* **Hvad er web**

Internettet er et forbindelsesnetværk, altså et netværk mellem en masse IP-adresser. Internetter fungerer på et globalt niveau imellem millioner af computer og andre elektroniske enheder. Med internetter er det muligt at få adgang til næsten alt information, kommunikere og mange andre ting. Som del af internettet er The World Wide Web, forkortet WWW. World wide web er en samling af hjemmesider lageret på webservere og forbundet til lokale computere via internettet. For at kunne tilgå en webside på www, skal man bruge en web-browser såsom Google Chrome, med flere. WWW blev opfundet i 1989 af en brite ved navn Tim Berners-Lee. Den egentlige ide med hans opfindelse var at gøre det nemmer at dele viden og informationer mellem videnskabsmænd på tværs af forskellige universiteter og institutter verden over. Berners-Lee mente at alle mennesker skulle kunne have gavn af www gratis og uden begrænsninger. Derfor oprettede man World Wide Web Consortium, W3C, som den dag i dag stadig sørger for at holde Webbet opdateret.

Så opsummeret er internettet muligheden for at sende informationer rundt blandt en masse knudepunkter, og skabe et netværk.

Webserver er en computer der oplagrer og udlevere data på internettet. Da webserveren er på internettet, kan den kommunikere med andre enheder der også er forbundet til internettet. Når man åbner en webside, sender man en anmodning til den server som opbevarer websiden, som dernæst svarer webserverne ved at åbne siden du vil se. Dette kaldes http anmodning og svar. Serveren som oplagrer dine data, er hardware mens systemet og dens programmer inden i computeren er software. Klienter er programmer der kobler sig til servere og udveksler information.

* Html

HTML betyder Hyper text markup language, og bruges til at strukturere indholdet i et dokument der senere bliver til vores hjemmeside. Med HTML kan man kun bestemme strukturen og ikke udseendet på ens tekst. HTML fungerer ved at man har forskellige tags som gør forskellige ting i vores dokument. Først starter man med et starttag <tagnavn>, dernæst skriver man sin tekst og slutter derefter med et slut tag </tagnavn>.

Når man starter sit HTML dokument, starter man med et <!DOCTYPE html> tag, som fortæller browseren hvilken type dokument den skal forvente. Dernæst kommer på dokumentet kommer et html tag der skal indeholde hvilket sprog som websiden skal bruge, og hvilken vej teksten skal læses. Tagget kan se således ud, <html lang="da" dir="ltr">.

Efter at have beskrevet hvilken type dokument man arbejder med samt sprog og læseretning, kommer hovedet af dokumentet. I <head> tagget skriver man forskellige oplysninger om dokumentet. I dette eksempel ses først <meta charset="utf-8"> som er en unicode, som betyder at den kan oversætte næsten alle tegn til ens webside. Dernæst kommer en titel, som er den man kan se i ens browser. Herefter kommer et link til et stylesheet, som indeholder designet til websidet. Det næste er kommet tagget <meta name="viewport" content="width=device-width, initial-scale=1.0"/>, som fortæller browseren hvordan den skal kontrollere sidens dimensioner og størrelser.

Et billede, der indeholder tekst

Automatisk genereret beskrivelse

Det næste tag på ens HTML dokument er body-tagget. Den indeholder alle de visuelle elementer af websiden. Det er her strukturen af dokumentet bliver bygget op. Placeringen af elementerne afgør hvor på siden de vil dukke op på websiden.

|  |  |
| --- | --- |
| **Tags** | **Beskrivelse** |
| <H1>…..<H6> | **Overskrifter**- (indeksene af strukturen) (h1-stort og vigtigt), (h6- lille og mindre vigtigt) |
| <HR> | **Horizontal rules**- bruges til at adskille indhold (har ikke en closing tag) |
| <P> | **Paragrafer**- bruges til hvert enkelt afsnit |
| <br> | **Break** - Bruges til at skille afsnit/ læsbart |
| <strong> … </strong> | **Bold**- (understrege) |
| <em>… </em> | **Italic**-(understrege) |
| &nbsp; | **Mellemrum**- til flere mellemrum |
| <img src=”*url*” alt=”*some\_text”*> | **Billede**- Når der skal tilføjes et billede |
| <Q> og <Blockquote> | <**q**> - bruges til korte citater  <**blockquote**> - bruges til at definere citater fra en anden kilde |
| <!--…--> | **Comments** -Identificere forskellige sektioner |
| <a> | **HTML links** -bruges til at lave links i en website |
| Target- attribute:  \_ blank  \_self  \_parent  \_top  framename | Angiver, hvor det linkede dokument skal åbnes   * + Åbner det linkede dokument i et andet vindue   + Åbner det linkede dokument i det samme vindue, når der klikkes   + Åbner det linkede dokument i den overordnede ramme   + Åbner det linkede dokument i hele vinduets krop   + Åbner det linkede dokument i en navngivet ramme |
| <ul> | Unordered list - bruges til at gruppere et sæt af relaterede varer, i en random rækkefølge |
| <ol> | Ordered list / numbered list - bruges til at liste varer i en bestemt/specifik rækkefølge |
| <dl> | Description list - bruges til at vise navn/ termer og deres definitioner eller tidspunkter/ begivenheder |
| <header> | Bruges til at indikerer elementer som er henvist i header sektionen |
| <nav> | Bruges til at indeholder navigations elementer |
| <section> | Bruges til at definere sektionerne i et dokument, som kapitler, header, footer eller andre sektioner i et dokument |
| <footer> | Bruges til at indikerer elementer som er henvist i footer sektionen |
| <main> | Bruges til at definere det vigtigste indhold af et dokument |
| <div> | Bruges til at definere en sektion af et dokument |
| <span> | Bruges til at definere sektion i en sektion i et dokument |
| <link> | Bruges til at definere relationen mellem et dokument og en ekstern ressource (styles) |

Ordforklaring

* Ip - internet protokol
* WWW - world wide web
* W3C - World Wide Web Consortium
* HTML - hyper text markup language
* CSS - cascading style sheets
* HTTP - hypertext transfer protokol

* CSS

CSS står for cascading style sheets, og beskriver hvor HTML elementerne bliver vist visuelt. Et CSS kan kontrollere flere websider på én gang. Så mens HTML bruges til at arrangere indholdet på siden, bruges CSS til fx fonts, farver, baggrund ovs. CSS kan tilføjes til dit dokument på tre forskellige måder: inline, internal og external.

|  |  |
| --- | --- |
| **Inline**  <p style=“color: blue”>My blue paragraph</p> | Tilføjes direkte i et HTML tags, ved brug af style attributter |
| **Internal/ embedded**  <head>  <meta>…</meta>  <title>...</title>  <style>  p {  color: red;  }  a {  color: purple;  }  </style> | Bruges for hele siden og er placeret i ”head” element. Style tags er omsluttet af alle styles på siden. |
| **External**  <!DOCTYPE html>  <html lang=“en”>  <head>  <title>CSS Example</title>  <link rel=“stylesheet” href=“styles/style.css”  type=“text/css”> | Et separat dokument som gemmes som .css. Linkes til i HTML dokumentet. |

I CSS bruger man selectors, modsat til HTML som bruger tags. Selectors er titlerne givet til stylesne ved internal og external stylesheets. For hver selector er der egenskaber inden for krøllede parenteser ({ }). Der gives en værdi til egenskaben efter et kolon (:). Semikoloner (;) bruges til at adskille egenskaberne.

En selector viser hvilket element i HTML filen vi arbejder med. Dernæst skal man haven en property samt en value. Property er hvad du vil ændre fx font størrelse, mens value er den værdi du vil ændre propertyen til en proberty og en value, kaldes sammen for en declaration.

Et billede, der indeholder tekst

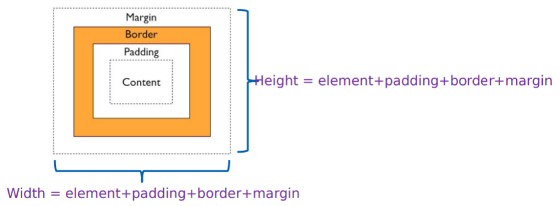
Automatisk genereret beskrivelse

Der er forskellige måder man kan ramme elementer til styling på.

|  |  |  |
| --- | --- | --- |
| ID selector | Vælger elementet med det specifikke id. Bruges i css ved at sætte et # foran det specifikke id. | #navigation {  background-color: blue;  } |
| Class selector | Vælger elementer med samme Class. Bruges i css ved at sætte et punktum foran ens class. | .udvikling{  padding: 20px;  margin-top: 50px ;  } |
| Universal selectors | Vælger alle elementer på en gang. Bruges i css ved at sætte en \*. | \* {  color: blue;  } |
| Element selector | Vælger alle elementer med et specifikt elementnavn. | p {  background-color: yellow;  } |

CSS Box Model

CSS box model bruges når elementer fra HTML skal indkapsles. Til box model bruger man margins padding, borders og elementet der skal indkapsles.



|  |  |
| --- | --- |
| **Begreb** | **Beskrivelse** |
| Content | Selve teksten eller billedet der skal indkapsles |
| Padding | Området rundt om elementet som er klar og ikke har nogen farve |
| Border | Kan være den farvede kant som er rundt om content og padding. |
| Margin | Det klare område udenfor border. |

Der kan indtastes en max-width, som bruges til at sætte en maksimumstørrelse på bredden, som gør at bredden passer også selvom man gør browseren mindre.

Der findes forskellige muligheder for border design.

Et billede, der indeholder bord

Automatisk genereret beskrivelse

Border-style ses overfor og border-width bestemmer hvor tyk border’en skal være. Border-color, bestemmer farven af border’en. Man kan med border-radius bestemme om border’en skal have afrundet hjørner.

TEXT ALIGNMENT, DECORATION, TRANSFORMATION OG INDENTATION

Text alignment, er hvor teksten kan blive placeret i enten venstre eller højre side, centreret eller justified. Ved text align bruges følgende deklaration

* Text-align: left;

Text decoration, er dekoration af teksten. Deklarationerne kan se ud som følgende -

* Text-decoration: none;
* Text-decoration: overline;
* Text-decoration: underline;
* Text-decoration: line-through;

Text-transform bruges til at bestemme om alt tekst skal være med capslock, små bogstaver eller om hvert ord skal starte med stort. Deklarationerne kan se ud som følgende -

* Text-transform: uppercase;
* Text-transform: lowercase;
* Text-trnasform: capitalize;

Tekst indentation bruges til at forskyde første linje af et tekststykke ind.

* Text-indent: value px;

**CSS FONTS**

* Font family, er skrifttypen
* Font style, består af normal, italic og oblique
* Font size, man skriver hvilket skriftstørrelse, teksten skal have (skriver man ikke noget bliver det automatisk størrelse 16 px).
* Font weight, man kan skriver font-weight: lighter; også bliver teksten tyndere.
* Font variant, Fx font-variant: small-caps; hvilket gør at alle bogstaver er med capslock, men hvor de bogstaver der normalt af med lille bogstav, er bliver et mindre bogstav med capslock.

Jo flere breakpoints der laves, jo flere skærmstørrelser. Typisk laver man breakpoints til tre størrelse skærme; mobil, tablet og web. Når man arbejder med responsivt design er det vigtigt at arbejde mobile first, da det kan være svært at tilpasse informationer til begrænset plads. Derud over bruges mobile enheder oftere end computere, så derfor er det vigtigt at ens webside ser vel designet, først og fremmest på mobilversionen.

Ved et breakpoints skrives en min-width skærmstørrelse. Denne min-width betyder at når skærmen er større end den givne værdi skal den opføre sig på en måde, fx:

Den bestemmer at når skærmen er mere end en bestemt værdi, skal den opføre sig på en måde.