|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Visualisering  V3.1 | | | |  |
| **Emne:** | SASS/SCSS | **Opgave:** | Funktioner og if/else |
| **Formål:** | * Jeg kan anvende funktioner i SCSS | | |
| * Jeg kan anvende if/else i SCSS | | |
| * Jeg kan anvende løkker i SCSS | | |

**Opgave 5 – SCSS og funktioner / if-else**

I denne opgave skal du lave to funktioner: en, der kan udregne en baggrundsfarve og en, der kan udregne bredden på et element (funktionen genbruger den matematiske udregning fra opgave 4).

Du skal lave en side, som viser 9 firkanter i forskellige blå farver. Disse skal vises i fem kolonner på en desktop skærm og i tre kolonner på en mobil skærm.

Desktop:



Mobil:



1. Opret en ny index.html fil og en tilhørende style.scss
2. Start med at linke til dit stylesheet. Husk, at du skal linke til style.css
3. I HTML filen skal der være en overliggende div i klassen ”kasser”. Under denne skal der ligge 9 divs, som tildeles klasserne: kasse-1, kasse-2, kasse-3 osv… til kasse-9
4. Din SCSS-fil skal indeholde variablen $blue, som indeholder grundfarven hsl(202, 74%, 64%). Denne bruger vi som grundlag for den baggrundsfarve, der skal genereres til kasserne
5. Du skal bruge to mixins:
   1. box - som genererer dine 9 kasser ved at sætte deres højde, bredde og baggrundsfarve
   2. imagegrid - som sætter kasserne op i det grid, du specificerer – afhængigt af din viewport bredde  
        
      Begge disse mixins skal kaldes i en for-løkke, så det kan gøres for alle 9 kasser
6. Du skal bruge to funktioner:
   1. fancyColor - som generer en mørkere og mørkere farve hver gang den bliver kaldt og returnerer farven
   2. calculateWidth - som udregner bredden på din kasse på samme måde, som du gjorde i opgave 4 og herefter returnerer resultatet
7. I din mixin ”box” skal bredden overføres, som du har udregnet i calculateWidth. Højden er afhængig af, om mixin’en er blevet kaldt fra en stor eller en lille skærm (større end 800px bred). Hvis den er blevet kaldt fra en stor skærm, skal højden på dine kasser være bredden gange 1,77. Hvis den er blevet kaldt fra en lille skærm, skal højden på dine kasser være bredden divideret 1,77. På den måde får vi kasser, der er næsten firkantede både på en mobil-skærm og en desktop-skærm.  
     
   Herudover skal du sætte baggrundsfarven på din kasse, som den er blevet udregnet i din fancyColor funktion.
8. Din fancyColor funktion skal gøre baggrundsfarven 5% mørkere hver gang, den bliver kaldt
9. Dit imagegrid skal være 3 kolonner med 3% margen på mobil og 5 kolonner med 2% margen på desktop

**EKSTRAOPGAVE**

Hvis du bliver færdig med ovenstående opgave, kan du udvide opgaven med følgende funktionalitet:

1. Opret 7 ekstra div'er i dit HTML-dokument og tildel dem klasserne op til kasse-16
2. Du skal nu lave en ny variabel - $total - hvor man kan angive, hvor mange kasser (op til 16), som man gerne vil have vist. Så hvis man skriver $total: 9, så bliver der kun vist 9 kasser og hvis man skriver $total: 12, så bliver der vist 12 kasser
3. Sørg for at din for-løkke og din kode kun viser det antal kasser, som man har bedt om