før det bliver uoverskueligt. En anden ting er, at vi faktisk bruger 2, <h1>-overskrifter, men det er kun den ene som er tilpasset. Man kan godt kopiere det tilpassede element. Således at vi får 2 <h1>-elementer som er lige lange. Det vil gøre vedligeholdelsen utrolig besværlig, fordi man skal rette to steder, hvis man vil ændre

noget i overskriften. Derfor er løsningen at tilføje <style></style> i <head> sektionen. Se hvor <style> er placeret i figur 4.

Inden for de to <style> tags, kan du nu definere hvordan din h1 overskrift skal se ud, så der gælder det samme for alle <h1>-overskrifter. Husk at slette dine style attributter i dine <h1>-tags. Du vil nu opleve at din skrift skifter til blå og bliver en smule større, hvis du bruger eksemplet her:

```
<style>
  h1{
    color:blue;
    font-size:130px;
    border:solid 10px;}
 </style>
```

Figur 5, Attributter i <style>- sektionen i <head>

Lektion 4

Undersider

Vi er jo ved at udbygge siden til at være en egentlig hjemmeside for Julius. Så jeg tænker at det vil være passende med en underside som fortæller lidt om katteracen Maine-Coon.

Lav en ny side i din editor, hvor du følger skabelonen fra fig. 4. Udfylder med alle <meta> tags data og gem siden som: ”om-maine-coon.html ”

I din index.html kan du nederst tilføje et link til den nye side.

```
<p>Klik her, for mere information om racen: <a href="om-maine-coon.html">Maine Coon</a><p>
```

På adressen https://eastcape.dk/hhx/maine-coon/maine-coon.zip kan du hente en zippet fil med billeder og tekst som du skal bruge. Og på samme adresse kan du se mit forslag til en formattering https://eastcape.dk/hhx/maine-coon/maine-coon.pdf

Opgave

Tilpas teksten med <h1>,<h2>,<h3>,<p>-tags og indsæt billeder med <img>-tag. Husk du kan med fordel bruge et -container element til at indeholde ens formatering som f.eks centrering.

Font

Når vi taler om font, mener vi den måde bogstaverne ser ud på. Vi kender måske alle comics sans, som ofte bliver brugt i invitationer til børne fødselsdage. Fonten ser ud som var den skrevet i hånden.

Dette er comic sans. Det er en font som ligner håndskrift.

Times, helvetica, serif, sans serif er alle font-familie navne og der findes et utrolig stort antal fonte og rigtig mange af disse er gratis af bruge.

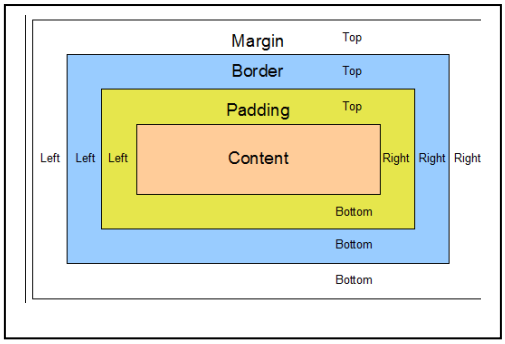
Prøv at tilføje atributten: font-family: "Gill Sans", sans-serif; -til <h1>-tagget. Og se hvordan overskriften skifter udseende.

```
<h1 style=” font-family: "Gill Sans", sans-serif;”>
```

Grunden til at der er to font navne er, at det fungerer som et fall-back system. Det betyder at hvis ikke den første font er installeret, bruger systemet den næste font. Jeg angiver altid min fall-back font som en generisk font fordi de som oftest altid er installeret på computeren.

Margin og padding

Vi har tidligere arbejdet med attributten border som laver en ramme rund om vores indhold. Men som du kan se på Figur 6, er der udover Border, også Padding og Margin. Margin kender I. Det er margin som gør at teksten ikke står helt ud til kanten på papiret. Padding er, hvor tæt resten af teksten står på elementet.



Figur 6, Margin, border, padding og indhold

Først ville det være dejligt hvis vi kunne få teksten lidt væk fra kanten. Det gør vi med margin. For at give teksten margin i begge sider skal vi bruge to linjer. I mit eksempel har jeg brug for at give <h3>,<h4> og <p> margin. <h1> og <h2> har jeg centreret. Derfor kan jeg tilføje følgende til min <style>-sektion.

```
h3,h4,p{
  margin-left: 20%;
  margin-right: 20%;
}
```

For at vise hvordan padding virker, kan vi tage et billede og tilføje en border på 10px.

```
<img src="maine-coon-race.jpg" alt="maine-coon" style="border:solid 10px; ">
```

Se hvordan billedet for en ramme omkring og hvor tæt det ligger til billedet. Tilføj nu padding på 50px og se hvordan rammen bliver skubbet væk fra billedet.

```
<img src="maine-coon-race.jpg" alt="maine-coon" style="border:solid 10px;padding:50px ">
```

Baggrund og tekst farve

Det første billede jeg har brugt i min løsning, er en kat på hvid baggrund. Man kan ikke se hvor billedet starter eller hvor det ender. Det kan man så med billede nummer to, det sorte billede. Medmindre, at man ændrer baggrunden til sort. Problemet er bare, at ændrer vi baggrunden til sort, vil vi ikke kunne se teksten. Derfor er vi også nød til at ændre farven på teksten til hvid. Jeg tilføjer også lige lidt padding, bare for æstetikkens skyld.

```
<div style="background-color: black; color:white; padding: 2%">
```

Du ved nu hvad meta data er og hvor man skal skrive det i sit dokument.

Du har også lært hvordan man fortæller hvilket sprog siden er skrevet i.

Du har lært at style attributten kan ændre på udseende af tekste og at vi har en bestemt sektion i <head> hvor vi kan lave generelle ændringer.

Du kan linke til en underside og du kan ændre den fonten på din side.

Nu kan du kende forskel på margin, border, padding og content af et html element.

Du kan bruge container tags til at ændre baggrunds og tekst fave for en enkelt del på en side.

Lektion 5

HTML sorteret og usorteret Liste

Når vi skriver tekster, har vi tit behov for at lister. Der findes to typer lister. Sorterede lister og usorterede lister. For det enkelte element i listen bruger vi det samme tag: <li>, list item. Det er altså container tagget som bestemmer type af liste. for ordered list, eller <ul> for unordered list.

```
<ol>
  <li>Cykel</li>
  <li>Pedal</li>
  <li>Cykelstyr</li>
  <li>Ringeklokke</li>
</ol>
```

Figur 7, sorteret liste

```
<ul>
  <li>Cykel</li>
  <li>Pedal</li>
  <li>Cykelstyr</li>
  <li>Ringeklokke</li>
</ul>
```

Figur 8, usorteret liste

Med mindre at vi angiver andet, vil hvert punkt i listen være angivet med en rund, sort cirkel - bulletpoint. Der findes en værdi til style-attributten som bestemmer listens layout. Værdierne kan være: disk, circle, square og none.

```
<ul style="list-style-type:square">
```

En liste som menu:

Vandrette lister kan blive brugt til at lave menuer! Opret et nyt dokument, gem det som menu.html. Start med at hente HTML-skabelonen fra figur 4, og tilføj følgende i din <body>-sektion:

```
<h2>En menu</h2>
<p>Dette er et eksempel på, hvordan man kan lave en menu i HTML:</p>

<ul>
 <li><a href="#Julius">Julius</a></li>
 <li><a href="#OmJulius"> Om Julius </a></li>
 <li><a href="#OmMaineCoon">Om Maine Coon</a></li>
 <li><a href="#Kontakt">Kontakt Julius</a></li>
</ul>
```

Figur 9, Menu ved brug af ID

Hvis du gemmer filen, og åbner den i din browser, vil du se at du har lavet en liste bestående af links.

Hvis du undrer dig over, at href værdien ikke er en url, men derimod starter med et #, er det godt observeret. Det er sådan, at hvis man angiver en href tag med et # - kaldes det et page anker. Det betyder at man kan springe til det sted på siden som menupunktet henviser til. Derfor skal vi i vores html-kode fortælle hvor ”ankeret” er kastet. Det gør vi med ”id ”-attributten. Typisk vil vi angive det i en overskrift, men det kan være i hvilket som helt element på siden.

```
<h2 id=”julius”>
```

Hvis du hellere vil linke til en anden side er det selvfølgelig navnet på filen du skal skrive i <a> tagget.

```
<ul>
 <li><a href="index.html">Julius</a></li>
 <li><a href="OmJulius.html"> Om Julius </a></li>
 <li><a href="OmMaineCoon.html">Om Maine Coon</a></li>
 <li><a href="Kontakt.html">Kontakt Julius</a></li>
</ul>
```

Som det første skal vi fjerne punkterne foran de enkelte list-elementer. Det vil vi gøre i <style>-sektionen. Her tilføjer vi:

```
ul{
  list-style-type: none;
  }
```

Her udover skal baggrunden farves orange, som Julius samt vi fjerner alt luft omkring med margin og padding. Jeg bruger en attribut som vi ikke har brugt før: overflow: hidden;. Den gør at hvis et af elementerne fylder mere, vil der ikke komme en scrollbar eller andet. Elementet er tvunget til at blive inden for de rammer som vi har defineret.

```
ul{
  list-style-type: none;
  margin: 0;
  padding: 0;
  background-color: orange;
  overflow: hidden;
  }
```

For at gøre listen horisontal bruger vi en float probberty på alle list elementer, <li>. Tilføj til <style>-sektionen:

```
li {
    float: left;
}
```

Gen filen og åben den i din browser – se hvordan alle list elementer nu står på linje. Men hyperlink er som standart blå og understreget, og det er ikke pænt i en menu. Derfor skal vi style, hvordan hyperlink i lister skal se ud, men kun i lister – hyperlink andre steder skal beholde deres udseende. Det kan vi gøre ved i vores <style>-sektion af kombinere <li> med <a> attributten. Nu kan vi style hyperlinks. Teksten skal være centreret, der skal være en padding på 16px, teksten skal være hvid og vi vil fjerne understregningen af hyperlinket. Det gør vi med text-decoration: none;. Som det sidste skal vi sikre os at elementet bliver vist samle. Det gør vi med display: block;. De andre attributter kender du og kan selv tilføje.

Erstat de tre linjer med de rigtige attributter. Du kan ikke kopiere linierne fra boksen, men du er selv nød til at finde ud af hvad der skal stå!!

```
li a {
  ”tekst centreret”;
  ”padding 16px”;
  ”hvid tekst”;
  text-decoration: none;
  display: block;
}
```

For lige at gøre det rigtigt lækkert, når man scroller med menuen, kan mail tilføje:

```
html{

  scroll-behavior: smooth;

}
```

Nu har du et forslag til hvordan man laver en menu med page anker og hyperlink

Du kan lave sorterede og usorterede lister

Du har brugt nogle af de attributter som vi har lært tidligere. Ændre farven på teksten og baggrundsfarven.

Du ved nu hvordan man kan ændre udseende på hyperlinks

Lektion 6

CSS ekstern, intern og indlejret

Indtil nu har vi set på to måder at style vores elementer på. Vi kan indlejre vores style attribut i HTML elementet, eller vi kan skrive det i <style>-sektionen i vores <head>-sektion. Når man skriver det i <style>-sektionen, kalder man det intern CSS. CSS betyder Cascading Style Sheets. Men vi kan faktisk også gemme det eksternt, i sin egen fil, med den fordel at flere HTML-filer kan bruge det samme style-sheet.

Vi kopierer alt hvad der står mellem de to <style>-tag i <head>-sektionen og gemmer det i en ny fil. Filen kalder du f.eks. ”mystylesheet.css”.

Nu kan du inkludere dit stylesheet i alle dine html filer. Det gør du i **<head>**-sektionen:

```
<link rel="stylesheet" type="text/css" href="mystylesheet.css">
```

Husk, da det er en relativ url vi bruger, skal html og css filen være i samme mappe.

Husk at tilføje css filen til din index.html fil!

Opgave:

Du skal nu udvide Julius hjemmeside. Index.html skal være hoved siden. Du skal ved hjælp af alt det du har lært indtil nu, få den til at se ”pæn ud”. Du skal implementere menuen fra opgaven vi har lavet tidligere og du skal lave de tre sider som menuen henviser til: om Main Coon, om Julius og en siden hvor der står kontakt informationer.

Classes og ID i CSS

Classes, er et ord inden for datalogi, som man bruger til at betegne flere elementer med de samme egenskaber. På dansk oversætter vi det med klasse. ID, identificerer det enkelte element. Man kan sammenligne det med jeres hhx klasse, det er en samling af elever. Skal jeg sige noget om alle elever, så vælger jeg hele klassen, men hvis jeg vil sige noget om den enkelte elev, bruger jeg ID.

I opgaven om Main-Coon katte har i brugt ID til at lave jeres menu (se Figur 9). Hvis ikke du har lavet opgaven, så kan du hente min version her: https://eastcape.dk/hhx/julius-site/julius-site.zip

I har i opgaven lavet page-anker, hvor i bruger ID til at definere hvor på siden menuen skal sende jer hen.

```
<h3 id=”kap4”>kapitel 4</h2>
```

I har give alle <h3> elementer et forskelligt ID, som gør at vi kan pege direkte på det ene element. I kan nu tilføje en linje CSS som kun én h3 overskrift rød, den for kapitel 3.

I din mystylesheet.css –fil skal du i bunden tilføje:

```
#kap3{
  color: red;
}
```

Husk at rette, hvis har kaldt dit ID noget andet. Bemærk at jeg sætter et #-tegn foran. Det betyder at det er et ID. Ønsker vi at ændre flere h3 overskrifter, men ikke dem alle, kan man bruge klasser. For at fortælle at html-elementet tilhører en klasse, tilføjer vi en attribut med en værdi på samme måde som med ID. Jeg tilføjer en class=”subheading”, til alle mine <h3> overskrifter som jeg vil ændre.

```
<h3 class=”subheading”>kapitel 4</h2>
```

I mystylesheet.css kan du nu skrive ændringerne for din klasse. Klasser angiver med et . foran.

```
.subheading {
  font-style: italic;
  font-size: 25px;
}
```

Opgave:

Du skal færdig gøre internetsiden for Julius, så den minimum fortæller noget om racen, om Julius og hvordan man kommer i kontakt med ham.

Du ved nu hvad sorterede og usorterede lister er og hvordan man bruger dem til at lave en menu.

Du har også lært hvordan man laver et page-anker og hvordan man bruger det på en side.

Du har lært hvad indlejret CSS er og hvordan man linker til en CSS-fil

Men vigtigst af alt, så ved du hvad et ID er. Hvordan man kan bruge det til identifikation af et html-element.

Du ved også hvad en klasse er og hvordan man bruger klasser til at skrive css til en gruppe af html-elementer.

Lektion 7

Javascript

Javascript er et af de grundlæggende værktøjer til webdesign. Vi skal derfor prøve at lege lidt med javascript. Javascript er indlejret i html dokumentet. Man kan bruge JavaScript som man bruger attributter

til html-elementer. Læs følgende eksempel som bruger <buttom> html-elementet. Eksemplet skriver en tekst på skærmen, og når man trykker på knappen ændre teksten sig.

```
<!DOCTYPE html>
<html>
<body>

<h2>Hvad kan man bruge JavaScript til?</h2>

<p id="demo">Med JavaScript kan du ændre indholdet af html-elementer.</p>

<button type="button" onclick='document.getElementById("demo").innerHTML = "Og det fede er, at det er on-the-fly!"'>klik her!</button>

</body>
</html>
```

I Atom editoren: opret en ny fil og giv den et navn med ekstensionen .html. Kopier eksemplet herover, og se om du kan få det til at virke i din browser.

```
<!DOCTYPE html>
<html>
<body>
  <div style="text-align:center">
    <h2>Hvad kan man bruge JavaScript til?</h2>
    <p>JavaScript kan ændre HTML attributværdier.</p>
    <p>I dette eksempel JavaScript ændrer værdien af src (source) attributtem af et image-tag.</p>
    <button onclick="document.getElementById('myImage').src='pic_bulbon.gif'">Tænd lyset</button>
    <img id="myImage" src="pic_bulboff.gif" style="width:100px">
    <button onclick="document.getElementById('myImage').src='pic_bulboff.gif'">Sluk lyset</button>
  </div>
</body>
</html>
```

Du kan finde de to billeder som du skal bruge her:

- https://eastcape.dk/hhx/javascript/pic_bulbon.gif
- https://eastcape.dk/hhx/javascript/pic_bulboff.gif

Opgave:

Forklar hvordan eksemplet virker. Hvad det gør og hvorfor det virker. Når du har luret det, kan du kopiere det til en ny fil i Atom ☺

Funktioner

Det er ikke ofte, at man kan indlejre JavaScript på den måde som jeg lige har vist. Ofte skal der flere linjer kode til. Måden vi løser det på er ved at <script>-sektion i vores header, på samme måde som vi har en <style>-sektion. Men til forskel fra <style>-sektionen, må <script> også gerne være i vores <body>-sektion.

Når vi har flere linier JavaScript, opretter vi en funktion som kan udføre den opgave som vi vil løse med JavaScript. For at vi kan skal en forbindelse mellem der hvor vi skriver koden og der hvor vi skal bruge den.

```
<!DOCTYPE html>
<html>
 <head>
  <script>
   function myFunction() {
    document.getElementById("demo").innerHTML = "Trylle trylle - nu er teksten ændret.";
   }
  </script>
 </head>

 <body>
  <h1>Dette er et HTML Dokument</h1>
  <p id="demo">Her er et afsnit</p>
  <button type="button" onclick="myFunction()">Klik her</button>
 </body>
</html>
```

Giver vi funktionen et navn. I dette eksempel har jeg kaldt functionen for: myFunction(). Paranteserne bruges til at sende information til funktionen, men det skal vi ikke bruge endnu. Resten af koden er, på samme måde som når vi skriver CSS, skrevet imellem to tuborgparanteser {}.

Når vi så vil bruge JavaScript koden, kan vi i HTML elementet angive en attribut som definerer en bestemt handling. F.eks når man med musen klikker på en knap. Herefter skal værdien fortælle hvilken funktion vi vil kalde.

```
<button type="button" onclick="myFunction()">Klik her</button>
```

Nu har vi skabt en forbindelse mellem knappen vi trykker på og den JavaScript kode vi har skrevet.

Opgave:

Kan du ændre koden ved at lave en funktion mere som laver teksten tilbage til det oprindelige?