HTML, CSS, Editor, Kommentarer:

**HTML:**

HTML står for ”Hyper Text Markup Language”, og bruges til at opbygge skelettet af en hjemmeside. Dvs. at dets primære formal er at få tekst, billeder, og afsnit ind på siden. Man kan derfor ikke have en hjemmeside uden at mindst have *noget* HTML.

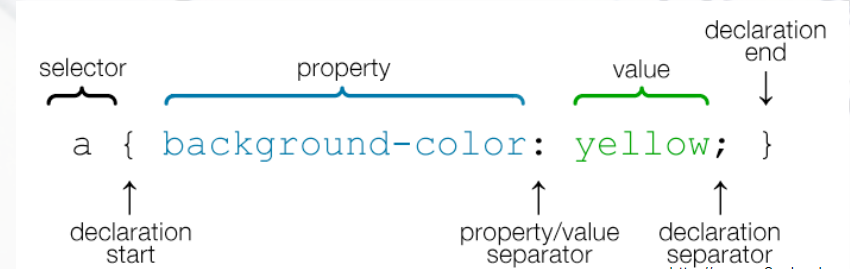
For at kunne gøre brug af HTML til opbygge en hjemmeside, så skal man bruge en ”editor”. En editor er et tekstredigeringsprogram hvori man skriver sin kode (eks. HTML, CSS, JavaScript, C#, C++ og mange flere). Hvilken editor man skal vælge, er helt til op en selv. I kodeverdenen er det en smagsmag, alt efter om man er til teksttype, farve, tema, layout, osv. Det kommer simpelthen an på, hvad man er mest til, og hvad der passer en bedst. Personligt så bruger jeg Visual Studio Code, fordi den har de funktioner og det layout jeg er mest vant til.

**CSS:**

CSS står for ”Cascading Style Sheets”, og er det computersprog der bruges til at style alt HTML. Hvis HTML er skelettet på hjemmesiden, så er CSS’en huden, øjnene, ørene, munden, hovedet, armene, benene osv. Herudover så bestemmer den også farverne, formerne og placeringen af disse ”kropsdele” på hjemmesiden. CSS er stylisten der har smag indenfor alt mode, makeup, tøjstil, og farver. Uden CSS så ville hjemmesiden se meget kedelig og død ud. Det ville bare være sort tekst på en hvid baggrund, uden noget struktur, og det ville køre i en lang række nedad, og slet ikke være ordentligt sat op. CSS er absolut nødvendigt, hvis ikke for farver, former, og design, så i det mindste for at kunne få det til at fungere på forskellige skærmstørrelser, eller sagt på en anden måde, ”mobile first.

”Mobile first” er den tilgang man tager til en hjemmeside med henblik på at opbygge den til mobilen først. Dette vil ikke kun gøre det nemmere at arbejde med tablet og desktopstørrelsen senere hen, men også eliminerer problemer som kunne have opstået, skulle man have startet med desktopversionen først. Mobile first er en tilgang med henblik på at gøre sin hjemmeside 100% mobil venlig først, før alt det andet.

Syntaks for CSS (regel for hvordan det skal skrives op, og i hvilken rækkefølge) ser sådan her ud:



**Editor og kommentarer:**

Editoren skriver i ren tekst og uden formatering (medmindre man har specielle extensions i en given editor, der gør således at man får formatering og andre funktionaliteter på sin tekst). Editoren hjælper også med at highlighte kode og er derfor en stor hjælp hvis man skulle skrive forkert. Den hjælper med at understrege koden således at man nemt kan fine hvilke vendinger og kodesætninger der har mangler, eller er forkerte.

Herunder også linjenummer, hvilke konflikter det har med andre kode, og med en rød markering på ”scrolleren” (det lille vindue der viser alt kode, som man kan scrolle op og ned i.) Derudover så er editoren også meget god til at ”write ahead”, hvilket vil sige at den vil give forslag på at afslutte en kodesætning, alt efter hvad man skriver. Dette er også med til at gøre kodeskrivning nemmere, ikke kun for professionelle, men også begyndere.

Nogle editors har også en funktion der tillader programmøren at se et ”preview” på hvordan siden vil se ud, med den kode man har indtil videre. Dette kan være meget dejligt at have integreret i sin editor, da man hurtigt kan se hvordan hjemmesiden ser ud, om der er mangler, og måske style ting på en anden måde en man lige tænkte over.

Når man skal kode, er det en god idé at træne sin kodeskik. Dvs. at skrive koden i små afsnit ad gangen, lave indrykninger med tabulatoren (”TAB”), linjeskift når det er nødvendigt, og ikke mindst **kommentarer!** Det er *vildt* vigtigt at skrive kommentarer til sin kode, da det ikke kun vil hjælpe en selv i fremtiden, skulle man rette eller tilføje noget til koden, men det vil også hjælpe med at gøre koden mere læsbar for andre, skulle de være nødt til at arbejde på den, eller hjælpe dig selv med den. Det gør det også meget nemmere at finde rundt i koden, og giver koden en meget bedre struktur. Så derfor husk altid at kommentér, kommentér, og atter kommentér!

Tags, Attributter, Elementer, Semantik, SEO:

**Elementer:**

For at kunne bygge en hjemmeside op på en ordentlig måde gennem HTML, så skal man gøre brug af ”tags”. Tags er med til at definere hvilket stykke kode er hvad, dvs. om det er f.eks. en paragraf, titel, undertitel, artikel, billeder mm. Vi refererer til et stykke kode som har et start, og et slut tag, som et ”element”. Man kan derfor også sige at en hjemmeside består af rigtig mange elementer

Et eksempel på et element kunne se således ud:

**<h1> Sikkerhed <h1/>**

**Tags:**

Tagget der er blevet brugt i dette eksempel, er et overskriftstag der hedder ”h1”. Tagget starter ved ”<h1>”, og har et slut tag ved <h1/>. Tags følger denne regel, lige meget hvilket slags tag der er tale om. I ovenstående eksempel betyder det at alt hvad der står imellem dette starttag, og sluttag, vil stå som en overskrift på hjemmesiden. I dette tilfælde vil vi få en overskrift hvor der står ”Sikkerhed”. Der findes flere overskriftstags og angives med ”<h1>, <h2>, <h3>, <h4>, <h5>, <h6>.

**Semantik og SEO:**

Grunden til at de har forskellige numre er fordi det er med til at indikere størrelsen af overskriften på hjemmesiden. Derudover så er det også med til at skabe semantik på hjemmesiden. At noget er semantisk, betyder egentlig bare at noget har mening/betydning. Hvis et tag er semantisk, så fortæller det noget om dets indhold.

Eksempel:

<div>: fortæller ingenting om sit indhold

<table>: fortæller noget om sit indhold idet den markerer alt inden i til at være en ”tabel”, hvilket er også hvad navnet på tagget selv indikerer.

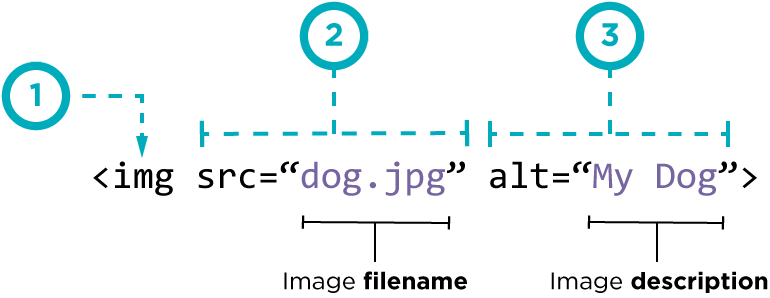
Semantik, eller semantiske tags, er med til at give hjemmesiden struktur, og gør det nemmere for søgemaskiner, at finde hjemmesiden, også kendt som ”SEO”. SEO står for ”Search Engine Optimization”, og betyder i bund og grund at det er med til at gøre det nemmere, og mere relevant for søgemaskiner at finde lige præcis DIN hjemmeside først, over andre hjemmesider. Dette er selvfølgelig relevant for at give din hjemmeside mest mulige besøgere, og kliks.

Følgende semantiske tags er gode at bruge, for at have sin hjemmeside SEO optimeret:

* **<section>:** Inddeler koden i sektioner, og giver mere overblik over de forskellige afsnit i HTML’en
* **<header>:** Bruges til at indikere ”hovedet” på hjemmesiden, og er det der ligger øverst på siden.
* **<footer>:** En footer bruges til at indikere det nederste på hjemmesiden eller ”fødderne” på den. Det står altid i bunden af en hjemmeside.
* **<form>:** Forms bliver brugt til at modtage bruger input, i form af små tekstbokse, hvori brugeren kan bl.a. skrive sit navn/adresse/alder, eller noget helt andet.
* **<table>:** Ligger lidt i navnet idet at det er en tabel som bliver inddelt med et yderligere tag der hedder <tr>, som står for ”table row”.
* **<article>:** Er med til at specificere selvstændigt indhold, og burde give mening på egen hånd uden noget andet yderligere.

Her er også nogle eksempler på andre tags som også bliver brugt, men som ikke er semantiske:

* **<p>:** Er en paragraf, og er stort set bare en linje tekst, der ignorerer linjeskift og mellemrum i koden.
* **<div>:** Bruges som regel til at indikere en sektion i koden som ikke nødvendigvis er nødt til at være semantisk, opfører sig mere eller mindre som <p> tagget.
* **<span>:** Ligner meget <div> tagget, men forskellen ligger i at den bruges til at lænke eksempelvis billede og tekst sammen, således at de står på samme linje. Derudover så er span tagget et inline element, mens div er et block element.
* **<img>:** Bruges til at indsætte et billede ind på hjemmesiden. Filen udpeges med en ”src” attribut, og man skriver stien til billedet mellem de to anførelses tegn.



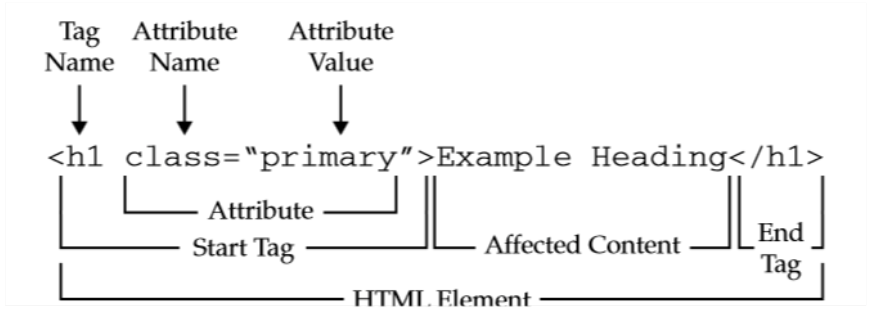
* **<li>** (herunder også <ol> og <ul>): Bruges til at lave lister på hjemmesiden. <ol> bliver brugt til at lave en ”ordered list” hvilket vil sige en liste med nummering på, den kunne f.eks. se sådan her ud:

1. Item
2. Item
3. Item
4. Item

En liste uden nummering bruger <ul> i stedet, som står for ”unordered list”, og kunne se sådan her ud:

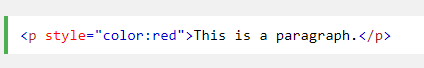
* **Item**
* **Item**
* **Item**
* **Item**

**Attributter:**



Attributter giver HTML elementer noget at relatere til. Dvs. at man nu kan nemmere, mere specifikt pege på hvilket element det er man snakker om. Man kan give et element en attribut som kunne eksempelvis være et ID, eller en klasse. En attribut består af to dele, et attributnavn, og en attributværdi. Navnet indikerer hvilken slags attribut der er tale om, mens værdien af attributten bruges til at pege på selve elementet med den specificerede værdi.

Eksempel:



*Her er attributnavnet ”****style****” og attributværdien ”****color:red****”.*

**Links:**

Links i HTML koden fungere på den måde at man skriver hvad der skal være klikbart imellem to <a> tags, det kunne f.eks. se sådan her ud:



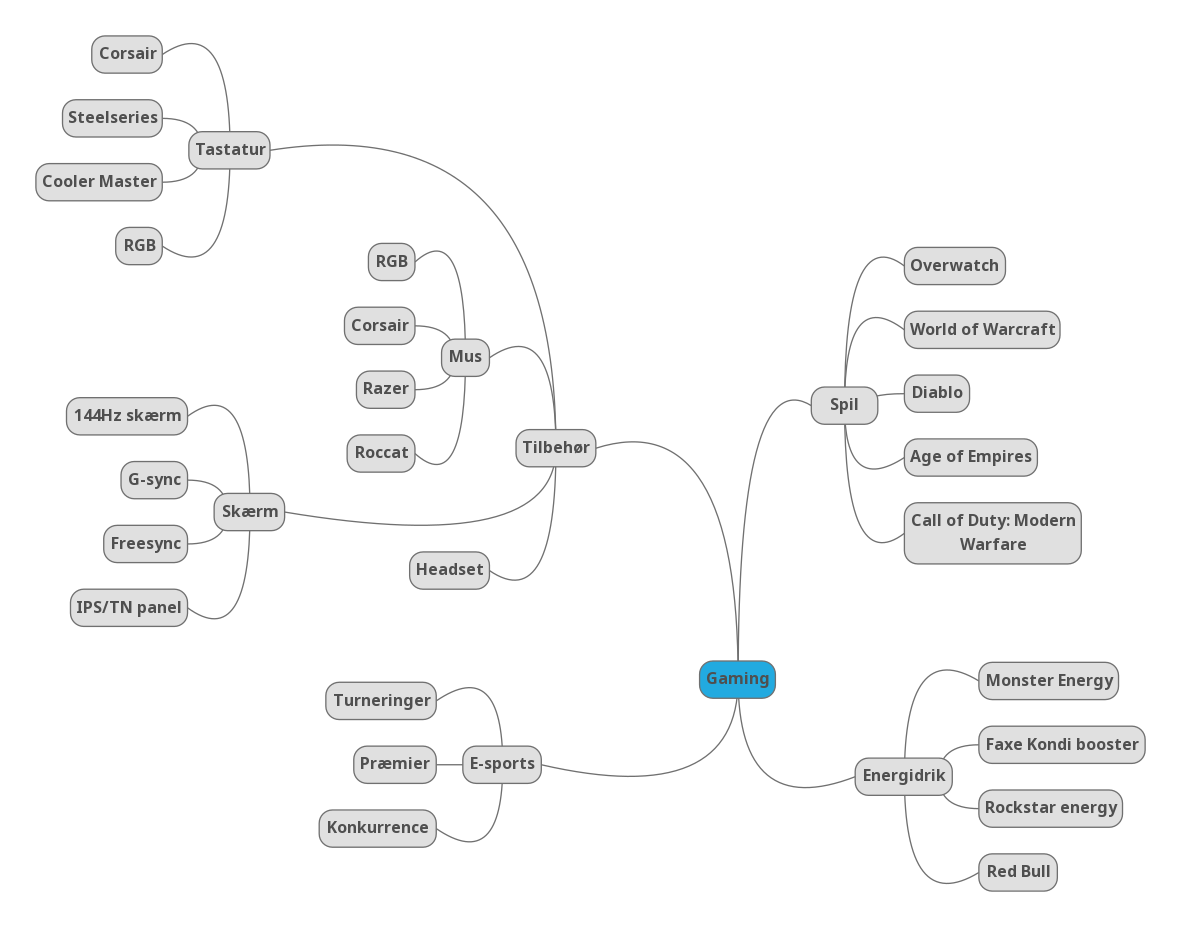
Her er “<a>” tagget starten og ”</a>” enden af elementet. Den røde tekst, er den hjemmeside som man bliver taget til når man klikker på linket, mens den grønne tekst er linkets navn (det man faktisk ser på hjemmesiden som blå highlighted tekst og som man kan klikke på).

**Mindmap:**

En mindmap minder meget om brainstorming. Det er en notationsform som holder fast på sammenhængen mellem de forskellige ord i mindmappet. Når man laver en mindmap så brainstormer man for at kunne finde på nye ord der relaterer sig til emnet, eller allerede nedskrevne ord. En mindmap kan med fordel også bruges med grafik/billeder og behøvest ikke kun at være tekst. Man kan bruge end mindmap til mange forskellige ting såsom:

* Repitition
* Overblik
* Brainstorming
* Sammenhæng
* Struktur
* Associationer

Et eksempel på en mindman kunne se sådan her ud:



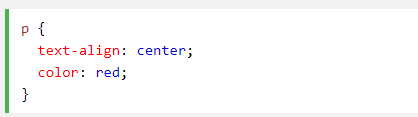
*Lavet af Ahmad Azzam.*

Selectors (Simple, Combinator, Pseudo):

**Simple selectors:**

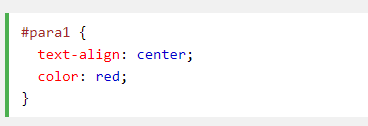
”**Selector**” er den første del i et stykke CSS-kode, og indikerer *hvor* i HTML’en der skal styles på. Herefter kommer ”**property**”, den indikerer *hvad* CSS’en skal style. Til sidst kommer ”**value”**, som egentlig bare er en værdi for property, og fortæller den *hvor meget* eller *i hvilken grad* HTML’en skal styles. Selve selectoren kan være flere forskellige ting. Den kan enten være et tag direkte fra HTML’en, eller et ID (ID’et vælger alle HTML elementer med samme ID, som i CSS’en, og angives med et ”#” før navnet på ID’et), eller en klasse (klassen vælger alle HTML elementer med samme klasse som i CSS’en og angives med et ”.” før klassenavnet på klassen).

**Element selector eksempel:**



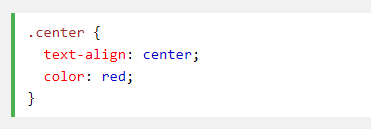
*Her er elementet ”p” valgt i CSS’en, og vil nu derfor style alt der har tagget <p> i HTML’en. Dette stykke CSS vil centrere teksten og give den en rød farve.*

**ID selector eksempel:**



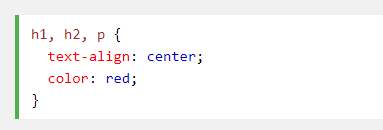
*I dette eksempel vælges alle HTML elementer med ID’et ”para1”. Her vil de blive stylet til a være centreret og have en rød farve.*

**Class selector eksempel:**



*Klasserne med navnet ”center” vil blive valgt i HTML’en, og vil blive centreret og få en rød tekst på hjemmesiden.*

**Grouping selector eksempel:**



*Udover ovenstående så kan man også gøre brug af grouping selectors. Her kan man vælger flere elementer samtidigt, og style dem alle sammen på samme måde, uden at skulle have til at skrive mange linjer kode i CSS.*

**Combinator selectors:**

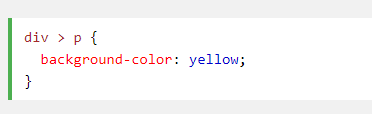
Herudover er der også ”combinator selectors” som vælger mere en bare én ting fra HTML’en, dette kunne være flere forskellige elementer, eller elementer der er ”child” af et andet element (hvilket vil sige at et element er et nedarvet element af et andet.) En ”combinator” er noget som beksriver forholdet mellem to selectors. Man kan derfor nemmere spotte i CSS’en hvilket element der er valgt, og i forhold til hvad.

**Decendant selector eksempel:**



*Vælger alle <p> tags, som ligger* ***inde i*** *<div> tags og giver dem en gul baggrund.*

**Child selector eksempel:**



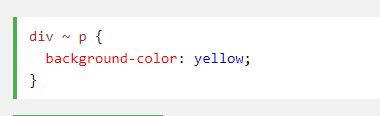
*Denne selector vælger alle <p> tags som er ”child” af <div> tags i HTML’en, og giver dem en gul baggrund. Dvs. at de <p> tags som bliver valgt,* ***skal*** *ligge inde i <div> tags i HTML’en.*

**Adjacent selector eksempel:**



*Denne type selector vælger det første <p> element i HTML’en som kommer* ***lige efter*** *<div> tagget. Alle andre <p> tags som kommer herefter blive ignoreret.*

**General sibling selector eksempel:**



*General sibling selectoren vælger i dette eksempel alle <p> tags som er søskende til <div> tags. Dvs. at her vil alle <p> tags som kommer* ***efter*** *<div> tags blive valgt.*

**Kilder:**

* <https://da.wikipedia.org/wiki/HTML>
* <https://da.wikipedia.org/wiki/CSS>
* <https://www.w3schools.com/>
* <https://www.netkablet.dk/internettet-skrfter-mobile-first-design/>
* <https://cdo-curriculum.s3.amazonaws.com/media/uploads/img_tag.png>
* *Intro til HTML.pdf*