

Slideserie: CSS3 Grid

DS3103 Webutvikling

Høst 2022

Rolando Gonzalez

Innhold slideserie






- Hva er CSS3 Grid?
- Sette opp grid
- Sette elementer i grid og angi størrelse
- Diverse teknikker i sammenheng med CSS3 Grid

Hva er CSS3 Grid?

- CSS3 Grid er et todimensjonalt gittersystem for å plassere HTML-elementer på en nettside.
- Med todimensjonalt menes at man kan angi rader og kolonner.

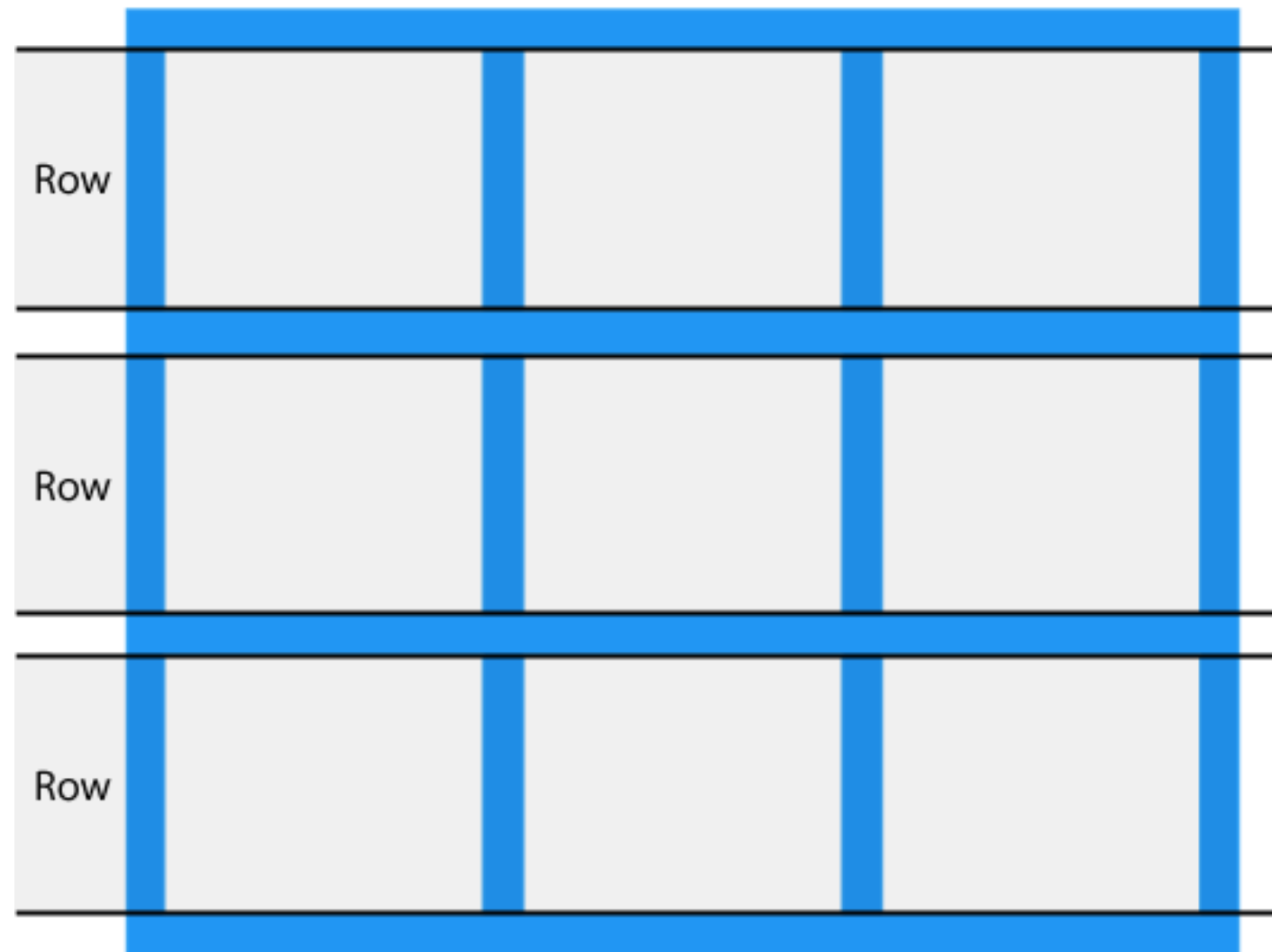
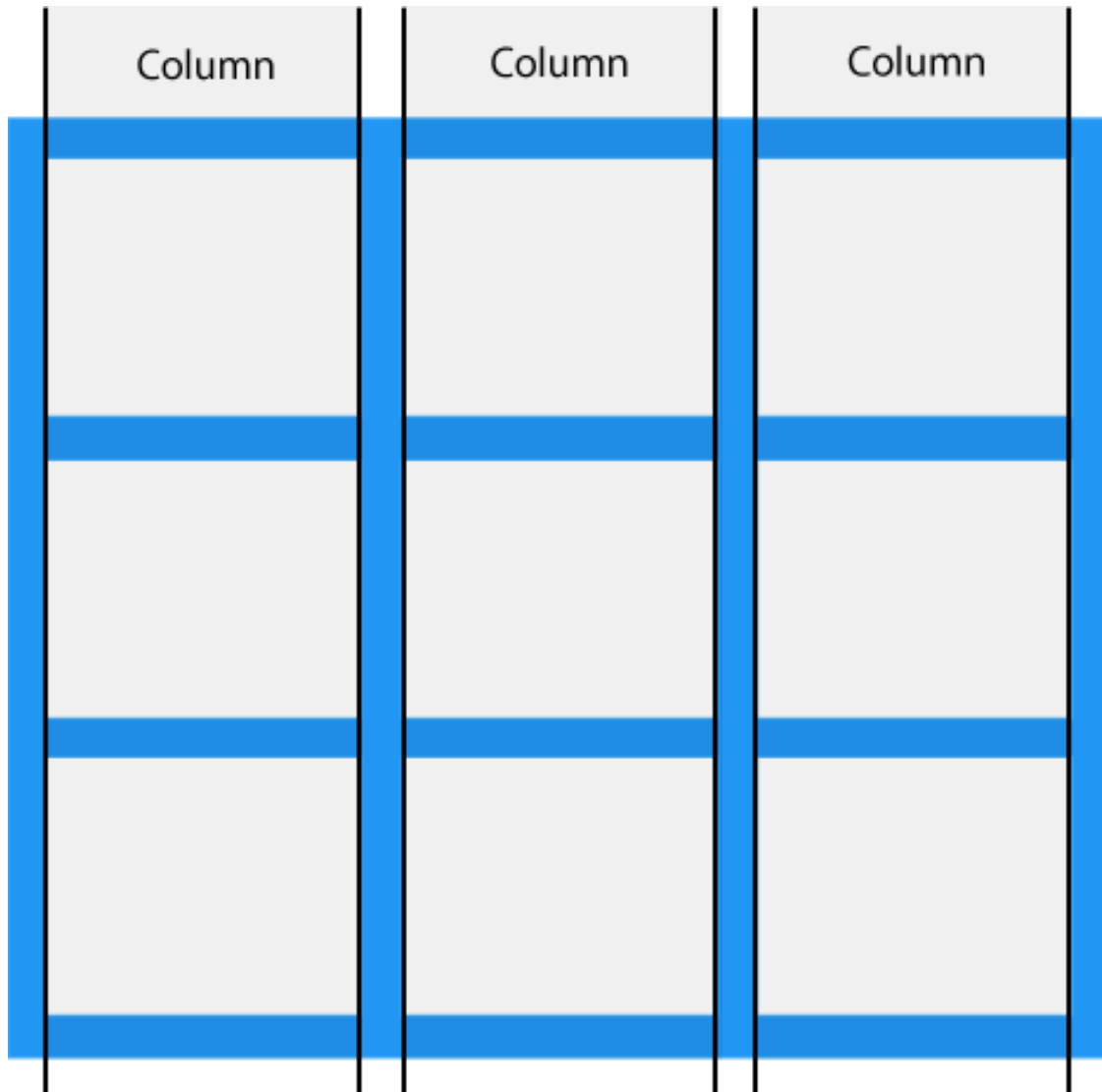
Browser Support

The grid properties are supported in all modern browsers.

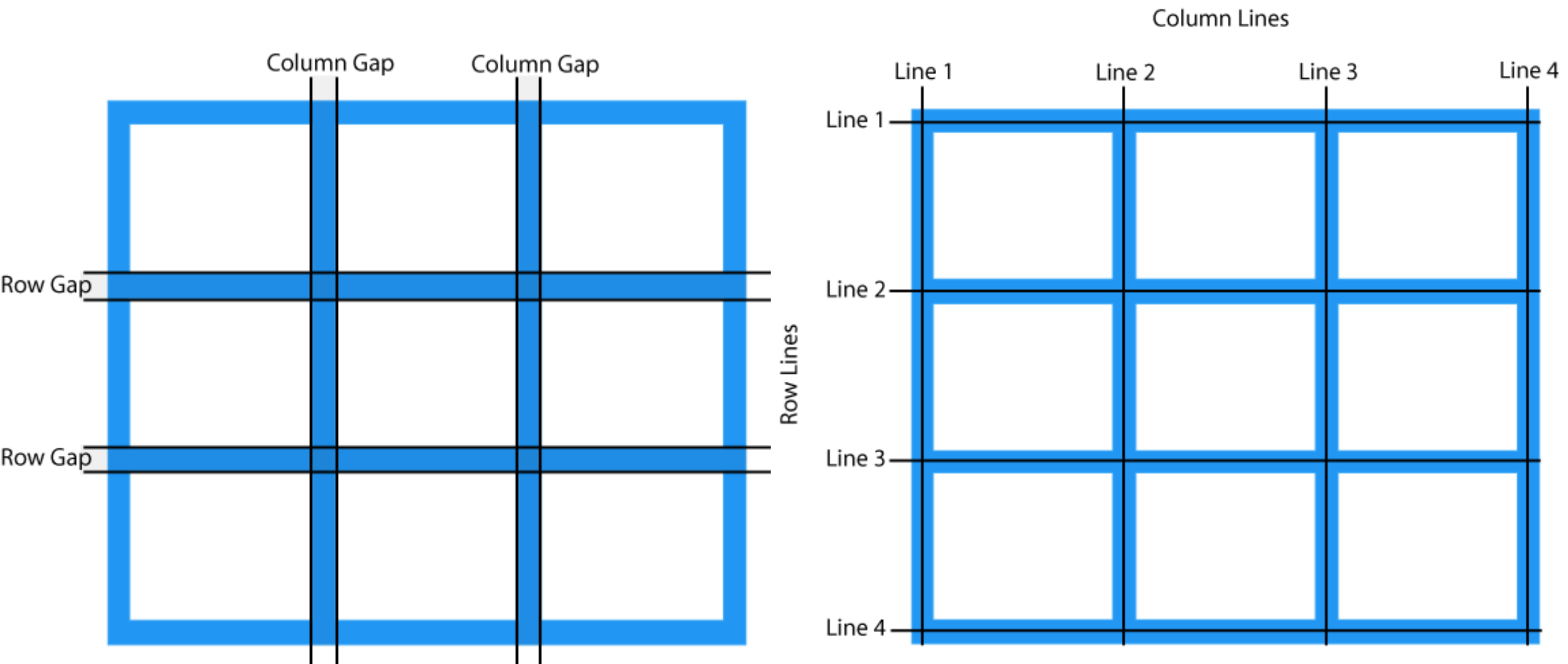
				
57.0	16.0	52.0	10	44

Skjerm bilde, Browser Support: hentet fra https://www.w3schools.com/css/css_grid.asp, [16.08.20]

Hva er CSS3 Grid?



Hva er CSS3 Grid?



display: grid

- Det hele starter med at man angir et område, som inneholder elementer som skal få grid layout, med display: grid. Dette området er gjerne et område som viser flere produkter eller annen type innhold som det er flere av.

```
<section class="products-section"></section>
```

```
.products-section{  
  display: grid;  
  /*annen kode*/  
}
```

grid-template-columns

- Ved å ta i bruk grid-template-columns kan du definere malen for antall kolonner og deres størrelser på siden.
- I kodeeksempelet under angir vi at .products-section skal deles opp i 12 kolonner.
- Vi må senere angi hvor mange av de 12 kolonnene som hvert element skal strekke seg over

```
.products-section{  
  display: grid;  
  grid-template-columns: repeat(12, 1fr);  
}
```

grid-gap

- For å bestemme mellomrommet som skal være mellom kolonnene og radene benytter vi oss av egenskapen grid-gap.

```
.products-section{  
  display: grid;  
  grid-template-columns: repeat(12, 1fr);  
  grid-gap: 1rem; /* kan også bruk f.eks. px */  
}
```


Sette størrelsen på en boks i grid

- I de forrige slidene har vi delt opp nettsiden i et usynlig gittersystem som er delt opp i 12 kolonner. Nå kommer delen hvor vi må bestemme hvor bred hver «boks» i gittersystemet skal være.

```
.products-section{
  display: grid;
  grid-template-columns: repeat(12, 1fr);
  grid-column-gap: 1rem; /* kan også bruk f.eks. px */
}

.product-box{
  grid-column: auto / span 6;
  /* auto angir at hver produktboks plasserer seg på første ledige plass i grid-
  systemet. span angir antall fraksjoner hver boks skal ta opp i bredden */
}
```

Om fractions

- Fractions (fr) er en enhet som lar oss angi (relative) størrelsesforhold.
- I eksempelet under er totalen "3" og hver kolonne er like stor
grid-template-columns: 1fr 1fr 1fr;
- I eksempelet under er totalen "4" og den midterste vil være dobbelt så stor som de to ytre.

grid-template-columns: 1fr 2fr 1fr;

- Vi kan også skrive det følgende for repeterende størrelser:

grid-template-columns: repeat(3, 1fr);

- Og evt. mer komplekst:

grid-template-columns: repeat(2, 1fr 2fr);

Diverse andre teknikker i CSS Grid

Et fleksibelt bildegalleri

- Ved å kombinere grid-template-columns og grid-column kan man lage fleksible bildegallerier bestående av bilder med varierende størrelser.
- Klassene settes direkte på bildene.
- grid-auto-flow: dense; vil fylle viewport med bilder uten å lage større tomrom mellom bildene.

```
#landscape-images{
  display: grid;
  grid-gap: 15px;
  grid-template-columns: repeat(auto-fit, minmax(200px, 1fr));
  grid-auto-rows: 200px;
  grid-auto-flow: dense; /*viktig!*/
}

.wide{
  grid-column: span 2;
}

.tall{
  grid-column: span 2;
}

.wide-and-tall{
  grid-column: span 2;
  grid-row: span 2;
}
```

auto-fit vs. auto-fill

- Auto-fill vil opprette tomrom (kolonner) for nye store nok kolonner, selv om det ikke er noe element til å fylle tomrommene med.
- Auto-fit vil kun forholde seg til elementer som finnes og fylle ut hele bredden med dem.

grid-template-areas

grid-template-areas

- grid-template-areas er en måte å definere visuelt-tekstlig hvordan hovedoppsettet for en nettside skal være.
- Med "hovedoppsettet" menes hvor mye hver hoveddel (for eksempel main-page-header, main-page-nav, main-content, main-page-footer), skal ta opp.

grid-template-areas

- Koden i skjermbildet nedenfor viser hvordan vi har satt en container med diverse egenskaper for å få den synlig. Deretter er den satt til å være et grid.
- Griden skal bestå av 12 kolonner (columns) og 3 rader (rows)

```
#main-page-container{
  width: 978px;
  background-color: rgb(0, 0, 0);
  height: 800px;
  margin: auto;

  display: grid;
  grid-template-columns: repeat(12, 1fr);
  grid-template-rows: 80px auto 80px;
  grid-gap: 15px;

  grid-template-areas:
    "h h h h h h h h h h h h"
    "m m m c c c c c c c c c"
    "f f f f f f f f f f f f";
}
```


grid-template-areas

- Vi ser så et området som heter grid-template-areas med en del bokstaver.
- Hver bokstav står for noe: h = header, m = menu, c = content (main), f = footer.
- Det er 12 bokstaver per rad og 3 rader (angitt i forrige slide med columns og rows)
- Dette er en visuell representasjon hvor mye hver del av nettsiden skal ta opp.

```
grid-template-areas:  
  "h h h h h h h h h h h h"  
  "m m m c c c c c c c c c"  
  "f f f f f f f f f f f f";
```

grid-area

(fortsettelse av temaet grid-template-areas)

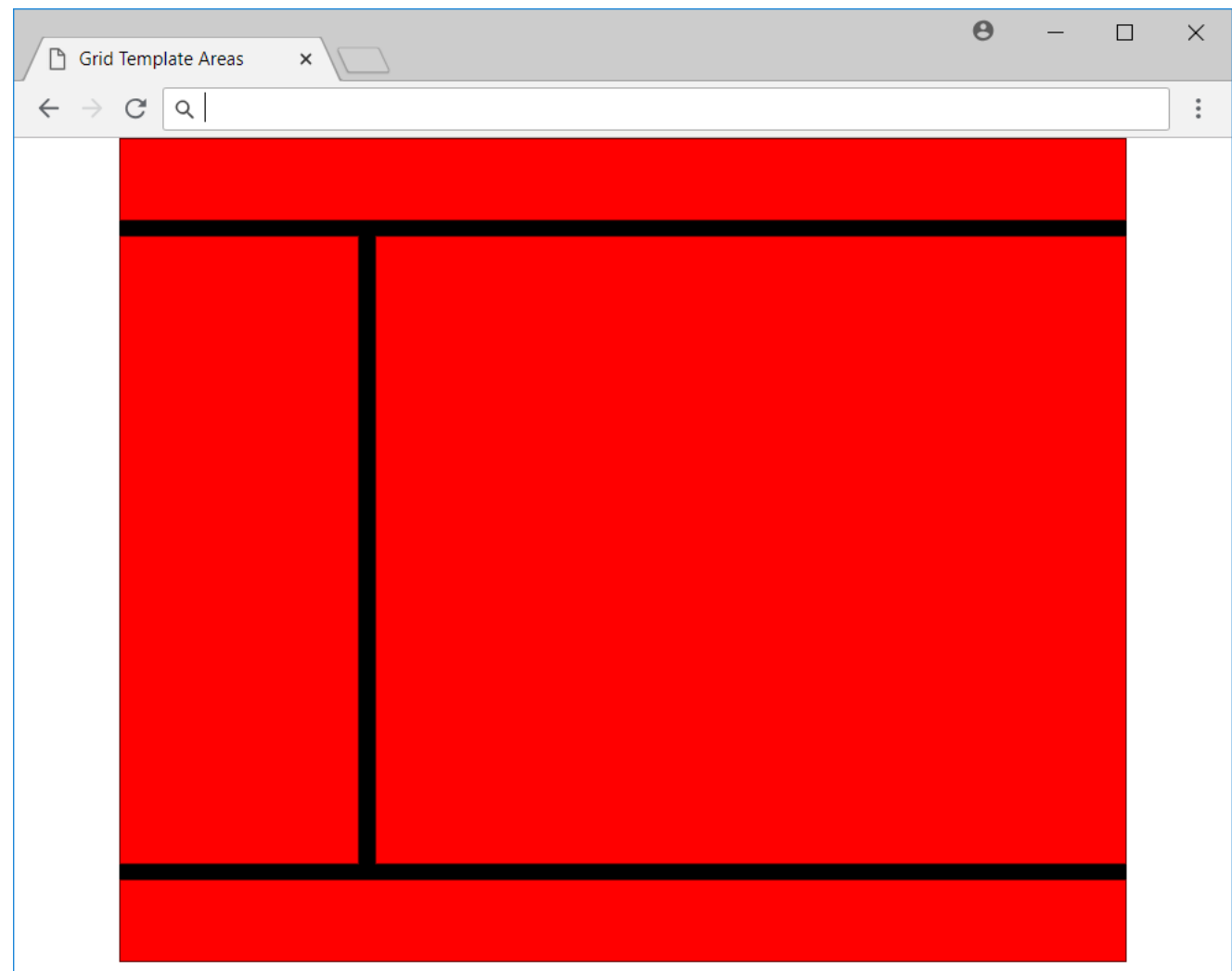
- Vi kan nå sette egenskapen grid-area med verdi som tilsvarer bokstaven i den visuelle representasjonen for å fylle området i nettsiden.

```
header{
  grid-area: h;
}

nav{
  grid-area: m;
}

main{
  grid-area: c;
}

footer{
  grid-area: f;
}
```



grid-template areas: tomme områder

- Merk at man kan angi at områder skal være tomme ved å legge inn punktum.

```
grid-template-areas:  
  "h h h h h h h h h h h h"  
  "m m m c c c c c c c c c"  
  ". . | f f f f f f f f f f";
```

Litteratur/referanser

- Mozilla (2018): Grid Layout, https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Grid_Layout, [last accessed 06.09.21]
- Per Harald Borgen (2018): CSS Grid Course, freeCodeCamp.org, [https://www.youtube.com/watch?v=t6CBKf8K_Ac], [last accessed 06.09.21]