

PRESENTASJON

BACHELOROPPGAVE VÅREN 2018 HØYSKOLEN KRISTIANIA STUDENT 288509

PROBLEMSTILLING

Hvordan kan man designe en nettbasert læringsplattform for å hjelpe unge grafisk design studenter (18-30 år) som synes koding er vanskelig, å lære HTML5 og CSS3 på en enkel måte de forstår, synes er engasjerende og er relevant for faget grafisk design?

VISUELL IDENTITET

< graphic
code >

< graphic
code >

< graphic
code >

OPEN SANS

Open sans brukes på hovedteksten. Den humanistiske sans serif fonten har åpne former og et vennlig, men nøytralt utseende. Fonten har god lesbarhet og er optimalisert for web. Jeg bruker regular, semibold og bold for å variere uttrykket slik at den kan bruke den på alle de typografiske nivåene.

Regular

AaBbCcDdEe
 1234567890
 ! « # % & / () = ? + * , . ; : -

Semibold

AaBbCcDdEe
1234567890
! « # % & / () = ? + * , . ; : -

Bold

AaBbCcDdEe
1234567890
! « # % & / () = ? + * , . ; : -

ROBOTO MONO

Roboto mono er fonten jeg bruker på koden på plattformen. Det er en monospaced font og den er optimalisert for å være godt lesbar på skjerm. Den har kjennetegn som gjør den til en god font for koding.

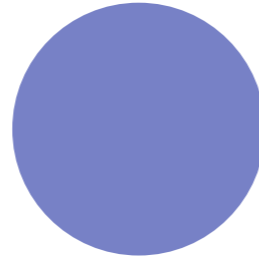
Regular

AaBbCcDdEe
 1234567890
 ! « # % & / () = ? + * , . ; : -

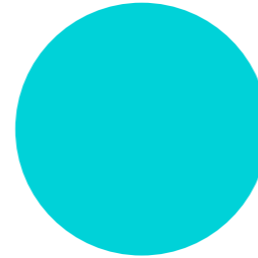
Farger

Hovedfarger

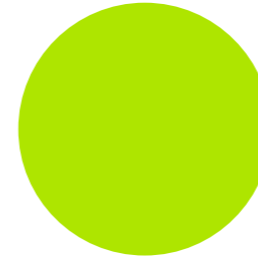
#7781C6



#00D2D8



#AEE500

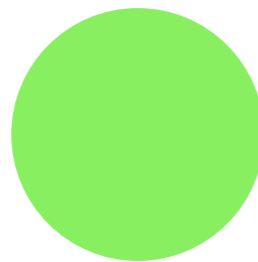


Sekundærfarger

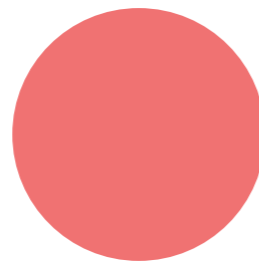
#5D9CFF



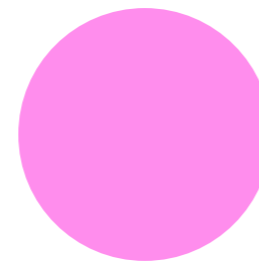
#87EF60



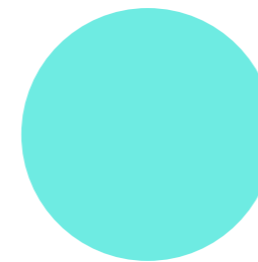
#F07272



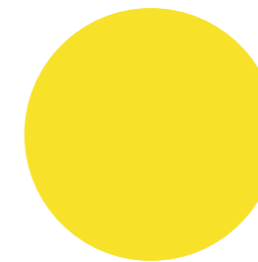
#FF8DED



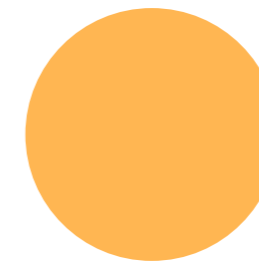
#6EEBE2



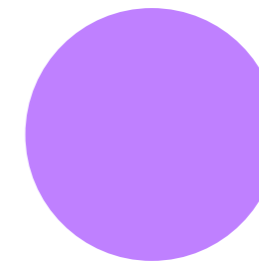
#F8E129



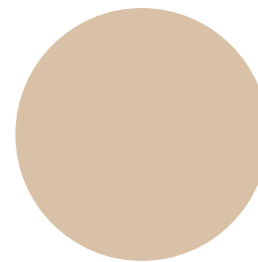
#FFB652



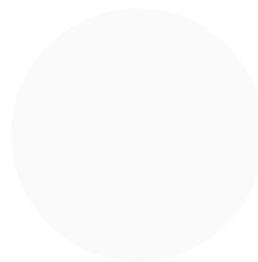
#BF80FF



#D8C1A7

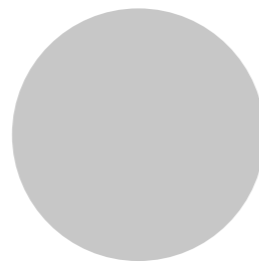


#FAFAFA



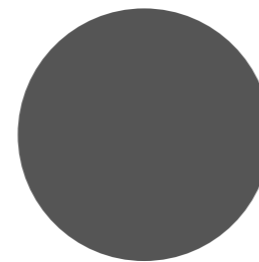
Bokser og bakgrunnsfarge

#C7C7C7



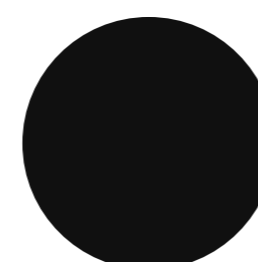
Rammer og skygge

#555555



Tekst

#101010



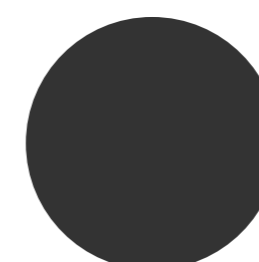
Kode område i kurs

#E2E2E2



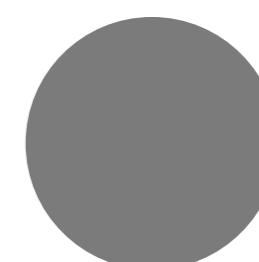
Tekst område i kurs

#333333

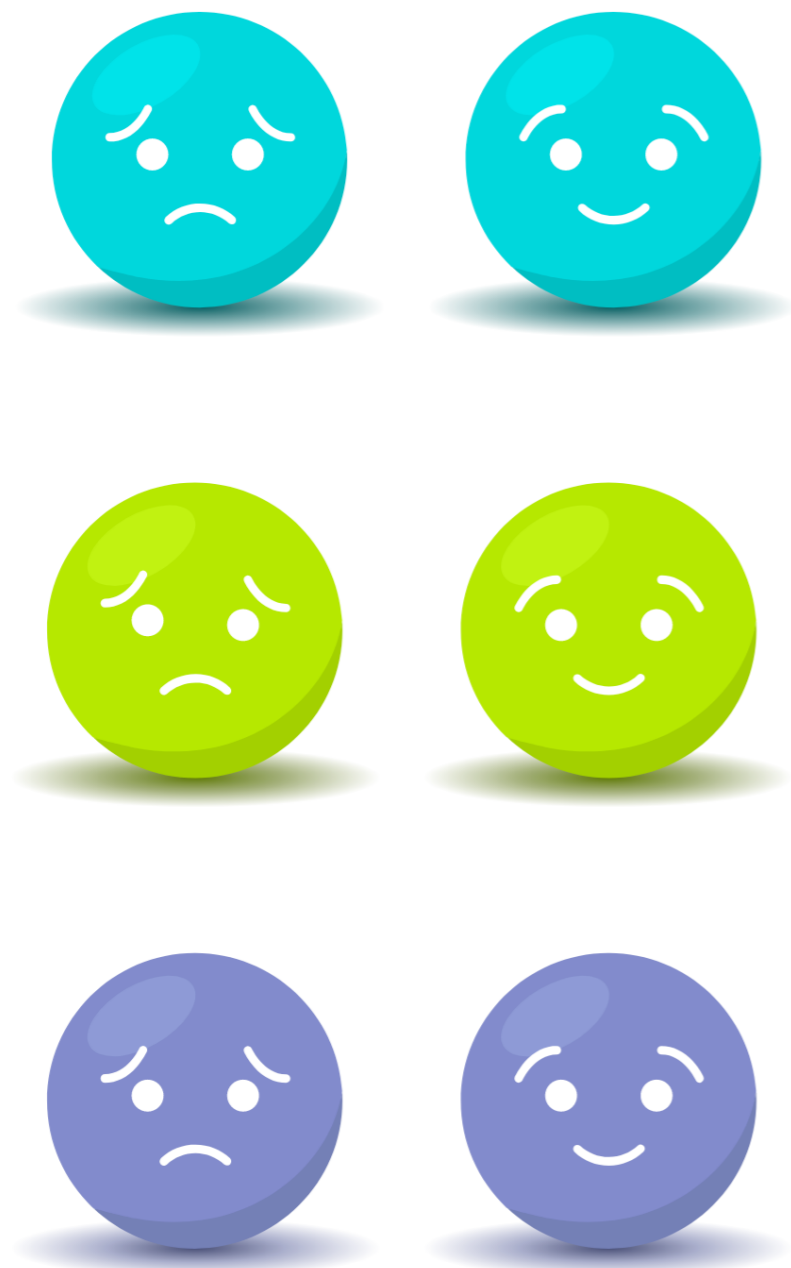


Ramme rundt kurs

#7B7B7B



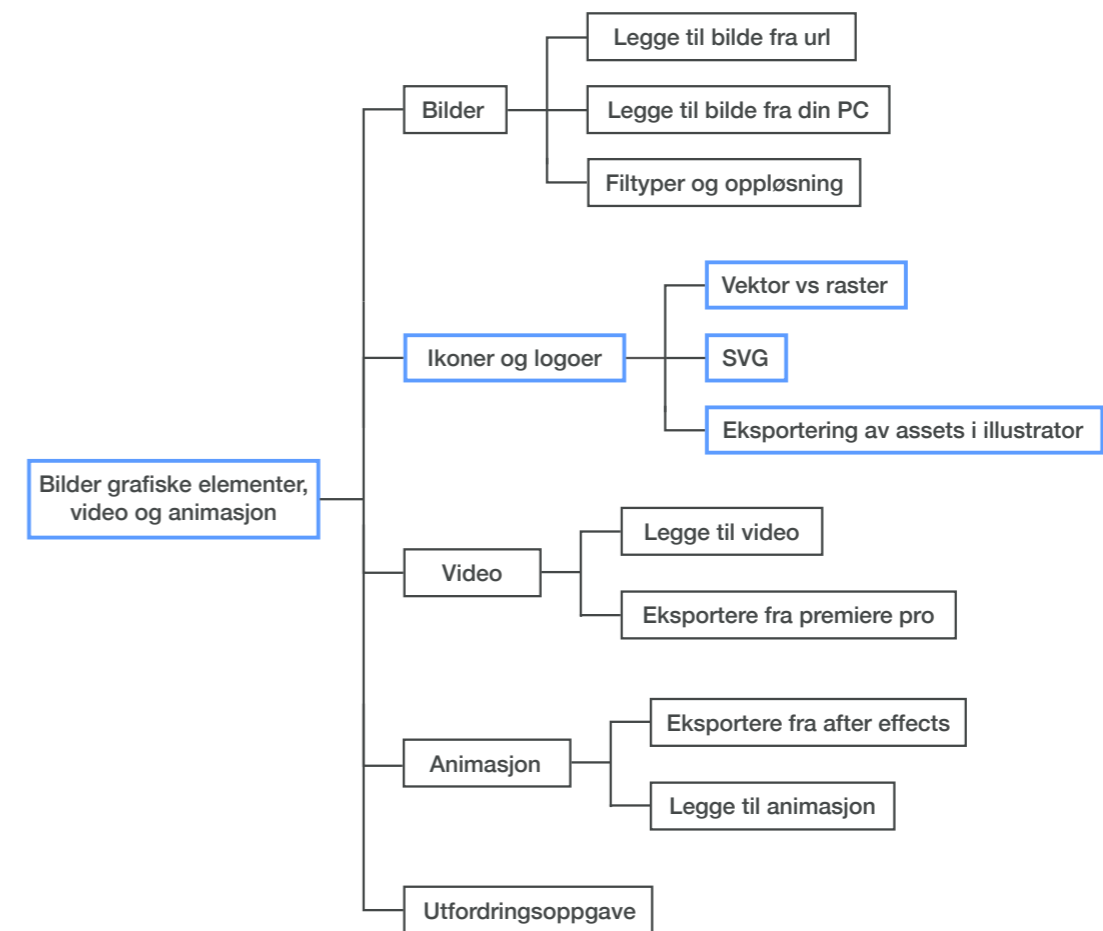
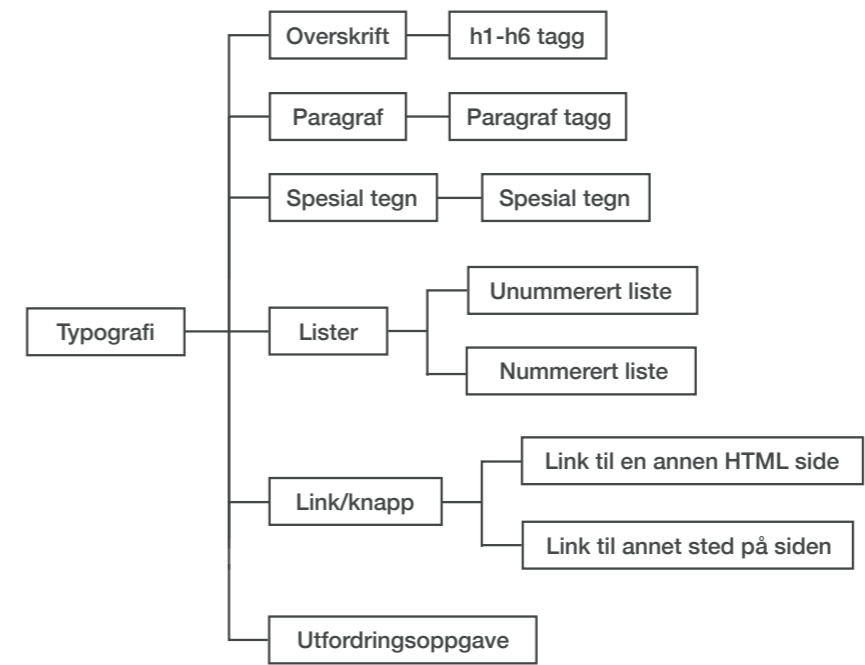
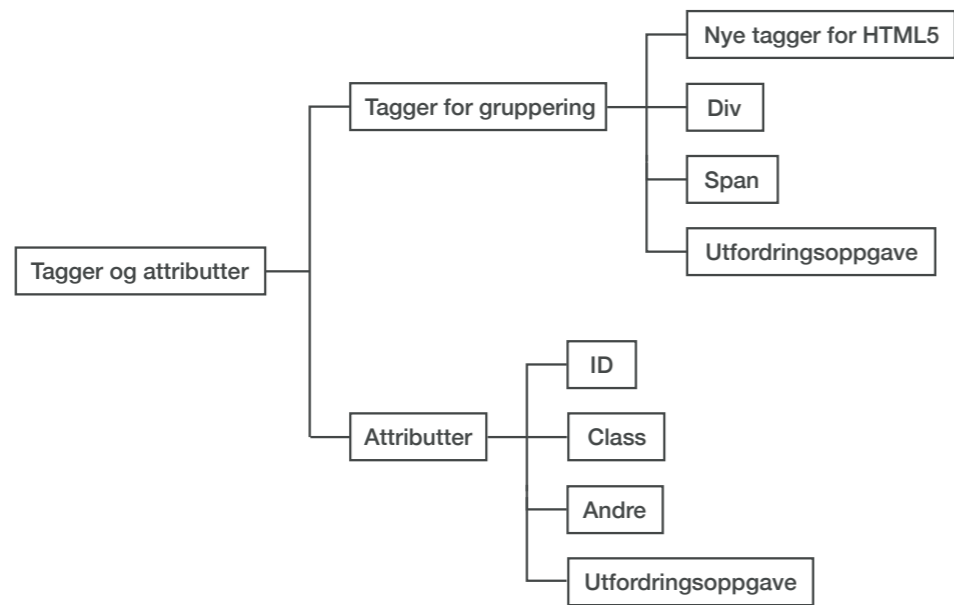
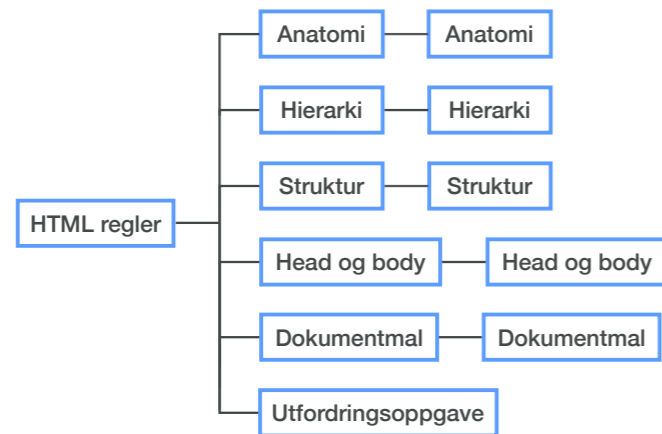
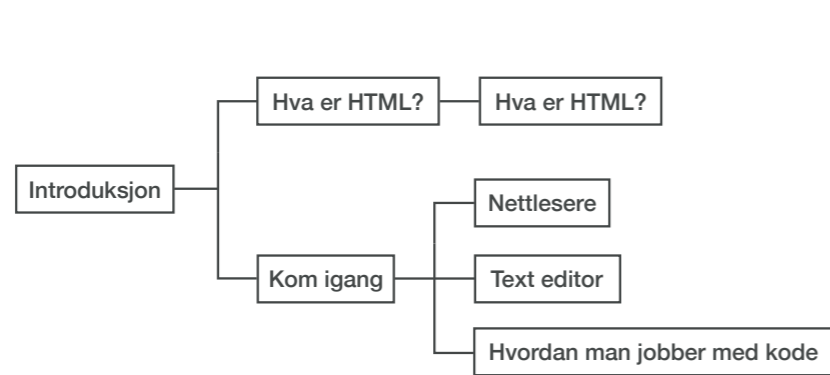
HTML og CSS faner i kurs



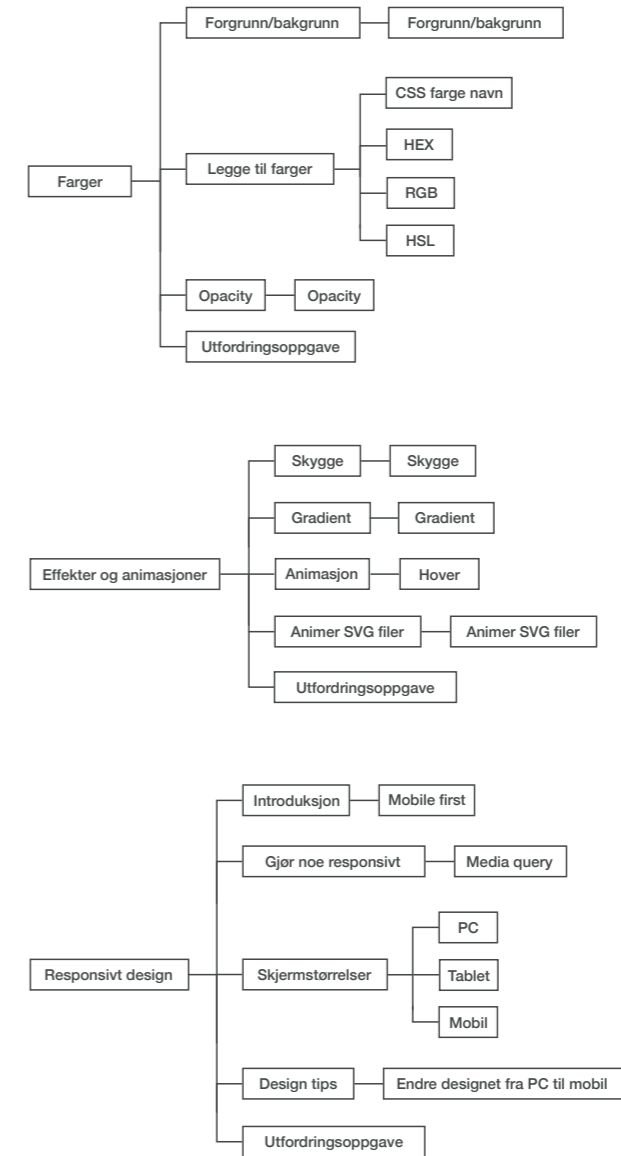
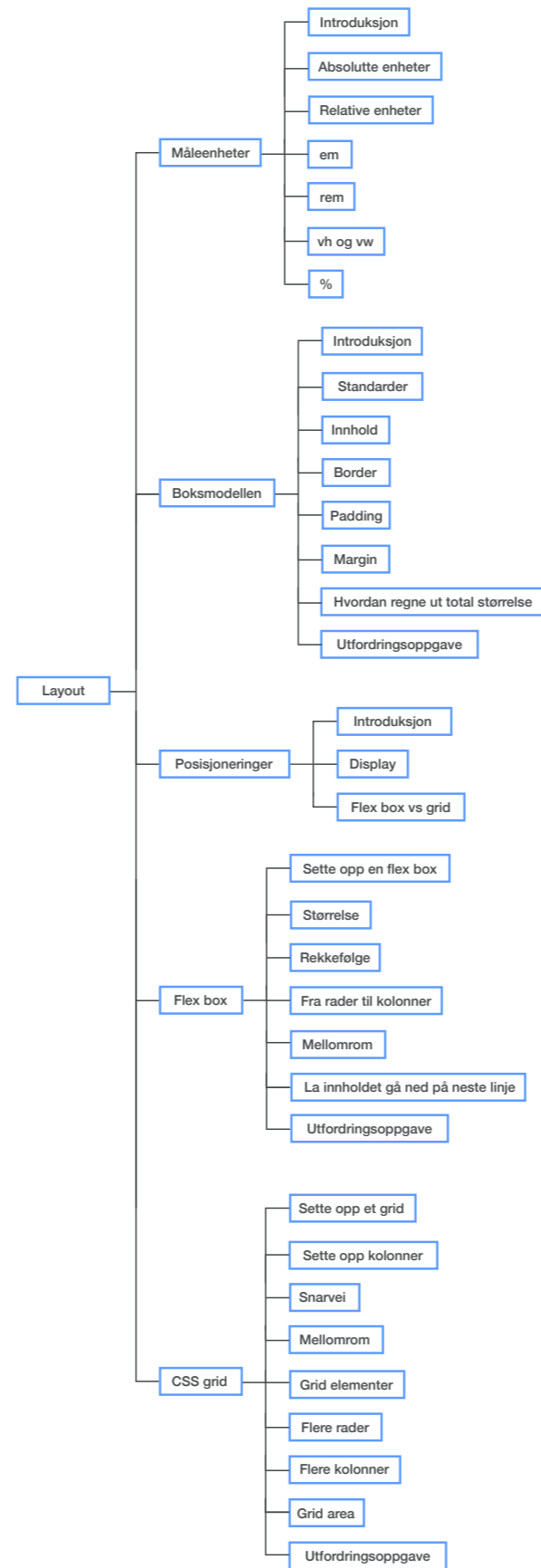
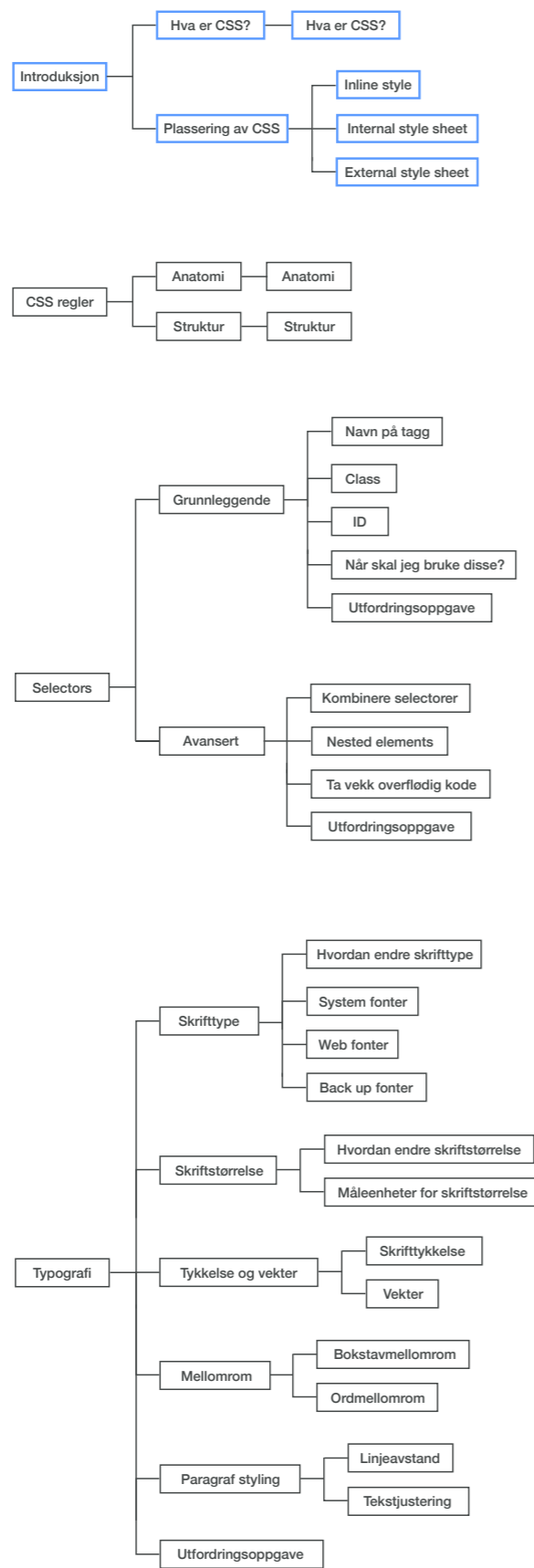


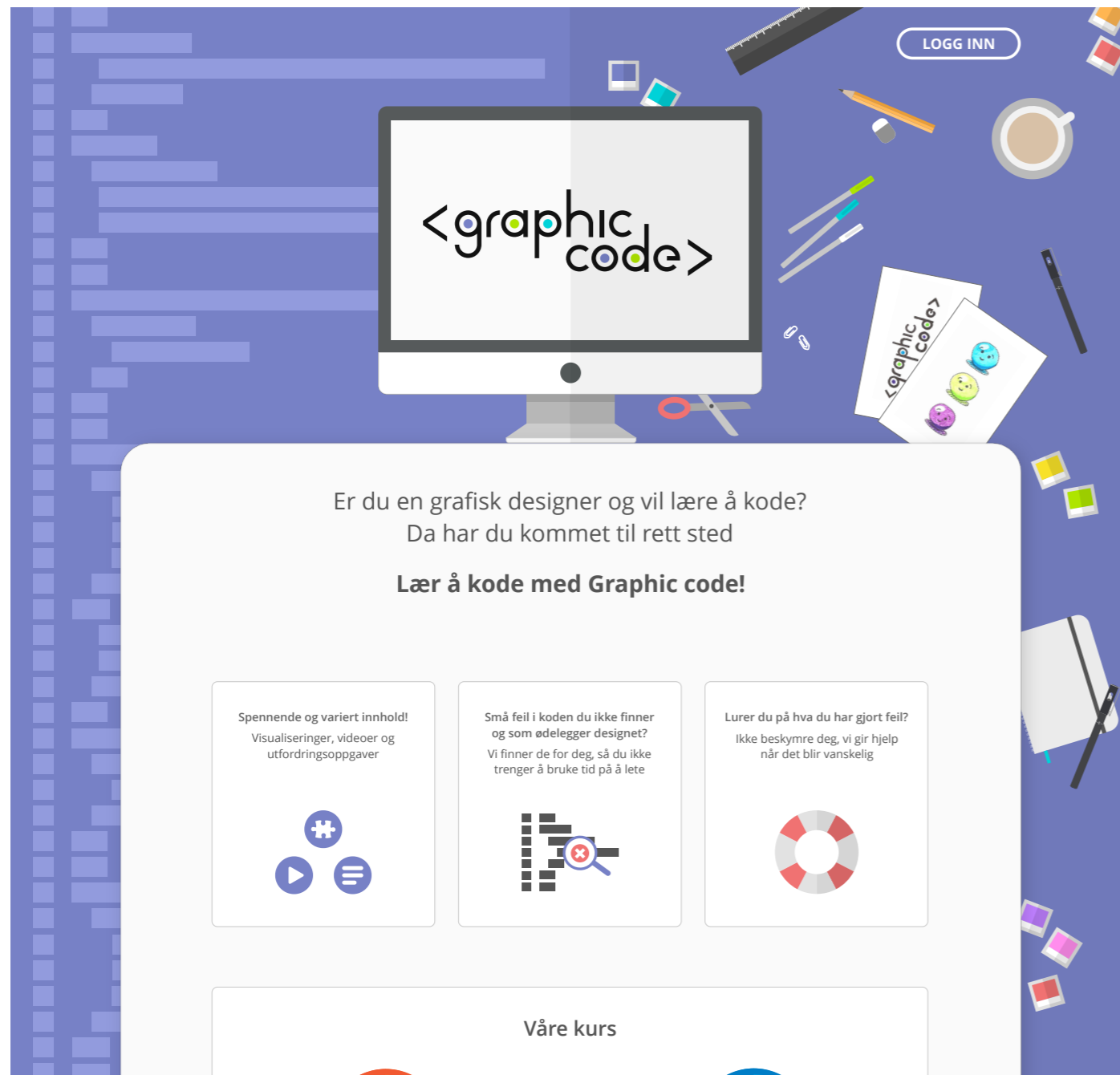
BRUKERGRENSESNITT

Innholdsplan - HTML



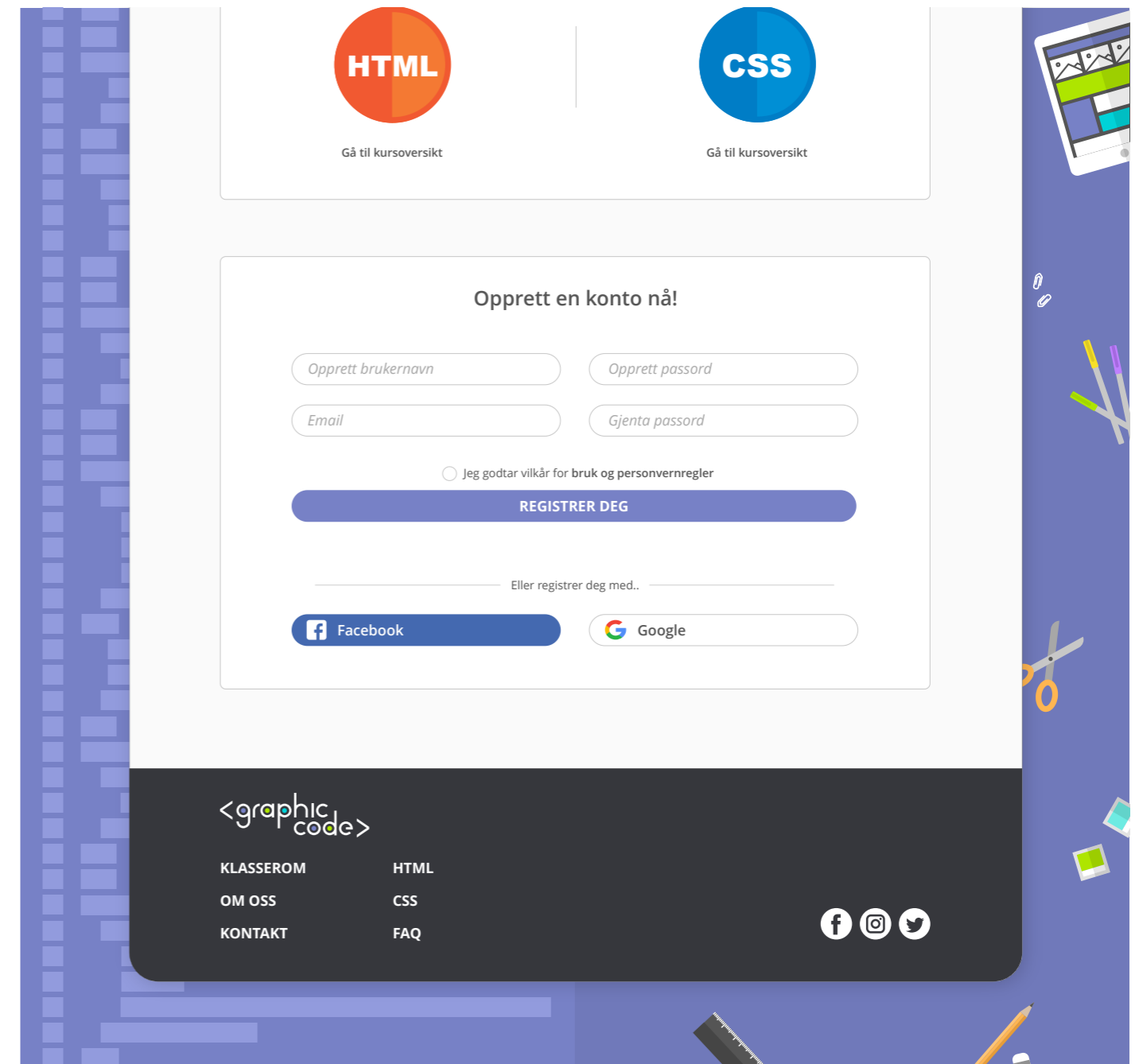
Innholdsplan - CSS





Index siden er den første brukeren møter. Den representerer kontrasten mellom koding (venstre side) og grafisk design (høyre side).

Man får en god forståelse for hva nettsiden handler om fra første stund. Det er lett å kunne se hva kursene inneholder, logge seg inn eller registrere seg.



The login page features a white card on a blue background. At the top, it says "Logg inn". Below this are two input fields: "Brukernavn/email" and "Passord". There are two radio buttons: "Husk meg" (selected) and "Glemt passord?". A blue button labeled "LOGG INN" is positioned below the radio buttons. Underneath, it says "Eller logg inn med..". There are two social login options: "Facebook" with its logo and "Google" with its logo. At the bottom of the card, it says "Har du ikke en bruker? Registrer deg HER".

The forgot password page features a white card on a blue background. At the top, it says "Glemt passord". Below this is a text prompt: "Skriv inn email adressen din nedenfor så sender vi et nytt passord til deg på mail". There is one input field labeled "Brukernavn/email". At the bottom of the card is a blue button labeled "SEND".

Innloggingen er enkel å forstå. Du kan logge inn med brukernavn/email og passord, Facebook eller Google.

The mail sent confirmation page features a white card on a blue background. At the top, it says "Mail sendt!". Below this is a blue circular icon with a white smiley face. Underneath the icon is the text "Skjekk innboksen din". At the bottom of the card is a blue button labeled "SEND".

<graphic code>

INNSTILLINGER FORUM LOGG UT

Sofie

HTML
Kurs i HTML tilrettelagt for nybegynnere. Kurset lærer deg reglene og strukturen for HTML, de viktigste taggene og hvordan man legger til innhold på en nettside.

FORTSETT KURSOVERSIKT

CSS
Kurs i CSS tilrettelagt for nybegynnere. Kurset lærer deg reglene for CSS og typografi, layout, farger, effekter og animasjon.

START KURSOVERSIKT

I **klasserommet** finner du oversikt over kursene og hvor langt du har kommet i de. Herfra kan du gå til innstillingene for å legge til et profilbilde og forsidebilde, du kan gå til forum, se kursoversikten og gå til bestemte deler av et kurs eller fortsette kurset der du avsluttet sist.

Kursoversikt - HTML

På **kursoversikten** til HTML er det fem deler. Hver del har sin egen illustrasjon som beskriver temaet.

Når du åpner en del, ser du alle underdelene av kurset. Om du leter etter noe spesifikt kan du gå hit.

Når en del er ferdig, bli den huket av.

5/5

BILDE, GRAFISKE ELEMENTER, VIDEO OG ANIMASJON

I denne delen lærer du om hvordan man legger til bilder, grafisk elementer, videoer og animasjoner til en nettside. I tillegg legges det vekt på hvordan du går fra design til kode.

- BILDE
- IKONER OG LOGOER
 - Vektor vs raster
 - SVG
 - Eksportering av assets i illustrasjon
- VIDEO
- ANIMASJON
- UTFORDRINGSOPPGAVE

2/5

HTML REGLER

I denne delen lærer du hvordan vi skriver HTML. HTML kan sammenlignes med å lære et nytt språk, der man har grammatikkregler man må forholde seg til. Et HTML element må være skrevet på en bestemt måte for å funke.

- ANATOMI
- HIERARKI
- STRUKTUR
- HEAD OG BODY
- DOKUMENTMAL
- UTFORDRINGSOPPGAVE

1/5

INTRODUKSJON

I denne delen får du en innledning i hva HTML er og hva det brukes til. I tillegg får du alle verktøyene du trenger for å komme igang å kode på egen hånd utenfor denne plattformen.

- HVA ER HTML?
- KOM IGANG

3/5

TAGGER OG ATTRIBUTTER

I denne delen lærer du om de vanligste taggene og attributtene vi bruker i HTML.

```

<div>
  <span>
    <header>
      <footer>
        <main>
          <nav>
            <section>
              <aside>
                class=«»
                id=«»
    
```

- TAGGER FOR GRUPPERING
- ATTRIBUTTER

4/5

TEKST

I denne delen lærer du om HTML taggene som handler om tekst og linker.

- OVERSKRIFT
- PARAGRAF
- SPELIAL TEGN
- LISTER
- LINK/KNAPP
- UTFORDRINGSOPPGAVE

Overskrift

Lorem ipsum dolor [LINK](#), consectetur adipiscing elit. Consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris. 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.

- Liste 1
- Liste 2
- Liste 3

[Knapp](#)

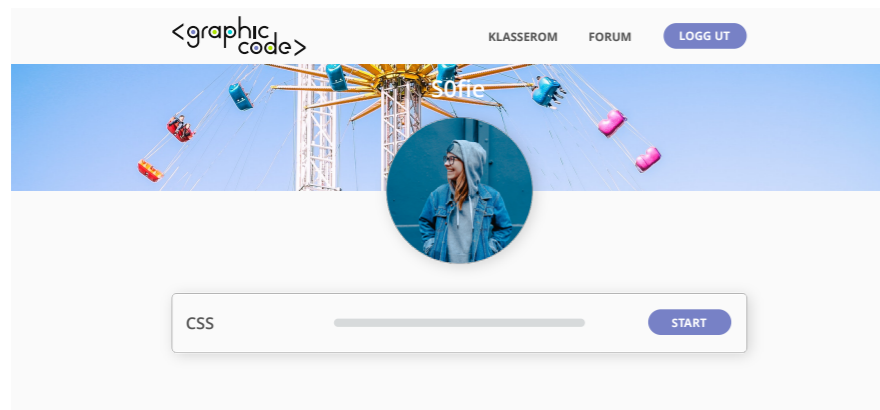
5/5

BILDE, GRAFISKE ELEMENTER, VIDEO OG ANIMASJON

I denne delen lærer du om hvordan man legger til bilder, grafisk elementer, videoer og animasjoner til en nettside. I tillegg legges det vekt på hvordan du går fra design til kode.

- BILDE
- IKONER OG LOGOER
- VIDEO
- ANIMASJON
- UTFORDRINGSOPPGAVE

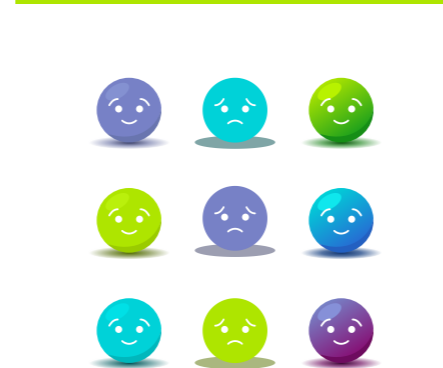
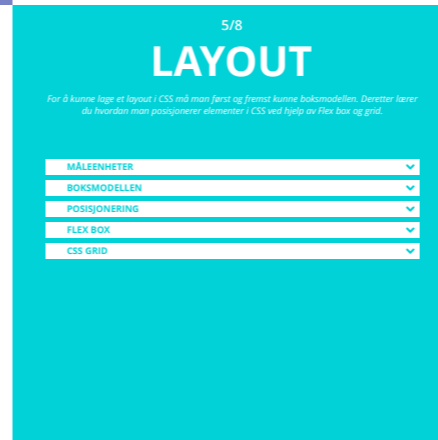
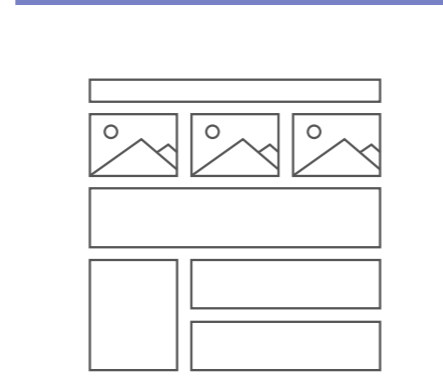
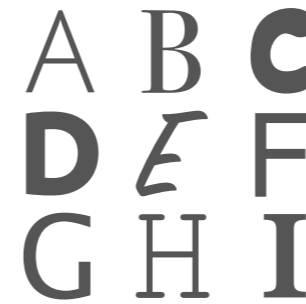
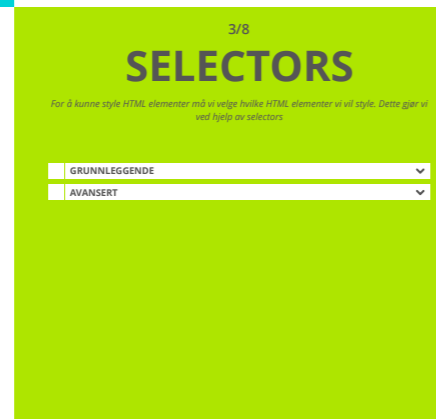
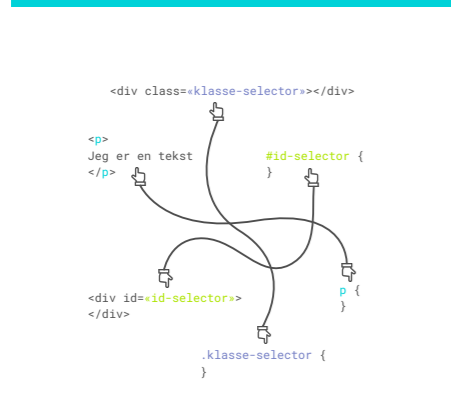
- HVA ER HTML?
- KOM IGANG



```
ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
  width: 200px;
  background-color: #91BF00;
}

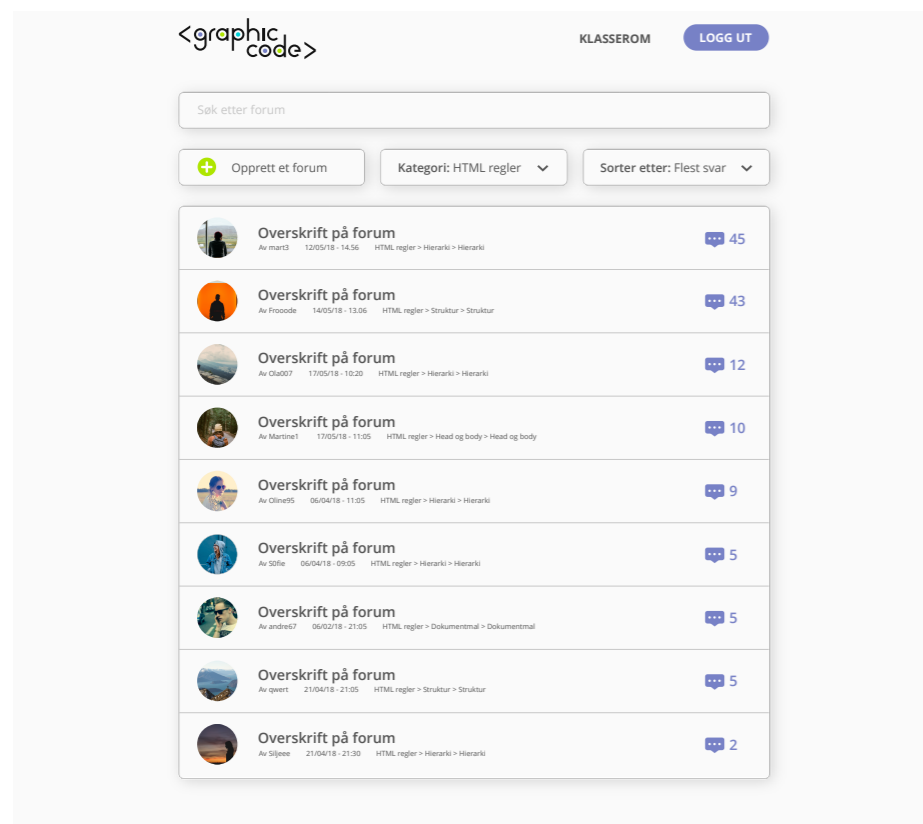
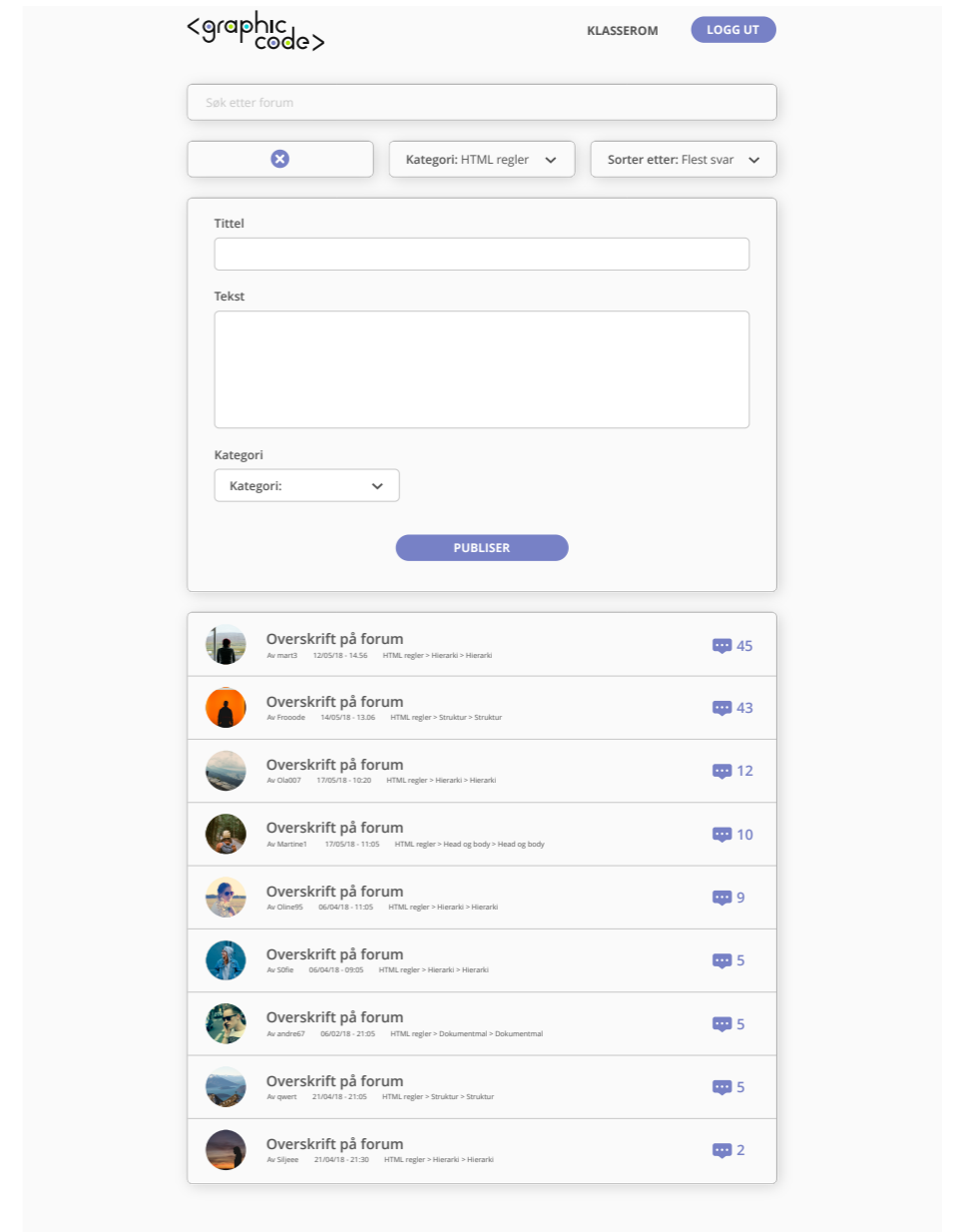
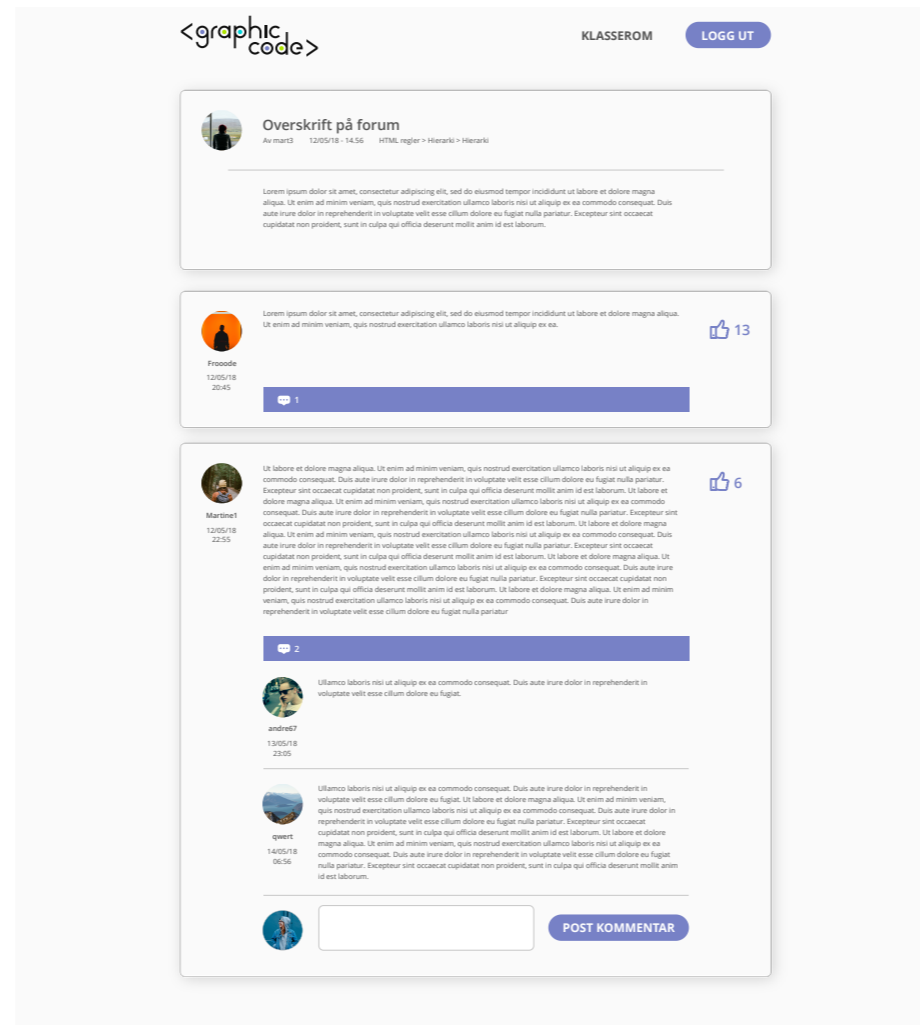
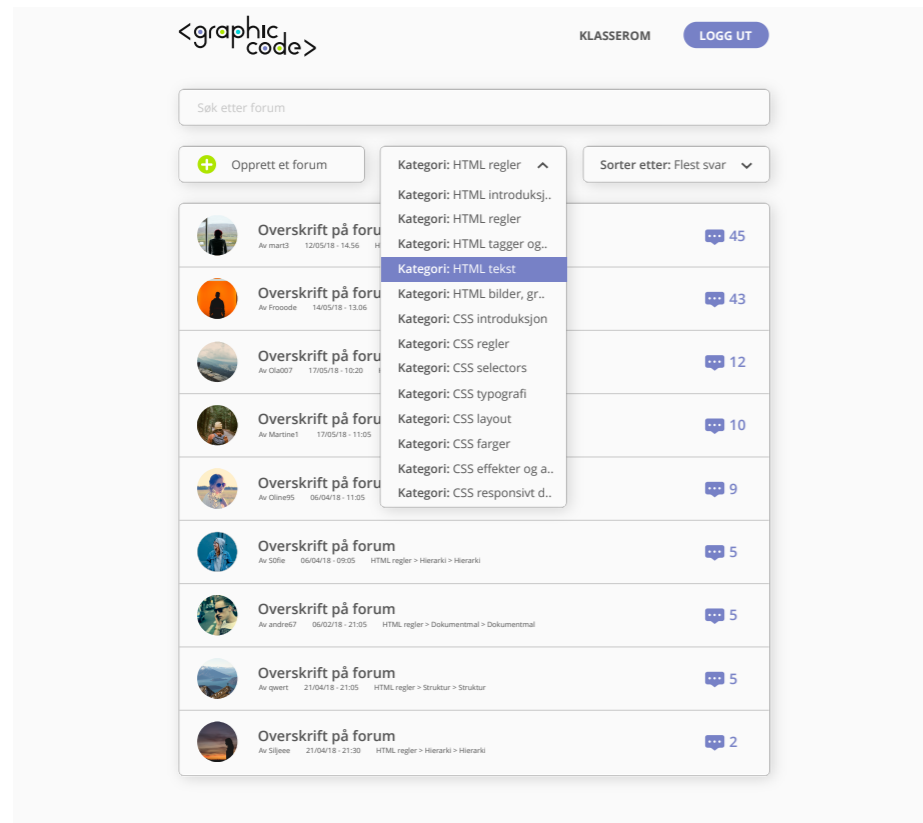
li a {
  display: block;
  color: #000;
  padding: 8px 16px;
  text-decoration: none;
}

li a:hover {
  background-color: #FF6955;
  color: purple;
}
```



På **kursoversikten** til CSS er det åtte deler. Hver del har sin egen illustrasjon som beskriver temaet.

Forum



I **forumet** kan du finne svar på ting man lurer på. Du kan søke etter de forskjellige kursdelene/kategoriene for å finne spørsmål folk stiller om den delen du holder på med.

Du kan også opprette et eget forum om du ikke finner svarene til spørsmålene du har.

The screenshot shows a course page for 'Struktur'. On the left, there's a text area explaining HTML structure and a code editor with HTML code. On the right, a browser preview shows a page with a header containing 'Dette er en overskrift' and a paragraph below it. The code in the editor is as follows:

```

1 <html>
2 <head>
3 </head>
4 <body>
5 <main>
6 <div>
7 
8 <h1>Dette er en overskrift</h1>
9 <p>Dette er en paragraf</p>
10 </div>
11 </main>
12 </body>
13 </html>
    
```

Below the code, there are task instructions: 'Strukturer koden i index.html' and a 'LES KODE' button.

The screenshot shows a course page for 'Sette opp kolonner'. On the left, there's a text area explaining how to set up columns using grid-template-columns and a video player. On the right, a browser preview shows a grid with three columns of different colors (cyan, purple, yellow). The code in the editor is as follows:

```

1 <div class="grid">
2 <div class="box-a">A</div>
3 <div class="box-b">B</div>
4 <div class="box-c">C</div>
5 <div class="box-d">D</div>
6 <div class="box-e">E</div>
7 <div class="box-f">F</div>
8 <div class="box-g">G</div>
9 </div>
    
```

Below the code, there are task instructions: 'Gi gridet to kolonner i hvilken som helst fr' and a 'LES KODE' button.

The screenshot shows a course page for 'La innholdet gå ned på neste linje'. On the left, there's a text area explaining flex-wrap and a code editor. On the right, a browser preview shows a flex container with three boxes labeled 'Box one', 'Box two', and 'Box three'. The code in the editor is as follows:

```

1 <div class="forelder">
2 <div class="barn 1">
3 <h1>Box one</h1>
4 </div>
5
6 <div class="barn 2">
7 <h1>Box two</h1>
8 </div>
9
10 <div class="barn3">
11 <h1>Box three</h1>
12 </div>
13 </div>
    
```

Below the code, there are task instructions: 'Legg til flex-wrap' and a 'LES KODE' button.


Selve kursene er bygd opp av **tre deler**. Den første delen er der all informasjonen du skal lære står. Den andre delen er der du koder for å løse oppgaven(e) du har fått nederst på siden. Den tredje delen er der det du har kodet vises.

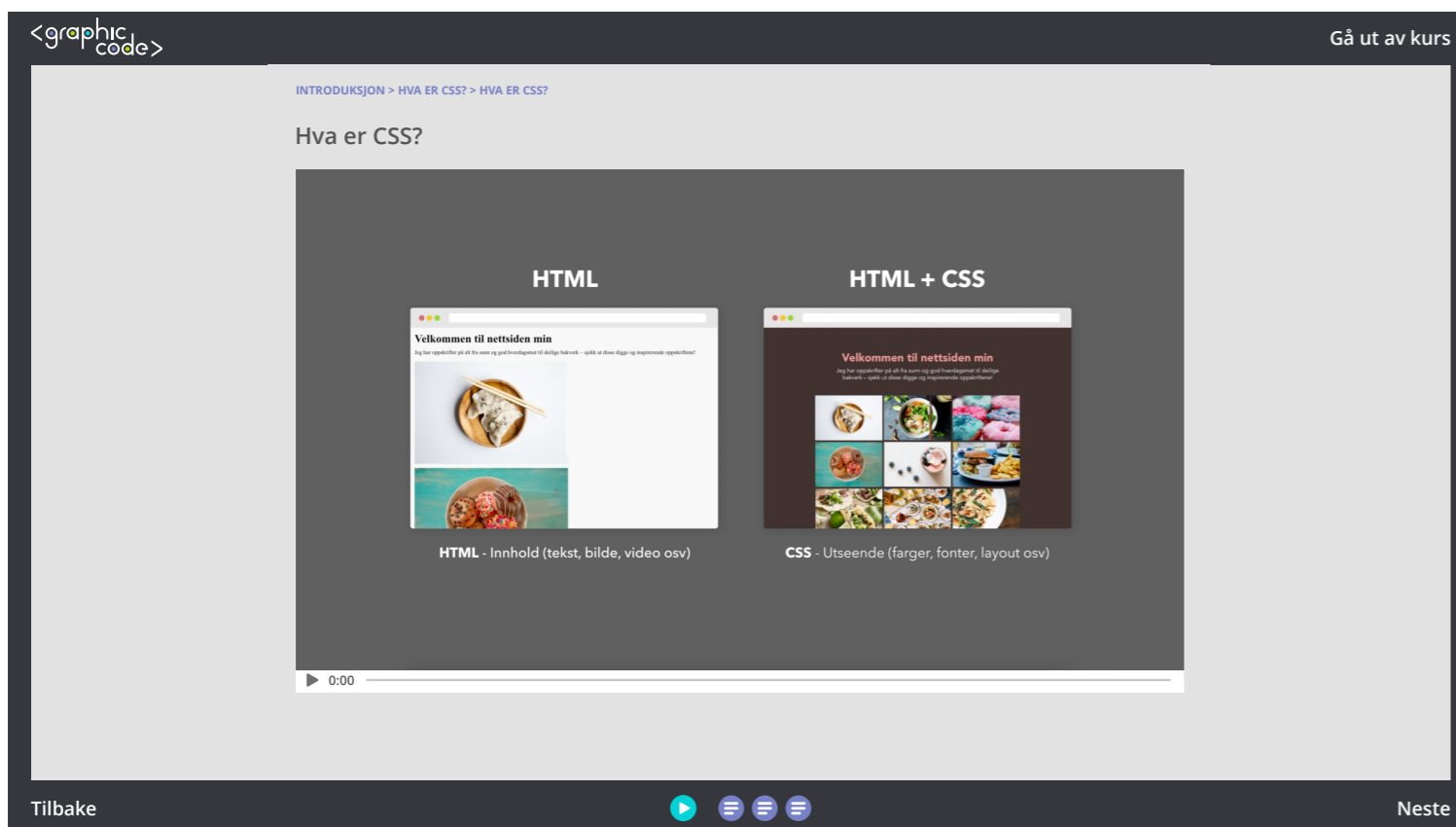
De fleste av sidene i kursene er bygd opp av **korte tekster** med **visualiseringer** og/eller **korte videoer**. På de fleste sidene er det oppgaver du fullfører før du går videre. Om du ikke vil lese teksten, er det et lyd ikon du kan klikke på for at teksten blir lest opp for deg.

The screenshot shows a task list with three items:

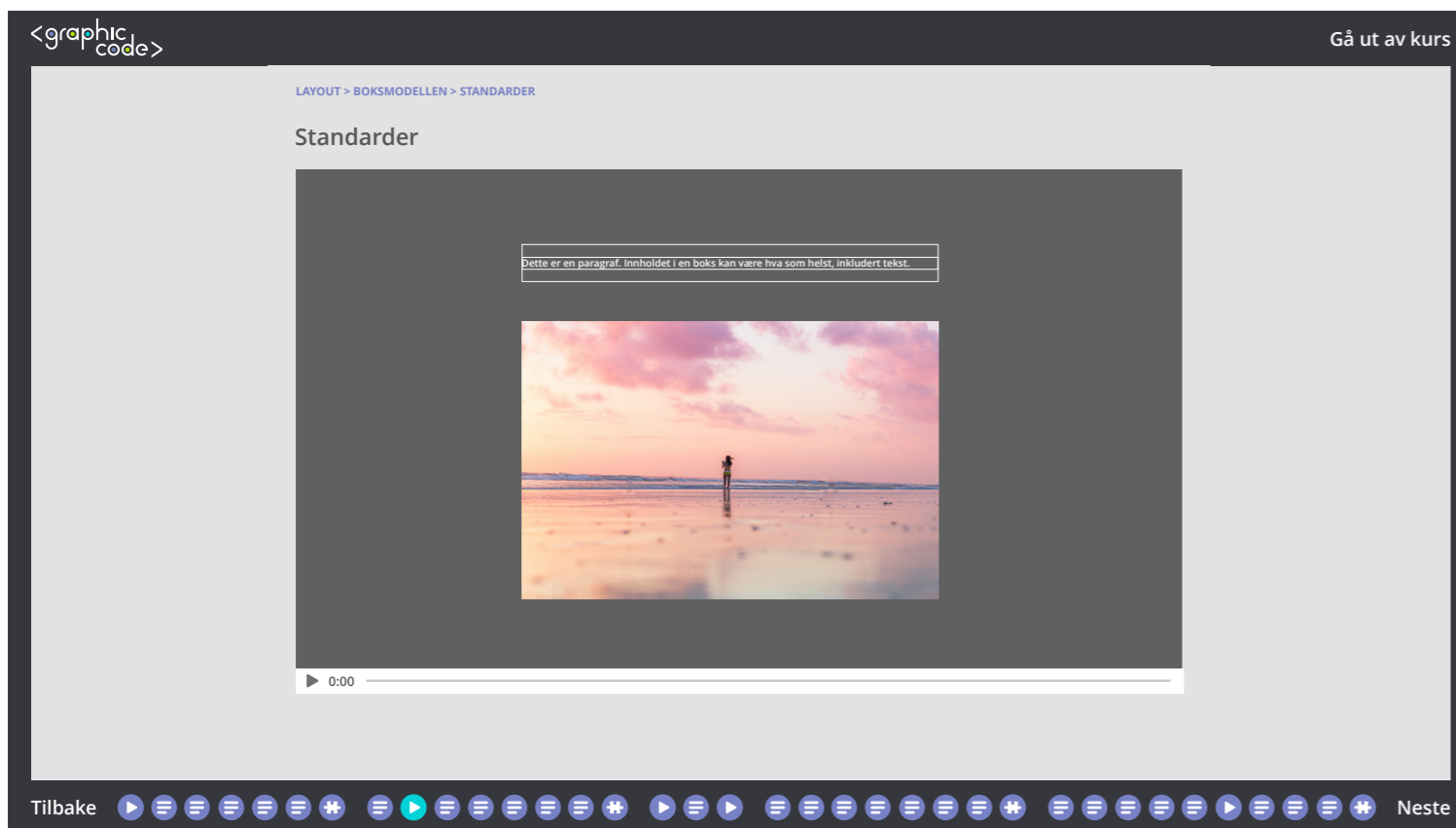
- ✓ Gi en rekkefølge til boks en, to og tre
- ✗ Gi en størrelse til boks en, to og tre HJELP HINT
- ③ Gi en størrelse til boks en, to og tre HINT

Below the list, there is a 'Bruk order:' button.

 Denne typen kursdel er markert med dette ikonet



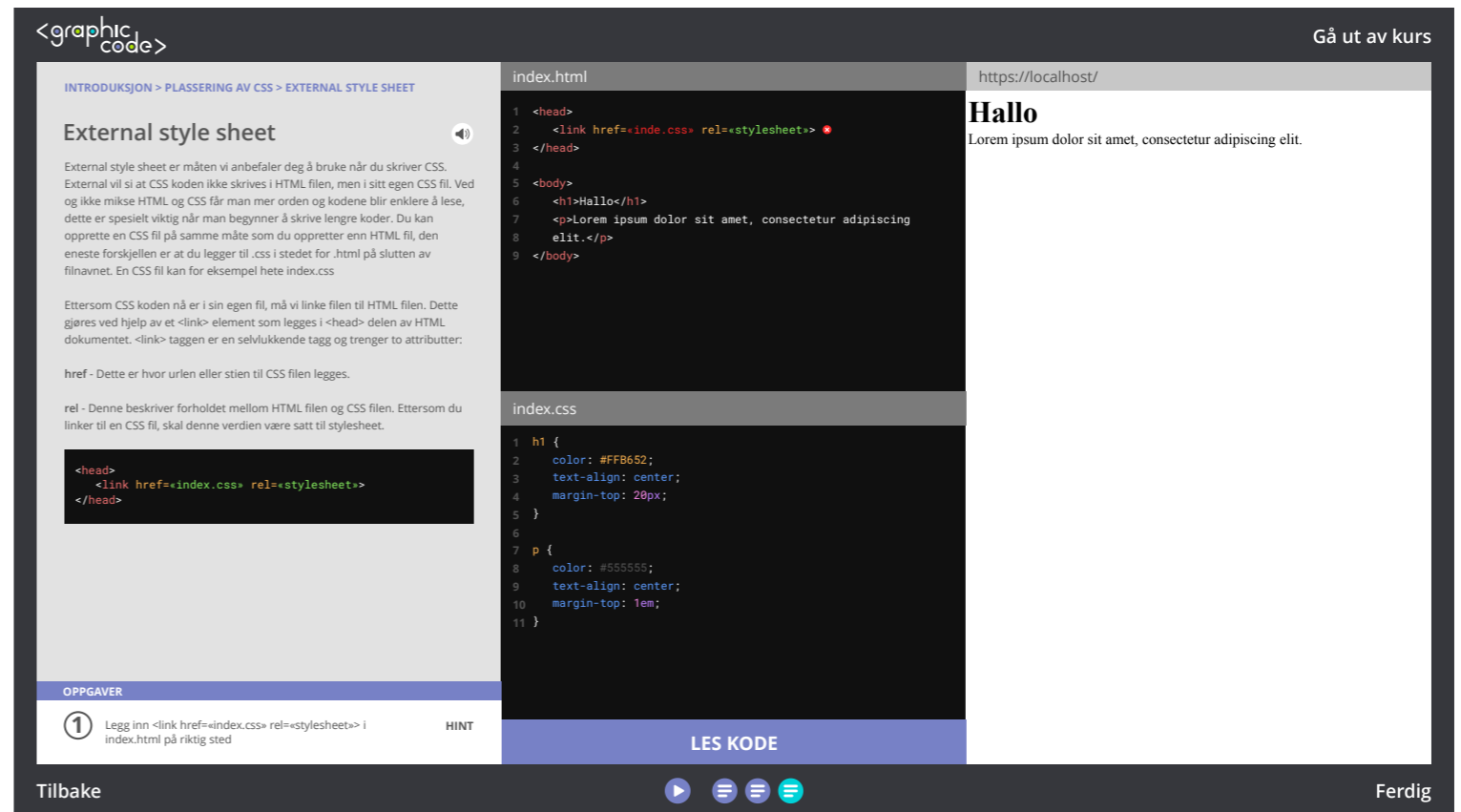
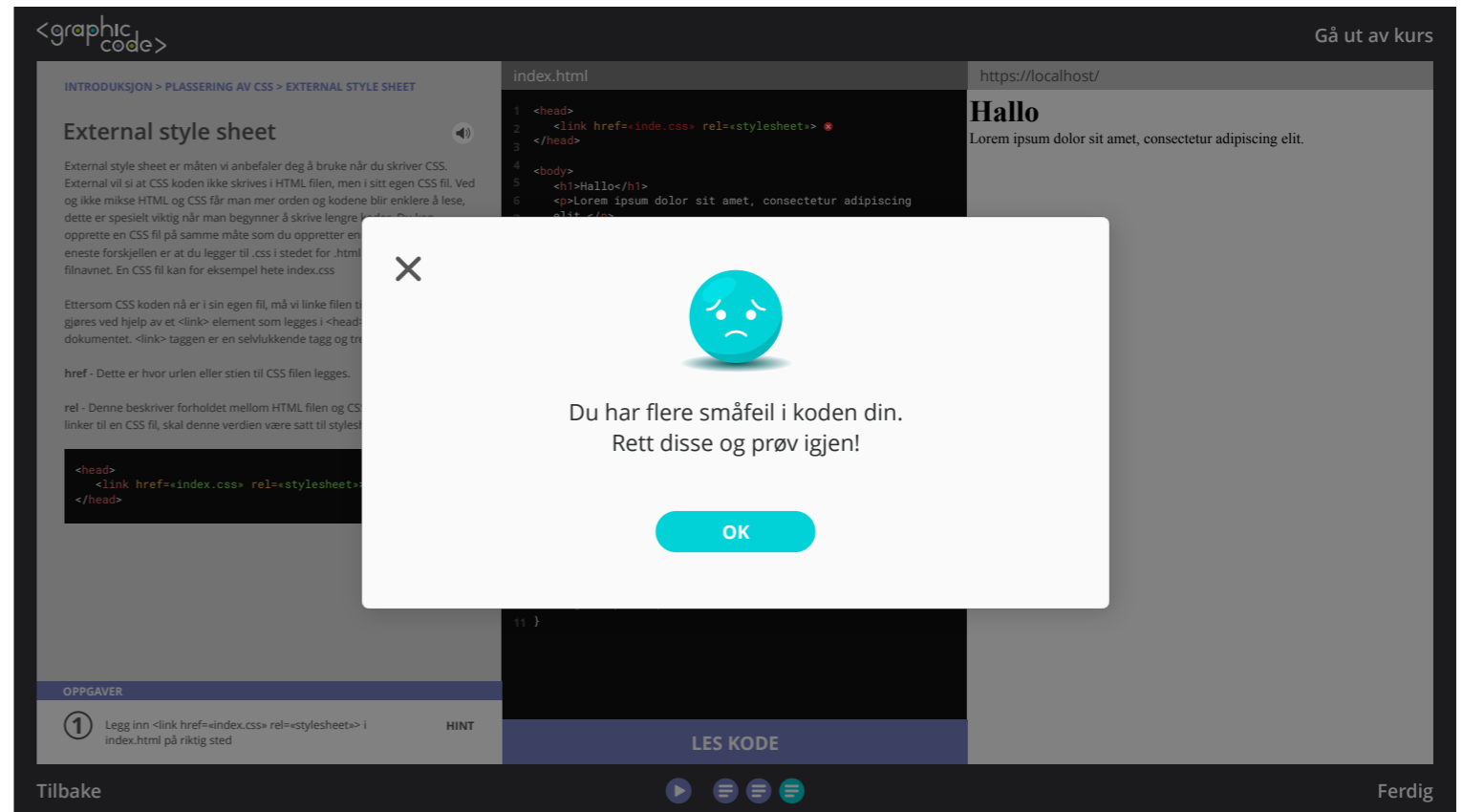
Mesteparten av kursene er bygd opp på den måten du så på forrige side, det gir brukeren mulighet til å finne informasjonen man trenger fort. Det er derimot noen deler av kurset der man bare trenger en forståelse for noe. Det kan være tilleggsinformasjon som ikke er så viktig eller en introduksjon for eksempel. Det er på disse delene det er **videoer**. Videoene er ment som en pause der man kan slappe av å høre på, før man går videre til neste del.



Denne typen kursdel er markert med dette ikonet



Om man har **småfeil** i koden når man trykker på "les kode", får man en melding der man får beskjed om å rette feilene og prøve på nytt. Hvor feilen(e) er vil bli markert med et ikon som viser et rødt kryss, i tillegg vil ordet som er feil være markert med den samme rødfargen. Småfeil vil si at man har skrevet en bokstav/tegn feil, har glemt en bokstav/tegn eller har en bokstav/tegn for mye. Småfeil er vanskelige å oppdage og gjør mange frustrert. Derfor finner plattformen de for deg.



Større feil i koden

index.html

```
1 <div></div>
```

index.css

```
1 div {
2   width: 145px;
3   height: 140px;
4   color: #F07272;
5   padding: 20px;
6   border: 10px solid #F8E129;
7 }
```

Du mangler:

-Margin

[PRØV IJEN](#)

[SE HVA ANDRE HAR SPURT OM DENNE OPPGAVEN I FORUM](#)

Om du har **større feil** i koden får du en melding der det blir vist hvor feilen er og eventuelt hva du mangler for å fullføre oppgaven. Du får mulighet til å prøve igjen etter å ha sett feilen eller så kan du se hva andre personer har spurt angående denne oppgaven i forumet. Om du velger det siste, kommer det opp et forum som bare inneholder forum om den spesifikke oppgaven du holder på med. På denne måten er det lett å finne svar, men om du ikke finner det, har du mulighet til å opprette et forum selv og spør om akkurat det du lurer på.

×

Overskrift på forum
Av andre67 06/02/18 - 21:05 Layout > Boksmodellen > Border 12

Overskrift på forum
Av Froode 14/05/18 - 13:06 Layout > Boksmodellen > Border 8

Overskrift på forum
Av Ola007 17/05/18 - 10:20 Layout > Boksmodellen > Border 5

Overskrift på forum
Av Martine1 17/05/18 - 11:05 Layout > Boksmodellen > Border 2

Overskrift på forum
Av Siljee 21/04/18 - 21:30 Layout > Boksmodellen > Border 0

[+ Opprett et forum](#)

×

Tittel

Tekst

[PUBLISER](#)

×

Overskrift på forum
Av mart3 12/05/18 - 14:56 HTML regler > Hierarki > Hierarki

>Lorem ipsum dolor sit amet, consectetur adipiscing 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.

Lorem ipsum dolor sit amet, consectetur adipiscing 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. [13](#)

Froode
12/05/18
20:45

[1](#)

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. 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. 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. 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. 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. [6](#)

Martine1
12/05/18
22:55

[2](#)

Ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat.

andre67
13/05/18
23:05

Ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat. 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. 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. [0](#)

qwert
14/05/18
06:56


[POST KOMMENTAR](#)

The screenshot shows the 'graphic code' interface. On the left, a task titled 'Utfordringsoppgave' asks to create a box that looks like a specific design. The design is a yellow-bordered box with a red inner box. Dimensions are provided: the yellow box is 110x37, the red box is 140x145, and there is a 20px padding between them. The yellow border has a color of #F8E129 and the red box has a color of #F07272. Below the design, it says 'Verdiene er i px'. The code editors show the following code:

```
index.html
1
2
3

index.css
1
2
3
4
5
6
7
div {
  width: 145px;
  height: 140px;
  color: #F07272;
  padding: 20px;
  border: 10px solid #F8E129;
}
```

På slutten av en del av kurset er det ofte **utfordringsoppgaver**. Utfordringsoppgavene er vanskeligere enn oppgavene på de andre sidene. Utfordringsoppgaven er til for å teste om du har forstått det du har lært så langt i kurset. Når du jobber med en utfordringsoppgave kan du kode og lese koden underveis. Når du føler du er helt ferdig kan du levere oppgaven for å se om du har gjort riktig eller feil. Har du gjort feil, vil du få en melding lignende den på forrige side (Større feil i kode). Om man har gjort riktig får man en melding som sier at du har gjort riktig og at du kan gå videre. På begge meldingene er det en kule som er enten glad eller lei seg. Om meldingen er positiv, er kula grønn, om den er negativ, er kula blå.

 Denne typen kursdel er markert med dette ikonet

Det er dessverre ikke helt riktig

```
index.html
1 <div></div>
2
3

index.css
1 div {
2   width: 145px;
3   height: 140px;
4   color: #F07272;
5   padding: 20px;
6   border: 10px solid #F8E129;
7 }
```

Du mangler:
-Margin

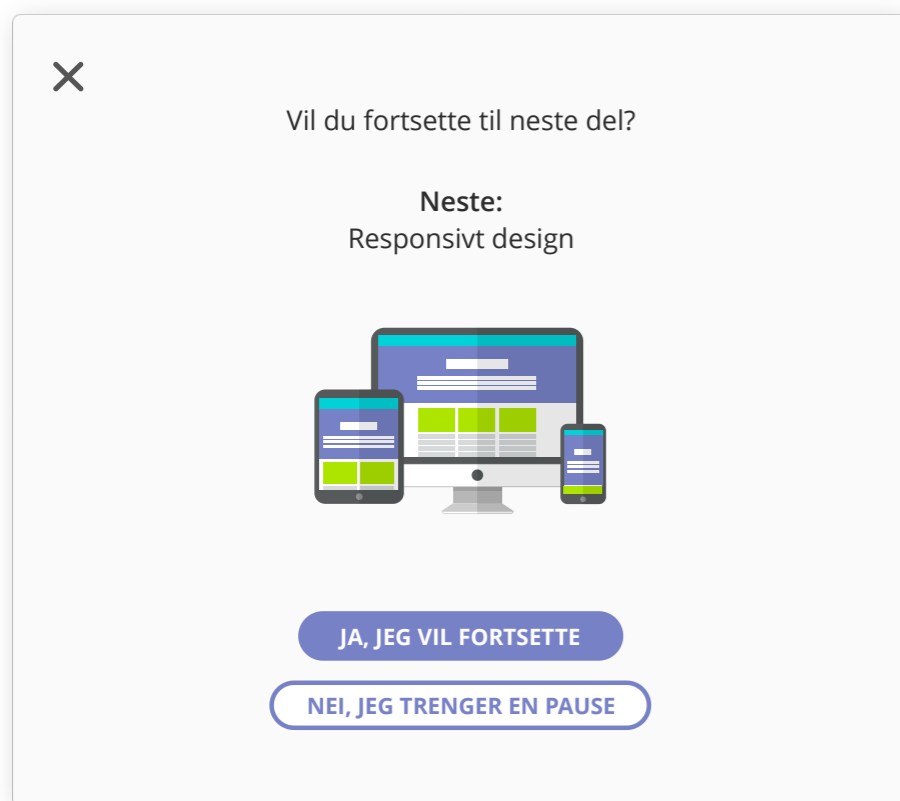
[PRØV IGJEN](#)

[SE HVA ANDRE HAR SPURT OM DENNE OPPGAVEN I FORUM](#)

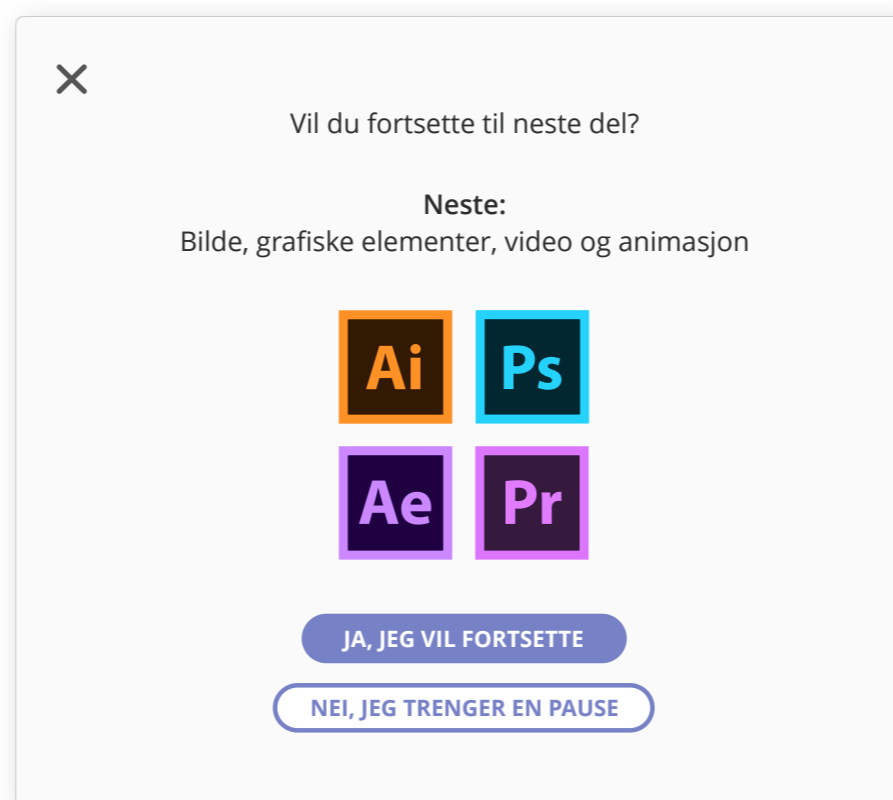
Helt riktig!

[GÅ VIDERE](#)

Neste del melding



Når man er ferdig med en del, får man beskjed om man vil gå videre til **neste del** eller å ta en pause



KURSINNHold

<graphic code> Gå ut av kurs

INTRODUKSJON > HVA ER CSS? > HVA ER CSS?

Hva er CSS?

<graphic code>

0:00

Tilbake Neste

<graphic code> Gå ut av kurs

INTRODUKSJON > Plassering av CSS > INLINE STYLE

Inline style

Det finnes tre forskjellige måter å sette inn et CSS stilark. Inline style blir veldig sjeldent brukt og er ikke til å anbefale. External style sheet derimot er standarden og det vi anbefaler, det er også dette vi vil bruke i dette kurset. Vi ønsker likevel å vise alle de tre forskjellige måtene.

Den første måten er inline, dette innebærer at man kan skrive CSS inne i HTML filen. Man gir et bestemt HTML element en attributt kalt style. Inne i denne attributten kan man fylle inn hvilken som helst CSS property (vi skal lære mer om CSS properties og anatomi senere). En inline style kan bli brukt for å gi en unik stil til et bestemt element.

```
<h1 style=color:blue; margin-left:30px;>This is a heading</h1>
```

Husk å legge til et semikolon på slutten av hver CSS property.

index.html

index.css

https://localhost/

LES KODE

Tilbake Neste

<graphic code> Gå ut av kurs

INTRODUKSJON > Plassering av CSS > INTERNAL STYLE SHEET

Internal style sheet

Internal style sheet er også inne i HTML filen, men i motsetning til inline style, lar den deg style flere HTML elementer på likt. Med internal style sheet kan man skrive CSS koder på en egen del av HTML filen ved å legge til et <style> element som man plasserer på innsiden av <head>.

Slik som dette:

```
<head>
<style>
</style>
</head>
```

Mellom style taggene kan du skrive CSS, slik som dette:

```
<head>
<style>
p {
color: red;
font-size: 20px;
}
</style>
</head>
```

index.html

index.css

https://localhost/

LES KODE

Tilbake Neste

<graphic code> Gå ut av kurs

INTRODUKSJON > Plassering av CSS > EXTERNAL STYLE SHEET

External style sheet

External style sheet er måten vi anbefaler deg å bruke når du skriver CSS. External vil si at CSS koden ikke skrives i HTML filen, men i sitt egen CSS fil. Ved og ikke mikse HTML og CSS får man mer orden og kodene blir enklere å lese, dette er spesielt viktig når man begynner å skrive lengre koder. Du kan opprette en CSS fil på samme måte som du oppretter en HTML fil, den eneste forskjellen er at du legger til .css i stedet for .html på slutten av filnavnet. En CSS fil kan for eksempel hete index.css

Ettersom CSS koden nå er i sin egen fil, må vi linke filen til HTML filen. Dette gjøres ved hjelp av et <link> element som legges i <head> delen av HTML dokumentet. <link> taggen er en selvlukkende tagg og trenger to attributter:

href - Dette er hvor urlen eller stien til CSS filen legges.

rel - Denne beskriver forholdet mellom HTML filen og CSS filen. Ettersom du linker til en CSS fil, skal denne verdien være satt til stylesheet.

```
<head>
<link href=index.css rel=stylesheet>
</head>
```

index.html

index.css

https://localhost/

Hallo
Lorem ipsum dolor sit amet, consectetur adipiscing elit.

LES KODE

OPPGAVER

1 Legg inn <link href=index.css rel=stylesheet> i index.html på riktig sted HINT

Tilbake Ferdig

<graphic code> Gå ut av kurs

LAYOUT > MÅLEENHETER > INTRODUKSJON

Introduksjon

ABSOLUTTE ENHETER:

cm mm pc
in px pt

RELATIVE ENHETER:

em %
vmax
ex
ch vh vmin
rem vw

0:00

Tilbake Neste


<graphic code> Gå ut av kurs

LAYOUT > MÅLEENHETER > RELATIVE ENHETER

Relative enheter

Elementer med en relativ måleenhet får størrelsen sin utifra noen annet. Relative elementer er basert på enten forelder elementets størrelse (%), font størrelsen (em, rem) eller viewport/størrelsen (vh og vw).

I forhold til absolute enheter kan relative måleenheter derimot skaleres opp og ned alt ettersom skjermstørrelsen endres. På denne måten kan man skape design som er responsivt og fungerer bra på alle flater.



0:00

Slik ser en nettside ut når man bruker relative enheter

index.html

index.css

LES KODE

https://localhost/

Tilbake Neste

<graphic code> Gå ut av kurs


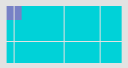
LAYOUT > MÅLEENHETER > ABSOLUTTE ENHETER

Absolutte enheter

Pixel (px) er den eneste absolute enheten vi skal ta for oss i dette kurset. Alle skjermer er bygd opp av små pixler. Om du bruker en enhet med dårlig oppløsning, kan vi tenke oss at et 1px er et pixel på skjermen din. For enheter med høyere oppløsning, slik som en Mac, tilsvarer 1px i CSS flere pixler på skjermen. Et pixel er en absolutt enhet og er konstant, det vil si at 50 pixler er alltid 50 pixler.


height: 1px;
width: 1px;

=

På skjermer med dårlig oppløsning er 1px i CSS det samme som 1px på skjermen. På skjermer med bedre oppløsning er 1px i CSS det samme som flere pixler på skjermen, i dette tilfellet fire.

Det er ikke anbefalt å bruke absolute måleenheter på digitale flater, dette er fordi skjermer varierer veldig i størrelse. Ved å bruke absolute enheter slik som pixler, får vi ikke et responsivt design, altså et design som kan skaleres opp og ned i størrelse og fortsatt se bra ut. For å få til dette bruker vi relative enheter som vi skal lære om i neste del.



0:04

Slik ser en nettside ut når man bruker absolute enheter

index.html

```
1 <div class="boks"></div>
```

https://localhost/

index.css

```
1 .boks {
2   color: purple;
3 }
```

OPPGAVER

1 Gi boksen en bredde på 20px og en høyde på 40px HINT

LES KODE

Tilbake Neste

<graphic code> Gå ut av kurs

LAYOUT > MÅLEENHETER > EM

Em

Måleenheten em er basert på font størrelsen til det nærmeste forelder elementet.

```
<div class="forelder">
  <p>Hallo</p>
</div>
```

```
.forelder {
  font-size: 20px;
}
p {
  font-size: 1em;
}
```

I dette eksempelet vil 1em tilsvare 20px fordi .forelder er p sin nærmeste forelder. Om man ikke legger til en font størrelse til forelder elementet vil 1em tilsvare 16px, dette er standarden.

STANDARD	FORELDER = 10PX
1em = 16px	1em = 10px
2em = 32px	2em = 20px
OSV...	OSV...

OPPGAVER

- 1 Sett font størrelsen til p lik 2em og se hvordan font størrelsen dobles

HINT

LES KODE

```
index.html
```

```
1 <p>Jeg er en tekst</p>
```

https://localhost/ Jeg er en tekst

```
index.css
```

```
1 p {
2 }
```

Tilbake NESTE

<graphic code> Gå ut av kurs

LAYOUT > MÅLEENHETER > REM

Rem

Rem fungerer på nesten samme måte som em. Rem står for «root em» og baseres på rot elementet sin font størrelse. Vi legger til en font størrelse på rot elementet ved å skrive det under .root slik som dette:

```
:root {
  font-size: 20px;
}
```

Nå vil alle CSS elementene du skriver rem på, være relativ til denne font størrelsen. Dette er annerledes fra em, der den baseres på nærmeste forelder.

ROT ELEMENTET = 20PX

1rem = 20px
2rem = 40px
3rem = 60px

OPPGAVER

- 1 Sett rot elementet til 10px
- 2 Gi em verdien som samsvarer med 80px til p elementet

HINT

LES KODE

```
index.html
```

```
1 <p>Jeg er en tekst</p>
```

https://localhost/ Jeg er en tekst

```
index.css
```

```
1 :root {
2 }
3
4 p {
5 }
```

Tilbake NESTE

<graphic code> Gå ut av kurs

LAYOUT > MÅLEENHETER > VH OG VW

Vh og vw

Vh og vw står for viewport height og viewport width, altså høyden og bredden på skjermen til enheten vår.

Høyde og bredde verdien er 100 totalt.

1vw = 1% av skjerm bredden totalt
1vh = 1% av skjerm høyden totalt

OPPGAVER

- 1 Få boks til å ta opp halvparten av skjerm bredden

HINT

LES KODE

```
index.html
```

```
1 <div class="boks"></div>
```

https://localhost/

```
index.css
```

```
1 .boks {
2   color: purple;
3   height: 50px;
4 }
```

Tilbake NESTE

<graphic code> Gå ut av kurs

LAYOUT > MÅLEENHETER > %

%

Vi bruker prosent når vi har et element som er relatert til et annet, slik som en forelder og barn. Vi kan si at barnet skal være så og så mange prosent av forelder elementet.

```
.barnebarn {
  width: 25%;
}
.forelder {
  width: 100vw;
}
```

OPPGAVER

- 1 Gi en bredde til forelder elementet
- 2 Gi forskjellige bredde verdier til barnet og barnebarnet og se hva som skjer

HINT

LES KODE

```
index.html
```

```
1 <div class="forelder"></div>
2
3 <div class="barn"></div>
4
5 <div class="barnebarn"></div>
```

https://localhost/

```
index.css
```

```
1 .forelder {
2   color: #002D88;
3   height: 50px;
4 }
5
6 .barn {
7   color: #AEE500;
8   height: 50px;
9 }
10
11 .barnebarn {
12   color: #778106;
13   height: 50px;
14 }
```

Tilbake NESTE

Introduksjon

Alle HTML elementer er bokser. I CSS bruker vi navnet «boksmodellen» når vi stiler disse boksene. CSS-boksmodellen er i hovedsak en boks som går rundt hvert HTML-element. Den består av: Selve innholdet, padding, border og margin.

Innhold - Selve innholdet i boksen, dette er hvor teksten og bildene er. For å spesifisere størrelsen til innholdet bruker man `height` og `width`.

Padding - Dette er det gjennomsiktige område mellom innholdet og border. Padding skaper luft mellom innholdet og rammen.

Border - Dette er en ramme som går utenfor padding. Rammen er synlig og kan designes på mange måter.

Margin - Dette er det gjennomsiktige området utenfor rammen. Margin skaper luft mellom flere ulike elementer så de ikke går inn i hverandre.

LES KODE

Standarder

Dette er en paragraf. Innholdet i en boks kan være hva som helst, inkludert tekst.

Tilbake **Neste**

Innhold

Alle elementers innhold har en høyde og en bredde. Som standard er disse målene satt til å romme akkurat det innholdet som er inni. Men vi kan gjøre endringer ved å sette en ny høyde verdi og bredde verdi på et element.

```

p {
  width: 300px;
  height: 100px;
}
    
```

```

1 ul {
2   background-color: midnightblue;
3   margin: 0;
4   padding: 0;
5 }
6
7 li {
8   text-decoration: none;
9   text-align: center;
10  background-color: powderblue;
11  color: black;
12 }
    
```

OPPGAVER

1 Gi en 100px bredde og 20px høyde til li **HINT**

LES KODE

Border

En border er en line som går rundt et element, som en ramme. Borders kan ha en width, style og color.

Width - Er tykkelsen på rammen. Kan være enten thin, medium, thick eller ha en verdi i pixeler.

Style - Er designet på rammen. Det er 10 forskjellige stiler å velge mellom: none, hidden, dotted, dashed, solid, double, groove, ridge, inset og outset.

Color - Er fargen på rammen. Det finnes flere forskjellige måter å bruke farger på i CSS, en av de er 140 innebygde fargekoder på CSS som du kan finne HER.

```

p {
  border: 3px solid coral;
}
    
```

I dette eksempelet har border en tykkelse på 3px, har en solid stil og en coral farge.

```

1 ul {
2   background-color: midnightblue;
3   margin: 0;
4   padding: 0;
5 }
6
7 li {
8   text-decoration: none;
9   text-align: center;
10  background-color: powderblue;
11  color: black;
12  height: 20px;
13  width: 100px;
14 }
    
```

OPPGAVER

1 Gi li en border med alle tre verdiene, du kan selv velge hvilke **HINT**

LES KODE

<graphic code> Gå ut av kurs

LAYOUT > BOKSMODELLEN > PADDING

Padding

Rammen er nå helt inn til innholdet, vi trenger derfor å legge på litt padding for å legge til mer luft i designet.

```

1 {
2   padding: 20px;
3 }
    
```

Denne koden legger til 20px på hver side av boksen, på toppen, bunnen, til høyre og til venstre.

Om du ønsker å bare ha padding på en side, kan vi bruke disse:

```

1 {
2   padding-top: 10px;
3   padding-right: 5px;
4   padding-bottom: 6px;
5   padding-left: 20px;
6 }
    
```

Om man derimot ønsker å ha padding på alle sidene, men forskjellige verdier, finnes det også en måte å gjøre dette på:

```

1 {
2   padding: 40px 50px 30px 10px;
3 }
    
```

I dette eksempelet er 40px topp, 50px er høyre, 30px er bunn og 10px er venstre. Retningen følger klokken.

```

1 <ul>
2   <li><a href=#Hjem>Hjem</a></li>
3   <li><a href=#Galleri>Galleri</a></li>
4   <li><a href=#Om oss>Om oss</a></li>
5   <li><a href=#Kontakt>Kontakt</a></li>
6 </ul>
    
```

```

1 ul {
2   background-color: midnightblue;
3   margin: 0;
4   padding: 0;
5 }
6
7 li {
8   text-decoration: none;
9   text-align: center;
10  background-color: powderblue;
11  color: black;
12  height: 20px;
13  width: 100px;
14  border: 2px solid white;
15 }
    
```

OPPGAVER HINT

1 Legg til padding på li

LES KODE

Tilbake Neste

<graphic code> Gå ut av kurs

LAYOUT > BOKSMODELLEN > MARGIN

Margin

Sist men ikke minst kommer margin. Margin er luften som ligger utenpå boksen, som lar deg definere hvor stort mellomrom det skal være mellom forskjellige elementer.

```

1 {
2   margin: 30px;
3 }
    
```

Margin fungerer på samme måte som padding, ved at man enten kan legge til samme margin på alt, eller spesifisere.

```

1 {
2   margin-top: 10px;
3   margin-right: 5px;
4   margin-bottom: 6px;
5   margin-left: 20px;
6 }
    
```

```

1 {
2   margin: 40px 50px 30px 20px;
3 }
    
```

Det som er forskjellen mellom padding og margin er derimot at det er mer vanlig å ha margin bare på noen sider og ikke alle.

```

1 ul {
2   background-color: midnightblue;
3   margin: 0;
4   padding: 0;
5 }
6
7 li {
8   text-decoration: none;
9   text-align: center;
10  background-color: powderblue;
11  color: black;
12  height: 20px;
13  width: 100px;
14  border: 2px solid white;
15  padding: 20px;
16 }
    
```

OPPGAVER HINT

1 Legg på margin på li slik at knappene får et mellomrom mellom seg

LES KODE

Tilbake Neste

<graphic code> Gå ut av kurs

LAYOUT > BOKSMODELLEN > HVORDAN REGNE UT TOTAL STØRRELSE

Hvordan regne ut total størrelse

Når du gir en høyde og bredde til et element, er det bare innholdet i boksen som vil bli denne størrelsen. For å regne ut den totale størrelsen på et element, må man derfor også inkludere padding, borders og margins.

La oss si at vi vil at <div> elementet skal ha en total bredde på 400px. Da kan vi sette opp noe slikt:

```

1 div {
2   width: 280px;
3   padding: 20px;
4   border: 10px solid blue;
5   margin: 30px;
6 }
    
```

```

1 <div class=boks></div>
    
```

```

1 .boks {
2   color: purple;
3   height: 50px;
4 }
    
```

OPPGAVER HINT

1 Gjør at total bredden til boksen blir 300px. Både width, padding og border må ha en verdi

LES KODE

Tilbake Neste

<graphic code> Gå ut av kurs

LAYOUT > BOKSMODELLEN > UTFORDRINGSOPPGAVE

Utfordringsoppgave

Lag en boks som ser ut som dette:

Verdiene er i px

```

1
    
```

```

1
    
```

OPPGAVER HINT


LES KODE **LEVER OPPGAVE**

Tilbake Neste

<graphic code> Gå ut av kurs

LAYOUT > POSISJONERING > INTRODUKSJON

Introduksjon



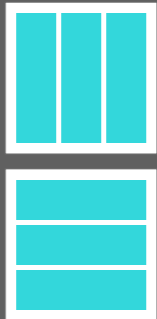
0:00

Tilbake Neste


<graphic code> Gå ut av kurs

LAYOUT > POSISJONERING > FLEX BOX VS GRID

Flex box vs grid



Flex box - Er best å bruke for å posisjonere elementer på et endimensjonalt layout, altså elementer som er plassert på en rad ELLER kolonner.



CSS grid - Er best å bruke for å lage layout til en hel side å inneholde alle delene av en nettside. Grid kan posisjonere elementer på et todimensjonalt layout med både rader OG kolonner.

0:00

Tilbake Neste


<graphic code> Gå ut av kurs

LAYOUT > POSISJONERING > DISPLAY


Display

Det første vi skal gå gjennom en display. Display er en property som beskriver hva slags type oppførsel et element skal ha. Vi skal gå gjennom tre forskjellige typer verdier; block, inline og inline-block. HTML elementer har en standard display verdi, paragrafer er for eksempel block elementer og linker er inline elementer. Vi kan endre display proprietien til et hvert element for å endre hvordan de oppfører seg.


Block - Block elementer starter på en ny linje og tar opp hele bredden. Se alle block elementer HER.



Inline - Inline elementer starter ikke på en ny linje og tar bare opp så mye høyde og bredde som den trenger. Se alle inline elementer HER.



Inline-block - Inline-block er en blanding mellom de to andre. Forskjellen mellom inline-block og block elementer er at inline-block elementer ikke starter på en ny linje.



index.html

```


1 <ul>
2 <li><a href=#hjem>Hjem</a></li>
3 <li><a href=#galleri>Galleri</a></li>
4 <li><a href=#omoss>Om oss</a></li>
5 <li><a href=#kontakt>Kontakt</a></li>
6 </ul>
                
```

index.css

```

1 ul {
2   background-color: midnightblue;
3   margin: 0;
4   padding: 0;
5 }
6
7 li {
8   text-decoration: none;
9   text-align: center;
10  background-color: powderblue;
11  color: black;
12  height: 28px;
13  width: 100px;
14  border: 2px solid white;
15  padding: 20px;
16  margin: 30px;
17 }
                
```

https://localhost/



OPPGAVER

1 Gi knappene en display verdi av inline-block og se hva som skjer. HINT

LES KODE

Tilbake Neste

Sette opp en flex box

For å lage en flex box, trenger du et forelder element med barn inni. Vi har satt opp dette for deg i index.html. Her har vi tre <div> elementer inne i forelder elementet. Hver div har en h1 og en paragraf.

For å gjøre dette om til flexbox, trenger vi å legge til display: flex; i forelder elementet, slik som dette:

```

forelder {
  display: flex;
}
    
```

index.html

```

1 <div class=forelder>
2 <div class=barn 1>
3 <h1>Box one</h1>
4 </div>
5
6 <div class=barn 2>
7 <h1>Box two</h1>
8 </div>
9
10 <div class=barn3>
11 <h1>Box three</h1>
12 </div>
13 </div>
    
```

index.css

```

1 .forelder {
2 }
    
```

https://localhost/

Box one
Box two
Box three

OPPGAVER
1 Gi forelder elementet display: flex; HINT
LES KODE

Størrelse

Det neste vi ønsker å gjøre er å bestemme hvor mye horisontal plass hvert element skal ta, dette gjør vi ved hjelp av flex propertien vi legger til på de forskjellige box elementene.

```

.barn1 {
  flex: 1;
}
.barn2 {
  flex: 1;
}
.barn3 {
  flex: 1;
}
    
```

index.css

```

1 .forelder {
2   display: flex;
3 }
4
5 .barn1 {
6   background-color: yellow;
7 }
8
9 .barn2 {
10  background-color: red;
11 }
12
13 .barn3 {
14  background-color: green;
15 }
    
```

https://localhost/

Box one Box two Box three

OPPGAVER
1 Gi en størrelse til de tre boksene HINT
LES KODE

Rekkefølge

Vi kan endre rekkefølgen på elementene ved å bruke order propertien. Order: 1 vil si det første elementet og order:3 er det siste. Dette er nyttig fordi vi trenger ikke lenger å legge elementene i riktig rekkefølge i HTML dokumentet.

```

.barn1 {
  flex: 1;
  background-color: blue;
  order: 2;
}
.barn2 {
  flex: 1;
  background-color: purple;
  order: 3;
}
.barn3 {
  flex: 1;
  background-color: green;
  order: 1;
}
    
```

index.html

```

1 <div class=forelder>
2 <div class=barn 1>
3 <h1>Box one</h1>
4 </div>
5
6 <div class=barn 2>
7 <h1>Box two</h1>
8 </div>
9
10 <div class=barn3>
11 <h1>Box three</h1>
12 </div>
13 </div>
    
```

index.css

```

1 .forelder {
2   display: flex;
3 }
4
5 .barn1 {
6   background-color: yellow;
7   flex: 1;
8 }
9
10 .barn2 {
11  background-color: red;
12  flex: 1;
13 }
14
15 .barn3 {
16  background-color: green;
17  flex: 1;
18 }
    
```

https://localhost/

Box one Box two Box three

OPPGAVER
1 Gi en rekkefølge til de tre boksene HINT
LES KODE

Fra rader til kolonner

Flex box gir deg rader som standard. Ved å sette verdien på flex-direction til column gjør det at du endrer fra rader til kolonner. Denne legges inn under forelder elementet.

```

forelder {
  flex-direction: column;
}
    
```

index.html

```

1 <div class=forelder>
2 <div class=barn 1>
3 <h1>Box one</h1>
4 </div>
5
6 <div class=barn 2>
7 <h1>Box two</h1>
8 </div>
9
10 <div class=barn3>
11 <h1>Box three</h1>
12 </div>
13 </div>
    
```

index.css

```

1 .forelder {
2   display: flex;
3 }
4
5 .barn1 {
6   background-color: yellow;
7   flex: 1;
8   order: 2;
9 }
10
11 .barn2 {
12  background-color: red;
13  flex: 1;
14  order: 3;
15 }
16
17 .barn3 {
18  background-color: green;
19  flex: 1;
20  order: 1;
21 }
    
```

https://localhost/

Box one Box two Box three

OPPGAVER
1 Gjør om fra rader til kolonner HINT
LES KODE

Mellomrom

For å få mellomrom mellom boksene kan vi i stedet for å skrive flex: 1 på hvert element, skrive flex-basis:

```
.barn1 {
  flex-basis: 30%;
}
```

Flex-basis fungerer på samme måte som vanlig prosent.

For å få mellomrom, legger vi inn justify-content i forelder elementet med space-between som verdi.

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

Med justify-content propertyen kan vi også bruke verdiene flex-start, flex-end, center og space-around. Disse fordeler innholdet på andre måter.

OPPGAVER

- Gjør om barne elementene til flex-basis: 30%; **HINT**
- Gjør at det blir mellomrom jevnt mellom elementene **HINT**
- Prøv ut de forskjellige justify-content property verdiene **HINT**

index.html

```
1 <div class="forelder">
2   <div class="barn 1">
3     <h1>Box one</h1>
4   </div>
5   <div class="barn 2">
6     <h1>Box two</h1>
7   </div>
8   <div class="barn3">
9     <h1>Box three</h1>
10  </div>
11 </div>
```

index.css

```
1 .forelder {
2   display: flex;
3 }
4
5 .barn1 {
6   background-color: yellow;
7   flex: 1;
8   order: 2;
9 }
10
11 .barn2 {
12  background-color: red;
13  flex: 1;
14  order: 3;
15 }
16
17 .barn3 {
18  background-color: green;
19  flex: 1;
20  order: 1;
21 }
```

https://localhost/

Box one
Box two
Box three

LES KODE

La innholdet gå ned på neste linje

Om vi ikke spesifiserer noe, vil flex box gjøre at elementene dine er på linje selv om skjermen blir så liten at alt går inn i hverandre. For å gjøre at elementene går ned på linjen under, bruker vi flex-wrap: wrap som legges i forelder elementet.

Standard:

```
.forelder {
  display: flex;
  flex-wrap: nowrap;
}
```

```
.forelder {
  display: flex;
  flex-wrap: wrap;
}
```

index.html

```
1 <div class="forelder">
2   <div class="barn 1">
3     <h1>Box one</h1>
4   </div>
5   <div class="barn 2">
6     <h1>Box two</h1>
7   </div>
8   <div class="barn3">
9     <h1>Box three</h1>
10  </div>
11 </div>
```

index.css

```
1 .forelder {
2   display: flex;
3   justify-content: space-between;
4 }
5
6 .barn1 {
7   background-color: yellow;
8   flex-basis: 30%;
9   order: 2;
10 }
11
12 .barn2 {
13  background-color: red;
14  flex-basis: 30%;
15  order: 3;
16 }
```

https://localhost/

Box one
Box two
Box three

OPPGAVER

- Legg til flex: wrap; **HINT**

LES KODE

Utfordringsoppgave

Sett opp en flexboks som ser ut som dette:

index.html

```
1
```

index.css

```
1
```

LES KODE **LEVER OPPGAVE**

Sette opp et grid

Det er mye å lære innenfor CSS grid, i denne delen går vi gjennom det grunnleggende. Om du ønsker, kan du utforske CSS grid mer på egen hånd.

For å sette opp et grid, trenger du å sette opp et forelder element med barn inni i HTML. Se HTML filen der vi har satt dette opp for deg.

Det neste vi må gjøre er å gi forelder elementet en property kalt display, med grid som verdi.

```

index.html
1 <div class="grid">
2 <div class="box-a">A</div>
3 <div class="box-b">B</div>
4 <div class="box-c">C</div>
5 <div class="box-d">D</div>
6 <div class="box-e">E</div>
7 <div class="box-f">F</div>
8 <div class="box-g">G</div>
9 </div>

index.css
1 .grid {
2 }
3
4 .grid div:nth-child(even) {
5   background-color: blue;
6 }
7
8 .grid div:nth-child(odd) {
9   background-color: green;
10 }
    
```

OPPGAVER

1 Gi grid elementet en display property satt til grid

HINT

LES KODE

Sette opp kolonner

Vi kan starte med å sette opp kolonner. For å gjøre dette legger vi inn grid-template-columns under forelder elementet grid.

```

index.html
1 <div class="grid">
2 <div class="box-a">A</div>
3 <div class="box-b">B</div>
4 <div class="box-c">C</div>
5 <div class="box-d">D</div>
6 <div class="box-e">E</div>
7 <div class="box-f">F</div>
8 <div class="box-g">G</div>
9 </div>

index.css
1 .grid {
2   display: grid;
3 }
4 .grid div:nth-child(even) {
5   background-color: blue;
6 }
7
8 .grid div:nth-child(odd) {
9   background-color: green;
10 }
11 }
    
```

OPPGAVER

1 Gi griddet to kolonner i hvilken som helst fr

HINT

LES KODE

Snarvei

I stedet for å skrive disse lange kodene, finnes det måter å forkorte de. I stedet for å bruke grid-template-columns og grid-template-rows, kan vi bruke grid-template, som samler begge to, slik som dette:

```

index.html
1 <div class="grid">
2 <div class="box-a">A</div>
3 <div class="box-b">B</div>
4 <div class="box-c">C</div>
5 <div class="box-d">D</div>
6 <div class="box-e">E</div>
7 <div class="box-f">F</div>
8 <div class="box-g">G</div>
9 </div>

index.css
1 .grid {
2   display: grid;
3   grid-template-columns: 1fr 1fr 1fr;
4 }
    
```

I tilfeller der alle kolonnene er like store, kan vi gjøre det enda mer kompakt ved å bruke repeat.

```

index.css
1 .grid {
2   display: grid;
3   grid-template-columns: repeat(3, 1fr);
4 }
    
```

Ved å skrive repeat, etterfulgt av hvor mange ganger man vil at det skal gjentas (her er det 3) og hvor store delene er (her er det 1fr), får man en mindre kode å forholde seg til.

OPPGAVER

1 Gjør om koden så den blir mer kompakt

HINT

LES KODE

Mellomrom

I layouten vi har laget så langt mangler vi en viktig del, mellomrommene mellom elementene våre. Dette kan vi fikse lett med grid-gap, denne legges også til forelder elementet. For å bare ha mellomrom mellom enten kolonner eller rader, bruk grid-row-gap og grid-column-gap i stedet.

grid-gap: 10px; - alle mellomrommene mellom rader og kolonner er 10px

grid-gap: 10px 30px; - radene vil ha 10px og kolonnene vil ha 30px

grid-column-gap: 10px - alle mellomrommene mellom kolonnene vil være 10px

grid-row-gap: 10px - alle mellomrommene mellom radene vil være 10px

```

index.html
1 <div class="grid">
2 <div class="box-a">A</div>
3 <div class="box-b">B</div>
4 <div class="box-c">C</div>
5 <div class="box-d">D</div>
6 <div class="box-e">E</div>
7 <div class="box-f">F</div>
8 <div class="box-g">G</div>
9 </div>

index.css
1 .grid {
2   display: grid;
3   grid-template-columns: repeat(3, 1fr);
4 }
    
```

OPPGAVER

1 Legg til et 20px mellomrom mellom både radene og kolonnene i gridet.

HINT

LES KODE

Grid elementer

0:00

Tilbake NESTE

Flere rader

En måte å få et element til å ta opp flere rader på er ved hjelp av grid-row. Derne blir lagt til i hvert enkelt element.

```

    .box-a {
      grid-row: 1 / 3;
    }
  
```

Grid-row har to tall. Det første tallet er hvor raden starter, det andre tallet etter skråstreken er hvor raden slutter.

grid-row: 1 / 3;

0:04

Se for deg at gridet har usynlige linjer slik som dette. Når man velger for eksempel 1/3, går gridet fra søk 1 til søk 3, noe som utgjør 2-bokser.

OPPGAVER

- 1 Velg en av boksene og få den til å ta opp minst to rader.

HINT

LES KODE

```

    index.html
    1 <div class=grid>
    2 <div class=box-a><A</div>
    3 <div class=box-b><B</div>
    4 <div class=box-c><C</div>
    5 <div class=box-d><D</div>
    6 <div class=box-e><E</div>
    7 <div class=box-f><F</div>
    8 <div class=box-g><G</div>
    9 </div>

    index.css
    1 .grid {
    2   display: grid;
    3   grid-template-columns: repeat(2, 1fr);
    4   grid-gap: 20px;
    5 }
  
```

Tilbake NESTE

Grid area

I stedet for å bruke grid-row og grid-column for å spesifisere hvor mye plass hvert element skal ta, kan vi bruke grid-area. Grid area er en effektiv måte du kan lage et layout for alle delene av nettsiden.

1. Start med å dele opp nettsiden i forskjellige deler i HTML, slik vi har gjort for deg.
2. Deretter legger vi inn display: grid; til forelder elementet grid, slik vi har gjort tidligere.
3. Så legger vi inn proprietien «grid-area» til alle elementene vi har inni gridet. Verdien skal være det du ønsker å kalle dette elementet, det er altså best å bruke klassenavnet du allerede har gitt den. Gjør dette på alle elementene.

```

    .box-a {
      grid-area: box-a;
    }
    .box-b {
      grid-area: box-b;
    }
    osv...

    .grid {
      display: grid;
      grid-template-columns: 1fr 1fr 1fr 1fr;
      grid-auto-rows: minmax(100px, auto);
      grid-template-areas:
    }
  
```

Nå har vi opprettet de forskjellige grid areas (grid områdene) og det neste vi skal gjøre å fordele de rundt om i gridet der vi ønsker. Dette gjør vi ved å plassere grid-template-areas: under forelder elementet.

Verdiene vi skal putte inn i denne proprietien beskriver hvordan vi vil at layouten skal se ut. Ettersom vi har 5 kolonner, skriver vi 5 verdier på hver linje (a a a c). Hvor mange linjer vi har kommer an på hvor mange rader vi vil

```

    grid-template-areas:
    +box-a box-a box-b box-c+
    +box-a box-a box-b box-c+
    +box-d box-d box-d box-g box-g+
    +box-e box-f box-f box-g box-g+;
  
```

Vi fjerner inn navnet til delene der vi vil ha de.

Om man ønsker å ha et tomt område i gridet uten noe innholdet, skriver man et punktum, slik som dette:

```

    +. box-a box-a box-b +.
    +. box-a box-a box-b box-c+
    +box-d box-d . box-g box-g+
    +box-e box-f box-f . .+;
  
```

OPPGAVER

- 1 Legg inn grid-area og lag navn til alle boksene
- 2 Legg inn grid-template-areas under grid
- 3 Sett opp gridet på den måten du ønsker

HINT

LES KODE

```

    index.html
    1 <div class=grid>
    2 <div class=box-a><A</div>
    3 <div class=box-b><B</div>
    4 <div class=box-c><C</div>
    5 <div class=box-d><D</div>
    6 <div class=box-e><E</div>
    7 <div class=box-f><F</div>
    8 <div class=box-g><G</div>
    9 </div>

    index.css
    1 .grid {
    2   display: grid;
    3   grid-template-columns: repeat(4, 1fr) / repeat(3, 1fr);
    4   grid-gap: 20px;
    5 }
    6 .box-a {
    7   grid-row: 1/3;
    8 }
  
```

Tilbake NESTE

Flere kolonner

Vi går det samme med kolonner som vi gjorde med rader, men skifter ut grid-row med grid-column.

```

    .box-a {
      grid-column: 1 / 3;
    }
  
```

grid-column: 1 / 3;

0:04

Du kan gi både grid-column og grid-row property til et element slik at den tar opp både rad plass og kolonne plass. Slik som dette:

```

    grid-column: 1 / 3;
    grid-row: 1 / 3;
  
```

OPPGAVER

- 1 Velg en av boksene og få den til å ta opp minst to kolonner

HINT

LES KODE

```

    index.html
    1 <div class=grid>
    2 <div class=box-a><A</div>
    3 <div class=box-b><B</div>
    4 <div class=box-c><C</div>
    5 <div class=box-d><D</div>
    6 <div class=box-e><E</div>
    7 <div class=box-f><F</div>
    8 <div class=box-g><G</div>
    9 </div>

    index.css
    1 .grid {
    2   display: grid;
    3   grid-template: repeat(4, 1fr) / repeat(3, 1fr);
    4   grid-gap: 20px;
    5 }
    6 .box-a {
    7   grid-row: 1/3;
    8 }
  
```

Tilbake NESTE

<graphic code> Gå ut av kurs

HTML REGLER > ANATOMI > ANATOMI

Anatomi

HTML består av elementer som beskriver og strukturerer innholdet på en nettside. Et element består av disse tingene:

Åpningstagg - På bildet ser vi en <p> åpningstagg. Denne <p> står for paragraf og viser at dette er et paragraf element. Vi har mange forskjellige tagger i HTML for forskjellig innhold.

Innhold - Etter åpningstagen kommer innholdet i taggen. I dette tilfellet er det en setning. Innholdet i taggen kan være andre ting også, dette skal vi lære mer om senere.

Avslutningstagg - Helt til slutt har vi en avsluttende tagg, den ligner mye på åpningstagen, men har en skråstrek i seg.

Ta en titt på bildet og lær deg hvordan et element ser ut.

Element

<p> Dette er en setning </p>

Åpningstagg
Innhold
Avslutningstagg

LES KODE

Tilbake
▶ ◀ ⏪ ⏩
Neste

<graphic code> Gå ut av kurs

HTML REGLER > HIERARKI > HIERARKI

Hierarki

```

graph TD
    div -- "barnet til / forbærer til" --> h1
    div -- "barnet til / forbærer til" --> p
    h1 <--> |"søsken"| p
    
```

Tilbake
▶ ◀ ⏪ ⏩
Neste

<graphic code> Gå ut av kurs

HTML REGLER > STRUKTUR > STRUKTUR

Struktur

Når vi skriver HTML kode, er det viktig at vi har orden i alle taggene. For at vi skal oppnå dette, bruker vi innrykk og linjeskift.

```

<div>
  <h1> Dette er en overskrift </h1>
  <p> Dette er en paragraf </p>
</div>
  
```

Her ser vi samme kode som var visualisert på forrige side, et div element med to barn som er på samme nivå og derfor er søsken. Fordi h1 og p elementet er inni div elementet, har vi lagt til et innrykk på disse.

Vi kan også ha barn inne i andre barn også, for hvert barn vi lager legger vi til et innrykk:

```

<html>
  <body>
    <header>
      <div>
        <h1> Dette er en overskrift </h1>
        <p> Dette er en paragraf </p>
      </div>
    </header>
  </body>
</html>
  
```

LES KODE

Tilbake
▶ ◀ ⏪ ⏩
Neste

<graphic code> Gå ut av kurs

HTML REGLER > DOKUMENTMAL > DOKUMENTMAL

Dokumentmal

Når vi setter opp et HTML dokument er det noen elementer man må inkludere. Vi har laget en mal for deg som du kan kopiere og bruke som en start når du koder HTML.

```

<!DOCTYPE HTML>
<html>
  <head>
    <meta charset="utf-8">
    <title> Tittelen til nettsiden går her </title>
  </head>
  <body>
  </body>
</html>
  
```

<!DOCTYPE html> - Først har vi en «document type declaration», også kalt doctype. Ved å skrive dette forteller vi nettseseren at dette er et HTML dokument. Alle HTML dokumenter starter med dette som første kode.

<html> </html> - Deretter har vi en åpnings og avslutningstagg for html. Html taggen inneholder alle andre HTML tagger unntatt Doctype. På innsiden av html taggen finner vi head og body taggene.

<head> </head> - I neste del vil du lære hva head taggen er for.

<body> </body> - I neste del vil du lære hva body taggen er for.

<meta charset="utf-8"> - Dette er en selvlukkende tagg, noe som betyr at den ikke har en avslutningstagg. Meta charset angir tegnkodingen for HTML-dokumentet og skal helst stå først under <head>.

<title> </title> - Dette er en tittel tagg som gir en tittel til nettsiden.

LES KODE

Tilbake
▶ ◀ ⏪ ⏩
Neste

<graphic code> Gå ut av kurs

HTML REGLER > HEAD OG BODY > HEAD OG BODY

Head og body

Som vi så på forrige side er <head> og <body> tagger som går under <html> taggen. Alt som går under <body> taggen er innhold vi kan se på en nettside. <head> taggen inneholder informasjon vi ikke kan se, slik som tittel, scripts, styles, meta data og så videre. <body> er den største delen i HTML dokumentet og inneholder masse mer enn <head> gjør.

```
<html>
  <head>
  </head>
  <body>
    Nettsidens innhold går her...
  </body>
</html>
```

index.html https://localhost/

LES KODE

Tilbake Neste

<graphic code> Gå ut av kurs

HTML REGLER > UTFORDRINGSOPPGAVE

Utfordringsoppgave

Sett opp dette hierarkiet i HTML

```
graph TD
  html[html] --- head[head]
  html --- body[body]
  head --- title[title]
  body --- div1[div]
  body --- div2[div]
  div1 --- p[p]
  div1 --- h1[h1]
  div2 --- h2[h2]
```

index.html https://localhost/

LES KODE LEVER OPPGAVE

Tilbake Ferdig

<graphic code> Gå ut av kurs

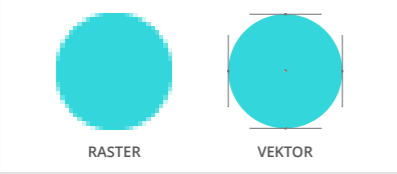
BILDE, GRAFISKE ELEMENTER... > IKONER OG LOGOER > VEKTOR VS RASTER

Vektor vs raster

Logoer, ikoner og andre grafiske elementer blir laget i vektor grafikk. Dette er ulikt raster grafikk som består av pixler.

Vektor - Vektor grafikk bruker linjer og former for å forme et bilde og er kalkulert med matematikk. Vektor grafikk kan skaleres opp og ned i størrelse uten å miste noe av kvaliteten. SVG er et vektor filformat man kan bruke på nett og passer bra til logoer og enkle grafiske elementer.

Raster - Raster bilde består at et grid med små pixler. Raster bilder mister kvalitet når man skaleres og passer best til fotografier og bilder med mye



RASTER VEKTOR

index.html https://localhost/

LES KODE

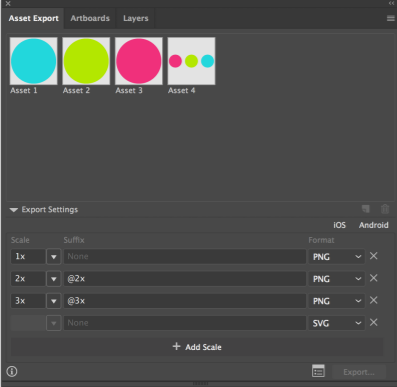
Tilbake Neste

<graphic code> Gå ut av kurs

BILDE, GRAFISKE... > IKONER OG LOGOER > EKSPORTERING FRA ILLUSTRATOR

Eksportering fra illustrator

Når man designer flere ikoner og grafiske elementer på en nettside i illustrator, finnes det en måte å eksportere alt på ikt. Gå til window > asset export



For å legge til elementer her, drar man de bare over til denne boksen. Om elementene ikke er gruppert sammen, vil de separeres slik vi ser på Asset 1, 2 og 3. Vi kan enten gruppere elementene sammen eller dra de over mens man holder inne alt knappen for å få resultatet du ser på Asset 4. Her kan man også velge hvilken filtype man ønsker og legge til så mange man vil.

index.html https://localhost/

LES KODE

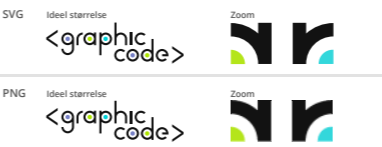
Tilbake Neste

<graphic code> Gå ut av kurs

BILDE, GRAFISKE ELEMENTER, VIDEO OG AN... > IKONER OG LOGOER > SVG

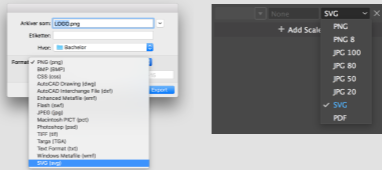
SVG

SVG lar deg bruke vektor grafikk i HTML filen din. SVG står for scalable vector graphic og består av et sett koder som forteller nettleseren hvordan elementet skal se ut. SVG er fantastisk å bruke på logoer og enkle grafiske elementer som ikoner der man ønsker å opprettholde kvaliteten. En annen fordel med SVG er at de kan animeres (! «CSS > Effekter > Animere SVG filler» vil vi gå mer inn på dette). Ettersom SVG er kode, vil det ikke ta lang tid å laste inn på nettsiden, så lenge man ikke har laget noe veldig detaljert med masse linjer og former.



Hvordan lager jeg en SVG fil?

1. For å lage en SVG må du først bruke et vektorbasert designprogram, som for eksempel illustrator.
2. Deretter må du eksportere det man har laget som en SVG fil. Det gjør du på samme måte som du eksporterer til andre filformater med illustrator.
3. Til slutt legger du til SVG filen på samme måte som du lærte i «HTML > Bilder > Legge til bilde fra din PC».



index.html https://localhost/

LES KODE

Tilbake Neste