

Del 1 - Flervalgsoppgave

1. b)
2. b)
3. a)
4. c)
5. d)
6. c)
7. c)
8. c)
9. a)
10. b)
11. a)
12. b)
13. a)
14. a)
15. a) // d)
16. a)
17. c)
18. a)
19. c)
20. c)

Del 2 - Spørsmål til obligatorisk oppgave

a. Refleksjon på obligatorisk oppgave 1:

- Hovedsakelig så er den største endringen fra Oblig 1 til Oblig 2 å gjøre nettsiden responsiv. Dette ble gjort ved å bruke media query (@media screen). Den gamle koden fra oblig 1 ble satt til å kun vises på desktop view, altså på skjermer over 1024px. Ny kode ble skrevet for mobile og tablet view. Denne nye koden ble for det meste delt av begge disse, bortsett fra en egen navigasjon, henholdsvis en dropdown liste kjapt kodet med jQuery og en "lightweight" version av desktop navigasjonen.
- Den mest interessante funksjonen jeg utviklet for nettstedet var på oblig 1 hvor jeg kodet en "lightbox" for galleriet utelukkende ved bruk av HTML og CSS. Dette krever at alle bildene i galleriet ligger to plasser i HTML koden. Lightbox bildene er i utgangspunktet skjult på siden ved bruk av CSS syntaksen "display:none". Lightboxen blir vist ved at man trykker på bildet i galleriet, dette bildet har en id som henviser til "lightbox-bildet". Denne operatoren setter da CSS syntaksen for dette bildet til "display:block" og bildet blir vist. Disse bildene har også stilregler som setter en svart heldekkende, men fortsatt litt gjennomsiktig bakgrunn i tillegg til å ha stilregelen "z-index: 999" som gjør at dette elementet alltid ligger helt på toppen, altså at det ikke har noen synlige elementer foran seg. Valgte å gjøre dette fordi bildene i galleriet i utgangspunktet var små og ville ha muligheten til å vise disse bildene i en større størrelse. Jeg brukte ikke ferdige "thumbnails" for dette og valgte heller å skalere, croppe og sentrere bildene ved bruk av flere CSS stilregler.

Hvis man trykker på den svarte rammen rundt bildet blir lightboksen gjemt igjen ved at den setter CSS syntaksen tilbake til "display:none". Dette krevde dog ganske mye kode, og jeg vil nok si at det å gjøre det i Javascript eller jQuery er en mye mer effektiv måte, men det var dog ikke lov på den første obligatoriske oppgaven.

b. Prototyping / Designprinsipp

- Persona - Først ville jeg bygget en fiktiv persona, kort fortalt er det litt som å skrive en CV/Søknad for en fiktiv person, i dette tilfellet hvor man skal finne og reservere et rom for Bø campus er det vanlig å tenke seg at det er en student det er snakk om, denne studenten er trolig mellom 18 og 30 år. Man kan da også gå utifra at personen er nogenlunde teknologisk anlagt eller i det minste vandt med dagens mobile løsninger. Basert på denne personaen(eller flere personaer) så blir en gruppe med brukerkrav formulert, disse kravene blir deretter tatt med videre til prototyping. Grunnen til at man gjør dette er fordi man ikke skal ta utgangspunkt i hva utvikleren/designeren selv mener at man trenger, siden disse ofte er langt mer teknologisk anlagt en brukeren i tillegg til å ha en mye bredere forståelse for hvordan produktet vil fungere og skal brukes siden det faktisk er de som har laget den.
- Prototyping - En prototype blir laget, f.eks på papir. Denne prototypen blir designet basert på brukerkravene som er satt av personaene. Et av disse kravene kan f.eks være at brukergrensesnittet skal være simpelt og oversiktlig. Man trenger da kanskje ikke 10 forskjellige knapper på hovedsiden med snareveier til de forskjellige sidene på den mobile nettsiden. Det holder kanskje med ett par knapper på siden. En knapp for å reservere en type rom, og en annen knapp for å reservere en annen type rom. Forskjellige typer rom kan f.eks være rom med eller uten tavle/TV.
- Universell webdesign - Når man skal utvikle en nettside er det viktig å ta hensyn til at det skal kunne brukes av alle uavhengig av f.eks funksjonshemming, omgivelser, utdanning osv. Det kan da i dette tilfellet være smart å ta hensyn til spesielt fargeblinde og svaksynte. Dvs. at man holder fargekoding til et minimum, evt. utvikle alternativer for fargekoding som er lett forståelig for fargeblinde. For svaksynte så kan det bety at det er best å holde evt. tekst kort og konkret, slik at denne teksten kan gjøres større.

Del 3 - Kode feilsøking og fullføring

a.

- Mangler <!DOCTYPE html> ???
- <header> . Headeren ligger under <head>, det skal den ikke, den skal ligge under <body>
- Gitt inline stilregler flere plasser, i , og <video>. Blir uoversiktlig og kronglete å redigere. Bedre å ha all stildefineringen i et eksternt stilark(CSS fil), noe som allerede er linket til og blir brukt for dette dokumentet.
- Strukturen på navigasjonen ser litt rar ut. Hvorfor liste?, hvorfor "float:right" på ett element? Men det avhenger av stildefinering i CSS filen.
- Skal <source> stå inne i <video>? <source> skulle ikke vert der i det hele tatt. fjern <source> slik at bare attributtene står inne i <video>. Skal <video> avsluttes med </video>?
- <video controls>, skulle vært noe mer enn bare controls, trenger definisjon av hvilke controls?
(note: ikke brukt video taggen selv, men noe ser rart ut, derfor listet)
- class=".gamelink"... skal ikke være et punktum(.) forran gamelink. Dette punktet er forbeholdt å referere til en class i CSS.
-
 </head>
 <body>

<div id="container">
 <header>
 This is a place holder for the header
 </header>
 <div class="left">
 <div class="leftbox">

 Chapter 1
 Chapter 2
 Chapter 3
 Chapter 4
 Chapter 5

 </div>
 <div class="leftbox">

 Helge Kassin
 Anh Nguyen Doc
 Tor Lønnestad

 </div>

 </div>

 <div class="right">
 <div class="rightbox">
 <p>
 Two of the most important characteristics of good design are discorerability and
 line2

 line3

 line4

 </p>
```

```

 </div>
 <div class="rightbox">
 <p>
 Donald Norman (born December 25, 1935) is the director of The Design Lab at University of
 line2

 line3

 line4

 line 5
 </p>
 </div>

</div>

</body>
</html>
```

CSS: side.css (note: For det meste brukt % fordi det blir stress å "telle pixler", vet at enkelte plasser så vil prosent ikke helt fungere, men i det minste antyder det hvordan jeg tenker.)

```
body {
background-color: darkgrey;
}
```

```
#container {
width: 50%;
margin: auto;
}
```

```
header {
width:100%;
height: 300px;
}
```

```
.left {
display: inline-block;
margin: 0;
background-color: white;
width: 25%;
padding: 5px;
}
```

```
.leftbox {
border: solid lightgrey 3px;
}
```

```
.leftbox ul {
width: 20%;
}
```

```
.right {
display: inline-block;
margin: 0;
```

```
background-color: lightgrey;
width: 75%;
padding: 5px;
}
```

```
.rightbox {
border: solid red 5px;
padding: 5px;
}
```

