Intro

Jeg har udviklet min hjemmeside i Hyper Text Markup Language (html) og Cascading Style Sheets (css), hvori jeg har gjort siden responsiv ved brug af media queries der ændrer layoutet. Hjemmesiden er udviklet ud fra en prototype som jeg tidligere har lavet i Adobe Xd. Jeg har brugt versionsstyring med et GitHub Repository, selvom jeg har arbejdet alene og til sidst har jeg udgivet min hjemmeside til mit one.com internet domæne.

De relevante links ses herunder:

Website: <https://www.prb.software>

Prototype: <https://xd.adobe.com/view/840a1c9a-f459-4d98-808e-e5259ba9cff6-d159/>

GitHub Repository: <https://github.com/PxArchive/Hjemmeside-projekt>

Forklaring af eventuelle ændringer siden prototype

Siden den tidligere version af prototypen, havde jeg faktisk arbejdet videre på selve prototypen før dette projekt, da den manglede en række vigtige ting. Derfor beskriver jeg herunder hovedsageligt forskellen på de to prototypeversioner i stedet for forskellen på prototypen og selve websiteimplementeringen, da jeg formåede at implementere langt det meste af den nye prototypes design, i det endelige website. Jeg nævner dog ikke de ting, som jeg ikke har ændret, f.eks. mine valg af typefaces.

Color palette & design

Først og fremmest færdiggjorde jeg den color palette som jeg ville bruge til websitet. Jeg endte med at bruge fire farver; sort, hvid, orange og lysebrun.

Sort giver indtrykket af modernitet og professionalisme

Hvid for at holde det rent og simpelt

Orange, da den associeres med varme, glæde og optimisme

Lysebrun for at give et venlig, ærlig indtryk

De indtryk som farvene og mit website generelt ikke endte med at dække over, var det med at den skulle være sjov/unik. Det var heller ikke det vigtigste indtryk så jeg vurderede at det var i orden, ikke at fokusere på det til den endelige version. Dog holdte jeg fast i designprincippet Consistency, ved at definere mine farver som variabler i mit stylesheet, som jeg så refererede til, alle steder hvor jeg gav elementer farver. Det gør også at hvis jeg i fremtiden vil ændre farverne til mit website, skal jeg blot ændre dem ét sted.

Yderligere justerede jeg også farvernes hue, saturation og brightness, for at sikre tilgængelighed, hvilket jeg opnåede ved at bruge tilgængelighedsværktøjerne på color.adobe.com. Jeg vil påpege, at den orange og lysbrun jeg har valgt, ikke egner sig til normal brødtekst, da deres contrast-ratio ikke er høj nok, hvilket gør dem svære at læse. Derfor har jeg udelukkende brugt dem til overskrifter, former og fed skrift. Yderligere skaber farverne ingen problemer for farveblinde.

Layout og elementer

Den største ændring ift. layout jeg lavede er, at jeg har tilføjet en dekorativ overskrift, som står vertikalt i venstre side af alle mine sites. Personligt syntes jeg den får min website til at føltes mere moderne og spændende. Derudover har jeg også lavet alle billeder afrundet firkanter, i stedet for at det var en blanding af firkanter og cirkler som det var før. Jeg tilføjede også en ’kontakt mig’ knap på forsiden og About siden for at gøre det lettere for brugere at finde mine kontaktoplysninger.

Sidst men ikke mindst har jeg også simplificeret portfolio siden, således at alt om mine projekter ligger på én side, i stedet for at jeg skal tilføje separate sider til hvert nyt projekt. Dette gør det også lettere for brugere at navigere, da de blot skal scrolle ned for at se det hele. Der hvor det er relevant, vil jeg selvfølgelig også tilføje links inde i tekststykkerne om projekterne, hvis det nu omhandler et projekt som jeg har udgivet et sted på internettet, som f.eks. på itch.io.

Indhold og funktionalitet

Da jeg prioriterede at færdiggøre funktionaliteten og stylingen af elementerne til min website, nåede desværre ikke at tilføje indhold til andet end forsiden og den menu som kan tilgås på mobile devices. Men nu hvor det meste af funktionaliteten og stylingen er på plads, forventer jeg at det bliver forholdsvis let at tilføje det manglende indhold. Det er værd at nævne at websitet tilpasser sig størrelsen af browser vinduet og ændre layoutet hvis den bliver smal nok, hvilket er tilfældet på mobile devices.

Forklaring af versionsstyring

Under udviklingen af min hjemmeside har jeg gjort brug af versionsstyring for at løbende kunne gemme min fremgang. Til dette har jeg brugt GitHub Desktop, hvori jeg har lavet og pushet et par commits om dagen. Da jeg arbejder individuelt på dette projekt og kun har arbejdet på det fra en computer, har jeg ikke haft brug for at pulle fra mit repository. Hvis jeg evt. havde lavet nogle ændringer i dokumenterne til mit website fra min bærbar computer, kunne jeg have lavet et commit med den version fra min bærbar computer og pushet den version til mit GitHub repository. Jeg ville så senere kunne hente den version ned fra mit GitHub repository til min stationær computer derhjemme og arbejde videre på den version, derfra.

Forklaring af kode bag mest udfordrende element på hjemmesiden

Det mest udfordrende ved at lave mit website, var – og stadig er – at justere og tilrette elementerne, så de er placeret korrekt ift. prototypen. Jeg endte med at bruge rigtigt lang tid på det dekorative element, som var en ekstra overskrift, der var roteret 90 grader mod uret, således at den stod vertikalt, til venstre for sidens indhold.

Problemet var ikke at vende selve teksten, men at vende den, uden at forskyde andre elementer på siden.

Min løsning var, at først lave en division til alt det midterste indhold på min side, med en klasse jeg kaldte for ”content”. Derunder tilføjede jeg yderligere to divisions med hver deres klasse, ”.vertical” og ”.horizontal”, for at adskille det dekorative vertikale element fra det horisontale indhold, med brødtekst og billeder. Desuden lod det mig også placere de vertikale og horisontale elementer på midten af siden, ved siden af hinanden ved at benytte flex systemet.

Herefter kunne jeg i mit stylesheet definere hvordan teksten i den vertikale division skulle vises vertikalt ved at bruge ”transform: rotateZ(-90deg)”, men det var stadig ikke nok, da selvom overskriften blev vist vertikalt, så blev den stadig betragtet af browseren som var den horisontal og skabte dermed et mellemrum mod højre, der forskudte alt det horisontale indhold.

For at undgå forskydningen, fandt jeg frem til at man skulle definere det vertikale tekstelement som ”writing-mode: vertical-rl”, hvoraf ”rl” betegner hvilken retning teksten skal fortsætte hvis der er mere end en linje, altså om det skal stå over (til venstre for) eller under (til højre for) teksten. Men da alle mine overskrifter kun fyldte en linje uanset, var det ikke vigtigt om jeg valgte ”rl” eller ”lr”, men hvis de fyldte mere end en linje, ville teksten fortsætte til højre for / under den forrige linje med ”vertical-rl”.

Da dette nu gjorde teksten vertikal, skulle jeg ikke længere bruge ”transform: rotateZ()” på ”.vertical” klassen, men i stedet på selve teksten inde i den klasse, hvilket jeg definerede som ”.vertical-text”. Jeg kunne have brugt en selector som ”.vertical p”, men jeg foretrak bare at give selve tekstelementet sin egen klasse, da det var mere overskueligt for mig.

Med writing-mode sat til vertikal, var teksten nu vertikal, dog var den nu sat til at læses fra top til bund, og da jeg ville have det omvendt, skulle selve teksten nu spejlvendes, hvilket jeg gjorde ved at rotere den 180 grader. Med det gjort kunne man nu læse teksten fra bunden mod toppen.

Desuden var det også problematisk at få den tynde linje under teksten til at stå vertikalt, da det umiddelbart ikke er muligt at rotere eller sætte ”writing-mode” til vertical for et <hr> element, så jeg endte med bare at lave en division der var 1px bred og 100% høj (indefor .vertical divisionen) og så sætte dens baggrundsfarve til sort. På nogle browsere som Firefox forsvinder linjen dog nogle gange afhængig af browserens skalering, så jeg vil i fremtiden prøve at finde en bedre løsning.

Det var den forkortede version af, hvordan jeg løste forskydningsproblemet, da jeg også prøvede at løse det på andre måder, som f.eks. at sætte ”transform-origin” til at være det nederste venstre hjørne af tekstelementet og derefter rotere det med ”transform: rotateZ()”, men det ledte til andre layout problemer.

Afsluttende overvejelser

Jeg er overordnet set, meget tilfreds med min hjemmeside og vil arbejde videre på den i fremtiden, så jeg kan få tilføjet det manglende indhold, og gøre at websitets layout er bedre tilpasset til mobile devices.