

Link til webserver:

<http://ahma0324.web.eadania.dk/Eksamens%20project/Laeringsportfolie/>

Link til GitHub:

<https://github.com/Ahamseru/Laeringsportfolie>

Om portfolien.

For at kunne få overblik over portfolien så bliver man nødt til at inddele emnerne i mindre bider, og finde ud af hvad der hører til hvad. Man bliver simpelthen nødt til at strukturere og kategorisere sit arbejde, og det er lige præcis hvad jeg har gjort. Jeg gik meget enkelt ind på Moodle, og fik kigget under vores "MMD 2019-21 Grundlæggende Faglighed - 1. semester" forløb. Herunder fik jeg kigget under hver enkel dag, og spurgte herefter mig selv følgende:

- Hvem har været underviser for dette emne?
- Er dette emne værd at skrive om i sig selv? (om der er snak om et overordnet emne, eller et underemne)
- Hører det til "HTML, CSS, Layout og Foto", "Design", eller "Kommunikation"?

På den her måde kunne jeg nemt og hurtigt se hvordan tingene skulle inddeles. Det var meget nemt for mig at inddele de forskellige dage, ind over de forskellige lærer, og på den måde fik jeg samtidigt også inddelt indholdet allerede der, i tre overordnede kategorier. Herunder fik jeg yderligere sat mig ind i de enkelte dage, og hvad hver dag indeholdt af emner. Ud fra disse emner, fik jeg dannet mig nogle relationer i hovedet, og kategoriserede dem således, under de underemner der findes på min hjemmeside.

Eksempel: det giver mening at sætte HTML og CSS sammen i et overordnet emne, da de har relationer til hinanden, og bakker hinandens emner op. Dog giver det ikke meget mening for mig at kyle begge emner ind under samme underkategori, da de hver især også skiller sig fra hinanden på forskellige måder. Jo, de er begge to kodesprog, men indholdet, og kodelogikken bag det, er fuldstændig anderledes, og man kan i nogle tilfælde også sige meget mere om det ene, frem for det andet.

Noget helt andet der også gav mening for mig at sætte ind under samme underkategori, var "Internettets historie, Mindmap, Server/Klient, og Design thinking". I mit hoved så ligger alle disse emner under at være meget mere teksttunge, fremfor at være grafik og eksempel baseret. Derfor gav det mere mening for mig at lægge disse emner sammen, under samme underkategori.

Når det kommer til designet af hjemmesiden, så ville jeg ikke noget vildt, som jeg kunne se at tidligere 1. semester studerende har gjort. Jeg ville gøre det meget simpelt, således at selv en bedstemor ville kunne finde ud af at klikke sig ind på de rigtige emner og læse om dem, uden at skulle bruge mere end to klik. Flashy farver, og kasser over det hele er slet ikke min stil, og jeg valgte

i stedet for at fokusere på at gøre knapperne store og tydelige til hvert underemne, og herudover også (personligt så er det her en MUST på alle hjemmesider, ligemeget indehold) implementere dark mode baggrund, således at det er lettere for øjnene at læse stoffet, uden at det skærer i øjnene, når man sidder om aftenen og eksempelvis får lyst til at kigge stoffet igennem. Jeg overvejede også at gøre baggrundsfarven for teksten mørk, og implementere en fuld dark mode på hele siden, men så tænkte jeg at det ville ramme designet lidt for hårdt, og folk er allerede mere vant til at læse sort på hvid tekst.

Jeg startede de første to uger (cirka), med at få et overblik over emnerne. Derudfra begyndte jeg at simpelthen med at skrive en hel vild masse, og satte designet af til side, til senere. Jeg var selv ret forbløffet over hvor meget jeg egentlig fik skrevet, på relativt kort tid, og om så få emner. Da jeg så fik nået et punkt i min tekst hvor jeg tænkte "puha, det er godt nok meget tekst", så begyndte jeg at udligne det hele ved at begynde så småt på designet af hjemmesiden. Til dette havde jeg selvfølgelig også lavet nogle wireframes før jeg overhovedet gik i gang, for at jeg ikke skulle rode for meget rundt i hvordan det nu egentlig skulle se ud.

Så begyndte jeg at arbejde mig ud fra mine wireframes, og fik dannet et skelet på relativt kort tid. Det der egentlig drillede mest, var struktureringen af indholdet under de forskellige dropdowns. Der var altid noget som blev forskudt for meget til venstre eller til højre, og stylingen af det passede heller ikke rigtig med det jeg forestillede mig. Det tog en masse research og ekstra læsning for at få det sat op på den måde som det ser ud nu. Jeg vil også sige at jeg fik da også holdt mange pauser alligevel. Der var flere dage hvor jeg simpelthen ikke lige magtede at få lavet noget, grundet træthed, og dårlig døgnrytme. Så det hjalp meget at komme væk fra det hele i et stykke tid, for så at vende tilbage til det et par dage efter. I sidste ende fik jeg dannet mig et godt billede af hvordan det hele skulle stå og være, og da jeg fik det meste af CSS-stylingen på plads, så kom resten helt automatisk. Jeg blev færdig med hele min hjemmeside + alt ovenstående d. 06/01/2020.

Hjemmeside indhold (kun tekst)

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 finde 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 klicks.

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 anførelsestegn.

https://cdo-curriculum.s3.amazonaws.com/media/uploads/img_tag.png

- **"li"** (herunder også og): Bruges til at lave lister på hjemmesiden. 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 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 fungerer 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 "" slutningen 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:

- **Repetition**
- **Overblik**
- **Brainstorming**

- Sammenhæng
- Struktur
- Associationer

Et eksempel på en mindmap 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 stilet til at 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 vælge 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 end 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 nedrøvet element af et andet.) En "combinator" er noget som beskriver 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.

Kaskaderegler, Specificity, Source order, Farver, Skrifttyper:

Når man snakker kaskaderegler, så snakker man om regler for hvordan CSS-koden skal opføre sig i visse situationer. Disse kaskaderegler skal definere hvordan to eller flere modstridende linjer i koden skal, opføre sig på hjemmesiden, og hvem af dem der faktisk får prioritet.

Specificity:

For at kunne forstå *hvorfor* et stykke CSS-kode, ikke gider at blive anvendt på hjemmesiden, så skal man forstå specificity. Specificity er det regelsæt i CSS-koden som angiver hvilken style bliver anvendt på hvad. Det er også med til at organisere koden, og give den struktur, således at den bliver mere læsbar. Sagt på en anden måde, så er der visse selectors der har forrang (højere "vigtighed") end andre selectors. Man inddeler som regel specificity i fire forskellige selector grupper (rangen på de forskellige selector er også i rækkefølge):

- **Inline Styles**
- **ID'er**
- **Klasser, attributter, pseudo-klasser**
- **Elements og pseudo-elements**

Source order:

Source order omhandler deklareringer i CSS-koden, og hvilken deklarerung man har lavet senest. Hvis der nu skulle stå at man styler headeren ét sted i CSS'en, men samtidig længere nede i koden, stå et nyt stykke kode for samme header, så vil det seneste stykke kode tage prioritet. Det man har skrevet sidst, er også det CSS'en vil gå efter. Specificity har altid forrang over source order. Der er dog én undtagelse og det er brugen af "!important". Den går lige forbi disse slags regler, men er også samtidig med til at ødelægge ideen og funktionaliteten af kaskadereglerne, og dermed også ruinere pointen med kaskaderegler som en helhed.

Box model:

Når der snakkes om "Box model" så snakkes der om hvordan forskellige HTML-elementer repræsenteres i forskellige slags kasser. Der findes to forskellige typer, "inline" og "block". Inline er når boksene placerer sig horisontalt efter hinanden, mens block er når kasserne placeres ovenover hinanden, eller i en stak. Der findes også noget der hedder "inline-block" hvilket er en kombination af de to fornævnte box models. Denne type boks, placere boksene horisontalt efter hinanden (inline), men beholder størrelsen af block boksen (block). For at kunne bruge dette i CSS'en, så kan man skrive eksempelvis:

display: block/inline/inline-block

Box properties:

I modsætning til "Box model", så er "Box properties" de forskellige boksens egenskaber, idet de alle har følgende:

- **Content: (indhold)**
- **Padding: (afstanden mellem ramme og indholdet)**
- **Border: (kan slå synlighed på kant fra eller til)**
- **Margin: (afstanden mellem ramme og andre elementers margen)**

Farver:

Måden man styrer farver på i CSS'en er faktisk ret nemt og ligetil. Man kan enten vælge at skrive navnet på farven direkte (på engelsk), eller man kan vælge at gøre det med mere præcision med disse tre metoder:

- **RGB: rgb(255, 255, 255)**
- **HEX-kode: #000000**
- **HSL: (100%, 100%, 100%)**

RGB:

Den øverste måde at skrive farver på i CSS (RGB), fungerer på samme måde som det lyder. RGB står for "Red, Green, Blue", og hver af de tre værdier kan gå fra 0-255. Jo højere et nummer, jo mere mæthed i farven. Den første værdi er **rød**, den anden **grøn**, og den tredje **blå**.

Man kan bruge en RGB calculator på w3school's hjemmeside

HEX-kode:

HEX-kode er en anden måde at skrive farver på. Her bestemmes værdien gennem et "#" efterfulgt af en kombination af bogstaver og tal. Ingen kan huske alle værdierne i hovedet og hvilken farve de står for, men i eksemplet for oven, bliver farven **sort**.

Der er en HEX-kode "pick a color" som man kan lege med på w3schools hjemmeside.

HSL:

I HSL arbejder man med procenter. Dvs. at man styrer hvor meget Hue (nuance), Saturation (mæthed), og Lightness (lyshed) Alle værdier går fra 0% - 100%.

Der findes en "HSL calculator" på w3school's hjemmeside som man kan bruge

Skrifttyper:

En god hjemmeside er en hjemmeside med relevante skrifttyper. Alle skrifttyper kan være gode, det er bare et spørgsmål om hvilke skrifttyper man skal bruge. Man bliver nødt til at sætte sig ned og tænke over sit site, og hvad det egentlig handler om, og derudfra selv bestemme hvilken skrifttype man gerne vil have til overskriften, underoverskrifter, brødtekst, titler, undertitler osv. Kan man anvende den rigtige skrifttype til den rette hjemmeside, så vil man også få en glad læser, og det vil samtidigt være nemt og roligt for øjnene at læse siden.

Der vælges som regel højst, 2-3 skrifttyper på en hjemmeside, dog er det også muligt at skabe en hjemmeside med blot 1 skrifttype. Det kan dog gå hen og blive lidt ensartet, og kedeligt at læse. Det er ikke alle computere der har installeret de samme slags fonte, så man skal derfor ud og lede efter nye fonte på nettet. Det kan muligvis være gennem "Google Fonts", eller "Adobe Typekit".

Google Fonts har en nem og gratis løsning til folk der gerne vil have et hav af valg, af skrifttyper. Skrifttypen selv bliver hostet af Google, og er meget let at implantere ind på sin hjemmeside. Adobe Typekit Har valgt at inddele de skrifttyper de har i "packs" og "family", alt efter hvad man er ude efter. Herefter kan man vælge at tilføje dem til sin Adobe konto, hvor man derfra kan importere og implementere dem ind i sin kode.

Positionering, Float og Clear:

Positionering i CSS foregår gennem fem forskellige måder (udover grids og andet lignende), og de lyder som følger:

- **Page flow** - Bliver positioneret fra toppen mod bunden.
- **Relative** - Minder om page flow, men er forskudt en smule.
- **Absolut** - Ikke med i flowet, men følger stadig ved at scrolle på siden.
- **Fixed** - Er ikke med i flowet, og følger heller ikke med ved at scrolle
- **Float** - Heller ikke med i flowet, og andre elementer kan flyde udenom.

Float er specielt idet at elementet flyder over et andet. Man kan specificere hvilken side elementet skal flyde hen til. Indholdet i det overordnede element vil flyde udenom, således at det ikke ligger oveni hinanden. Bruges mest til at placere billeder, og andre elementer ved siden af hinanden.

Clear hjælper med afgrænsningen af hvilke sider der må røre andre elementer oveni det overordnede element. Kan have følgende values: "left", "right", "both", og "none"

Versionsstyring:

Versionsstyring eller "version control" på engelsk, omhandler at holde styr på sine filer, og hvilken version af dem der er den seneste. Det kan være alt fra navngivning af mapper og filer, til at gemme noget i skyen gennem en service som Onedrive, Google Drive, eller lignende. Indenfor kodning bliver der typisk brugt hjemmesider som "GitHub", Gitlab, "BitBucket", og mange andre lignende. Disse hjemmesider har også typisk en software version, som er med til at gøre det endnu lettere, og hurtigere at opdaterer sine filer og mapper. Softwaren, er ligesom "onlineversionen", med til at holde styr på hvilken fil der er ændret, slettet, opdateret osv., og er altid i synkronisering med versionen i skyen.

Versionsstyring bliver brugt af alle i hele verden, til alverdens projekter, men bliver brugt især meget af it-folk. Det kan være alt lige fra:

- **Organisationer**
- **Hobby-kodere**
- **Programmører**
- **Firmaer/Iværksættere**

Og bliver brugt til udvikling af bl.a.:

- **IT-systemer**
 - **Apps**
- **Hjemmesider**
- **Frameworks**
- **Hobby-udvikling**
- **Nye løsninger af problemer**

Git:

Git er en protokol, hvilket vil sige at det er en "standard". Git er ikke det samme som GitHub og skal ikke forveksles med det, men er i stedet et versionsstyringssystem. Man kan selv opsætte en Git-server, eller man kan vælge at gøre brug af tjenester som Github, Gitlab, Bitbucket, mm. Git blev oprettet og skrevet af Linus Torvald, og var oprindeligt lavet til Linux kernel, som er et open-source, unix lignende operativsystem.

GitHub:

GitHub er en webplatform, dog også med en pc-klient version der tillader at man kan oprette repositories (en form for bibliotek for filer og mapper). GitHub har en masse features, der hjælper med projektstyring, samarbejde, version kontrol, og meget andet. Det er et socialt medie idet at man kan dele kode med andre, og gøre brug af andres kode, helt frit og kvit. Det blev lavet med den intention at folk videregiver deres kode, eller hjælper hinanden med at arbejde på projekter, hvad end det er private eller offentlige opgaver. GitHub er blandt et af de mest populære kodedelingssite og ville være lidt mærkeligt hvis man ikke kendte til det som programmør. Det er også et site som man kan bruge som en portfolio for den samling af kode, filer, og mapper man nu efterhånden

får lavet, gennem sin karriere. Der er også mulighed for private repositories, til lukkede og egne projekter.

Versionsstyringsbegreber:

Repository (repo) – et bibliotek for alle filer, mapper. Bruges som en stor projektmappe.

Commit (at comitte) – at gemme sine seneste ændringer i projektmappen (repository), med mulighed for at tjekke hvilke ændringer der er blevet lavet, og dermed også mulighed for at gå et skridt tilbage, skulle der være noget galt med den seneste version.

Pull – at nedhente de seneste ændringer fra repository, og merger dem med sine egne.

Pull request – en anmodning om at kunne pulle fra skyen.

Branch – en sidegren som arbejdes på, før man merger ind i *master branch* (hovedgrenen). Det er en kopi af den seneste version i skyen, og man berører ikke hovedgrenen imens man arbejder på denne forgrening

Master branch – hovedgrenen i versionstyringen, og er den endelige version af koden. Man skal kun merge, hvis man er sikker på at det ikke vil få konflikter, og man har fået testet alt hvad man vil teste af.

Lavet af Ahmad Azzam.

Layout, Grids, og Informationsarkitektur:

At kunne designe en hjemmeside er ikke bare noget man lige gør over en dag eller to, det handler om meget mere end det. Først og fremmest er der de forskellige skærmstørrelser på de forskellige enheder, og hver enhed har et sæt antal pixels i højden og i bredden. Som webdesigner så ønsker man at brugeren får så optimal en oplevelse som muligt uanset hvor henne i verden de er, og ligegyldigt hvilken enhed de er på. Samtidigt så ønsker man også som designer at skulle kode så lidt som muligt, men stadigvæk med det bedste resultat som kan opnås. Der skal kun være én version

af hjemmesiden, men den skal samtidigt også kunne tilpasse sig de forskellige skærmstørrelser "on the go". Herudover så ønsker man også som designer at sit site har en god SEO. For at kunne opnå denne tilfredsstillelse så skal man gøre brug af "@mediaqueries". @mediaqueries bruges som et breakpoint, således at når man når til en vis skærmstørrelse (i bredden) så skifter siden layout, således at det bedre kan tilpasse sig hjemmesiden på den givne enhed.

Layout:

Layoutet defineres som regel i fire overordnede former:

- **Statisk (fixed width)**
- **Flydende ("liquid layout", flydende og elastisk)**
- **Responsiv (tilpasser sig skærmstørrelsen "on the go")**
- **Adaptiv (blanding mellem statisk og responsiv)**

Et statisk webdesign omhandler brugen af en fast bredde på hjemmesiden. Dvs. at denne type design ikke vil ændre størrelse ved forskellige skærmstørrelser. Alt ekstra plads på skærmen, vil blive til tomt plads, med noget mellemrum. På mobile enheder skaleres dette ved at zoome ind. Det var engang det dominerende design, og designeren havde også samtidigt mere kontrol over resultatet. Dog fungerer dette meget dårligt på alle andre skærmstørrelser end den tiltænkte.

Flydende webdesign er meget mere tilpassende idet at det selv tilpasser sig til bredden, og man angiver samtidigt bredder hyppigt i procenter. Her kan elementer blive meget små og smalle og der er samtidigt grænser for hvor småt disse elementer kan gå. Et fluid webdesign bliver normalt valgt når indholdet og funktionaliteten på sitet, vægtes højere end den grafiske del. Det flydende webdesign udnytter pladsen godt, og giver også brugeren en bedre oplevelse.

Det responsive layout består meget typisk af et fluid layout, herunder samtidigt under en række regler som ændrer på skærmbredden afhængigt af hvilken enhed man som bruger er på. Her bliver der gjort brug af "media queries" (@media) i CSS, og det er altid samme HTML og CSS-filer uanset hvilken enhed der bliver brugt. Det er den foretrukne måde i dag at lave hjemmesider på, og er også blevet en standard. Det er ikke særligt svært eller vanskeligt, men kræver dog en masse planlægning og omtanke. Herunder skal man også gå ind med mindsettet "**mobile first**" da det ellers vil få store konsekvenser senere hen skulle man have sprunget over, eller glemt dette trin. Det er altid nemmest at designe til noget der har begrænset plads **FØRST**, før man overhovedet går i gang med en skærmstørrelse hvor der kan være meget mere plads.

Det adaptive layout er noget der ligger imellem en statisk og en responsiv tilgang. Består selv af en række statiske layouts og styres af @mediaqueries. Dog ville der være for lidt eller for meget plads på skærme der ikke rammer lige præcis de breakpoints der er i koden. Samtidigt er det noget man vælger ret sjældent hvis der ellers er mulighed for at gøre det responsivt. Bruges dog ofte til at konvertere et statisk layout om til et responsivt layout, uden at skulle skrive det hele om igen. Her er tanken, i modsætning til det responsive, "**Mobile last**".

Grids:

Grids bruges til at opsætte hjemmesiden således at elementerne er fordelt på den måde som man egentlig vil have. Grids indeholder både rækker og kolonner som man kan udfylde med plads, og man kan samtidig fortælle CSS'en hvor meget plads et element skal fylde, gennem disse rækker og kolonner. Grids bygger videre på flexbox (et-dimensionelt), som ellers ligner grids rigtig meget, men hvor, flexbox i stedet er indhold baseret, så er grids container baseret. Grunden til at man bruger grids i dag, er fordi at der aldrig rigtig har været en god måde at opsætte sin side på før. Funktioner som float og clear, har været alt for vanskelige at gøre med, og har stort set været "trial and error", hele vejen igennem. Før i tiden skulle man selv specificere stort set alt, der var meget mere fokus på elementerne selv, i stedet for positioneringen af dem. I dag kan grid systemet, selv udregne bredden af elementerne i hver celle.

Vi skulle som øvelse på klassen prøve at sortere, otte rækker således at de så ud som slutresultatet vi fik fra Niels. Her er de otte rækker usorteret:

Og her er rækkerne igen, men sorteret og farvet som de skulle være (af mig):

Måden man definerer et grid på er at man udvælger et element som eksempelvis "< section>" og bruger det som en "container". Heri skriver man "display: grid;". Nu er dette element klar til at vise

sit grid, når vi får det udfyldt med elementer. Indtil videre hjemmesiden og dets layout ikke være ændret.

I dette eksempel udvælges elementet med klassen "grid-container" til at være vores grid skelet

Herfra skal vi til at angive kolonnerne og definere dem således at hjemmesiden ved, hvor mange og hvor bredde vores kolonner skal være. Dette gøres eksempelvis på denne måde:

Hvor antallet af input efter kolon definerer hvor mange kolonner vi skal have (en kolonne med 1fr og henholdsvis 3fr i størrelse, alt i alt to kolonner). Her betyder "fr", "fractions", eller "dele", og definerer hvor meget plads en kolonne skal have. Ønsker vi flere kolonner en disse to, så tilføjer vi en ekstra "fr" for at introducerer en kolonne mere. Skulle antallet af elementer ikke gå op, vil griddet selv introducerer en ny række, hvor de resterende elementer kan stå i stedet. Der kan også på samme måde angives hvor mange rækker man vil have, og hvor store de skal være. Et eksempel på dette kunne se sådan ud:

Hvor griddet vil ændre sig til at se sådan her ud i stedet for:

Herudover er set også muligt at definerer hvor meget en kolonne skal spænde over. Man kan altså definere en "grid-column start:" og en "grid-column-end:", hvor numrene indtastet efter hvert eksempel, vil være den linje i kolonnen man starter fra, og slutter på henholdsvis.

Og vil få griddet til at se sådan her ud i stedet:

Dette kan selvfølgelig også gøre med rækkerne hvor man i stedet skriver "grid-row-start:" og "grid-row-end:".

Alt i alt så er griddet bare meget nemmere når det kommer til struktur, og der er herudover også meget nemt at vedligeholde. Det anvender også navngivning af grid-elementer, og er derfor også nemt at definere og opsætte. Et eksempel på navngivningen af de forskellige elementer kunne se sådan her ud:

Hvor "grid-template-areas" definerer hvor meget et element, som har det givne bogstav skal fylde i griddet (både i rækker og kolonner). Det er også muligt at lade et felt i griddet stå tomt ved at indtaste et "." i stedet for et bogstav.

Ud fra dette eksempel:

Så vil griddet komme til at se sådan her ud i stedet for:

Informationsarkitektur:

Informationsarkitektur eller "IA", omhandler organisering og strukturering af information om et produkt eller service. Meningen med informationsarkitektur er at hjælpe brugeren med at finde sit mål, hurtigt og effektivt på en hjemmeside. Det er meget vigtigt at forstå at IA ikke er grafisk design, og skal faktisk overvejes før man overhovedet begynder med at designe et website.

"Good information architecture enables people to find and do what they came for. Great information architecture takes find out of the equation: The site behaves as the visitor expects. Poor or missing information architecture neuters content, design, and programming and devalues the site for its owners as well as the audience it was created to serve. It's like a film with no director. The actors may be good, the sets may be lovely, but audiences will leave soon after the opening credits."

- *Jeffrey Zeldman (stifter af A List Apart)*

LATCH:

En klassisk måde at organisere information på, er gennem "LATCH" metoden, som står for følgende:

- **Location**
- **Alphabet**
- **Time**
- **Category**
- **Hierarchy**

Denne type organisering er oprindeligt inddelt af Saul Wurman.

At prøve og organisere noget ud fra tilfældighed, er ikke en organiseringsmetode!

Eksempel på Location:

Eksempel på Alphabet:

Eksempel på Time:

Eksempel på Categories

Eksempel på Hierarchy:

Gangstertesten:

Typiske teknikker og metoder man kan bruge til at teste, om man har fået lavet en god IA, er ved at eksempelvis køre "gangstertesten" Gangstertesten går ud på at man skal forestille sig at man ligger i et bagagerum, hvorefter man bliver løsladt af gangsterne, for så at kunne finde vejen, hjem eller til et andet sted man gerne vil hen. På denne måde kan man teste kvaliteten af IA og navigeringen på siden. I praksis så kunne man vælge en hvilken som helst tilfældig side, med et tilfældigt produkt, og derfra orientere sig på følgende:

- Hvor er jeg henne på siden?
- Hvordan kommer jeg til forsiden?
 - Er der hovedsektioner?
- Søgning (hvor og hvad kan jeg søge på?)
- Eventuelt en "Du er her" indikator?

Dette er også kendt som *kortlægning*. Herudover så ville det være meget godt hvis der var en eller anden form for *mærkning*. Der burde altid være labels og skiltning som kan hjælpe brugeren med at identificere sin position på hjemmesiden. Knapper, ikoner, banners mm. er alt sammen med til at vej vise brugeren og give et hurtigt overblik på hvor man er, og hvor man gerne vil hen.

Når man snakker om IA, så vil man også komme ind omkring flows og flowcharts. Et flow, eller et flowchart er et diagram der visualiserer hvordan en bruger af et system, vil navigere rundt, og udføre forskellige opgaver. Et flowchart er opbygget af tre elementer:

- Trin: Her vil brugeren allerede være i gang med sit flow, og et trin kunne eksempelvis være et informativt trin. Dette kunne være når der udveksles information til brugeren eller når systemet vælger at svare tilbage på en brugerhenvendelse. Udfyldning af en formular kunne være et eksempel under dette element.
 - Veje: er med til at beskrive processens retning og illustreres med pile.
- Beslutninger: Her er det brugerens tur til at tage et valg der vil være med til at afgøre hvilken vej processen skal fortsætte i. Her er beslutninger illustreret typisk med ja/nej spørgsmål.

Herudover så er det også muligt at designe wireframes. Wireframes er skabeloner med minimale detaljer af hver side, på hele hjemmesiden. Den viser meget enkelt, hvad der er på hver side, og

hvad der er muligt at interagere med på siden. Et eksempel på et wireframe for en forside kunne se sådan her ud:

Internettets historie, Server/Klient og Design thinking:

Internettets historie:

Det hele startede tilbage i 1957 det var nemlig her man startede med at tænke på om computeren kunne håndtere flere opgaver end én ad gangen, i stedet for at skulle sende én mand en ad gangen for at sende ét styks information. Den 4. oktober 1957, under den kolde krig, blev der sendt en satellit til rummet af Sovjetunionen, USA blev hurtigt bange for at der skulle ske et "missile gap" (frygten for at Sovjetunionen ville snart have flere missiler end dem selv, hvilket var noget af det allervigtigste på det tidspunkt). For at USA kunne komme på banen igen, så valgte de at lave et stort computernetværk som det kaldte "ARPANET", og det valgte de at gøre gennem en plan de kaldte for "DARPA". Man delte netværket op i fire forskellige dele. Den militære del "RAND Corporation", den kommercielle del "NPL", og den videnskabelige del "CYCLADES". Disse dele ville senere hen blive grundlaget for det internet vi kender i dag.

Manden der stod bag det egentlige internet som vi kender i dag, hedder Tim Berners-Lee og var en del af det vi kalder for "Hypertext Pioneers". Han fik endelig oprettet internettet som vi kender det i dag, i 1990. Det gjorde han på den måde at han brugte tidligere information fra andre "Hypertext Pioneers", både i sin og før hans tid, lagte det alt sammen, og fik endelig bragt Hypertext til internettet. Nu kunne man lige pludselig browse internettet med et unikt men dog et simpelt design, som tilladte at internettet som en helhed at blive adopteret af verden.

Herudover kunne man dele internettets generelle historie gennem tiden i fem forskellige dele:

Tidlige eksperimenter (1993-1995):

Her blev udviklingen af den første browser NCSA Mosaic etableret. Denne første periode handlede meget om hvordan et websted skulle formes og opbygges fra bunden. Den grafiske del blev holdt tilbage af de teknologiske begrænsninger, af HTML og browseren. Man ikke rigtig kontrollere font

type, eller størrelse på det tidspunkt. Heller ingen baggrundsfarve eller centrering af teksten var muligt.

Under konstruktionen (1996-2000):

Under denne periode steg antallet af brugere på webbet vanvittig meget. Det gav lige pludselige kommercielle muligheder gennem internettet, nu hvor det havde mange så mange brugere som aldrig før. Her blev Amazon i 1994 lanceret som et af de allerførste webshops på nettet. Websteders antal blev ved med at stige, og brugerne manglede nu en side hvor de kunne søge efter noget mere specifikt, for at kunne komme hen til den rigtige hjemmeside. Det er under denne periode at Yahoo! kom på banen for brugerne. I 1996 blev Microsofts browser Internet explorer, i version 3.0 lanceret. Dette ville starte browserkrigen mellem Microsoft og Netscape. Måden Microsoft vandt på, var ved at installere browseren automatisk på brugernes nye computere, således at de havde det tilgængeligt med det samme. Netscape blev hurtigt presset ud af markedet, og Microsoft vandt i 2002 med at være på 96% på browsermarkedet.

I 1997 kom HTML-version 3.2 til verden, som gjorde det muligt at styre hjemmesidens baggrundsfarve. Herudover så gjorde Netscape 2.0 på dette tidspunkt, det muligt at integrere grafiske knapper. Dette første til et nyt område indenfor knapdesign.

Semantisk søgning (2000-2003):

Under denne periode mente eksperter at internettet ikke ville kunne klare årtusindskiftet. Hjemmesiderne var langsomme om at loade, og ville gøre det svært for brugeren at fastholde begejstringen mens de var på nettet. Det var her Google kom ind på banen med deres søgemaskine. Den var nemlig i stand til at filtrere mange websteder, og give brugeren præcis de de ledte efter, gennem en algoritme "Pagerank". Denne søgemaskine kunne tage brugeren med et klik, direkte hen til den ønskede hjemmeside. Pagerank, som også betyder "semantik" blev udviklet løbende, og fik hurtigt stor betydning for mange besøgende på Google. Ud af dette udsprang der en industri for søgemaskineoptimering, eller "SEO". Google valgte derfor at lancere en subgenre af søgemaskiner der omhandlede semantisk søgning.

Det sociale web (2003-2010):

Det der for alvor skubbede det sociale web i gang, var begyndelsen af deling af små videoklip der gik viralt, sjove billeder, beskeder osv. Det foregik bl.a. på YouTubes side, som blev skabt i 2005. Formålet med siden var at kunne dele videoer med andre brugere. En video som gik viralt meget hurtigt som hed "Charlie bit my finger", nåede hurtigt på de 800mio visninger, og gjorde YouTube til et godt eksempel på brugergenereret indhold på et socialt medie. I 2007 kom det første mobile netværk, som blev opkaldt for "Twitter". Det var skabt til at dele en sms-tekst på højst 140 karakter. I første omgang undrede man sig over hvordan man kunne overhovedet skrive noget der ville give mening på så få karakter. Det viste sig dog at det hurtigt blev noget meget populært bl.a. kendte og berømte, og trak selvfølgelig almene brugere med sig ind på siden.

Det sociale web (2010 -):

Gennem Apples iPhone og iPad blev der introduceret en ny kategori af web-brugere, de mobile web-brugere. Den revolutionære berøringsskærm gjorde det muligt at styre alt på de mobile enheder gennem et enkelt klik med fingeren, på skærmen. Gennem det mobile netværk, åbnede der nye muligheder for navigation, og tredjepartsprogrammer, gennem Apples "Appstore". Samme år valgte Google at lancere deres eget styresystem som de kaldte for "Android", og fik meget hurtigt overtaget størstedelen af markedet, pga. at det var billigere at købe en mobil/tablet med Android, end en mobil/tablet med iOS. Efter at nu mange brugere har fået fat i en mobil løsning, så skulle man også samtidig finde en løsning til at præsentere hjemmesider på en ny og kompakt måde, på den lille mobilskærm. Designet skulle være konsistent, men stadig dynamisk, således at det kunne tilpasse sig, ligemeget hvilken skærm hjemmesiden måtte blive præsenteret på.

Server og Klient:

Server:

En server er essentielt en computer (server) der deler sit indhold og ressourcer med andre computer (klienter). De mest hyppige slags servere som folk kender til, er dataservere, og filservere. Disse slags servere holder på en mængde data/filer, som klienten så kan tilgå ved at sende en efterspørgsel til serveren, hvorefter klienten får svar på at den efterspurgte data/fil kan tilgås eller ej. Det er muligt for en computer at både være en server og en klient på samme tid. Her er nogle eksempler på forskellige slags servere:

- **Fil-server**
- **Print-server**
- **Web-server**
- **Mail-server**
- **FTP-server**

Klient:

En klient er en computer som bruges til at tilgå en server. Klienten kobler sig til serveren og udveksler data med serveren. Et par eksempler på dette kan se således ud:

- **Webserver kommunikerer med en web-klient (webklienten kunne være en server)**
- **FTP-server kommunikerer med en FTP-klient (det kunne være eksempelvis et program)**
- **Print-server kommunikerer med en print-klient (print-klienten er en del af styresystemet og er normalt usynligt.)**

Design thinking og Useability:

Når man gerne som designer vil finde ud af hvad brugeren ønsker sig, er det en god ide at dykke dybere ned i brugerens tanker og forståelse for produktet. Man skal som designer kunne komme ind i hovedet på brugeren, og finde ud af helt præcist hvordan og hvorledes de gerne vil have at slutproduktet skal se ud. For at kunne opnå dette mål findes der en metode der hedder "Design thinking". Gennem design thinking så vil man som designer være i stand til at forstå stort set alt omkring et emne, ved at have dykket dybere ned i det, gennem de fem følgende metoder indenfor design thinking:

- 1. Empathize**
- 2. Define**
- 3. Ideate**
- 4. Prototype**
- 5. Test**

Empathize:

At "empathize" er at sætte sig ind folks hoved. Det er simpelthen at finde ud af hvad der foregår i hovedet på en potentiel bruger, om et givet emne. Det er at stille spørgsmål som hvordan? Hvorfor? Hvem? Hvornår? Osv. Empathize er vigtigt for os som designere, fordi det tillader os at stille vores egne meninger og tanker omkring et emne til side, og lade andre komme med eksempler og ideer til hvordan DE ser på tingene i stedet for. At observere under empathize fasen er meget vigtigt, man kan få meget information bare at af kigge på hvad andre mennesker gør, indenfor en given situation.

Define:

Define betyder egentlig bare at skulle definere et decideret problem indenfor et givet emne. At kunne definere brugerens behov, og problemer. Man bruger informationen fra empathize fasen til at danne sig en ramme for brugerens ide af et produkt. Herunder gennemgår man sin information som man har fået samlet i empathize fasen, og sortere fra hvad man egentlig kan bruge, og hvad der ikke kan bruges i sammenhængen.

Ideate:

Under ideate fasen, er designeren klar til at generere ideer til produktet som slutbrugererne gerne skulle være ude efter, ud fra vores information fra empathize fasen. Her er det meningen at man skal tænke ud af boksen og kigge efter alternative løsninger til et problem. Herunder er det muligt at bruge brainstorming, eller fået lavet sig en mindmap.

Prototype:

Det er først når man som designer er nået til denne fase, at man kan begynde og tænke i løsninger, og prototyper. Før denne fase skulle hjernen kun være fokuseret på at tænke over tingene, men nu kan som designer begynder at bruge alt information man har fået samlet indtil videre, til at danne sig et eksempel på et slutprodukt, og derudfra lave en prototype. Det er her man skal begynde at identificere hvad der er godt, og hvad der er dårligt som et produkt, for den potentielle bruger.

Test:

Det giver lidt sig selv at man skal afprøve sin prototype, for at kunne se om den skal forbedres, eller om den har nogle mangler. Dette kan med fordel gøres, ved at teste slutproduktet på en potentiel fremtidig bruger. Herudfra vil man få meget god feedback på, hvad der kunne være inkluderet i slutproduktet, og hvilke ting der fungerer godt/dårligt.

Journey mapping:

Journey mapping handler om at kunne kortlægge en brugers interaktion med en given, app, hjemmeside eller lignende. Der er altså snak om brugercentrerede aktiviteter, hvor man følger brugeren fra start til slut. Herunder vil man få en meget præcis og detaljeret oversigt over hvordan en bruger interagerer med hvad, hvornår, og hvorfor. I design thinking har vi altid brugeren i hovedet, og det er altid dem, det handler om når det kommer til at designe et produkt. I stedet for at lave antagelser om hvad brugeren måske ville gøre i visse situationer, så kan vi få meget mere ud af, af at lade brugeren selv prøve at finde frem til det svar som de vil have, gennem interaktion, og gennem user mappings.

Under user mapping, er der to sektioner, eller to inddelinger som man følger parallelt med hinanden. "The process" og "customer journey map".

The process handler om den process som brugeren gerne vil igennem, og beskriver meget nøjagtigt, hvordan en bruger vil til at interagerer med hvad, for at kunne nå til det næste skridt.

Customer journey map handler om de følelser og tanker som en bruger går igennem, under disse forskellige trin, eller "touch points", for at kunne nå frem til sit mål. Det handler om at tænke højt, og få brugerens tanker nedskrevet, og eventuelt en smiley, der viser brugerens frustrationer, glæde, fortvivlelse osv., under hvert af disse touch points. Et eksempel på sådan en model kunne se sådan her ud:

Kvalitative undersøgelser:

Det der bl.a. kendetegner en kvantitativ tilgang til en undersøgelse, er at skulle afdække et område (emne) der kan kvantificeres. Dvs. at kunne undersøge noget som man derefter hurtigt kan få overblik over, og lægge nogle tal på (Hvor mange?). Det skal kunne måles på den ene eller anden måde, hvad end det er gennem skemaer, grafer, eller direkte tal på papirerne.

Fordele:

- Masser af data kan behandles på én gang.
- Det er meget let at få overblik over mange af svarene, og se hvad det mest populære svar er.
- Det er billigt for virksomheder og for private at lave en kvantitativ undersøgelse.
- Det er heller ikke særligt tidskrævende, modsat den kvalitative tilgang.

Ulemper:

- Det er som regel med korte svar, og de er derfor ikke uddybende nok.
- Det er også afhængigt af undersøgerens færdigheder og forståelse.
- Meget svært at se om folk svarer ærligt.
- Herudover er det også besværligt at få nye informationer ud af den spurgte.

Kvalitative undersøgelser:

At lave en kvalitativ undersøgelse er en meget smallere tilgang end den kvantitative måde. Man har fokus på en meget mindre gruppe, man bruger denne tilgang når man gerne vil være sikker, på at det er de rigtige uddybninger, og forklaringer folk kommer med. Spørgsmålene er normalt meget fordybende, og detaljerede, så man også kan få et fordybende og detaljeret svar tilbage. Observation af folk er også en af de værdier en kvalitativ undersøgelse har. Hvad det er? Hvorfor? Hvordan? Osv.

Fordele:

- Detaljerede og fordybende svar.
- Man kan få input som slet ikke ville falde en selv.
- Observation af tone, øjenkontakt og kropssprog som man ellers ikke ville få ved en kvantitativ tilgang.
- Derudover kan man være meget sikker på at folk er meget troværdige i deres, netop fordi det er normalt et face to face scenarie.

Ulemper:

- Skulle interviewet være over telefon, så ville man ikke kunne se hverken kropssprog, øjenkontakt, eller generelle adfærd.
- Tager som regel lang tid at fremstille og fremføre.
- Koster virksomheder meget tid, og mange penge.
- Ikke generaliserbar som den kvalitative undersøgelse.
- Heller ikke særligt repræsentativt.

Useability:

Useability omhandler brugbarheden af digitale produkter. Det handler bl.a. om hvad brug er og hvornår en bruger synes noget er brugbart. At arbejde med useability er at arbejde med hjemmesidens kvalitet og brugbarhed, ved at bruge en række principper og metoder til at opnå dette. Som designer skal man have en vis interesse indenfor useability, det er simpelthen essentielt for at kunne designe noget som overhovedet er brugbart. At ikke have nogen interesse indenfor useability, er det samme som at være ligeglad med sit website, og man kan derfor allerede der bare

droppe at overhovedet få lavet en hjemmeside. En dårlig useability på en hjemmeside kan ikke kun have konsekvenser for en selv som designer eller medarbejder i en organisation, men derimod i længden blive et stort problem for hele organisationen selv, idet at websitet er med til at repræsentere organisationen eller virksomheden. De tre kategorier som tilsammen udgør grundlaget for useability er følgende:

- **Funktionalitet**
- **Effektivitet**
- **Tilfredshed**

Funktionalitet:

Som bruger af et website, så har man et behov som man gerne vil have opfyldt, hvis websitet ikke opfylder disse krav, så vil brugeren bare lede et andet sted, og klikke væk fra siden. Hvis websitets funktionalitet ikke er brugervenligt, så vil brugeren tro at der er fejl på siden, og derigennem komme til at støde på misforståelser som ellers ikke skulle have været der. Det kan hurtigt blive for meget for brugeren, og for frustrerende. Det har dermed direkte betydning for websitets succes.

Effektivitet:

Brugerne vil gerne kunne udføre handlinger på hjemmesiden, nemt og hurtigt. Skulle dette ikke kunne lade sig gøre på siden, så vil brugeren igen finde et andet sted, og klikke væk fra siden. Det skal eksempelvis være nemt at kunne handle ind på en shopping side. Det skal være nemt at tilføje og fjerne vare fra indkøbskurven, det skal være nemt at søge efter vare, at kontakte support osv. Effektiviteten af hjemmesiden har noget at sige når det kommer til konkurrence mellem de forskellige organisationer, og er derfor meget vigtigt at det er i orden.

Tilfredshed:

Brugen af et website er en personlig oplevelse for den enkelte bruger. Brugen af hjemmesiden skal kunne gøre brugeren tilfreds. Tilfreds med designet, tilfreds med effektiviteten, og funktionaliteten. Er brugeren tilfreds med sitet, så føler de sig også godt tilpas på siden, hvilket vil sige at de nok vil komme igen på et andet tidspunkt. Det skal være lækkert at bruge siden, nemt, motiverende, og engagerende. En god brugervenlighed, og en god hjemmeside i det hele taget, giver brugeren en følelse for kvalitet. Ikke kun over hjemmesiden, men også over organisationen som en helhed.

Fotografering:

Et foto eller et fotografi kommer af de græske ord for at skrive/tegne. Et foto kan i dag tages med et kamera (professionelt), eller en mobil (amatør/professionelt.) Hvis man vil have gode billeder, så vil man nok også i de fleste tilfælde vælge et ordentligt kamera over noget som en mobil eller et engangskamera. For at kunne tage et ordentligt og godt billede har man først og fremmest brug for et kamera. Kameraerne i dag variere meget i alt fra lukketid til dybdeskarphe, ISO osv.

Objektiver og brændvidde:

Et objektiv er den linse man kan skrue på, foran på kameraet. Den er med til at bestemme brændvidden på det billede man tager. Brændvidden bestemmer hvor stort et udsnit af billedet man får med. Brændvidden betegnes i millimeter og et traditionelt normalobjektiv i kamera verdenen ligger på de 50mm. Jo større brændvidde jo mere snævert og smallere bliver billedet, men man kan til gengæld også zoome meget ind og fokusere på ét objekt. Jo mindre brændvidde, jo mere får man med på billedet, og man kan tage store og bredde billeder som dækker rigtig meget.

Lukker:

Et kamera har også en "lukker". Denne lukker har også en "lukkertid", som er med til at bestemme hvor skarpt et billede bliver, når man tager det. Jo længere tid lukkeren er åben desto længere tid har lyset til at komme ind på billedet, og desto mere bevægelsessløring og rystelser bliver opfanget i billedet. Hvis lukkertiden er meget hurtig, har lyset mindre tid til at blive opfanget på billedet, og billedet udstår også mere skarpt, og stillestående.

Her er et eksempel på en lang eksponering:

Og her er et eksempel på en kort eksponering:

Blænde:

Blænden i kameraet bestemmer hvor meget lys der bliver opfanget når billedet bliver taget. Den har derfor den rolle om at bestemme hvor kontrast fyldigt et, og hvor skarp en dybde man får på billedet. Dette kaldes også for en dybdeskarphe. Et eksempel på nogle blændetal kunne se sådan her ud, alt efter hvor står åbningen i blænden er:

Her er et eksempel med et højt blændetal (f22):

Og her er et eksempel med et lavt blændetal (f2.8):

ISO:

Kameraets lysfølsomhed bestemmes af en knap på digitalkameraet der hedder "ISO". ISO bestemmer hvor lysfølsom kameraet skal være, og dermed hvor lysfølsom billedet der skal fremkaldes også være. Jo højere en ISO jo mere lysfølsom er kameraet, og desto mere "støj" får man også med på billedet, i form af mange prikker (korn).

Eksempel på ISO:

Man kan hermed sige at man skal indstille sit kamera efter ønskede ISO, blænde, og lukkertid, så man får det bedste resultat som muligt. For at kunne få et ikke bearbejdet billede således at man selv kan indstille de tre forskellige værdier, så skal man sætte kameraet til at tage billeder i "RAW"-format. Skulle man lade det op til kameraet så vil man få et "lossy" JPEG-format, med meget støj i billedet. Dette er fordi man lader ISO, blænde, og lukkertid være helt op til kameraets automatiske system. Herudover kan man dirigere det ønskede resultat yderligere ved at bringe sit eget lys meget. Lyslægning kan ændre opfattelsen af billedet fuldstændig alt efter hvordan man vælger at bruge det. Her er et eksempel på hårdt vs. Blødt lys:

Farver:

Når det kommer til at designe hjemmesider, så er brugen af de rigtige farver, meget vigtigt. Man smider ikke bare to farver sammen og siger "så er den klaret". Det handler meget mere om psykologien bag farverne, og hvordan de er med til at hjælpe med at sende de rigtige beskeder til brugeren. Samtidig så har hver farve forskellig mening, alt efter hvilken person det er der kigger på dem. Det med at vælge farver er derfor ikke bare lige til, og det er heller ikke noget videnskab, det kommer derimod helt op til den enkelte, og hvordan de vælger at tage imod den her farve. Dog har man kommet med en general ide over hvordan farverne bliver opfattet af de fleste mennesker, og samtidig også blevet grundlaget for hvordan man bruger farverne i dag, i de forskellige designs.

Et eksempel kunne være at man prøver at designe en hjemmeside rettet mod kvinder, og det er derfor meget oplagt at vælge en farve som lilla, da den appellerer meget til kvinder, og er samtidig hadet af mænd. Her er et eksempel over de populære brands, og meningen med deres farvevalg:

Farvehjulet:

Når der snakkes om farver, og farvepsykologi, så kan man ikke komme udenom at nævne "farvehjulet". Farvehjulets oprindelse var fra 1600-tallet da Isaac Newton påviste at lyset brydes gennem en prisme, og man fik mange forskellige farver ud på den anden side af prismet. Eksperimenterne han lavede med prismet, fik ham til at konkludere at blå, gul, og rød var de primære farver, da disse farver ikke kunne blandes af andre farver. Farvehjulets opbygning er således blevet etableret på den her måde i dag:

Det er meget vigtigt at nævne, at farvesystemet som vi kender det i dag stadig ikke er absolut, men bygger derimod på mange teorier og opfattelser af forskellige mennesker, og kan betyde noget forskelligt fra kultur til kultur.

- **Primære farver:** rene farver
- **Sekundære farver:** Blandet af to primære farver
- **Tertiære farver:** blandet af én primær og én tertiær farve

Farverum:

Når vi snakker om farver, så er det faktisk tale om to forskellige farvesystemer. Et **additivt farvesystem** (RGB), som står for "Red, Green, Blue". Her er der snak om farver som benyttes på computeren og tv-skærme. Lys bliver tilføjet gennem kombinationen af farver, og kaldes derfor for et "additivt" farvesystem.

Et **subtraktivt farvesystem** derimod indeholder farverne (CMYK) som står for "Cyan", "Magenta", "Yellow". Her bliver lyset trukket ud, jo flere af farverne man kombinerer sammen. Disse farver bliver brugt mest til tryksager.

- **Kulør/HUE:** Er en graduering eller en variation af en farve
- **Mætning/Saturation:** Her snakker man om en kulør som bliver blandet med forskellige mængder af gråtone. Man kan også sige at farven får sig en "valør"
- **Lyshed/Brightness:** En farve bliver blandet med ren sort, eller ren hvid.

Farvesammensætning:

Farverne har også en position på farvehjulet, og alt efter deres placering, og brug af dem, så har man valgt at give de forskellige farvesammensætninger navne.

Monokrom: Én kulør.

Analog: Farven er indenfor et bredere spektrum, af lysheden og mætningen.

Komplementære: Farver der ligger direkte overfor hinanden på farvehjulet.

Split komplementære: Farver der ligger direkte overfor hinanden på farvehjulet, men med et bredere spektrum.

Triader: Tre farver som hver især ligger lige langt væk fra hinanden.

Tetriader (dobbelt komplementær): Fire farver som er komplementære til hinanden.

Webdesignerens farvepalette:

Som webdesigner er det meget vigtigt at man gør brug af hele sin farvepalette når man skal designe hjemmesider. Man skulle gerne på en hjemmeside have brugt RGB farver, farvehjulet til at determinere hvilke farver der ligger hvor, og dermed hvilke farver der hører sammen, valg af en grundfarve, og valg af en accentfarve.

Typografi:

Hvad er typografi egentlig? Typografi er IKKE kun skrifttypen på en tekst men derimod Fonte, kolonner, læsbarhed, layout, billeder og logoer. Typografi er kunsten i at kunne vælge den skriftstil, på den rigtige hjemmeside, til læseren. Det er at kunne determinere hvorfor netop det er **den** skrifttype man har valgt. Alt efter hvilken skrifttype man har valgt, og hvordan hvert bogstavs strøg er blevet brugt, så kan det få kæmpe betydning for forståelsen , i en given kontekst. Her er en simpel illustration af hvad fonte, og andre styles gør ved teksten:

Det er det at man bare kan skifte en lille ting, for så at kunne næsten ændre mening og/eller perspektiv over teksten. Man har derfor en vis mængde kontrol over hvordan læseren skal opfatte teksten. Det er det der gør typografi til et af designerens mest værdifulde værktøj.

Fontanatomi:

En linje med bogstaver har mindst fem linjer de forholder sig til når man skriver:

- **Baseline:** det er den linje som hele teksten sidder på
- **Cap height (eller cap line):** det er det højeste punkt på de store bogstaver
- **Ascender height (eller topline):** det er denne linje der viser hvor et bogstav (majuskler) som h, eller k rører deres højeste punkt. Faktisk så er den her linje lige en tand højere end det højeste punkt på de store bogstaver (Cap height).
- **X-height (eller midline):** denne linje viser hvor højden af små bogstaver (minuskler) går til, og (det ligger også i navnet) måles med bogstavet "x".
- **Descender height (eller beardline):** det er de bogstaver som går under baseline (bogstaver som p eller y), og viser hvor bunden af disse bogstaver ligger.

Typeface:

Når der snakkes om typefaces, så skal vi dykke ned i en skrifttype familie først. Under denne skrifttype findes der "weights". Disse weights er med til at bestemme om skrifttypen er fed, kursiv, normal, eller noget helt andet. Et typeface af en skrifttype familie, er **samlingen** af disse weights.

Leading:

Leading er med til at beskrive mellemrummet mellem to linjer tekst. Det er også kendt som "line-height" i CSS, og kan måles ved at først få fat i to baselines. Leading er stærkt stykke værktøj, idet det kan bestemme hvor læsbar en tekst bliver. Jo mindre en leading jo tættere står linjerne op ad hinanden, og man vil sandsynligvis som læser, næsten begynde at føle sig kvalt. Jo mere whitespace mellem linjerne, jo mere vil man som læser, føle at man har god luft til at læse. Dog kan det sagtens blive for meget af det ene eller det andet. En general regel er at jo flere store ord man har på en linje, jo mere leading skal man bruge. Derudover så skal man også forhøje leading hvis mellemrummet mellem bogstaver er stort. Jo mere mellemrum mellem bogstaverne, jo mere skal man forhøje leading. Eksempel på leading (se næste side):

- **Eksempel 1:** her er leading sat til at være en værdi der er i minus (en værdi der er mindre end skriftstørrelsen). Man kan se bogstaverne røre hinanden fra linje til linje, og dette vil selvfølgelig påvirke læsbarheden.
- **Eksempel 2:** her er leading sat til en ofte anbefalet værdi, som er 120% af størrelsen af skrifttypen. Dvs. hvis din skrifttype er 10pt, så skal leading være 12pt.
- **Eksempel 3:** nu er leading sat til 200% af skriftstørrelsen, og det er selvfølgelig får meget i denne kontekst. Det ødelægger fuldstændig flowet i teksten.

Tracking (eller afstand mellem bogstaver):

Det hedder "letter-spacing" i CSS, og et eksempel på tracking kunne se sådan her ud:

Tracking kan beskrives med ordene *løs* eller *stram*. At have en løs tracking er at have mere mellemrum mellem en gruppe bogstaver, mens en stram tracking er når bogstaverne står meget tættere op ad hinanden. Tracking har mere eller mindre de samme guidelines som leading har, og har også stor betydning for læsbarheden af teksten. Jo længere en linje tekst er, jo løsere tracking skal man bruge, og omvendt, jo kortere en linje, jo strammere en tracking.

Kerning:

Kerning er med til at bestemme afstanden mellem to bogstaver specifikt. Der bliver ofte en forvirring mellem kerning og tracking. Tracking er at se på bogstaverne samlet, og derudfra bestemmer

afstanden mellem dem, mens kerning er at kigge ned på det mikroskopiske niveau, og bestemme afstanden mellem to udvalgte bogstaver. Kerning er derfor muligheden for at justere afstanden mellem to bogstaver, for at gøre det lettere at læse teksten, og så man også får et godt flow. Husk altid, at man som læser ikke lægger mærke til god typografi, fordi at man som læser netop føler at det hele bare kører.

Alignment:

Alt efter hvordan man vælger at aligne sin tekst, så vil det selvfølgelig også få konsekvenser for hvordan en læser opfatter teksten, og om de føler at teksten er let at læse. At kunne bestemme alignment, er helt op til en selv, og kommer an på ens eget tema. Her er eksempler på hvordan man kan aligne teksten:

- **Flush left:** her er teksten forskudt til venstre, og er også den mest naturlige form for alignment, her i vesten.
- **Flush right:** teksten er forskudt mod højre, dette kan være med til at ødelægge flowet for os som læser fra venstre til højre, og skal ikke bruges som en norm, men derimod som et værktøj som kan skabe kontrast mellem de forskellige afsnit og linjer.
- **Justified:** er når starten og slutningen af en linje tekst, er begge aligned til venstre og højre. Selvom justified ser ret god ud i sig selv, så kan det være lidt svært at læse for nogen, da der er mindre visuel feedback på når en linje er slut.
- **Centered:** her er teksten centreret til midten af boksen i stedet for kanterne. Endnu en gang skal man passe på med hvordan man bruger centered, da det også kan være med til at ødelægge flowet af en tekst.

Her er et eksempel på ovenstående termer:

Når man skal skrive en tekst, er det derfor en god ide at overveje følgende ting:

- Hvilken skrifttype skal min titel have?
 - Hvilken skrifttype skal min brødtekst, og undertitler have?
 - Hvor skal jeg bruge bold (fed skrift), italic (kursiv skrift) eller light (normal skrift)?
-

Gestaltlove:

Gestaltlovene handler alle sammen om at kunne se helheder, at hurtigt kunne spotte ligheder, patterns og lignende. Menneskets hjerne er rigtig god til at fylde huller ud, med noget som den tror passer ind med resten, og er også derfor at vi ser ansigter, former og figurer i noget som ellers ikke skulle forestille dette. Hjerne prøver egentlig stort set at simplificere komplekse designs og billeder, ved at fylde alle huller ud således at den får noget der er organiseret, og giver mening, i stedet for noget tilfældigt. Under gestaltlovene findes der seks forskellige love og de er som følger:

- **Similarity (Lighed)**
- **Continuation (Kontinuation)**
- **Closure (lukkethed)**
- **Proximity (nærhed)**
- **Figure/Ground**
- **Symmetry and Order (symmetri og orden)**
- **Common fate**

Similarity (Lighed):

Når der snakkes om similarity, så snakkes der om visuelle elementer som er ens, hvad end det er i størrelse, form, farve eller retning. Alt der går indenfor disse attributter, vil hjerne vælge at gruppere som en gruppe. Farven er den bedste og stærkeste måde at skabe lighed på, eksempelvis så kan det bruges i link, eller i navigation, for at understøtte at alle med lige præcis den farve, er en helhed. Man kan derfor i design nemt fortælle en bruger på en hjemmeside, hvad det er der hører sammen, for så at gøre det nemt for brugeren at danne sig et overblik over disse elementer. Dette vil selvfølgelig også være med til at gøre bruget af sitet lettere og mere intuitivt.

Eksempel på similarity:

Continuation (kontinuation):

Objekter der ser ud til at være kontinuere i en lang række eller bue, kalder vi for "continuation". I design af en hjemmeside kunne de eksempelvis være paragraffer, billeder eller grafer (hvad end det er vertikale eller horisontale). Denne form for brug af design, er med til at kommunikere kvalitet, og relation til brugeren af sitet. Den hjælper også med at guide brugeren rundt, på en sti der er nem at følge. Eksempel på continuation:

Closure (lukkethed):

Closure omhandler dele af figurer eller billeder som kommer sammen, eller som står tæt op af hinanden for at kunne danne en helhed. Det er simpelthen ideen at hjernen skal kunne selv fylde hullerne ud i designet, for så at kunne skave en helhed. Det er meget ofte brugt i logo design, da det er der man kan lade hjernen løbe løbsk, med hvilket som helst design som det skulle være. Et eksempel på closure kunne se sådan her ud:

Proximity (nærhed):

Her er objekter der er nære eller tætte på os opfattet som at være relateret, i en højere grad end objekter der ligger langt væk. Det er faktisk ét af de første principper som rammer opfattelsesevnen i hjernen, og "vinder" over lighed i farver/kontrast. Ved at tilføje whitespace (mellemrum) mellem figurer og billeder, så kan vi som designere separere dem fra hinanden, og mindske relationen. Omvendt kan vi øge relationen ved at samle objekter sammen, og mindske mellemrummet imellem dem. Alt dette kan endda opnås uden brug af kasser, eller grænser rundt om figurerne, billederne osv. Et eksempel på proximity kunne se sådan her ud:

Figure/Ground:

Figure/Ground princippet ligner meget closure idet at den tager fordel i at hjernen selv skal fylde hullerne ud i designs og billeder osv. Her vil hjernen prøve at skelne mellem figurer og former der

ligger både i baggrunden og i forgrunden. Dette kan alt sammen blive meget brugbart idet at designeren kan vælge hvad der skal være fokus på, på et website. Det kunne eksempelvis være en fade effekt, således at baggrunden blev mere utydelig, imens forgrunden kommer frem i fokus. Et eksempel på denne lov kunne se sådan her ud:

Symmetry and Order (symmetri og orden eller prægnans):

Denne lov siger at hjernen vil prøve at opfatte figurer og former på en så simpel måde som muligt. Eksempelvis så vil ringene i det olympiske symbol, opfattes som en kollektion af fem cirkler, i stedet for tilfældige streger i buer. Ellers kunne man tage et logo jeg har brugt på forsiden for "Design" kategorien. Selvom figurerne overlapper, så kan man sagtens se at designet er lavet ud af en trekant, en firkant og en cirkel. Her er endnu et eksempel på symmetry and order:

Common fate:

Common fate var faktisk ikke med i gestalt teorien til at starte med, men man har siden da valgt at inkludere den med i UX-design. Loven siger at folk vil prøve at gruppere ting der ser ud til at vende den samme vej, og som har den samme retning (bevægelse). Dette kan også ses i vores natur som vi allerede kender den, og kunne eksempelvis være en flok fugle som flyver i formation i samme retning, eller en masse fisk som svømmer i samme retning. Her er et eksempel på common fate:

Grundlæggende kommunikation:

Som designere så er kommunikation et af vores vigtigste redskaber, for at kunne kommunikere det rigtige, til de rigtige personer, på den rigtige måde. Men hvad er kommunikation egentlig? Kommunikation er i sig selv en proces hvor vi meddeler til hinanden. Dette kan være i et hvilket som helst rum, og kommunikationen kan foregå på mange forskellige måder, men dog oftest gennem sprog. Ordet kommunikation kommer fra ordet "communicare" fra latin og betyder egentlig at kommunikere eller at gøre fælles. Vi kommunikerer gennem alt lige fra mimik, tøj, udseende, påklædning, kropssprog, stemmeføring mm., og hvad end det er bevidst eller ubevidst. Der er som udgangspunkt tre forskellige måder at kommunikere på, og de lyder som følger:

- **Verbal kommunikation (snak, stemmeføring)**
- **Non-verbal kommunikation - gestik (arme og kropssprog) , mimik (ansigtsudtryk)**
 - **Skriftlig kommunikation**

Kommunikation er både det vi siger og gør, det vi ikke siger og gør, dem vi siger det til (afsender, modtager, budskab), og vores perception af det. Herudover findes der også to forskellige slags meddelelser:

- **Kongruente meddelelser:** Det er meddelelserne via de forskellige verbale og nonverbale kanaler. Ex: "Det er meget hyggeligt det her" (smiler, glad, udadvendt). Her er modtageren tryk i det der siges, men også i det der menes.
- **Inkongruente meddelelser:** Her er meddelelserne gennem kanalerne (verbale og nonverbale), ikke i overensstemmelse med handlingen. I denne situation vil modtageren være usikker på med hvad der menes. Ex: "Du ser rigtig dejlig ud" (sur, kigger mærkeligt, ingen øjenkontakt).

Udover dette findes der også to forskellige former for kommunikation, både synkront og asynkront:

- **Synkron kommunikation:** Her snakkes der om at de kommunikerende skriver/taler sammen på samme tid.
- **Asynkron kommunikation:** Under disse omstændigheder så kan deltagerne sagtens kommunikere skriftligt til hinanden, men der er ofte en lille tidsforskydning, efter hver besked, i kommunikationen.

I kommunikation findes der også fem forskellige grundantagelse og de er som følger:

1. Det er ikke muligt at ikke ikke-kommunikere.

2. En hvilken som helst given kommunikation består af to dele, indholdet og oplysninger om forholdet mellem dem der kommunikerer.
3. Måden vi kommunikerer på og hvordan vi tolker hinandens kommunikation, som også er afhængigt af forholdet mellem de to parter (pauser, tegnsætninger, tone, osv.)
4. Folk kommunikerer gennem den digitale kode (hvad der siges) og den analoge kode (hvordan det bliver sagt, kropssprog, tonefald skrifttype og opsætning osv.)
5. Selve kommunikationsudvekslingen er enten gennem en symmetrisk eller asymmetrisk tilgang, afhængigt om lighed og magt mellem de to parter.

Vi kan altså som mennesker kun kommunikerer gennem det sprog vi har tilgængeligt. Der hvor sproget stopper, stopper muligheden for at skabe en forståelse for en given ting. Derudover så findes der også kommunikationsparadigmer, som er de forskellige måder at se verden på gennem kommunikation. Der er det samfundsvidenskabelig paradigme, hvor der er fokus på afsender og transmissionen, og det humanistiske paradigme, hvor der er fokus på modtageren og interaktionen.

For at gøre det lettere at kunne beskrive en kommunikationshandling, så har Harold D. Laswell kommet med sin berømte replik i 1948 (som man siden har fået lavet om til en brugbar model i dag), og han siger: "Hvem? Siger hvad? Gennem hvilken kanal? Til hvem? Med hvilken effekt?"

Ud over dette er der også i kommunikation blevet gjort brug af IMK-modellen (Interkulturelmarkedskommunikation). Her er teksten blevet sat i centrum, og man kigger også på teksten som andet end bare ord. Under denne model er afsender og modtager fremstillet på lige fod med hinanden, og modellen ser sådan her ud:

Herudover er der også indenfor kommunikation blevet gjort brug af AIDA modellen som lyder på følgende:

- **Attention (opmærksomhed):** Det er her man skal fange modtagerens opmærksomhed. Det skal være blikfangende, det skal være spændende, og måske endda noget som modtageren kan relatere til.

- **Interest (interesse):** Når opmærksomheden er skabt mellem afsender og modtager er det tid til at fordybe modtageren i emnet, og gøre dem mere nysgerrige og interesserede. En side, eller et produkt skal være ekstraordinært, der skal være en form for "spin" på det, for at skille sig ud af de andre lignende produkter.
- **Desire (ønske):** Efter at opmærksomheden og interessen er blevet fanget, skal man prøve at skabe et vis behov hos modtageren, modtageren skal ville det her, og også lysten til at have det. Der er tale om at lokke modtageren, fordi produktet ikke nødvendigvis sælger sig selv.
- **Action (handling):** Det er desværre ikke nok at kun skabe interesse for produktet, og måske endda et behov for det, det er også vigtigt at det faktisk bliver solgt. Her kan man med fordel bruge "Call to action" metoden, ved at eksempelvis sætte en timer på et godt tilbud, på et produkt. Eller måske noget helt andet der får brugeren til at tage handling så hurtigt som muligt.

Præsentationsteknik:

Hvad er præsentationsteknik egentlig, og hvorfor skal vi som multimediedesigner lære noget omkring det? Præsentationsteknik er egentlig bare at kunne præsentere noget på en god, underholdende, og effektiv måde, men hvordan gøres dette egentlig? Formålet med en præsentation i det hele taget, er at kunne informere modtageren og overtale (salg) dem. Herudover er det rigtig godt hvis man også kan uddanne og underholde modtageren samtidigt. Sidst men ikke mindst skal man også kunne skabe en vis fastholdelse og interesse hos modtageren.

Det er meget vigtigt at man som den der præsenterer emnet/produktet at man laver et rigtig godt førstehåndsindtryk. Faktisk så begynder kommunikationen mellem to parter, i det øjeblik de kigger på hinanden, og herfra er de første 20 sekunder altafgørende for hvordan en person synes om en anden. At kunne præsentere noget godt og grundigt, er ikke noget man er født med, det er noget man kan lære sig frem til, og blive god til. Dette kan gøres genne "retorik", hvilket i sig selv er talekunst, som taleren Gorgias bragte til Grækenland, fra kolonierne på Sicilien for ca. 2500 år siden. For at kunne få sig en god retorik (talekunst) så består "opskriften" af fem faser:

1. **Inventio:** Her skal alt stoffet findes, gennemsøges og forberedes. Essentielt, så skal man i gang med en brainstorm. Hvad skal præsentationen egentlig handle om, og hvad er formålet? Hvad er det man vil opnå oplægget, og hvor vil man kigge hen?
2. **Disposito:** her skal man disponere sit oplæg og finde ud af hvilken rækkefølge tingene skal komme i. Det skal hænge logisk sammen, og det skal præciseres, således at man ved hvilke argumenter man vil bygge på under fremlæggelsen.
3. **Elocutio:** under dette segment skal man finde ud af hvordan man vil inkludere Logos/Ethos/Pathos i sin præsentation. Herudover skal man også finde ud af hvordan et emne skal præsenteres, og hvilken sprogstil man gerne vil bruge.
4. **Memoria:** dette trin omhandler egentlig bare at kunne memorere og øve sig manuskript, således at præsentationen foregår let og elegant.
5. **Actio:** dette er delen hvor man faktisk kommer ud for at skulle præsentere, og er selve delen hvor man skal fremføre hvad man har forberedt under de fire tidligere trin.

En anden måde man kan inddele sin præsentation op i, er ved at bruge fiskemodellen eller bare "fisken":

Som man kan se ud fra modellen, så er fisken inddelt i et hoved, en krop, og en hale. Hovedet er indledningen af stoffet, og man "bider" sig fast i sin modtager. Kroppen er alt det der ligger i midten af præsentationen, og indeholder pointerne, og budskaberne. Halen er afslutningen og afrundingen på præsentationen, og man sikrer sig at alle er med på hvad der blev sagt.

Man kan også se på det som et talekort, hvor man inddeler indholdet i tre rubrikker. "exordium" som er alt det indledende til stoffet, "corpus" som er alle argumentationerne omkring emnet, og perratio som er afslutningen og afrundingen på ovenstående.

Herudover kan man yderligere forbedre sin præsentation ved at følge de 9 C'er. De lyder som følger:

- 1. Concise – gør det kort**
- 2. Compelling – gør det interessant**
- 3. Conceptual – gør det udetaljeret**
- 4. Customized – gør det til dit publikum**
- 5. Conversational – gør det åbent til yderligere debat og diskussion**
- 6. Clear – gør det forståeligt**
- 7. Credible – gør det troværdigt**
- 8. Concrete - gør det specifikt**
- 9. Consistent – gør det konsistent**

Når alt det så er sagt, så kan man stadig komme ud for angst, nervøsitet, og stress både før og under en præsentation. Måden man kan hjælpe med at forebygge dette, er ved at først og fremmest at øve sit stof. Herefter kan man så småt skabe sig en tryk over stoffet. Stol på dig selv, du kan det her! Visualisere ved fortællingen af stoffet. Skulle man være i stand til det, ville det være godt hvis man kunne integrere noget humor ind i sit emne, da humor fremføre grin, og smil. At smile er meget vigtigt da det siger noget om dig som den der skal præsentere, og at du er glad ved at være her, og sikker i din egen præsentation. Vejrtrækning er også en essentiel del af en præsentation, tag pauser, og træk vejret, det hjælper!

Kvantitativ og Kvalitativ empiri:

Hvad kendetegner kvantitative undersøgelser?

Det der bl.a. kendetegner en kvantitativ tilgang til en undersøgelse, er at skulle afdække et område (emne) der kan kvantificeres. Dvs. at kunne undersøge noget som man derefter hurtigt kan få overblik over, og lægge nogle tal på (Hvor mange?). Det skal kunne måles på den ene eller anden måde, hvad end det er gennem skemaer, grafer, eller direkte tal på papirerne.

Fordele:

- Masser af data kan behandles på én gang.
- Det er meget let at få overblik over mange af svarene, og se hvad det mest populære svar er.
- Det er billigt for virksomheder og for private at lave en kvantitativ undersøgelse.
- Det er heller ikke særligt tidskrævende, modsat den kvalitative tilgang.

Ulemper:

- Det er som regel med korte svar, og de er derfor ikke uddybende nok.
- Det er også afhængigt af undersøgerens færdigheder og forståelse.
- Meget svært at se om folk svarer ærligt.
- Herudover er det også besværligt at få nye informationer ud af den spurgte.

Hvad kendetegner kvalitative undersøgelser?

At lave en kvalitativ undersøgelse er en meget smallere tilgang end den kvantitative måde. Man har fokus på en meget mindre gruppe, man bruger denne tilgang når man gerne vil være sikker, på at det er de rigtige uddybninger, og forklaringer folk kommer med. Spørgsmålene er normalt meget fordybende, og detaljerede, så man også kan få et fordybende og detaljeret svar tilbage. Observation af folk er også en af de værdier en kvalitativ undersøgelse har. Hvad det er? Hvorfor? Hvordan? Osv.

Fordele:

- Detaljerede og fordybende svar.
- Man kan få input som slet ikke ville falde en selv.

- Observation af tone, øjenkontakt og kropssprog som man ellers ikke ville få ved en kvantitativ tilgang.
- Derudover kan man være meget sikker på at folk er meget troværdige i deres, netop fordi det er normalt et face to face scenarie.

Ulemper:

- Skulle interviewet være over telefon, så ville man ikke kunne se hverken kropssprog, øjenkontakt, eller generelle adfærd.
- Tager som regel lang tid at fremstille og fremføre.
- Koster virksomheder meget tid, og mange penge.
- Ikke generaliserbar som den kvalitative undersøgelse.
- Heller ikke særligt repræsentativt.

Valg af den kvantitative metode:

Man kan med fordel bruge kvantitativ metoden, når man vil gerne have hurtige svar, fra en stor målgruppe. Det kan være alt lige fra spørgeskemaer om skolelivet, til privatlivet. Den kvantitative metode anvendes når det er meget simple spørgsmål (ex. Ja/nej spørgsmål), "radio buttons" (vælge fra flere forskellige muligheder, eller når det er multiple choice spørgsmål, hvor mere end et svar er ok. Man skal være rimelig sikker på at når man laver spørgeskemaet, at det også bliver et samlet og repræsentativt resultat man får. Derfor bruger man denne metode når man vil gerne have bestemte og korte svar fra en stor mængde mennesker.

Valg af den kvalitative metode:

Den kvalitative metode anvendes når man ønsker et mere fordybende og præcist svar på bl.a. hvordan, hvorfor, hvor, eller hvad noget kan være. Denne metode er til dem der gerne vil observere alt lige fra kropssprog, til sprogbrug af den interviewede, imens man indsamler svar på spørgsmålene fra dem. På den her måde kan man virkelig få et godt indtryk af den interviewede, og dermed lære dem mere at kende, gennem deres øjenkontakt, kropssprog, tone osv. Dette er også en metode til at eliminere løgn og utroværdighed fra den interviewede, skulle dette være et "face to face" spørgeskema. Et eksempel på dette kunne være når man bliver fanget af en af dem som går rundt i gågaden, og gerne vil stille folk nogle spørgsmål, hvor de så kan stå overfor hinanden og snakke om emnet. Når vi snakker den kvalitative metode, så er det også meget vigtigt at have sat sig dybere ind i spørgsmålene, således at det ikke er "ja/nej" spørgsmål, men derimod uddybende såsom "hvordan, hvorfor, hvornår, hvorledes" osv.

Hvordan kan man kombinere kvalitative og kvantitative undersøgelser?

Ud fra ovenstående så kunne det tænkes at en måde man kunne kombinere begge metoder på, kunne være at tage selv ud og spørge folk ja/nej spørgsmål. På den måde kan man få lov til at "læse" og observere den man står overfor, og de kan samtidig give enkelt feedback tilbage. Dog stadig med et hvis kropssprog, øjenkontakt osv. På den her måde kan man kombinere noget fra begge verdener til sin undersøgelse.

En helt anden måde at kombinere både den kvalitative og den kvantitative metode på kunne også være at bruge en mix mellem både kvantitative og kvalitative spørgsmål. Det kunne f.eks. gøres ved at afvikle hele undersøgelsen på nettet, men hvor undersøgelsen både indeholder "single choice"/"multiple choice", og "plain text" bokse. Forstået på den måde at man kan svare meget enkelt (krydse af) på ét spørgsmål, og svare med fritext på et andet, således at man kommer i detaljer med sit svar.

Hvad betyder at transskribere?

At transskribere betyder at overføre noget fra ét tegnsystem til et andet, eksempelvis kunne dette være fra tale til skrift, eller at skrive tanker/data ned på et stykke papir, således at man har ordene i "fysisk form", at kunne læse ud fra.

Hvad betyder at meningskondensering?

Meningskondensering kommer fra ét af "Kvales 5 analyseformer" og skal forstås som ¹ *...en praktisk metode, hvor ...lange udsagn sammenfattes til kortere udsagn, hvor hovedbetydningen af det, der er sagt omformuleres i få ord. På den måde kan lange interview reduceres til korte formuleringer, hvor essensen af udsagnene er bevaret ufortolket*". Meningskondensering er også den første fase af fortolkningsprocessen, som også er Kvales første analyseform, for analyse af interviews.

Business Model Canvas (BMC):

¹ Podcasting - en undersøgelse af et nyt medies betydning i hverdagen – side 56

"Business model canvas", eller på dansk, forretningsforståelse/forretningsmodel omhandler forståelsen indenfor virksomheden som en helhed. Der er som udgangspunkt tre forskellige virksomhedstyper:

- **Produktionsvirksomhed:** en virksomhed som fremstiller eller forarbejder forskellige typer af varer.
- **Handelsvirksomheder:** under denne type virksomhed er der ingen fremstilling eller forarbejde, men i stedet bliver der købt og solgt varer.
- **Servicevirksomheder:** disse slags virksomheder sælger ydelser eller varer som ikke er fysiske og som ikke kan lagres. De sælger tjenesteydelser som andre virksomheder ikke kan tilbyde.

Men hvad er egentlig en forretningsmodel? Det er et stykke ledelsesværktøj som kan bruges til at bl.a. lave strategiske analyser, strategiske planlægninger af virksomheden. Derudover er der også udvikling af virksomhedens forretning. Det er at stille følgende spørgsmål vedrørende/omkring virksomheden:

- Hvor er virksomheden?
- Hvem er virksomheden?
- Hvad kan virksomheden gøre?
- Hvordan tænker virksomheden?
- Hvor vil virksomheden hen?
- Hvem er virksomheden skabt for?
- Hvorfor er virksomheden, hvor den er?
- Hvad skal virksomheden gøre nu?
- Hvordan får virksomheden udført det den vil?
- Hvem er det der skal gøre hvad, og hvornår?

- Hvordan kommer virksomheden videre herfra?

Herudover så handler BMC også om mission, vision, og værdier:

- **Mission:** her skal missionen skabes ud fra virksomhedens formål, klar og korrekt formuleres. Der skal inkluderes midlerne til at kunne nå i mål, hvad det er man som virksomhed gerne vil konkurrere på, og skulle gerne også danne grundlag for hvorfor virksomheden overhovedet eksisterer.
- **Vision:** under dette punkt snakker man om hvor virksomheden gerne vil hen, hvilke ønsker der er, og hvilke forventninger der hører til. Man snakker om at kigge langt ude i fremtiden, og at køre efter en fælles retning som virksomheden så skal arbejde efter at kunne nå, og komme igennem.
- **Værdier:** funktionen for værdierne og deres grundlag, er at kunne påvirke holdninger og meninger i virksomheden.

Derudover så har forskere kortlagt digitale forretningsmodeller og fundet 7 arketyper, og de er som følger:

1. **Den digitale købmand:** har en fysisk butik men salget foregår gennem internettet. Ud over dette så er alt med service, branding, varekataloger mm. Alt sammen også på nettet.
2. **Webshoppene:** denne sælger køber hos grossisterne og sælger direkte til forbrugerne gennem internettet. På denne måde undgår sælgeren fysiske butikkers omkostninger, og har også et større varesortiment.
3. **Den digitale producent:** sælgeren vælger at udvide sine produkter med online tjenester, med bl.a. IOT (Internet of things). Dette er med til at skabe digitale lock-in effekter, samt nye funktionaliteter.
4. **Softwareproducenten:** bygger på software med en licens eller direkte betaling. Opdateringer ruller ud gennem internettet, og har også en vis skalerbarhed.
5. **Den digitale platformsbygger:** bygger sin egen platform fra bunden, og lægger den op på internettet.

6. **Appopfindereren:** bygger app efter brugerens behov, og har samtidig lave startup costs.

7. **Wiki-skaberen:** her er det brugeren selv der er både skaberen, og også forbrugeren af slutproduktet. Indtjening sker via donationer, eller indirekte salg.

Business Model Canvas skabelon:

Selve BMC'en er inddelt ind i 9 kasser, og hver kasse er inddelt ind i en bestemt kategori.

Hvem/Salgs kasserne:

1. **Kunder:** dette omhandler hvem der bliver skabt værdi for, hvem de vigtigste kunder er, og hvilke kundesegmenter der kan defineres ud fra dette.

3. **Distribution:** her undersøger man hvilke kanaler kunderne foretrækker når det kommer til kommunikation. Hvordan der kommunikeres til kunder i dag, og hvilke kommunikationsveje der fungerer bedst.

4. **Relationer:** hvilke relationer er der, og skal der være til kunden? Undersøgelse af hvilke relationer som kunden foretrækker, og hvordan disse relationer vedligeholdes.

Hvad/Produkt kassen:

2. **Produkt værdi:** undersøgelse af hvilken værdi der skabes for kunderne. Hvilke af kundernes problemer der bliver løst, og hvilke behov der bliver opfyldt.

Hvordan/Organisation kasserne:

6. Nøgle ressourcer: et kig på hvilke ressourcer der skal til for bl.a. kunderelationerne, værdifaktorerne, vejen til kunderne, og indtægtskilderne.

7. Nøgle aktiviteter: hvilke former for aktiviteter kræver indtægtskilderne, værdifaktorerne, og vejen til kunderne

8. Nøgle partnere: undersøgelse af hvem der er nøgle partnere, hvem der er nøgle leverandør, hvilke nøgleressourcer leverer partner, og hvilke nøgleaktiviteter som partneren udfører.

DKK/Økonomi kasserne:

5. Indtægter: et nærmere kig på hvilke værdifaktorer som kunderne gerne vil betale for, hvor meget de er villige til at betale, hvilke forskellige indtægtskilder er mulige, og hvordan kunderne foretrækker at betale.

9. Omkostninger: hvor henne ligger de væsentlige omkostninger henne i forretningsmodellen? Hvilke ressourcer og aktiviteter er mest omkostningstunge.

Value proposition:

Value proposition er til stor hjælp for designere og virksomheder, idet at det hjælper med at skabe værdi for kunderne. Value proposition beskriver de fordele som kunderne kan forvente af et produkt eller en ydelse. Man slår simpelthen Business Model Canvas og Value proposition sammen for at få det bedste resultat og balance som muligt, mellem kunde og virksomhed. Under value proposition findes der tre undersegmenter som hedder kundeprofil, value map, og match.

Kundeprofil:

Overvejelser over hvad det er kunderne ønsker at få gjort ved et produkt. F.eks. om at en computer skal kunne trække særlige programmer, eller se ud på en bestemt måde, eller have et lavt

strømforbrug osv. Hvilke frustrationer kunderne oplever ved et produkt, det kunne eksempelvis være tid, penge, levetid, holdbarhed osv. Hvilke forventninger kunden har til at produkt, angående design, prestige, service mm.

Value map:

Value map hjælper med at få et overblik over hvilke produkter en virksomhed allerede er i besiddelse af, som kunne hjælpe kunden. Hvordan disse produkter og services kan hjælpe med kundens "pains" (frustrationer), og hvordan disse produkter samtidig kan skabe værdi/gains for kunderne.

Match:

Et "match" opstår når man har nået kundens begejstring over sin value proposition, hvilket sker gennem at udføre det som kunden spørge efter, lindrer deres frustrationer, og giver dem følelse af værdi, over et produkt. Match er meget svært at nå og vedligeholde, da hver kunde har et vist ønske, og et sæt problemer/frustrationer, der også kommer med. Det er derfor svært at gøre alle glade og tilfredse på samme tid, og samtidig også opretholde denne tilfredshed.

Ophavsret og licenseringsmetoder:

(Referér til "Ophavsret og licenseringsmetoder" opgaven, som ligger i et word dokument på siden under Kommunikation > Ophavsret og licenseringsmetoder).

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://en.wikipedia.org/wiki/Version_control
- Intro til HTML.pdf
- Intro til CSS.pdf
- Versionsstyring.pdf
- Responsiv i praksis.pdf
- Internettet og Serveren.pdf

- [wwwHistorieOnlineKommunikation.pdf](#)
- <https://da.wikipedia.org/wiki/Server>
- https://www.w3schools.com/css/css_grid.asp
- <https://www.webdesignerdepot.com/2018/09/grid-vs-flexbox-which-should-you-choose/>
- Grids2.pdf
- Introduktion til fotografering.pdf
- [https://da.wikipedia.org/wiki/Objektiv_\(optik\)](https://da.wikipedia.org/wiki/Objektiv_(optik))
- <https://www.interaction-design.org/literature/topics/design-thinking>
- <https://eadania.mrooms.net/course/view.php?id=1744#section-4>
- <https://www.designkit.org/>
- [Useability.pdf](#)
- <https://designmodo.com/color-psychology-web-design/>
- Farvelaere.pdf
- <https://www.webfx.com/blog/web-design/the-basics-of-typography/>
- Typografi.pdf
- <https://www.toptal.com/designers/ui/gestalt-principles-of-design>
- <http://informatik-gym.dk/glossary/gestaltlove/>
- 05_gestaltlovene.pdf
- <https://www.cuinsight.com/wp-content/uploads/2018/07/bigstock-214122001.jpg>
- Kommunikation MMD 1. semester kommunikationsteori.pptx (PowerPoint).
- <https://www.dagens.dk/aida-modellen>
- [Mette Hald \(2\).pdf](#)
- [Præsentationsteknik.pdf](#)
- Kommunikation i multimediedesign - 2.udgave - Redaktion Anne Mette Busch
- <https://metodenu.systime.dk/index.php?id=131>
- <https://metodenu.systime.dk/index.php?id=132>
- http://denstoredanske.dk/Samfund,_jura_og_politik/Sociologi/Sociologisk_metodologi/kvantitative_metoder
- <https://c-samf.systime.dk/index.php?id=716>
- https://ordnet.dk/ddo/ordbog?query=transskribere&entry_id=12002131
- <http://www.kommunikationsforum.dk/file.asp?id=2598> - Podcasting - en undersøgelse af et nyt medies betydning i hverdagen
- [Interfacedesign fra idé til prototype - Morten Rold \(side 222-223\).pdf](#)
- [Interfacedesign fra idé til prototype - Morten Rold \(side 93-94\).pdf](#)