

IT 1 - Sandvika VGS

Plassere innhold på nettsiden

Uke 37

Plan

- Plassere innhold på nettsiden
 - Normal flow
 - CSS-egenskapen display
 - Flexbox
 - Grid
 - Floats

Plan

- Plassere innhold på nettsiden
 - Normal flow
 - CSS-egenskapen display
 - Flexbox
 - Grid
 - Floats

Normal flow

- **Normal flow** er måten elementer legger seg på nettsiden når vi ikke har lagt til CSS for å endre plassering på nettsiden.

```
1  <p>I love my cat.</p>
2
3  <ul>
4    <li>Buy cat food</li>
5    <li>Exercise</li>
6    <li>Cheer up friend</li>
7  </ul>
8
9  <p>The end!</p>
```

I love my cat.

- Buy cat food
- Exercise
- Cheer up friend

The end!

Normal flow

- I normal flow legger noen elementer seg på en ny linje, disse kalles for «**block-elementer**»
 - F.eks.: headere, paragrafer og lister
- Andre elementer legger seg på samme linje, og kalles for «**inline-elementer**».
 - F.eks. lenker og bilder

Plan

- Plassere innhold på nettsiden
 - Normal flow
 - CSS-egenskapen display
 - Flexbox
 - Grid
 - Floats

CSS-egenskapen display

- Brukes til å endre plasseringen til elementer på nettsiden
- Eks:
 - I normal flow (uten endringer) er bilder «inline-elementer» dvs. at de ligger ved siden av hverandre på samme linje.
 - Prøv å endre bildene til «block-elementer».

```
img{  
    display: block;  
}
```

CSS-egenskapen display

- Det finnes veldig mange forskjellige display-egenskaper for å endre posisjonen til HTML-elementer

Value	Description
inline	Displays an element as an inline element (like). Any height and width properties will have no effect
block	Displays an element as a block element (like <p>). It starts on a new line, and takes up the whole width
contents	Makes the container disappear, making the child elements children of the element the next level up in the DOM
flex	Displays an element as a block-level flex container
grid	Displays an element as a block-level grid container
inline-block	Displays an element as an inline-level block container. The element itself is formatted as an inline element, but you can apply height and width values
inline-flex	Displays an element as an inline-level flex container
inline-grid	Displays an element as an inline-level grid container
inline-table	The element is displayed as an inline-level table
list-item	Let the element behave like a element

Eksempel

- Anta jeg ønsker å posisjonere følgende section-elementer sidelengs:

HTML

```
<section class="vedSiden"></section>
<section class="vedSiden"></section>
<section class="vedSiden"></section>
```

- Dette er «block-elementer» som legger seg under hverandre på hver sin linje
- Vi kan gjøre de om til **inline-block***, og sette passende margin for å plassere de pent sidelengs men i lengden blir dette tungvint, derfor bruker vi ikke så ofte «inline-elementer».

CSS

```
.vedSiden{
  background-color: lightblue;
  width: 300px;
  height: 200px;
  border: 1px solid black;
  border-radius: 5%;
  display: inline-block;
}
```

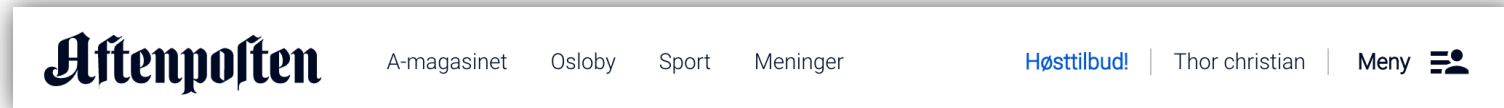
*inline-block er «inline-elementer» som har alle egenskapene til «block-elementer» bortsett fra ny linje.

Plan

- Plassere innhold på nettsiden
 - Normal flow
 - CSS-egenskapen display
 - Flexbox
 - Grid

Flexbox

- Flexbox brukes til en-dimensjonal posisjonering
 - Flexbox er mer fleksibel enn inline, med flere posisjoneringsegenskaper.



Flexbox

- Flexbox ligger rundt innholdet som skal posisjoneres

```
<section class="flexcontainer">  
  <section class="vedSiden"></section>  
  <section class="vedSiden"></section>  
  <section class="vedSiden"></section>  
</section>
```

- Flex settes på et element i CSS ved **display: flex;**

```
.flexcontainer{  
  display: flex;  
}
```

Flexbox

- Vi har nå masse egenskaper til posisjonering:

flex-direction brukes til å sette retningen på elementene.

```
.flexcontainer{  
  display: flex;  
  flex-direction: column;  
  + flex-direction: row-reverse;
```

```
.flexcontainer{  
  display: flex;  
  flex-direction: column-reverse;
```

```
.flexcontainer{  
  display: flex;  
  flex-direction: row;
```

justify-content brukes til å sette plassere elementene inne i flexboksen

```
.flexcontainer{  
  display: flex;  
  justify-content: center;  
}
```

+ justify-content: flex-end,
+justify-content: space-around;
+justify-content: space-evenly;

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

```
.flexcontainer{  
  display: flex;  
  justify-content: flex-start;  
}
```

- Det finnes mange flere egenskaper, sjekk internett! (men husk av vi foretrekker grid når vi plasserer i to dimensjoner)

https://www.w3schools.com/css/css3_flexbox.asp

Oppgave - navigasjonsbar

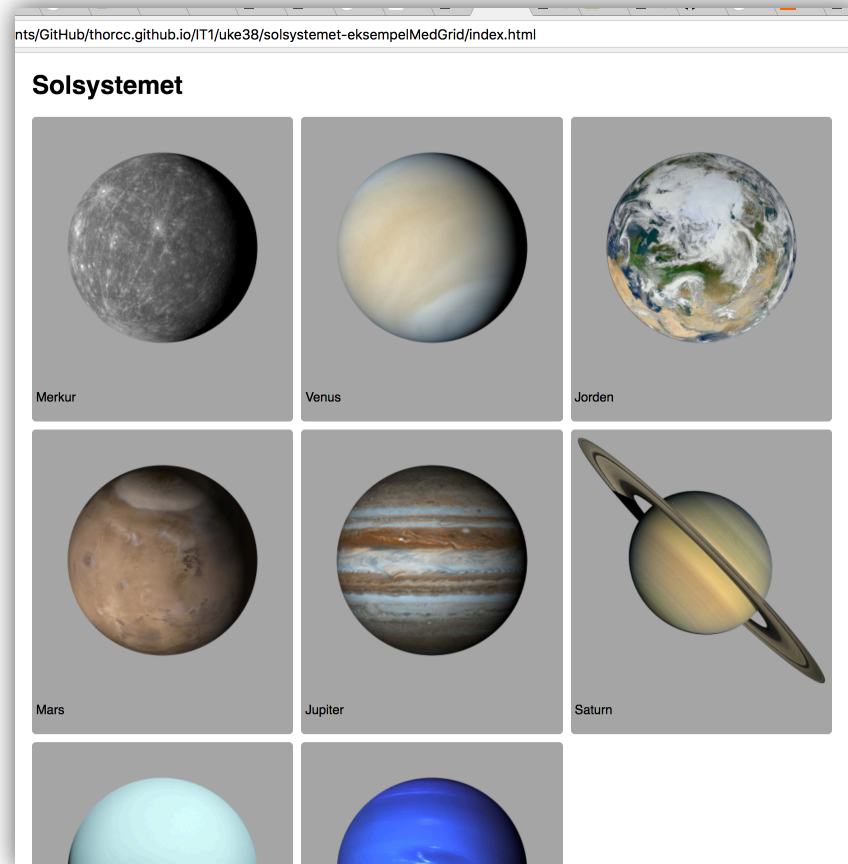
- Vi lager en navigasjonsbar
 - Lag fire-fem lenker på toppen av nettsiden din.
 - Bruk flexbox til å plassere lenkene pent sidelengs. Kombiner gjerne med width og margin på det omkransede elementet
 - Eksperimenter med text-decoration: none, farger og skriftstørrelse på lenkene dine
 - Legg på border-bottom slik at vi får en underlinje på hele navigasjonsbaren.

Plan

- Plassere innhold på nettsiden
 - Normal flow
 - CSS-egenskapen display
 - Flexbox
 - Grid

Grid

- Grid brukes til to-dimensjonal posisjonering



Grid

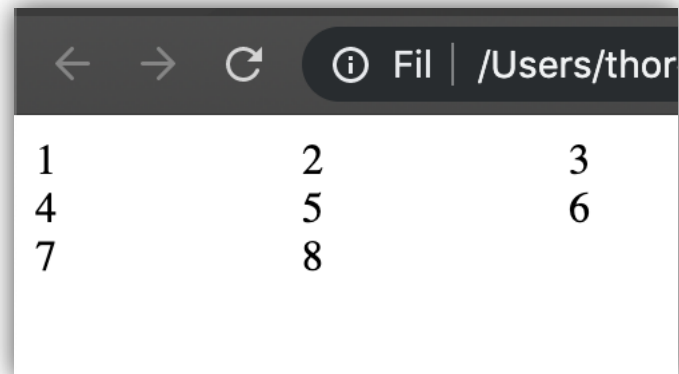
- Grid-elementet ligger rundt innholdet som skal posisjoneres

HTML

```
<section class="gridcontainer">
  <section class="gridelement">1</section>
  <section class="gridelement">2</section>
  <section class="gridelement">3</section>
  <section class="gridelement">4</section>
  <section class="gridelement">5</section>
  <section class="gridelement">6</section>
  <section class="gridelement">7</section>
  <section class="gridelement">8</section>
</section>
```

CSS

```
.gridcontainer{
  display: grid;
  grid-template-columns: 100px 100px 100px;
}
```



1	2	3
4	5	6
7	8	

Grid

- `grid-template-columns` bestemmer hvor mange elementer som skal være i bredden og størrelsen på disse.

```
grid-template-columns: 100px 100px 100px;
```

Gir tre elementer på 100px ved siden av hverandre

```
grid-template-columns: 100px 100px 400px 100px;
```

Gir fire elementer ved siden av hverandre, hvor element nummer tre er 400px mens de andre er 100px

```
grid-template-columns: 1fr 2fr 3fr;
```

Gir tre elementer ved siden av hverandre, hvor det første elementet tar opp én del, det andre to deler og det tredje tre deler. (fr = fraction)

Grid

- Tips:
 - Grid-gap gir margin mellom elementene i en grid. Eks: `grid-gap: 10px;`

Oppgave - solsystemet

- Du skal lage et nettsted om solsystemet
 - Nettstedet skal ha minst tre sider.
 - Lag en side som presenterer planetene i en grid.
 - Lag en navigasjonsbar som lenker mellom sidene
 - Bilder av planetene ligger i mappen «bilder»