

# CSS Flex

# Navbar-eksempel

DS3103 Webutvikling

Rolando Gonzalez

2022

# Innhold

- Hva er flex?
- Eksempelbruk av flex for enkel navigasjonsbar
- Oversikt egenskaper i flex
- Sentrere et element i en boks

# Hva er flex? – offisiell definisjon

- In the flex layout model, the children of a flex container can be laid out in any direction, and can “flex” their sizes, either growing to fill unused space or shrinking to avoid overflowing the parent. Both horizontal and vertical alignment of the children can be easily manipulated.

(<https://www.w3.org/TR/css-flexbox-1/>)

# Hva er flex?

- Flex er en teknikk for å plassere elementer ved siden av hverandre i bredden. Det typiske caset for bruk av flex er headerområdet på nettsiden hvor man har logo på venstre side, deretter for eksempel søkefelt i midten og navigasjonslenker på høyre side, alt i en rad.
- Mens man i det som heter CSS Grid jobber med to akser (rader og kolonner) jobber man med én akse i CSS Flex.

# Navigasjonsmeny med flex

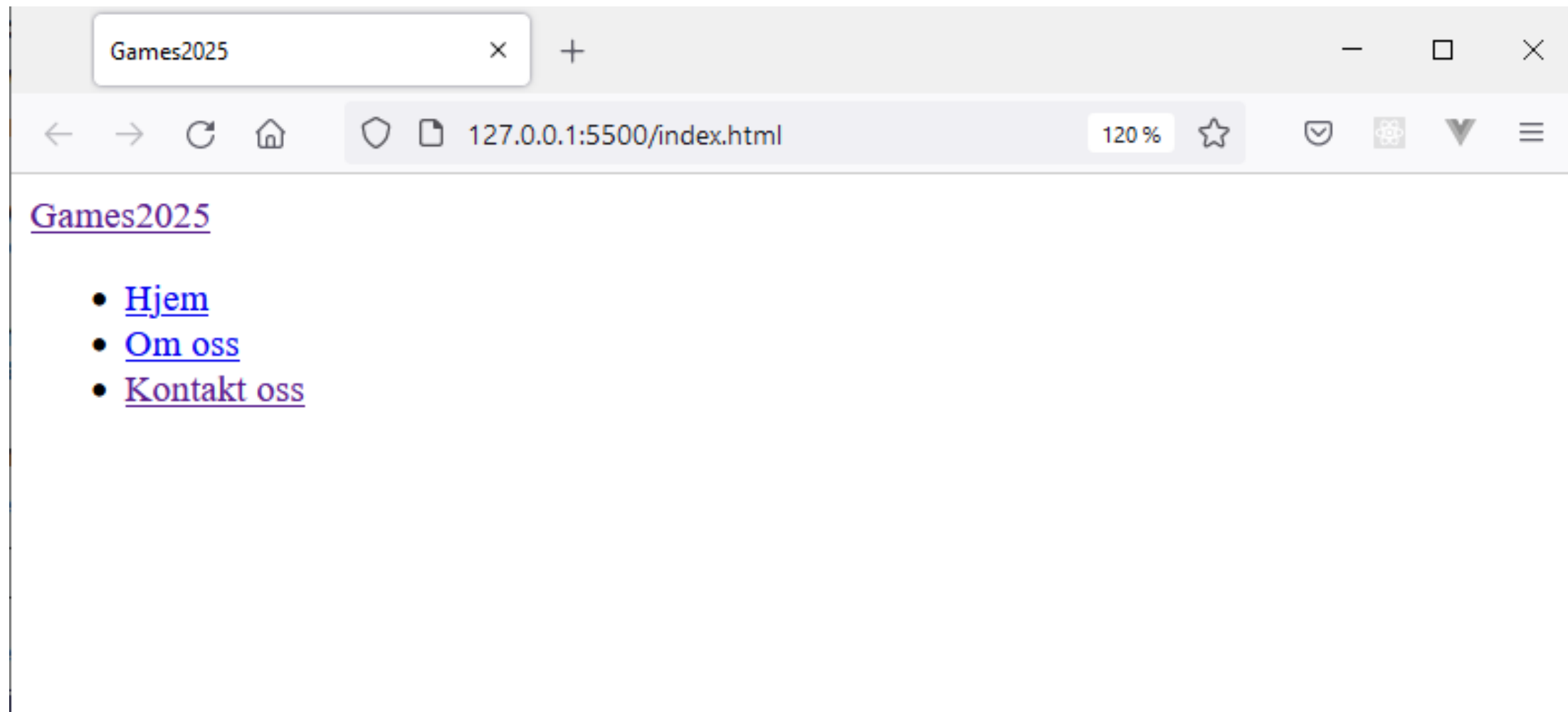
Eksempelkode, steg for steg, for plassering  
av logo og navigasjonsmeny med flex

# HTML

- Oppretter en <header> som inneholder logoen til nettsiden og en <nav> som inneholder en <ul>, som igjen inneholder lenkene våre.

```
11 <header class="main-header">
12
13   <a class="logo" href="index.html">Games2025</a>
14
15   <nav>
16     <ul class="main-navigation">
17       <li><a class="main-navigation__link" href="home.html">Hjem</a></li>
18       <li><a class="main-navigation__link" href="about.html">Om oss</a></li>
19       <li><a class="main-navigation__link" href="contact.html">Kontakt oss</a></li>
20     </ul>
21   </nav>
22
23 </header>
```

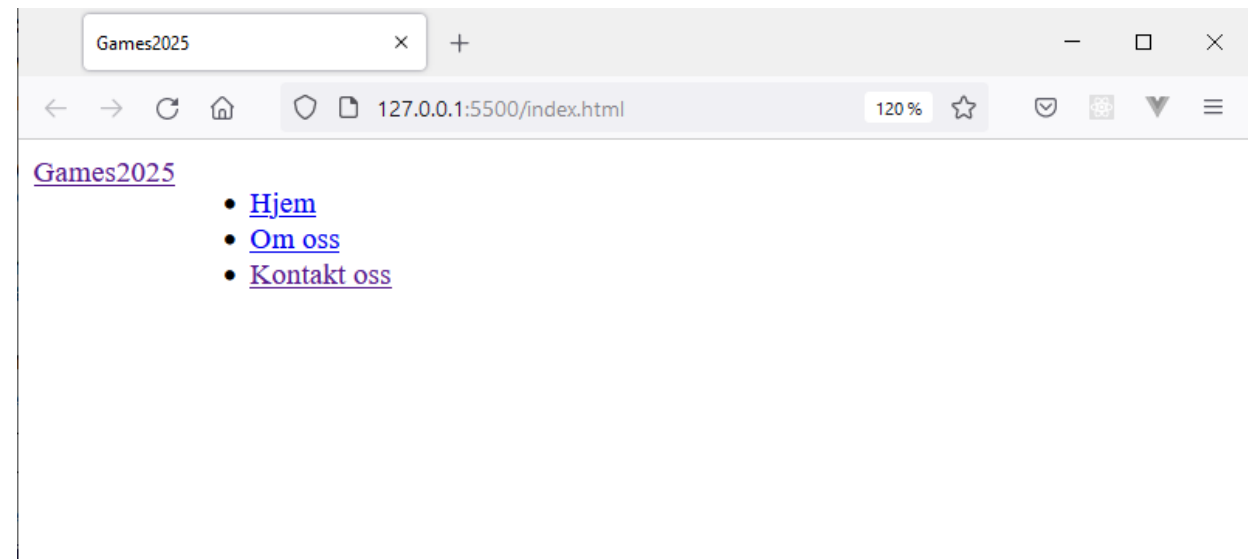
# Hvordan det ser ut så langt



# Opprettelse av flex container

- Det første steget er å gjøre `<header class=«main-header»>` til en flex container. Den inneholder elementene som plasseres ved siden av hverandre.
- Vi gjør `.main-header` til en flex container ved å bruke egenskapen `display: flex` og resultatet er at logo og navigasjon kommer side ved side

```
.main-header{  
  display: flex;  
}
```

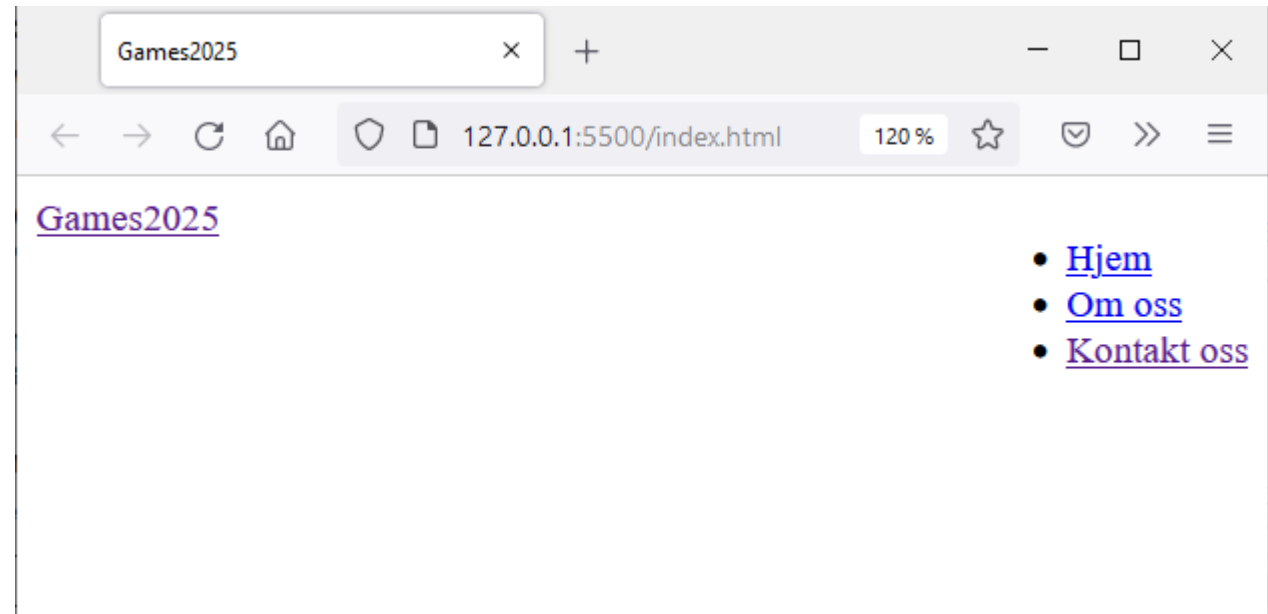




# Flytte logo og navigasjon til hver sin side

- Med justify-content kan man flytte alle elementer til høyre side (right), til midten (center) osv., eller som vist i koden ta hele bredden ved å sette rom mellom elementene med space-between

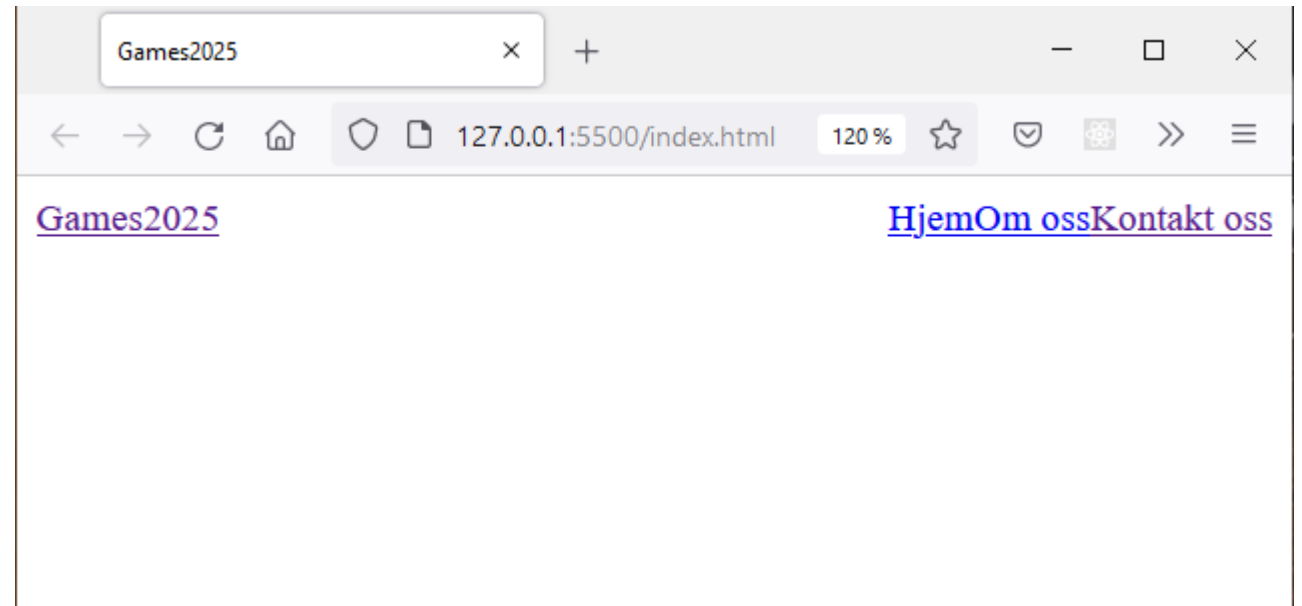
```
.main-header{  
  display: flex;  
  justify-content: space-between;  
}
```



# Flex for navigasjonsmenyen

- Flex fungerer veldig fint for navigasjonsmenyen hvis vi har tenkt å stilsette lenkene med bakgrunnsfarge og padding. Man gjør main-navigation til en flex container og lenkene kommer side ved side
- margin, padding og list-style brukes for å fjerne default liste-style. Merk at dette normalt gjøres i en egen såkalt «CSS reset»-del av CSS.

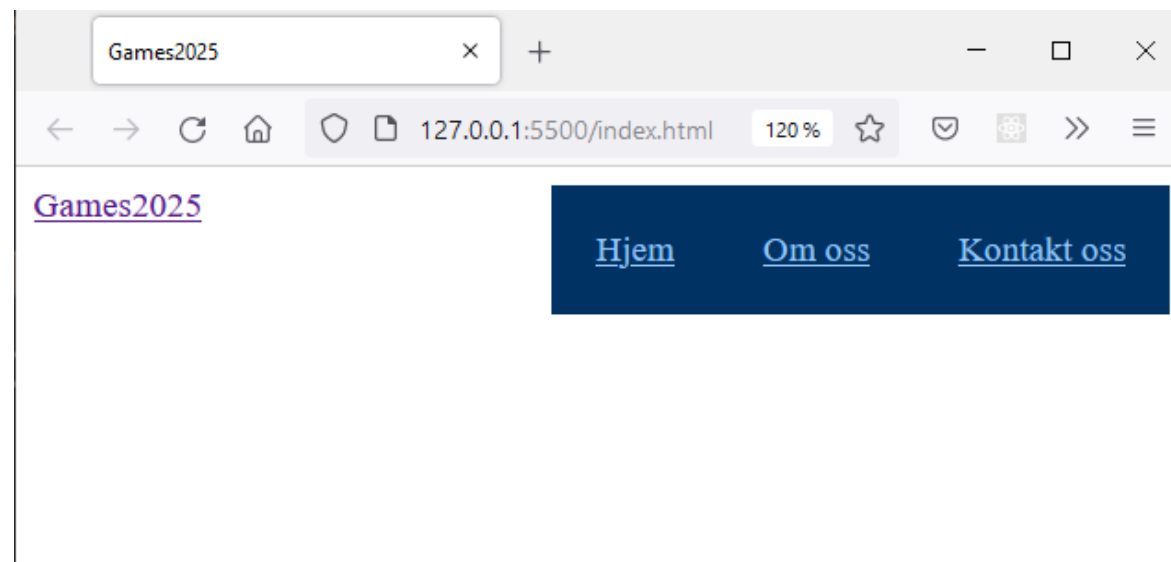
```
.main-header{  
  display: flex;  
  justify-content: space-between;  
}  
  
.main-navigation{  
  display: flex;  
  margin: 0;  
  padding: 0;  
  list-style: none;  
}
```



# Midtstille vertikalt 1 / 2

- Man ønsker å gi padding til elementer i navigasjonsmenyen og man kan ende opp med at noen elementer kommer høyere opp enn andre.
- I eksempelkodeen nedenfor har navigasjonslenkene fått padding og bakgrunnsfarge og kommer lavere ned enn det logoen gjør

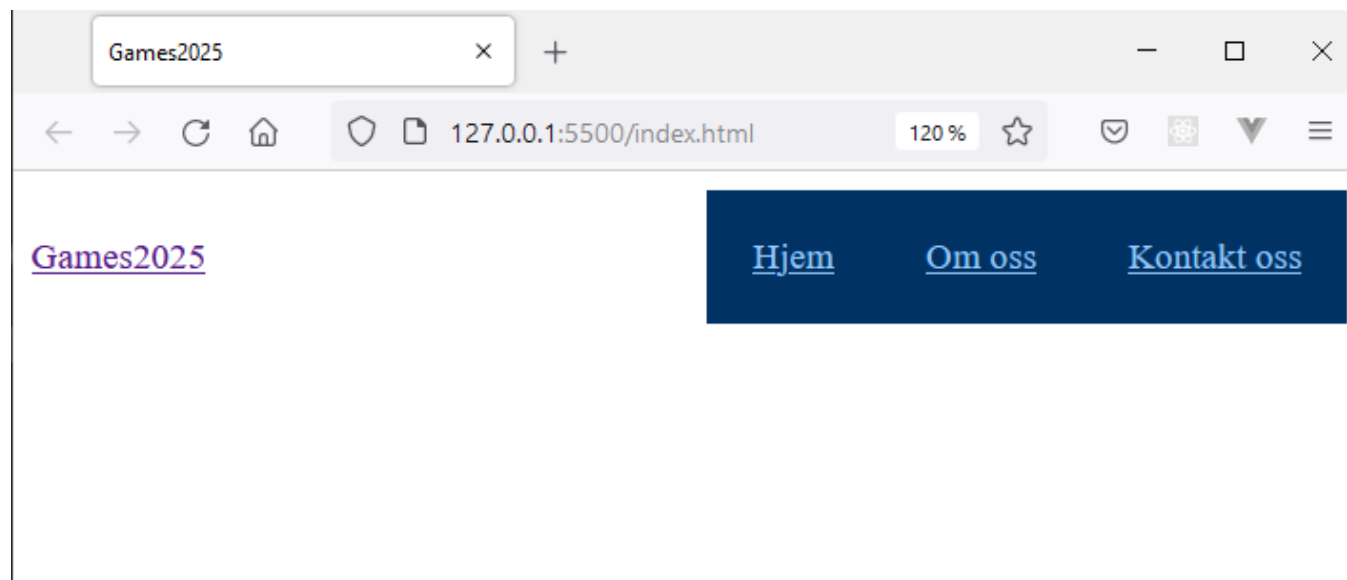
```
.main-navigation__link{  
  display: block;  
  padding: 20px;  
  background-color: ■rgb(0, 50, 100);  
  color: ■rgb(150, 200, 250);  
}
```



# Midtstille vertikalt 2/2

- I flex har vi egenskapen align-items som sørger for at elementene er midtstilt vertikalt – dvs. kommer på samme linje innenfor flex container-boksen.

```
.main-header{  
  display: flex;  
  justify-content: space-between;  
  align-items: center;  
}
```



# Komplett CSS-kode

```
.main-header{
  display: flex;
  justify-content: space-between;
  align-items: center;
}

.main-navigation{
  display: flex;
  margin: 0;
  padding: 0;
  list-style: none;
}

.main-navigation__link{
  display: block;
  padding: 20px;
  background-color: #0056b3;
  color: #cfe2f3;
}
```

# Flex-egenskaper

Property	Description
<u>display</u>	Specifies the type of box used for an HTML element
<u>flex-direction</u>	Specifies the direction of the flexible items inside a flex container
<u>justify-content</u>	Horizontally aligns the flex items when the items do not use all available space on the main-axis
<u>align-items</u>	Vertically aligns the flex items when the items do not use all available space on the cross-axis
<u>flex-wrap</u>	Specifies whether the flex items should wrap or not, if there is not enough room for them on one flex line
<u>align-content</u>	Modifies the behavior of the flex-wrap property. It is similar to align-items, but instead of aligning flex items, it aligns flex lines
<u>flex-flow</u>	A shorthand property for flex-direction and flex-wrap
<u>order</u>	Specifies the order of a flexible item relative to the rest of the flex items inside the same container
<u>align-self</u>	Used on flex items. Overrides the container's align-items property
<u>flex</u>	Specifies the length of a flex item, relative to the rest of the flex items inside the same container

[https://www.w3schools.com/css/css3\\_flexbox.asp](https://www.w3schools.com/css/css3_flexbox.asp)

# Sentrere et element i en boks med flex

- Hvis man ønsker å sentrere et element i midten av en boks kan man enkelt gjøre dette ved å på containeren rundt elementet sette `display: flex;`, `justify-content: center;` og `align-items: center;`

```
.container{  
  width: 100vw;  
  height: 100vh;  
  background-color:  rgb(30, 30, 30);  
  display: flex;  
  justify-content: center;  
  align-items: center;  
}  
  
.container .red-box{  
  width: 100px;  
  height: 100px;  
  background-color:  rgb(255, 0, 0);  
}
```

