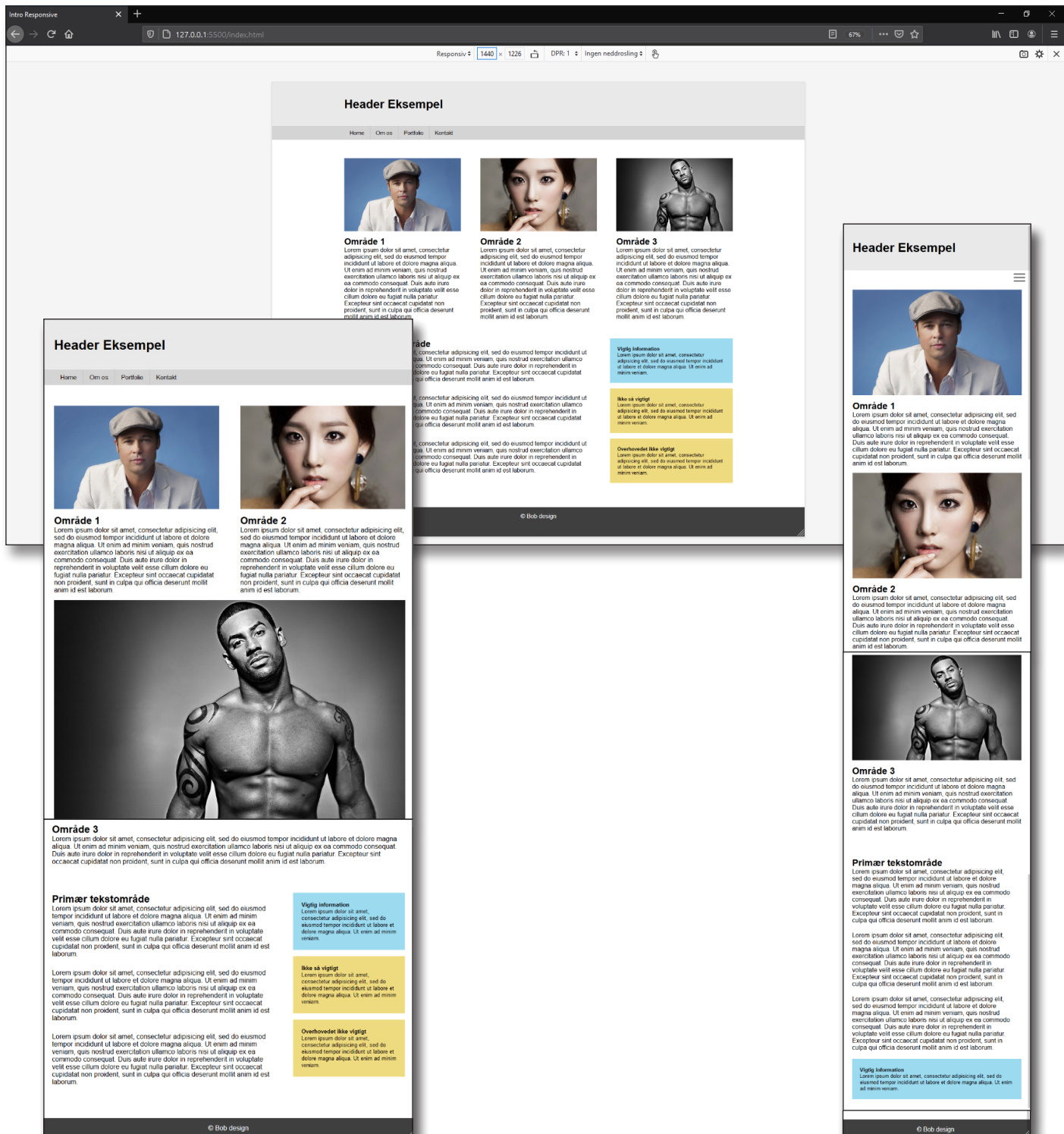


Intro til responsive design



Vi skal starte med at arbejde med Responsive webdesign.

Et mere teknisk udtryk vil være at vi skal arbejde med en introduktion til:
"Media Queries"

Media Queries giver os muligheden for at bruge flere CSS dokumenter på samme side afhængig af skærmstørrelsen.

Lad os komme igang!

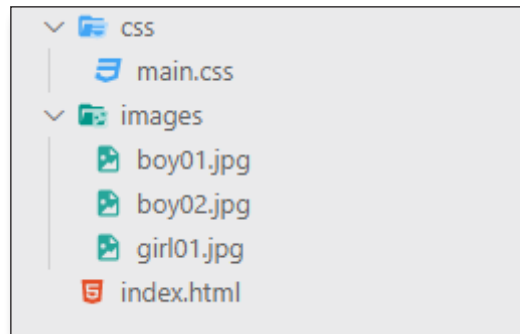
Du har modtaget en mappe som hedder "2020_IntroResponsive-ELEV" Den indeholder:

-----#•#-----#•#-----#•#-----#•#-----#•#-----#•#-----#•#-----#•#-----

index.html

--CSS-- »
main.css

--images-- »
boy01.jpg
boy02.jpg
girl01.jpg

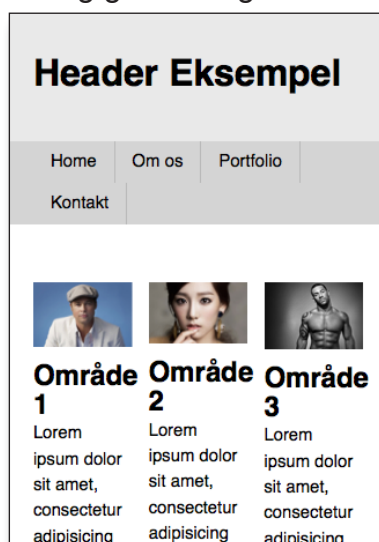


-----#•#-----#•#-----#•#-----#•#-----#•#-----#•#-----#•#-----

1

Start med at åbne "index.html" i din fortrukne html editor. Lav et preview i browseren, og som du kan se så er det en almindelig fungerende hjemmeside.

Hvis du gør skærmen mindre, så tilretter kolonnerne sig godt nok efter skærmen. Det ser bare ikke særlig godt ud og vil absolut ikke fungerer på en tablet eller smartPhone.



2

På billedet herunder ser du den HTML der skaber siden. Det er en simpel side, og som du kan se så er der allerede linket til et CSS dokument.

```
4 <head>
5   <meta charset="UTF-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Intro Responsive</title>
8   <link rel="stylesheet" href="css/main.css">
9
10  <link rel="stylesheet" href="css/responsive.css" media="screen and (max-width: 900px)">
11
12 </head>
```

3

Åben "**main.css**" i din editor og scroll ned til hvor css koden slutter.
Efter kommentaren skal du lave din første "**Media Query**"

En "**Media Query**" giver kommandoen til Browseren at den skal ignoreres indtil bestemte værdier er opfyldt. F.eks at skærmen er større eller mindre end en bestemt bredde.

I denne opgave starter du på stor skærm og arbejder dig ned mod Mobil-skærmen.

```
129
130 /* =====
131 RESPONSIVE CSS
132 =====*/
133
134 @media screen and (max-width: 900px){
135
136
137 }
138
139
```

Når din "Media query" er skrevet med "**max-width**" så er den aktiv så længe skærmen er MAX 900px bred, altså alt under 899px.
Hvis du havde skrevet den med "**min-width**", så ville den først være aktiv når skærmen er større end 900px.

4

Du skriver først en test-værdi for at se om din “media query” virker.

```

132
133 @media screen and (max-width: 900px) {
134     body {
135         color: red;
136     }
137 }
```

Gem! Se resultatet i browseren når du har genindlæst dokumentet.
Gør dit browser vindue mindre end 900px i bredden og se resultatet.

Header Eksempel

Home Om os Portfolio Kontakt



Område 1

Lorem ipsum dolor sit amet,
consectetur adipisicing elit, sed do
eiusmod tempor incididunt ut



Område 2

Lorem ipsum dolor sit amet,
consectetur adipisicing elit, sed do
eiusmod tempor incididunt ut



Område 3

Lorem ipsum dolor sit amet,
consectetur adipisicing elit, sed do
eiusmod tempor incididunt ut

Det er selvfølgelig ikke et resultat du skal bruge til noget, men kun en måde at se at du har forbindelse til din “media query” og at begrænsningerne på skærmstørrelsen virker.

Så din “media query” er ret vigtig. Tænk på det som et spørgsmål til browseren.

5

HVIS skærmen har en **max-width** på **900px** (altså mindre end 900px) så skal CSS dokumentet aktiveres ELLERS skal CSS dokumentet være inaktiv.

Lad os så komme igang med at skrive noget vi kan bruge.

Det første vi skal er at lave top sektionen om til 2 kolonner i stedet for 3, når skærmen er mindre end 900px.

Lad os tage et kig på hvad det er der laver 3 kolonner for os i HTML'en

```
<!-- one-third -->
<article class="one-third mobile-collapse">
  
  <h2>Omr&aring;de 1</h2>
  <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut
    labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco
    laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in
    voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat
    non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.</p>
</article>
```

Som du kan se så er det en almindelig div med en `class="thirds"` med `display: flex;`
Lad os se på hvad den gør i `"main.css"`

```
/*Thirds*/
.thirds{
  display: flex;
  justify-content: space-between;
  padding-bottom: 50px;
}

.one-third{
  flex-basis: 30%;
}
```

Som du kan se så opdeler den bare siden med 3 kolonner på hver **30%**
Der er afstand i bunden. Og bredden af kolonnerne kommer fra `flex-basis: 30%;`
Det er bare en mere "flex-agtig" måde at skrive `"width:"` på.

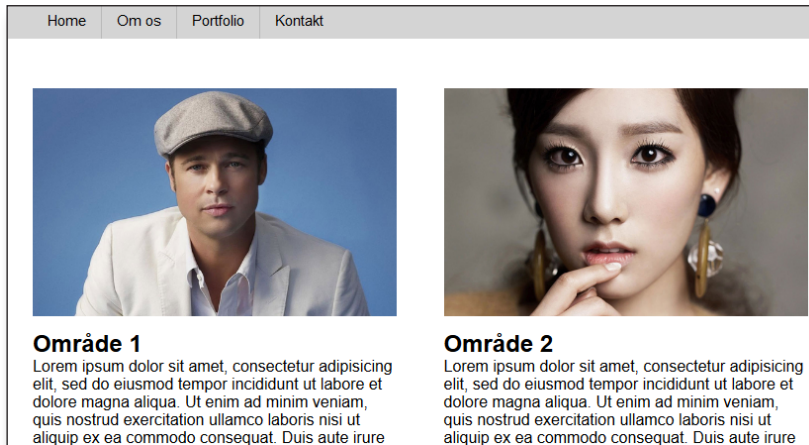
6

(HUSK! SLET BODY-FARVEN, DET VAR KUN EN TEST)

For at få 2 kolonner skal vi altså give `thirds` værdien `"flex-wrap: wrap;` og
lave om i `%` som styrer `one-third`. Derfor skriver du indenfor din nye `"media query"`

```
135 @media screen and (max-width: 900px){
136
137   .thirds {
138     flex-wrap: wrap;
139   }
140   .one-third {
141     flex-basis: 47%;
142     margin-bottom: 1rem;
143   }
144 }
145 }
```

Hvis du ser det i browseren, så kan du se at de 3 kolonner er væk, men der er et lille problem



Den sidste “**article**” med `<article class="one-third one-third-last mobile-collapse">` fylder det samme som de andre to.



Derfor giver vi “**one-third-last**” stilingen “**flex-basis:100%**” så billede og tekst fylder hele bredden

```
.one-third-last{
  flex-basis: 100%;
}
```

7

Nu har vi et “**Tablet**” design færdig. Men vi skal også bruge et design med kun en enkelt kolonne som egner sig bedre til smartPhones.

Efter den første “**media query**” slutter skriver du en ny “**media query**”
Den er til mobil-screen, og skal være for alle skærme mindre end 500px


```
@media screen and (max-width: 500px){  
  
}
```

Den kode vi skriver indenfor denne “**media query**” er først aktiv når reglen er opfyldt. så du skriver:

```
@media screen and (max-width: 500px){  
  
  .thirds {  
    display: block;  
    width: 100%;  
  }  
  
}
```

Refresh din browser, flyt størrelsen ned under **500px** i bredden og se effekten.

```
@media screen and (max-width: 500px){  
  
  .thirds {  
    display: block;  
    width: 100%;  
  }  
  .two-columns {  
    display: block;  
    width: 100%;  
  }  
}
```

Vi gør det samme for næste sektion af siden. **<section class="two-columns">**
Hvis vi også ændre den fra **display:flex;** til **display:block;** så bliver den også til et 1 kolonne design.

Vi har flere muligheder for at gøre siden rigtig responsive i designet.

F.eks. faktaboksene i bunden af designet. Den blå og de to gule.

Den blå er vigtig information, men de gule er ikke så vigtige, så dem kan vi godt fjerne når designet kun er på en kolonne. De har en **class** ved navn **hide-mobile** som ikke har været aktiv endnu, for de må jo gerne være med når der er plads til det, men nu kan vi skrive **display:none;** og fjerne dem ved skærmbredde under **500px**.

```
@media screen and (max-width: 500px){  
  
  .thirds {  
    display: block;  
    width: 100%;  
  }  
  .two-columns {  
    display: block;  
    width: 100%;  
  }  
  .hide-mobile {  
    display: none;  
  }  
}
```

8

Nu skal vi lige kigge lidt på navigationen.

Navigationen følger ikke særlig godt med når skærmen bliver mindre. Den fungerer egentlig fint i tablet-mode, men når vi kommer til mobile-mode, så vil det være naturligt at skifte til en burger-menu.

Indskreven i HTML er der gemt en burgermenu, som i bred format er `display:none`.

```
<div class="section-inner">
  <!--==== BURGER MENU ====-->
  <a href="#" class="toggle-button">
    <span class="bar"></span>
    <span class="bar"></span>
    <span class="bar"></span>
  </a>
  <!--=====END=====-->
```

9

Vi kan kalde på burgermenuen når skærmstørrelsen er under 500px, ved at sætte vores `` til `display:none`; og samtidig sætte burgermenu til `display:flex`;

```
@media screen and (max-width: 500px){

  .thirds{
    display: block;
    width: 100%;
  }
  .two-columns{
    display: block;
    width: 100%;
  }
  .hide-mobile{
    display: none;
  }
  nav ul{
    display: none;
  }
  .toggle-button{
    display: flex;
  }
}
```

Header Eksempel



main.css - - "media query"

```
129  /* =====
130  RESPONSIVE CSS
131  =====*/
132
133  @media screen and (max-width: 900px) {
134    .thirds {
135      flex-wrap: wrap;
136    }
137    .one-third {
138      flex-basis: 47%;
139      margin-bottom: 1rem;
140    }
141
142    .one-third-last {
143      flex-basis: 100%;
144    }
145  }
146
147  @media screen and (max-width: 500px) {
148    .thirds {
149      display: block;
150      width: 100%;
151    }
152    .two-columns {
153      display: block;
154      width: 100%;
155    }
156    .hide-mobile {
157      display: none;
158    }
159
160    nav ul {
161      display: none;
162    }
163    .toggle-button {
164      display: flex;
165    }
166  }
167
```