

INTRODUKTION GRIDS

“Webudvikling vil ikke være det samme efter...”



Dagens plan

- Nye pladser
- Opsamling fra sidst
 - Løse ender fra HTML og CSS
 - Opsamling på det sidste af CSS-gennemgangen
- CSS Grid System
 - Først lidt baggrund og CSS Grids og alternativer
 - Derefter fælles gennemgang (kod-med) af basisfunktionalitet i Grids
 - Et par øvelser - måske når vi lidt af dem her, ellers hjemmearbejde
- Responsive layout (indhold i separat præsentation)
 - Introduktion til layoutstrategier i på WEB
 - Introduktion til responsivt design og teknikker (simpelt eksempel)
 - Kort om designstrategier og hvorfor det betyder noget, hvordan man griber det an
 - Et lidt større eksempel, der kombinerer Grids og responsivt design
- Validering af kode (selvstudie)
- Mind Map (CSS)

Lad os se på det



Hvad er CSS Grid Model

- Den nyeste specifikation i CSS -standarden
- Omtales som det mest revolutionerende siden responsive webdesign
- Bygger delvis på Flexbox
- Nu understøttet i alle browsere



Hvorfor CSS Grids

- Der har ALDRIG været en GOD måde at placere elementer i CSS
- Funktioner er blevet brugt til mere end de var tænkt til (Hacks)
- Funktionerne hvor været meget vanskelige at bruge
 - Float og Clear har for mange været “trial and error”
- CSS flexbox er kun til “1-dimension” (retning)



Ikke alle har været glade for positioneringsmulighederne



En anden måde at tænke på

- Før:
 - fokus på elementet
 - eks. float mv.
 - bredder på elementerne
- Grid:
 - forældreelement
 - Grid “regner” selv bredder ud



Lad os komme i gang!

- En introduktion til de væsentligste egenskaber ved CSS Grids gennem kode-med-eksempler*
- Der gives eksempler på to forskellige tilgange



*eksemplerne kan også løses som selvstudie.

1
2
3
4
5
6
7
8

1
4
3
5
2
6
7
8

Basis HTML og CSS

(basis.zip)

```
1  <!DOCTYPE html>
2  <html lang="da">
3  <head>
4      <meta charset="utf-8">
5      <title>CSS Grid-eksempler</title>
6      <link rel="stylesheet" href="ovelse1.css">
7  </head>
8  <body>
9  <div id="mitgrid">
10     <div id="felt1">1
11     </div>
12
13     <div id="felt2">2
14     </div>
15
16     <div id="felt3">3
17     </div>
18
19     <div id="felt4">4
20     </div>
21
22     <div id="felt5">5
23     </div>
24
25     <div id="felt6">6
26     </div>
27
28     <div id="felt7">7
29     </div>
30
31     <div id="felt8">8
32     </div>
33     </div><!-- mitgrid -->
34 </body>
35 </html>
36 <!-- OBS: Div kan med fordel erstattes af semantiske elementer -->
37 <!-- OBS: Navngivningen af id'er (felt1..felt8) bør være
38 forklarende, og refere til indholdet. Eksempelvis "top-banner",
"navigation", "undermenu", "aktuelt_tilbud", "nyhedsartikel",
"kontakt_info" osv. -->
```

```
1  /* Mit CSS
2  (C) Copyright 2019: Niels Østergaard
3  Eksempler på CSS Grid */
4
5  body {
6      font-family: sans-serif;
7      background-color: #999999;
8      margin: 1em;
9  }
10
11 #mitgrid div {
12     font-size: 5em;
13     color: #000000;
14     text-align: center;
15     border-color: #000000;
16     border-width: 2px;
17     border-style: dashed;
18 }
19 |
```

1

2

3

4

5

6

7

8

Makker-øvelse

Gennemgå CSS linje for
linje

Diskuter CSS'et
Særligt linje 11



Et grid oprettes ved at definere en container-element som “grid”
- men det påvirker ikke layoutet

```
20 ▼ #mitgrid {  
21     display: grid;  
22 }
```

- Vi angiver kolonner med “grid-template-columns”
- enheden “fr” betyder fractions, altså dele
- Her er der to kolonner, hvor de har bredden 1 og 3
- ?
 - Vil det gøre en forskel at skrive 10 og 30?
 - Hvad hvis vi ønsker tre kolonner?

```
20 ▾ #mitgrid {  
21     display: grid;  
22     grid-template-columns: 1fr 3fr;  
23 }
```



- Hvis antallet af elementer ikke går op, startes automatisk på en ny række
- Bredderne for kolonerne holdes i alle rækker

```
20 ▼ #mitgrid {  
21     display: grid;  
22     grid-template-columns: 1fr 3fr 1fr;  
23 }
```

1	2	3
4	5	6
7	8	

- Som med bredder kan man der også angives højder med “grid-template-rows”

```
20 ▼ #mitgrid {  
21   display: grid;  
22   grid-template-columns: 1fr 3fr 1fr;  
23   grid-template-rows: 2fr 1fr 1fr;  
24 }
```



2 Måder at bruge CSS Grid

- Vi ser på et eksempel fra hver!
- Første eksempel:
 - Minder om Flexbox
 - Styrer fordelingen ved at angive den på hvert element
 - Tungt at ændre i
- Andet eksempel
 - Styrer fordelingen på forældrelementet, lettere at ændre
 - Kræver navngivning
-

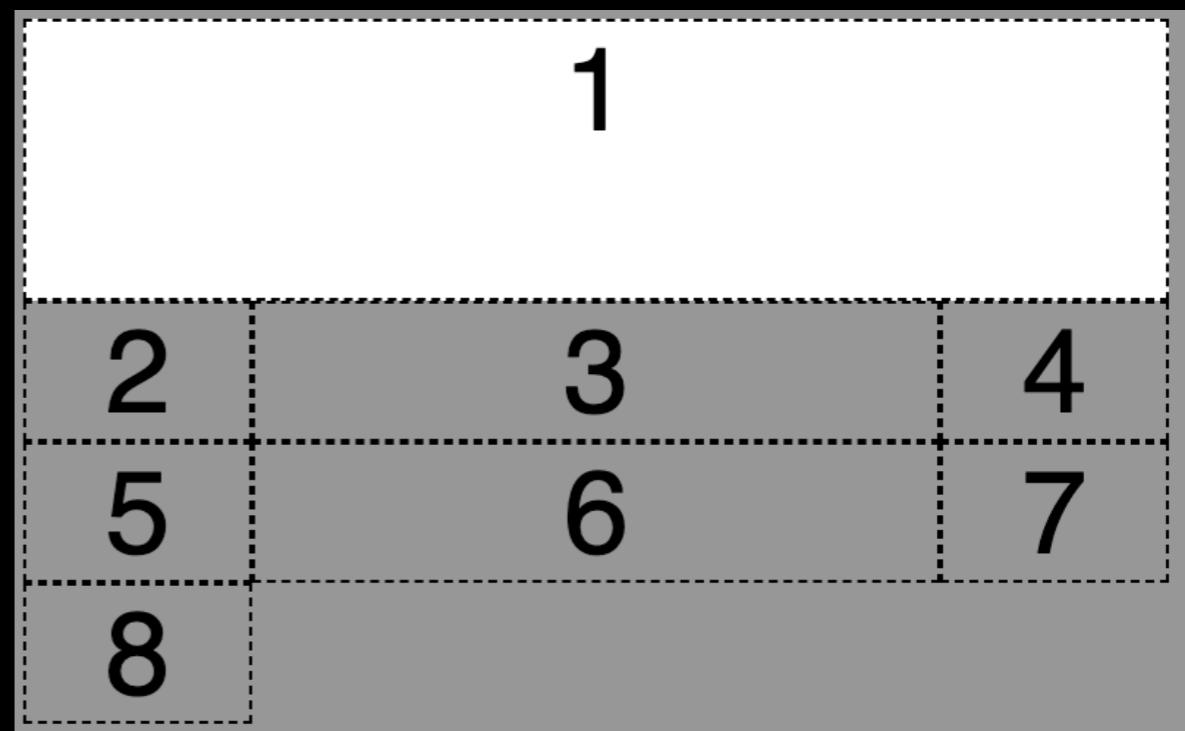
Makker-øvelse

- Hvordan udpeger vi elementet med 1-tallet?
- Hvor mange forskellige selectorer kan I komme i tanke om (der udpeger elementet med 1-tallet)?



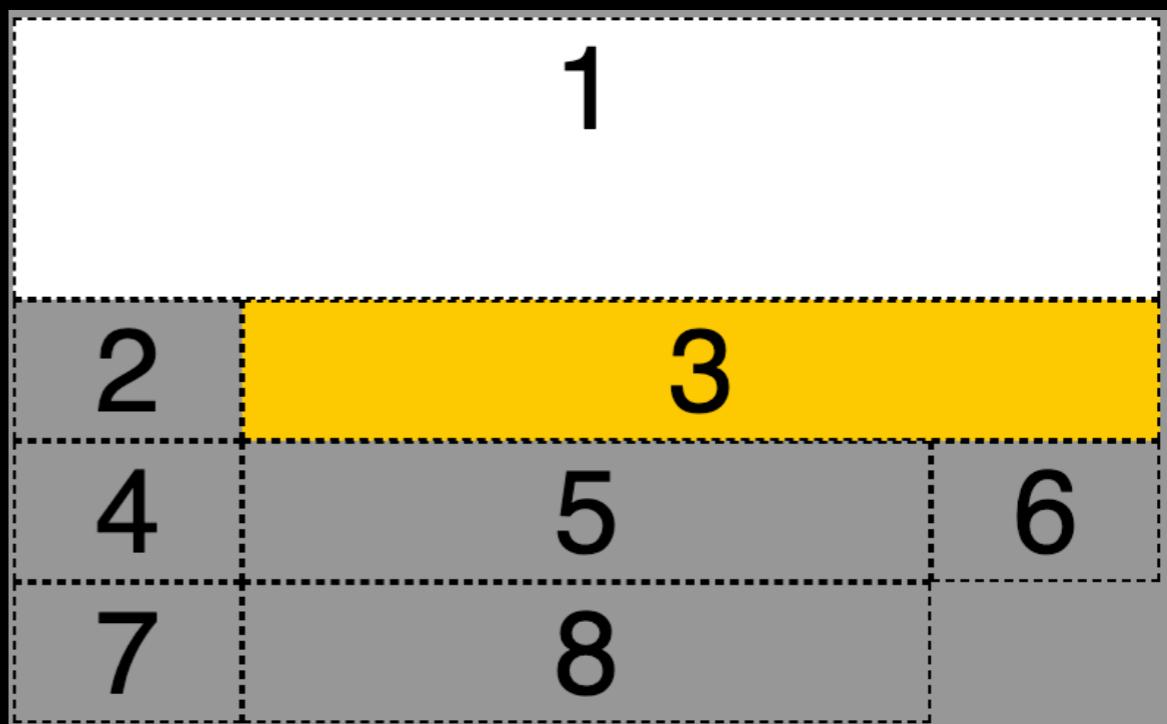
- Et element kan spænde over flere kolonner
- Alle linjer i et grid er nummereret fra 1
- Angiv start- og slut-linje
- OBS: Hvis der er angivet et sprog HTML, som læses fra højre mod venstre, tælles også den vej i grid'et!

```
25  
26 ▼ #mitgrid div:nth-child(1){  
27     background-color: #fff;  
28     grid-column-start: 1;  
29     grid-column-end: 4;  
30 }
```



- Starten behøver ikke være første kolonne
- Hver lodrette linje mellem kolonner tælles
- Her fra 2 til 4

```
32 ▼ #felt3 {
33   background-color: #ffcc00;
34   grid-column-start: 2;
35   grid-column-end: 4;
36 }
```

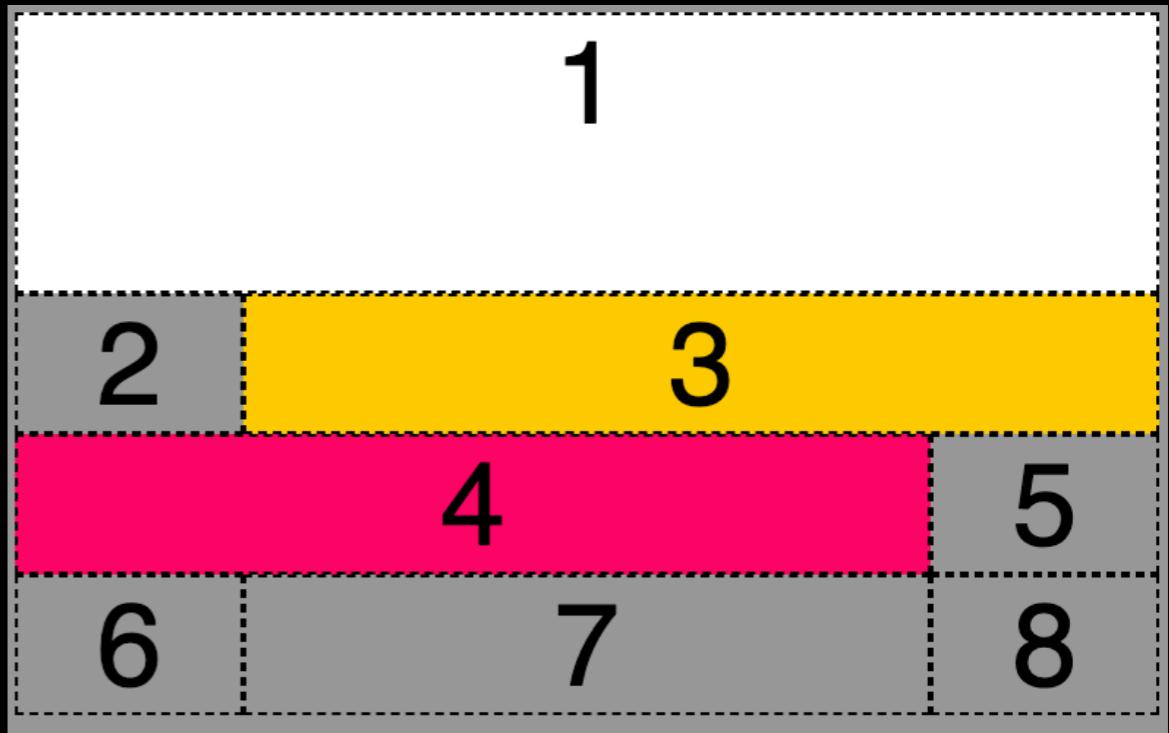


- Der findes forkortelser til mange ting i Grids
- “grid-column: 2/4” gør det samme

```
32 ▼ #felt3 {
33   background-color: #ffcc00;
34   grid-column: 2 / 4;
35 }
```

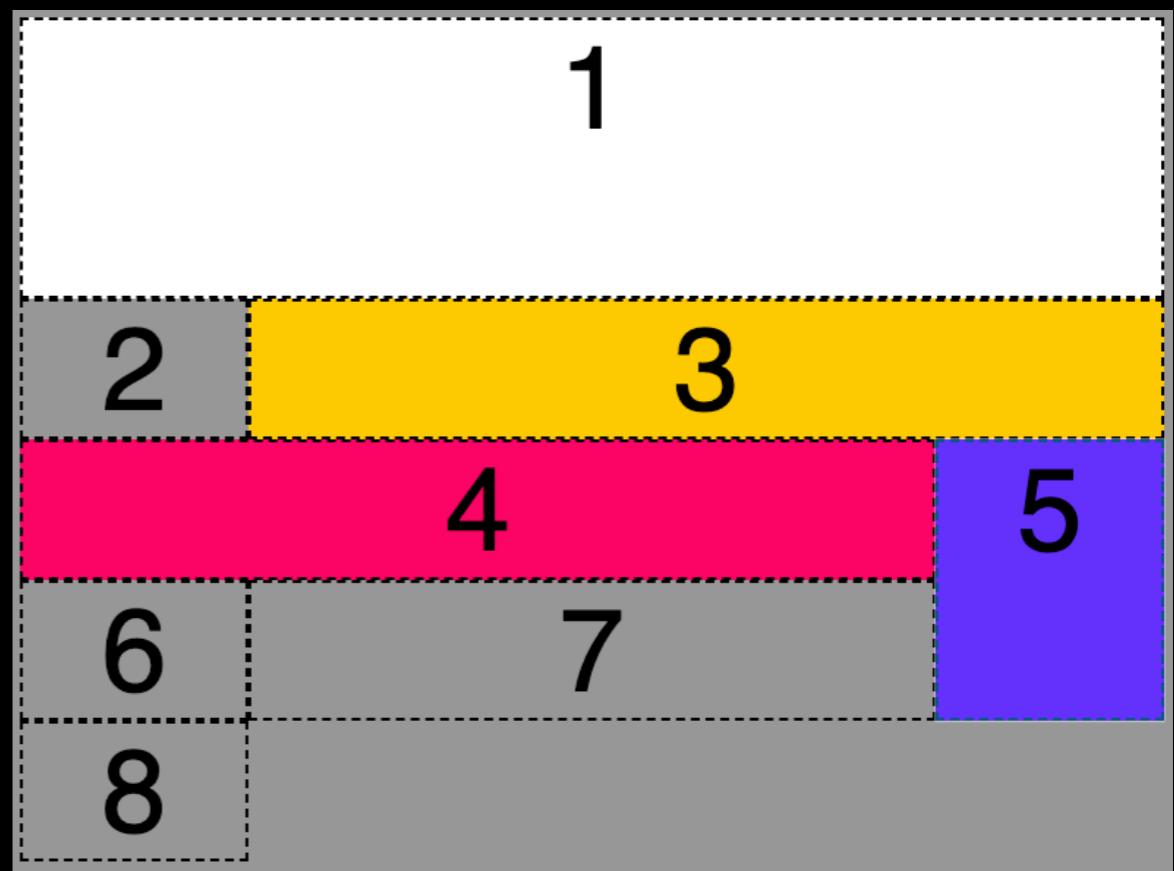
- Man kan også nøjes med at angive hvor mange kolonner den skal spænde over
- gøres i grid-column-end”
- der angives “span” og antallet af kolonner
- breddeangivelsen på den overliggende grid-container holder stadig

```
37 ▼ #felt4 {  
38     background-color: #f06;  
39     grid-column-end: span 2;  
40 }
```



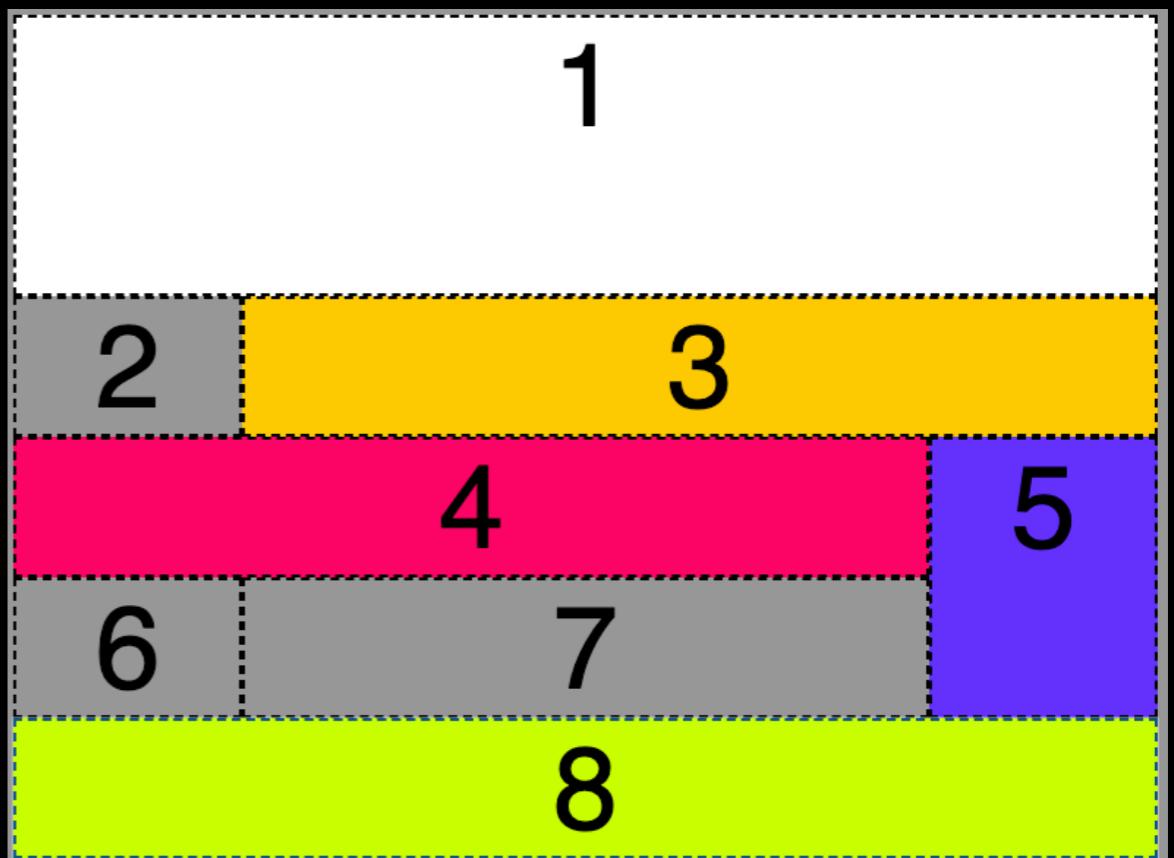
- Det er også muligt at spænde over flere rækker

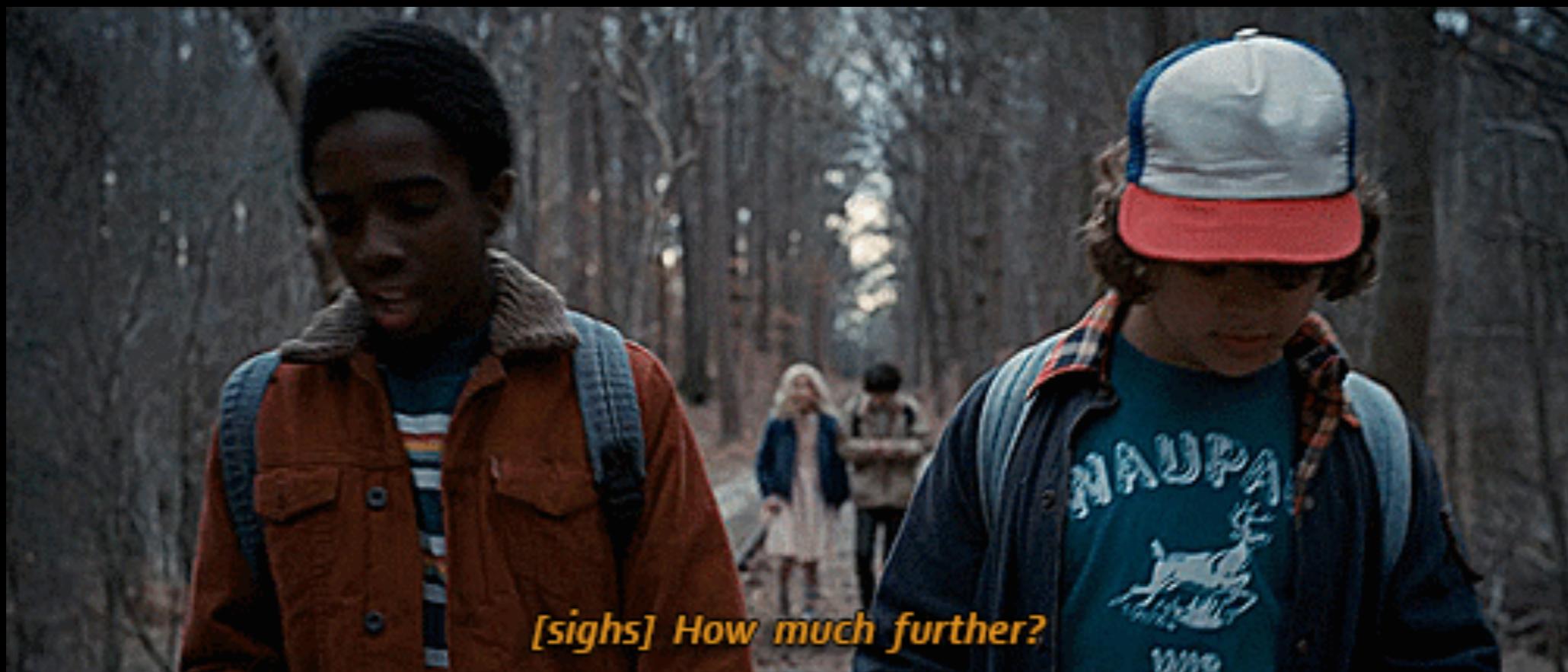
```
42 ▼ #felt5 {  
43     background-color: #63f;  
44     grid-row-end: span 2;  
45 }
```



- Hvad gør vi for at få felt8 til at fylde hele bredden som en footer?
- Kan man gøre det på flere måder?

```
47 ▼ #felt8 {  
48     background-color: #cf0;  
49     /* Hvad skal der stå her? */  
50 }
```





Fem minutter...

- så er der første omgang øvelser

- Det var første eksempel
- Nu til det andet



Gem dit CSS!

- Gem derefter under et nyt navn
- Angiv det nye navn i HTML-filen

Enklere

- Lettere at vedligeholde
- Det overordnede grid styres fra forældre-elementet
- Indhold stadig i de enkelte elementer
- Anvender navngivning af grid-elementerne (ud over id!)
- Vi navngiver den med bogstaver for at vise at det ikke er id
- Den metode som de fleste anvender



Named areas

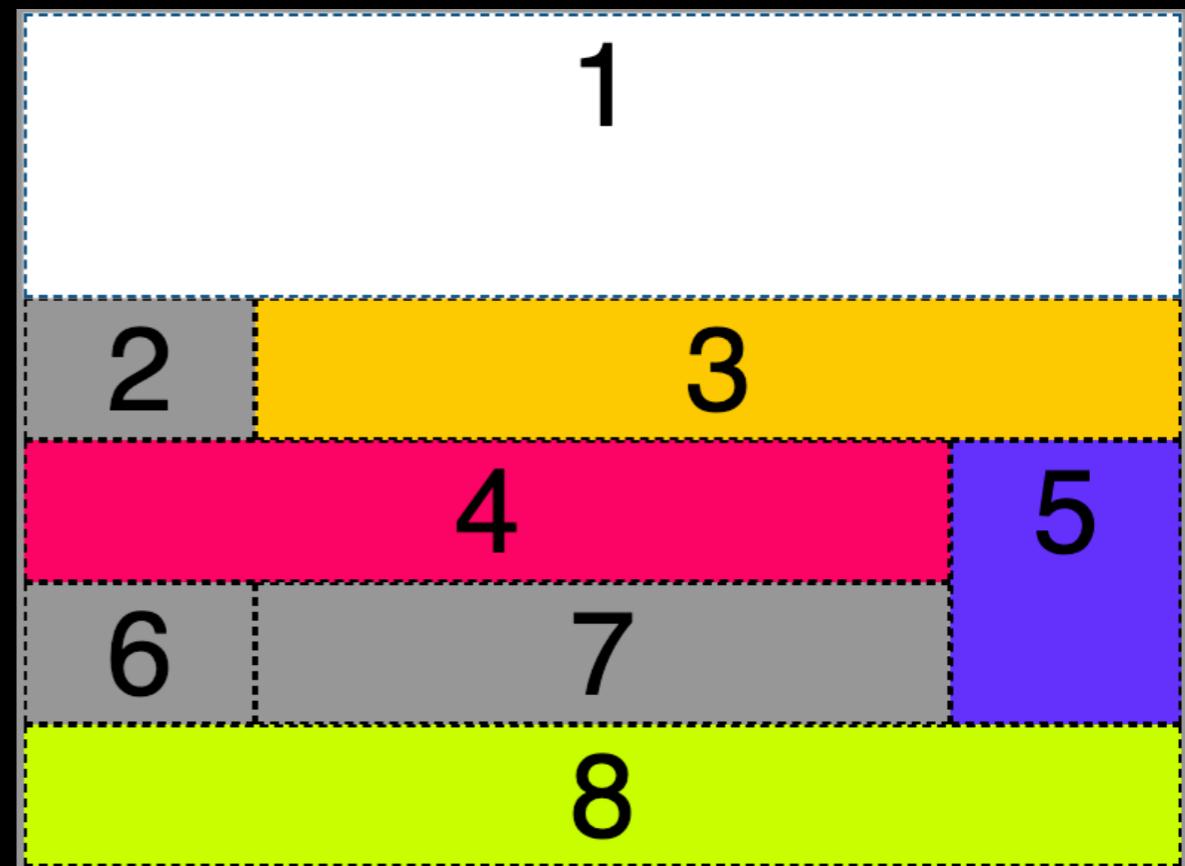
- Elementer kan navngives med “grid-area”
- Der kan derefter referer til navnet
- Her er det kun bogstaver, men det bør være sigende navne
- Eks: overskrift, sidebar, footer, foto, nyheder osv..
- (OBS: Du skal ikke bekymre dig om at alle felterne nu står oven i hinanden, det fikser vi nu..)

```
18    }
19
20 ▼ #mitgrid {
21     display: grid;
22     grid-template-columns: 1fr 3fr 1fr;
23     grid-template-rows: 2fr 1fr 1fr;
24 }
25
26 ▼ #mitgrid div:nth-child(1){
27     grid-area: a;
28     background-color: #fff;
29 }
30
31
32 ▼ #felt2 {
33     grid-area: b;
34 }
35
36 ▼ #felt3 {
37     grid-area: c;
38     background-color: #ffcc00;
39 }
40
41 ▼ #felt4 {
42     grid-area: d;
43     background-color: #f06;
44 }
45
46 ▼ #felt5 {
47     grid-area: e;
48     background-color: #63f;
49 }
50
51 ▼ #felt6 {
52     grid-area: f;
53 }
54
55 ▼ #felt7 {
56     grid-area: g;
57 }
58
59 ▼ #felt8 {
60     grid-area: h;
61     background-color: #cf0;
62 }
```

- I grid-containeren (som indeholder grid-templetten) tilføjes “grid-template-areas”
- Heri angives navne på de “areas der skal placeres” (vi har lige navngivet a-h)
- Der angives en række pr. linje
- I hver linje angives et element pr. kolonne
 - Hvis et element spænder over flere kolonner, gentages det
 - Hvis flere linjer er ens, der istedet justeres i grid-template-rows”

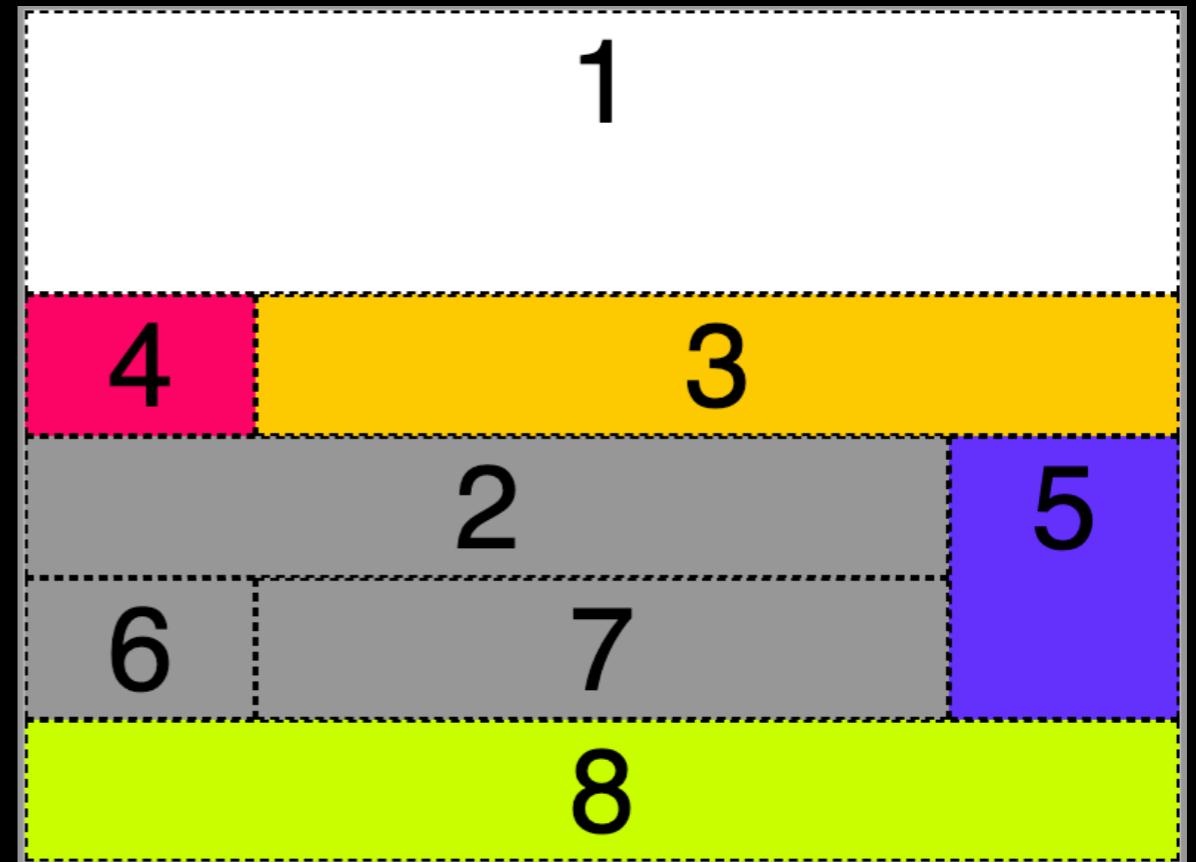
```

20 ▼ #mitgrid {
21   display: grid;
22   grid-template-columns: 1fr 3fr 1fr;
23   grid-template-rows: 2fr 1fr 1fr 1fr 1fr;
24   grid-template-areas:
25   "a a a"
26   "b c c"
27   "d d e"
28   "f g e"
29   "h h h"
30 }
```



```
20 ▼ #mitgrid {  
21     display: grid;  
22     grid-template-columns: 1fr 3fr 1fr;  
23     grid-template-rows: 2fr 1fr 1fr 1fr 1fr;  
24     grid-template-areas:  
25     "a a a"  
26     "d c c"  
27     "b b e"  
28     "f g e"  
29     "h h h"  
30 }
```

- Flyt nu om på to felter:
- Vi bytter om på felt 2 og 4 (b og d)



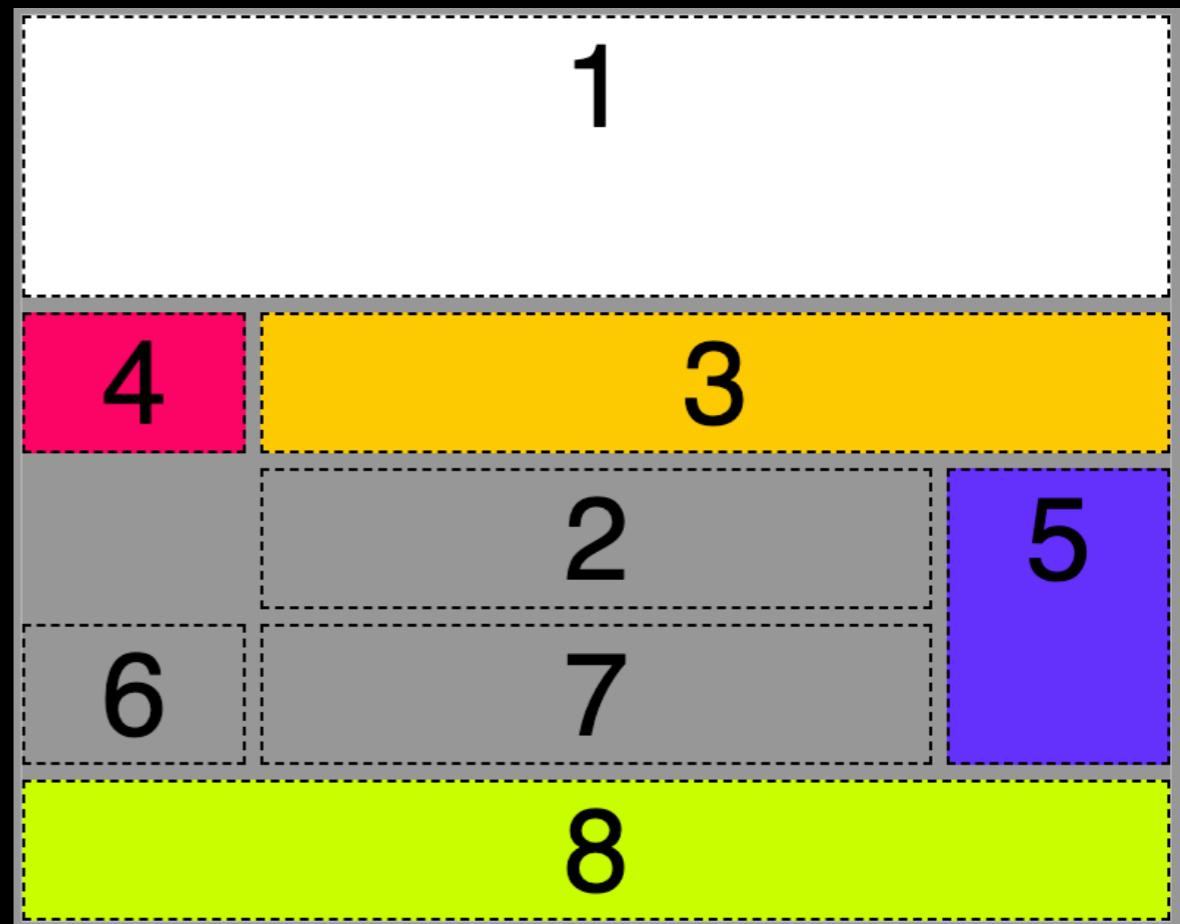
Kan du se fordelen?



- Man kan også lade et felt stå tomt
- Her lader vi kolonne 1 i række 3 (hvor 2 lige før dækkede både kolonne 1 og 2 stå tom, så 2 kun dækker kolonne 2 (den brede))
- Det gøres med et . (punktum)
- Tilføj “grid-gap” for at gøre det tydeligere

```

20 ▼ #mitgrid {
21   display: grid;
22   grid-gap: 10px;
23   grid-template-columns: 1fr 3fr 1fr;
24   grid-template-rows: 2fr 1fr 1fr 1fr 1fr;
25   grid-template-areas:
26     "a a a"
27     "d c c"
28     ". b e"
29     "f g e"
30     "h h h"
31 }
```



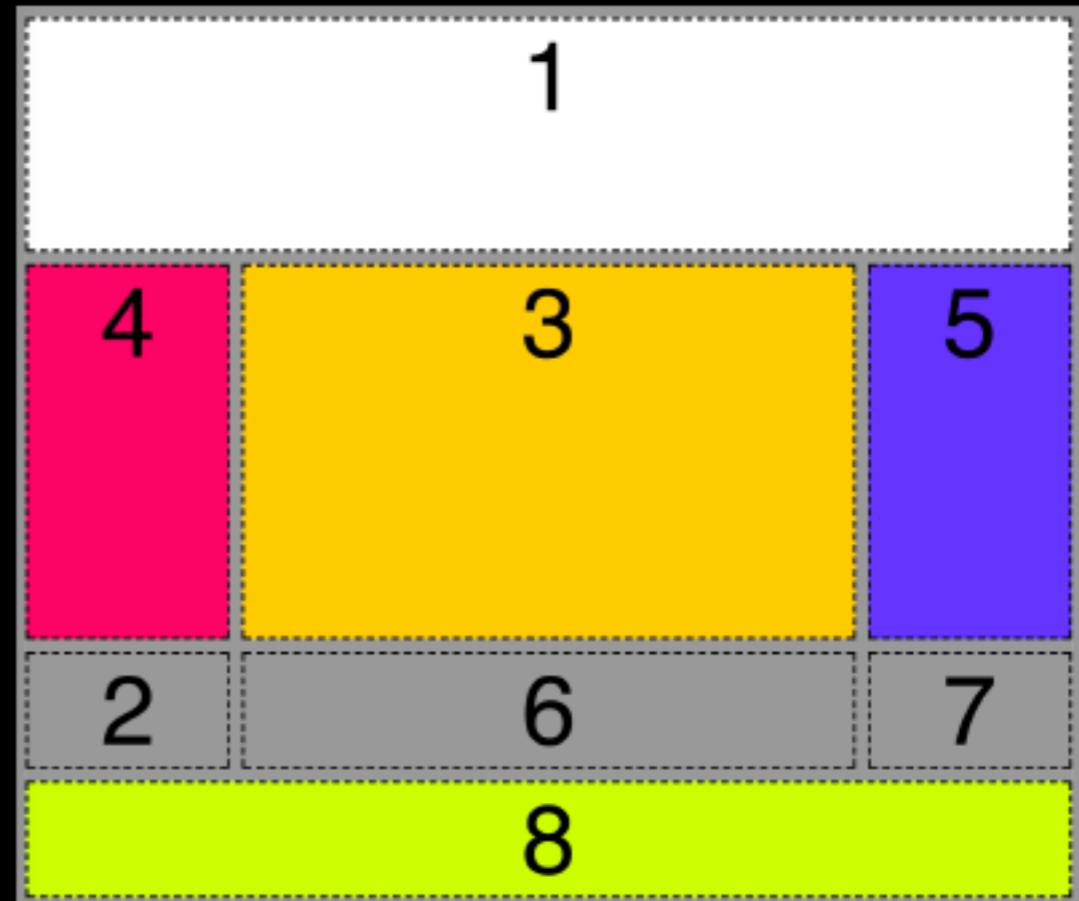
Nu til øvelser..

Husk at gemme en CSS for
hver besvarelse (du kan
senere bruge dem i din
læringsportfolio)



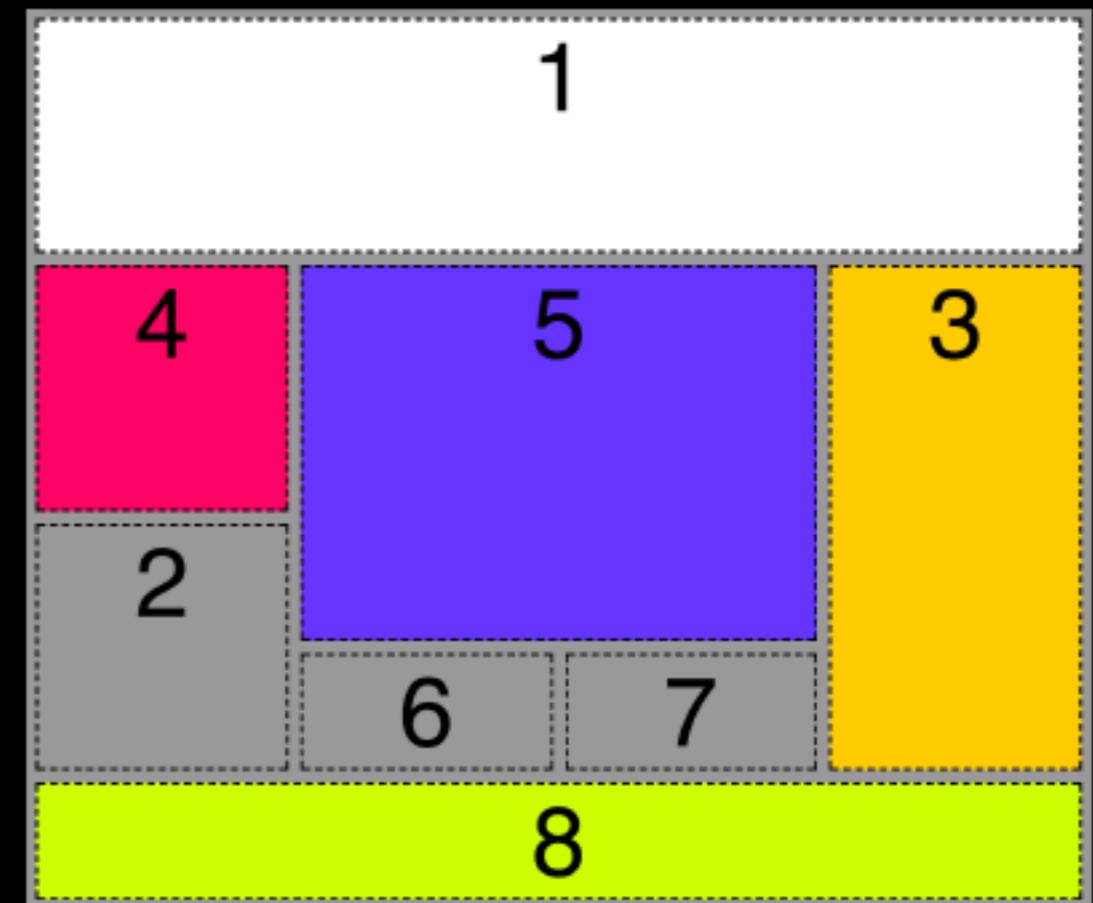
Øvelse 1

Husk nu at gemme CSS for
hver besvarelse



Øvelse 2

Husk nu at gemme CSS for
hver besvarelse



Øvelse 4 (Morsenøgle)

Her skal også laves HTML!

Tips:

Det kan blive lettere hvis man
læser om:

“repeat” i “grid-template-
columns” og -rows

+

:nth-child(even)/(odd)

E				T											
I		A		N		M									
S	U	R	W	D	K	G	O								
H	V	F	*	L	Æ	P	J	B	X	C	Y	Z	Q	Ø	CH

Litteratur

Rachel Andrew: <https://alistapart.com/article/the-new-css-layout-excerpt>
<https://alistapart.com/article/the-story-of-css-grid-from-its-creators>

Supplerende litteratur:

Rachel Andrew: <https://gridbyexample.com/>

Rachel Andrew: <https://youtu.be/N5Lt1SLqBmQ>

Daniel ?: <https://youtu.be/HgwCeNVPlo0>

K3...