

INTRODUKTION GRIDS

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



Planen

- Introduktion til CSS Grid Model
 - Et skridt på vejen mod at kunne lave responsive websider
 - I dag er der fokus på at opnå et statisk layout
 - Næste gang tilføjer vi responsivitet
- I dag
 - Først lidt baggrund
 - Derefter fælles gennemgang af basisfunktionalitet
 - Slutter med selvstændige øvelser

Hvad er CSS Grid Model

- Den nyeste specifikation i CSS-standarden
- Omtales som det mest revolutionerende siden responsive webdesign
- Nogen kan genkende lidt at tankegangen bag layout med tabeller
- Ikke fuldt understøttet endnu i alle browsere
 - MS EDGE forventes at understøtte det i version 16, som forventes at blive frigivet d. 17. oktober



ALLE er begejstrede for den nye CSS Grid Model

Hvorfor Grid?

- 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 har været meget vanskelige at bruge
 - Float og Clear har for mange været “trial and error”
- CSS Flexbox er kun tænkt til “1-dimension” (retning)

Følelsen når man arbejdede med float...



I relation til flexbox

- Grid bygger grundlæggende på ideen fra flexbox
- Men tænkt til håndtere “2 -dimensioner” (retninger)
- Med flexbox bestemmer indholdet en del (og har det ikke samme størrelse bliver det let vanskeligt)
- Grid tager udgangspunkt i layoutet
- Flexbox kan anvendes inde i grids

En anden måde at tænke på...

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

Lad os komme i gang!

En introduktion til de
væsentligste egenskaber
ved CSS Grids gennem
kode-med-eksempler*

*eksemplerne kan også løses som selvstudie.
Her tilrådes det at anvende litteraturen som
supplerende informationskilde



What to expect:

1
2
3
4
5
6
7
8

1		
4	3	
	2	5
6	7	
8		

Basis HTML og CSS



Find dem på: https://github.com/niosDania/CSS_Grids2017

```
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="mit.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
34     </div><!-- mitgrid -->
35 </body>
36 </html>
```

mit.css

```
1  /* Mit CSS
2  (C) Copyright 2017: 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  }
```

1

2

3

4

5

6

7

8

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

```
20 ▼ #mitgrid {  
21     display: grid;  
22 }  
23
```

- 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 sker der hvis vi ønsker tre kolonner?

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

1	2
3	4
5	6
7	8

- Hvis antallet af elementer ikke går op, startes automatisk på næste række
- bredderne for kolonerne holdes

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

1	2	3
4	5	6
7	8	

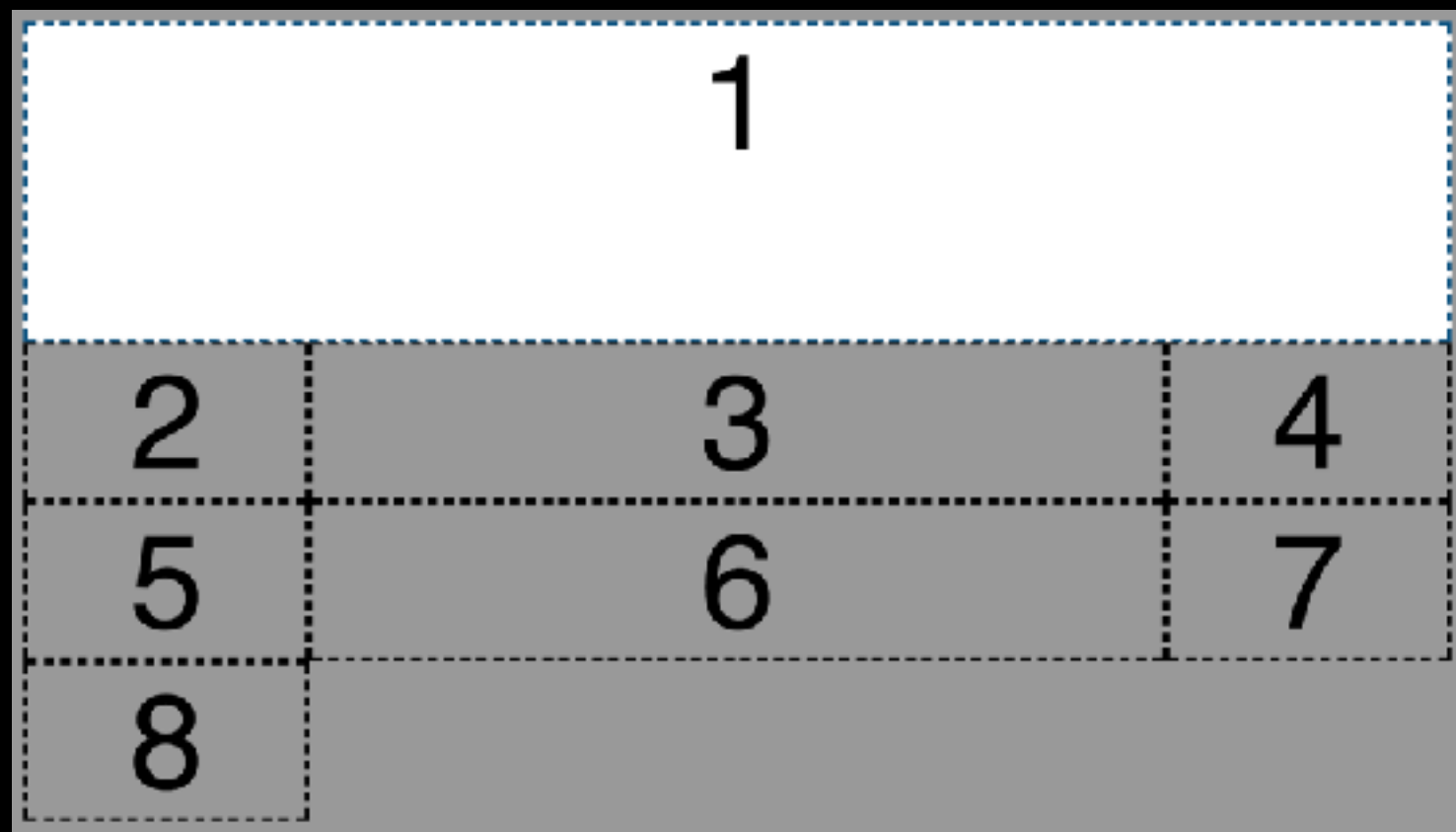
- Som med bredder kan der også angives højder med “grid-template-rows”

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

1	2	3
4	5	6
7	8	

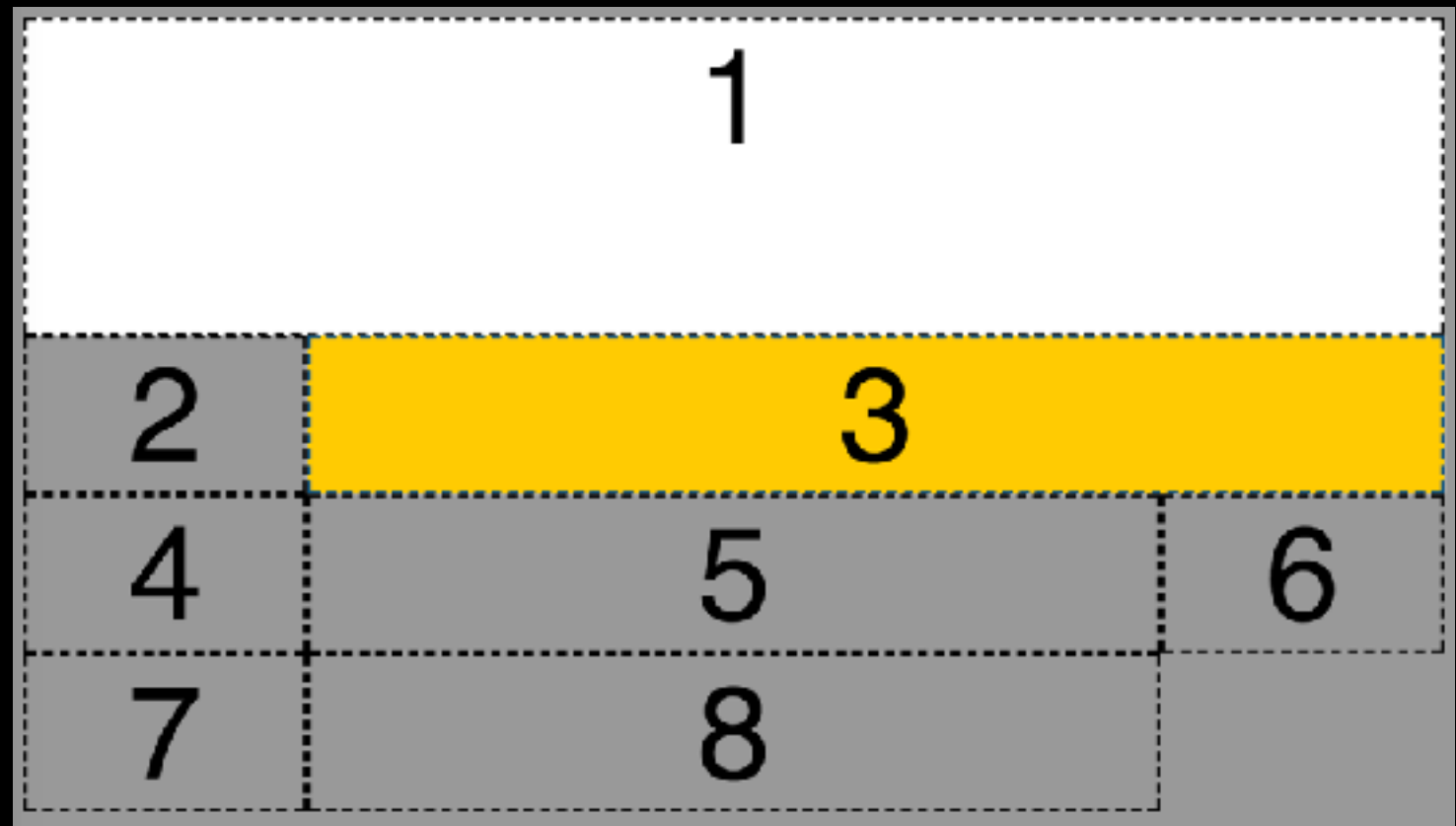
- Et element kan spænde over flere kolonner
- Alle linier i et grid er numereret fra 1
- Angiv start linie og slut linie
- OBS: hvis der er angivet et sprog der læses fra højre mod venstre, tælles også den vej!

```
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 linie mellem kolonner tælles
- Her fra 2 til 4
- Der findes forkortelser til mange ting i Grids
- “grid-column: 2 / 4” gør det samme

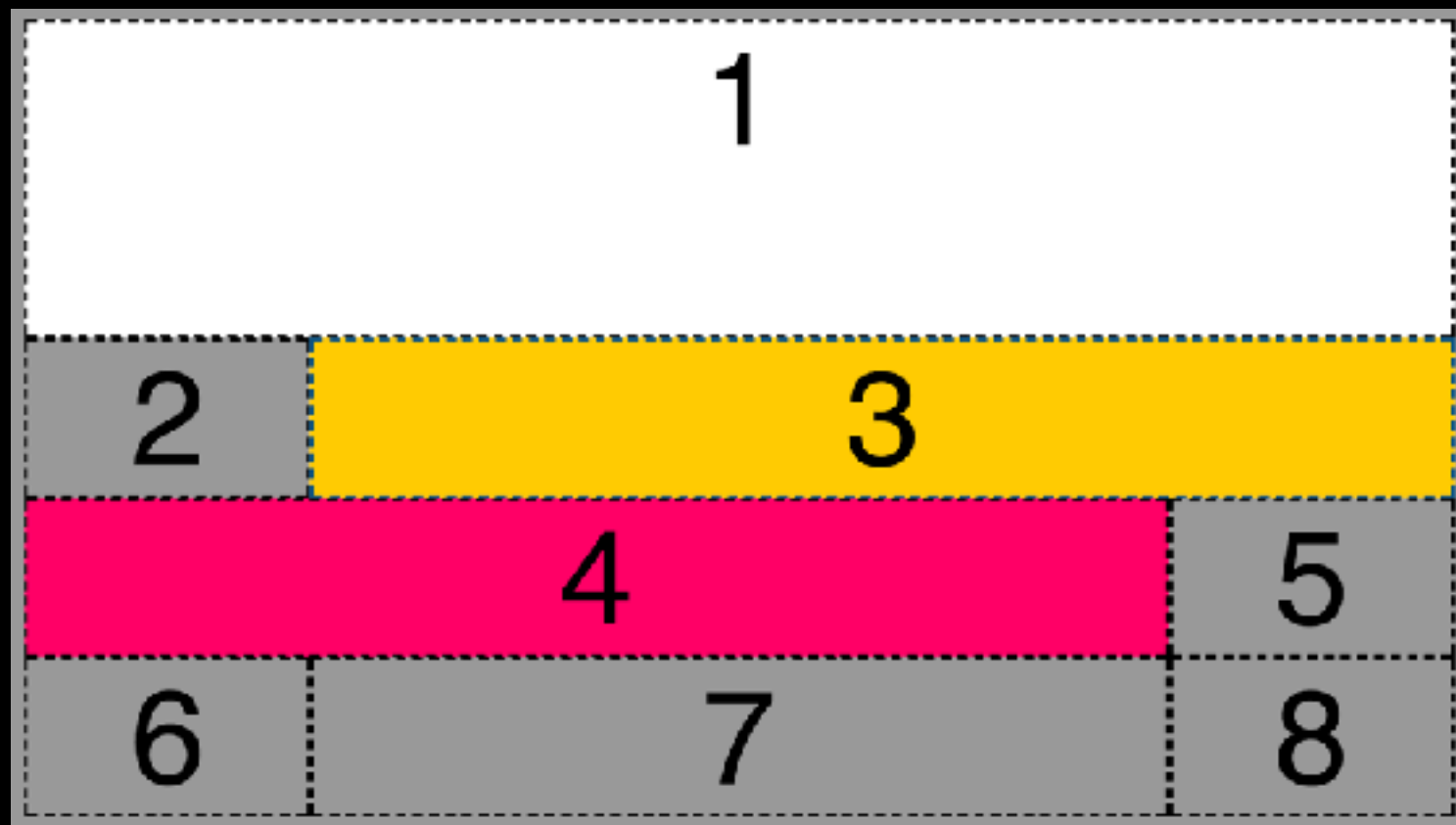
```
32 ▾ #mitgrid div:nth-child(3){
33     background-color: #FC0;
34     grid-column-start: 2;
35     grid-column-end: 4;
36 }
```



```
32 ▾ #mitgrid div:nth-child(3){
33     background-color: #FC0;
34     grid-column: 2 / 4;
35 }
```

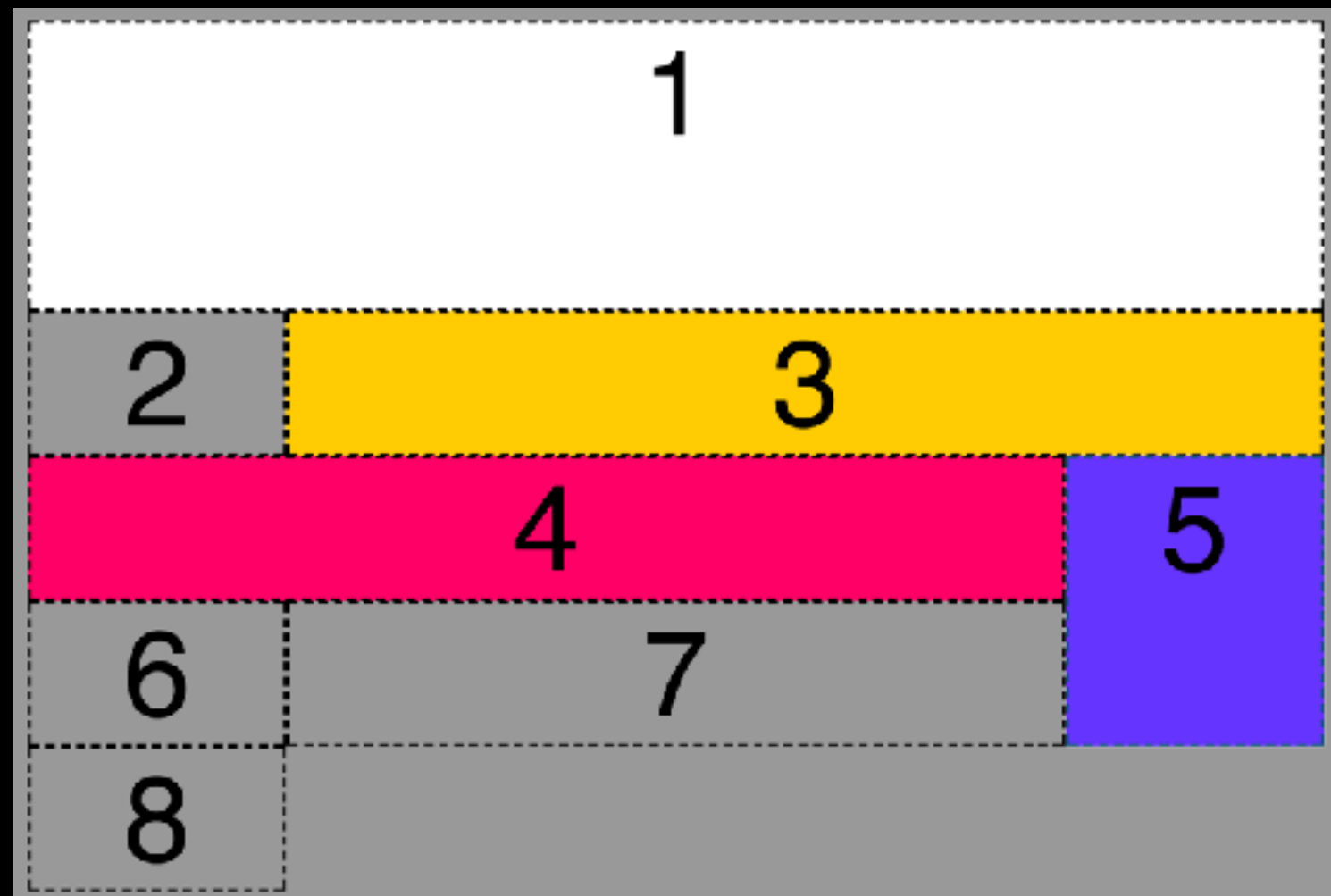
- Man kan også nøjes med at angive hvor mange kolonner den skal spænde over
- gøres i “grid-column-end”
- der angives “span” og antallet af kolonner
- bredde angivelsen på den overliggende grid-container holder stadig

```
38 ▼ #felt4 {  
39     background-color: #F06;  
40     grid-column-end: span 2;  
41 }
```



- Det er også muligt at spænde over flere rækker

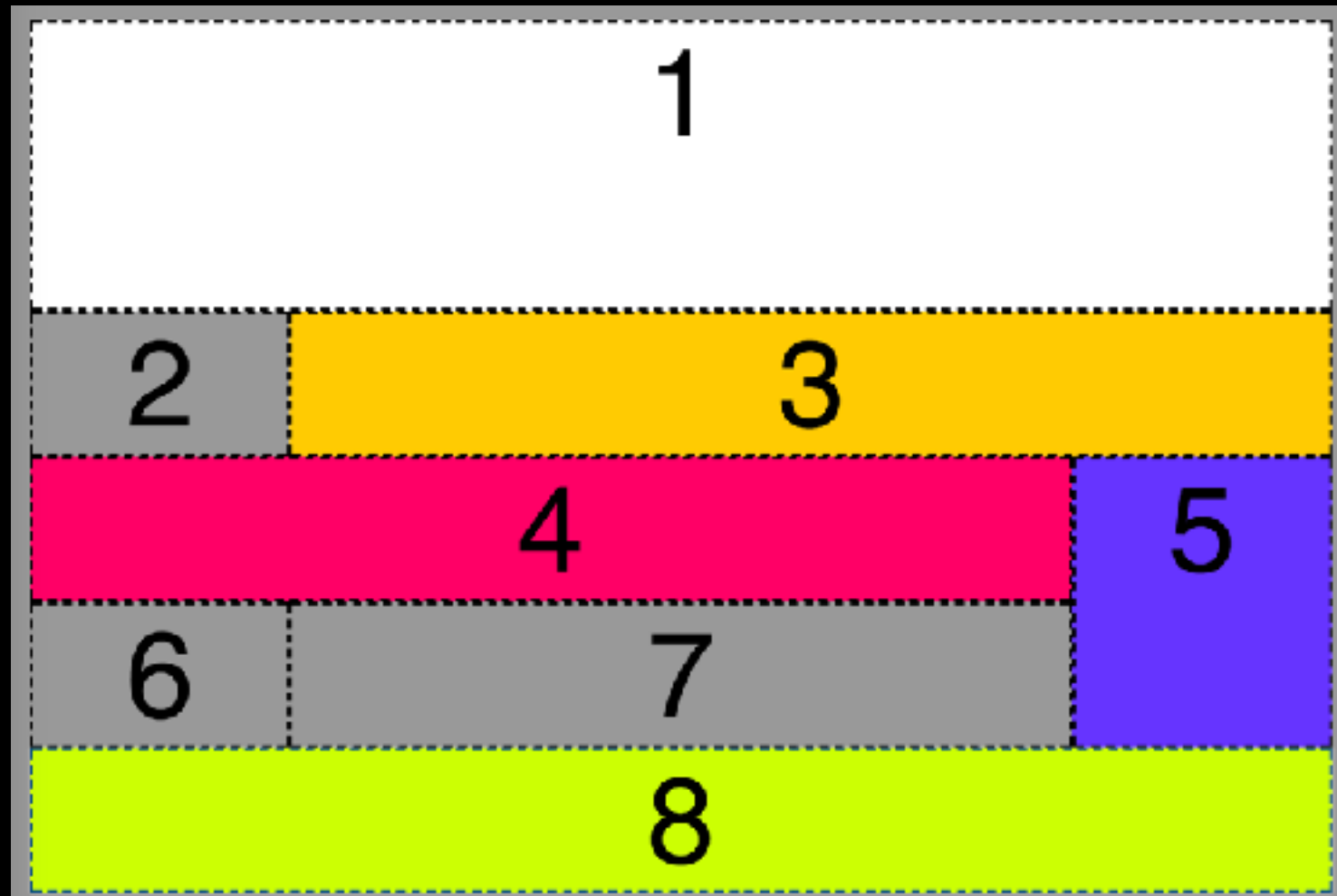
```
43 ▼ #felt5 {  
44     background-color: #63F;  
45     grid-row-end: span 2;  
46 }
```

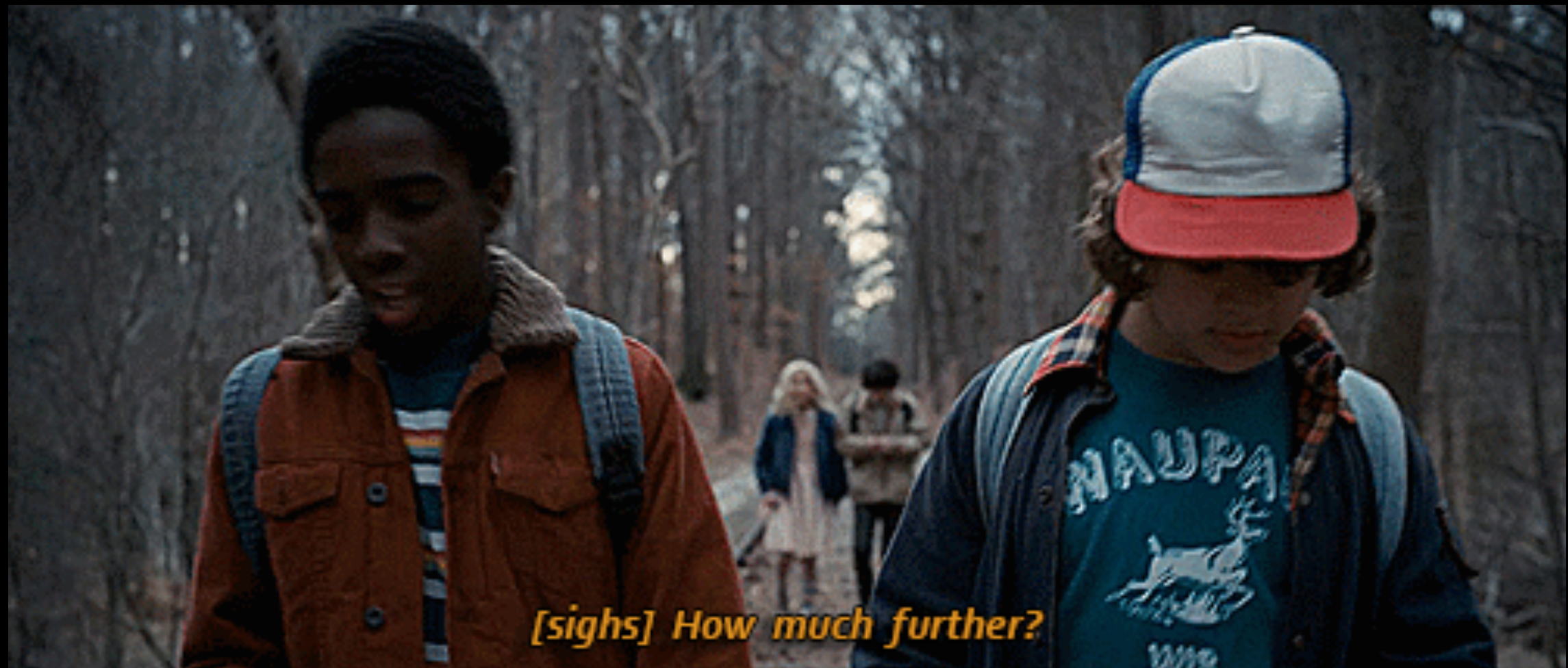


- Hvad gør vi for at få felt8 til at fylde hele bredden som en footer

?

```
48 ▼ #felt8{  
49     background-color: #CF0;  
50     /* hvad skal der stå her? */  
51 }
```





Fem minutter, så er der
første omgang øvelser..

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



Named areas

- Elementer kan navngives med “grid-area:”
- Der kan derefter refereres til navnet
- Her er det kun bogstaver, men det bør være sigende navne
- Eks: overskrift, sidebar, footer, foto etc.

```
32 ▼ #mitgrid div:nth-child(1){  
33     background-color: #FFF;  
34     grid-area: a;  
35  
36 }  
37  
38 ▼ #felt2 {  
39     grid-area: b;  
40 }
```


- Her er slettet alt positionerings-angivelse
- Hvert element har et navn (a-h) og nogle har en farve
- Filen hedder mit_andet.css

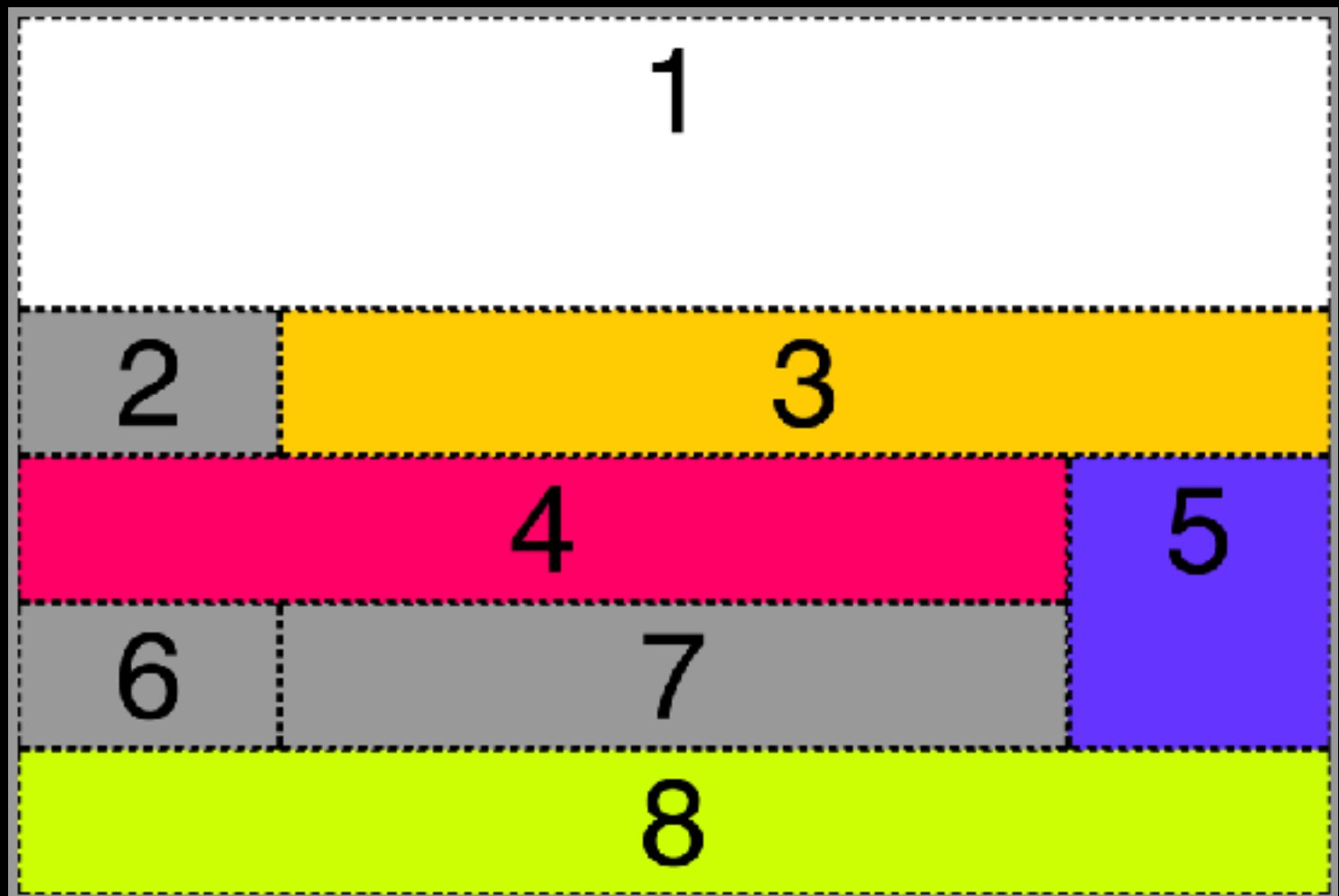
```
32 ▼ #mitgrid div:nth-child(1){
33     background-color: #FFF;
34     grid-area: a;
35 }
36
37 ▼ #felt2 {
38     grid-area: b;
39 }
40
41 ▼ #mitgrid div:nth-child(3){
42     background-color: #FC0;
43     grid-area: c;
44 }
45
46 ▼ #felt4 {
47     background-color: #F06;
48     grid-area: d;
49 }
50
51 ▼ #felt5 {
52     background-color: #63F;
53     grid-area: e;
54 }
55
56 ▼ #felt6 {
57     grid-area: f;
58 }
59
60 ▼ #felt7 {
61     grid-area: g;
62 }
63
64 ▼ #felt8{
65     background-color: #CF0;
66     grid-area: h;
67 }
```

- I selve grid-templaten tilføjes “grid-template-areas”
- Heri angives navnene på de “areas” der skal placeres
- Der angives en linie per række
- I hver linie angives et element per kolonne
 - hvis et element spredes over flere kolonner, gentages det
 - Hvis flere linier er ens, kan der i stedet 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 }

```

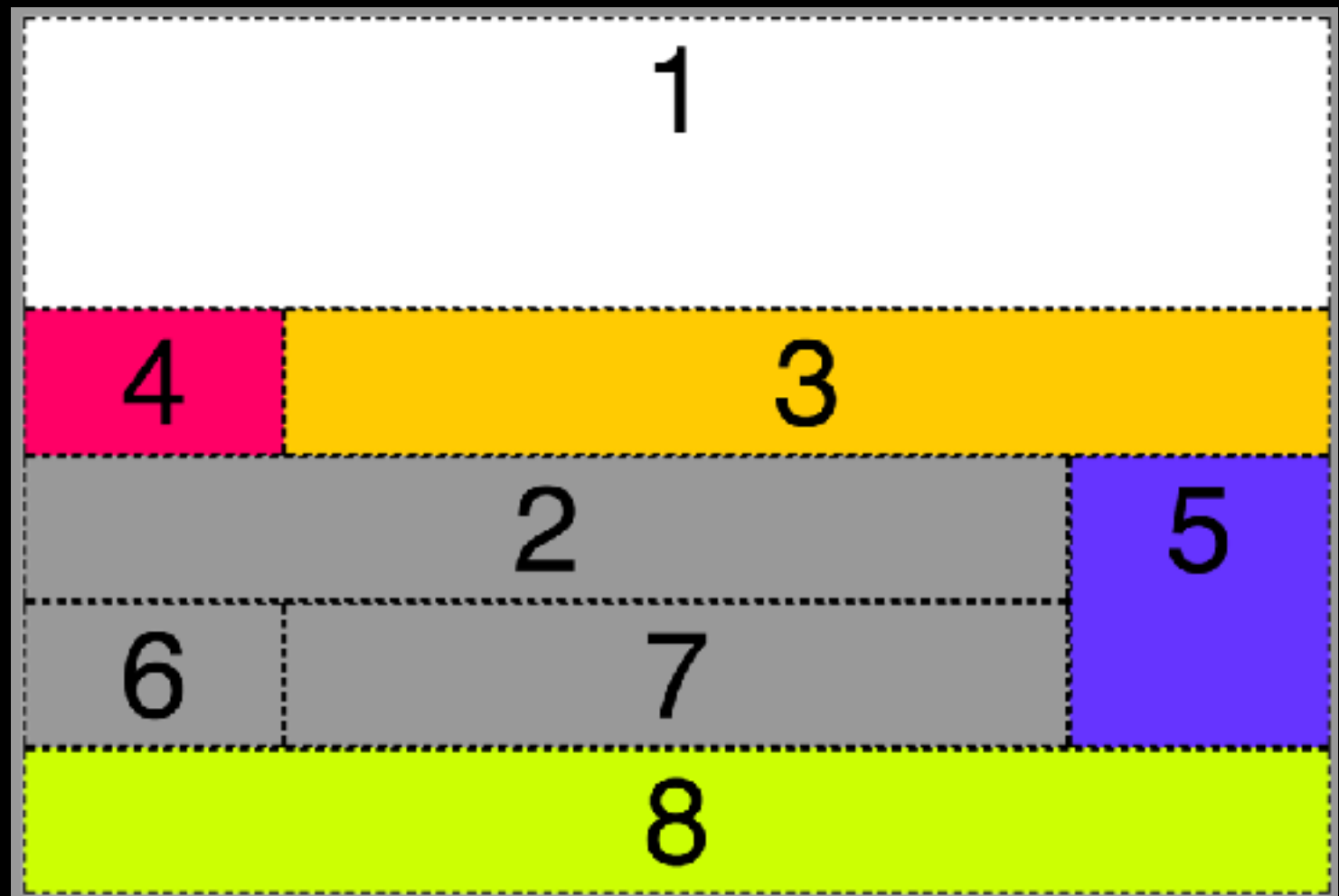


```

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 }
31

```

- Byt nu om på 4 og 2 (b og d)

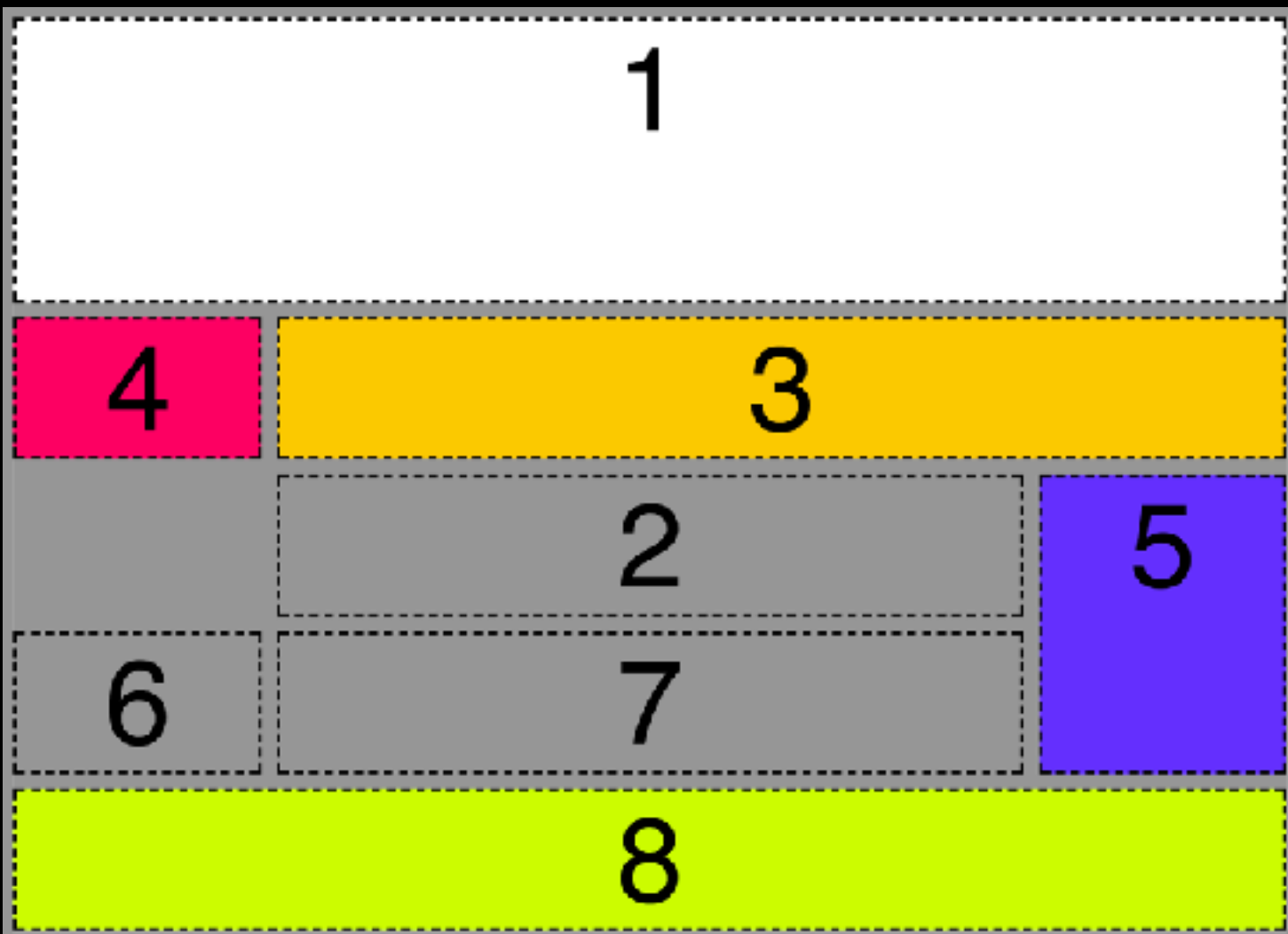


- Man kan også lade et felt stå tomt
- Her lader vi kolonne 1 i række 3 (hvor 2 lige før dækkede både kolonne 1 og 2) stå tom, så 2 kun dækker kolonne 2
- Det gøres med et . (punktum)
- Tilføj “grid-gap” for at gøre det tydeligere

```

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 }

```

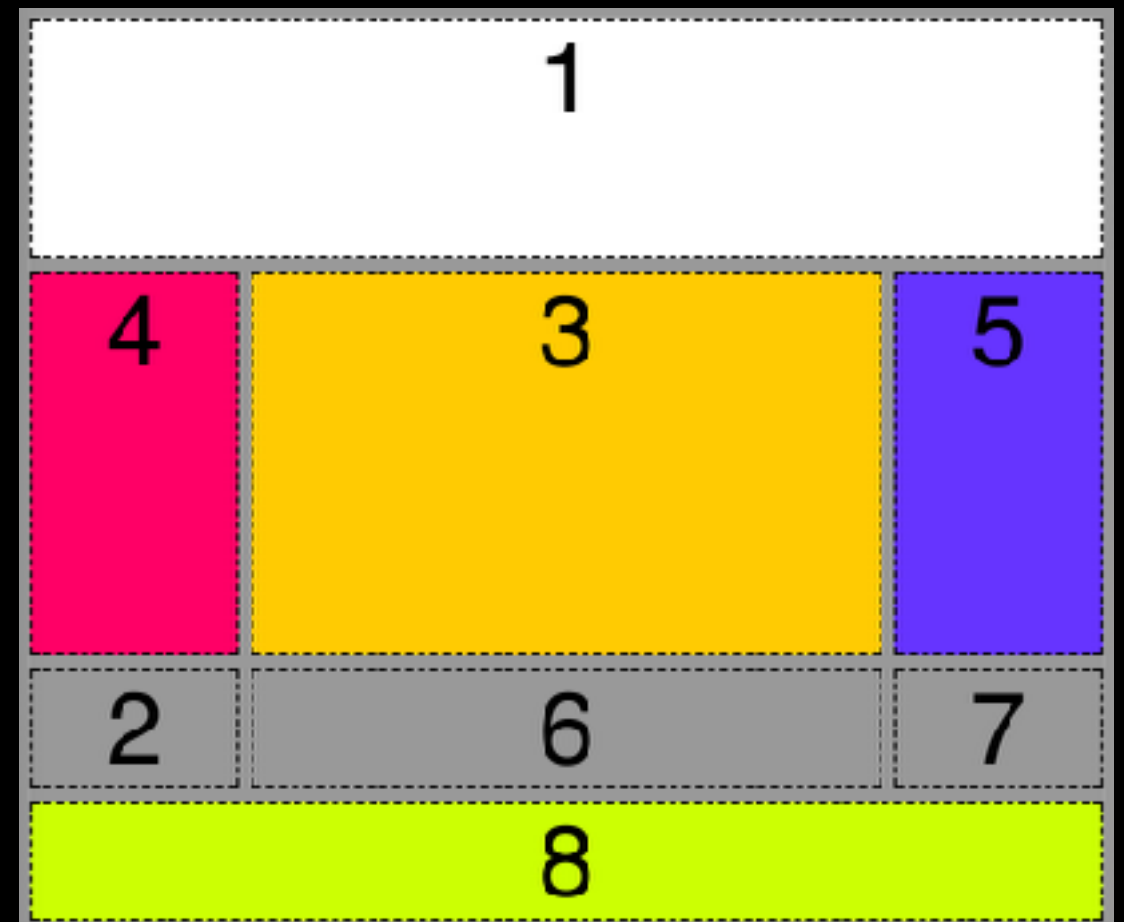


Nu til øvelser..
(Husk at gemme en CSS
for hver besvarelse)



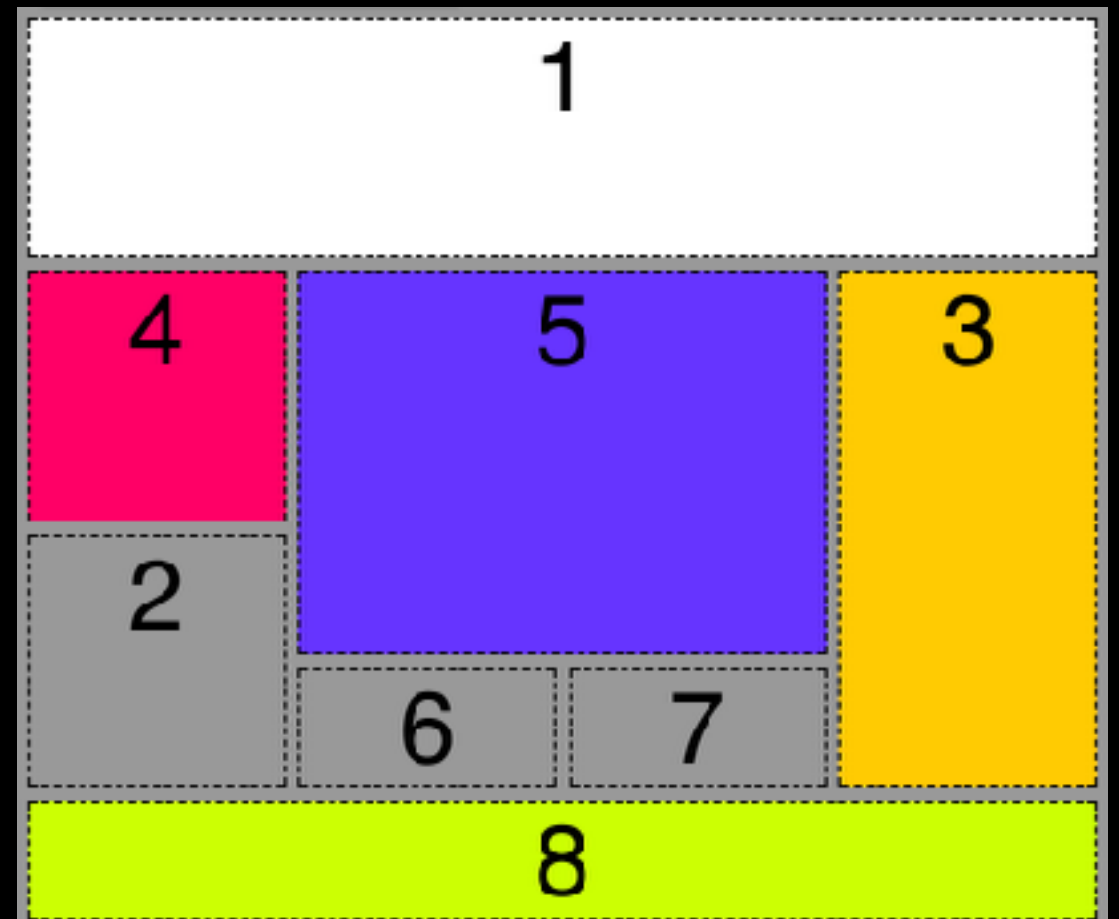
Øvelse 1

(Husk at gemme en CSS
for hver besvarelse)



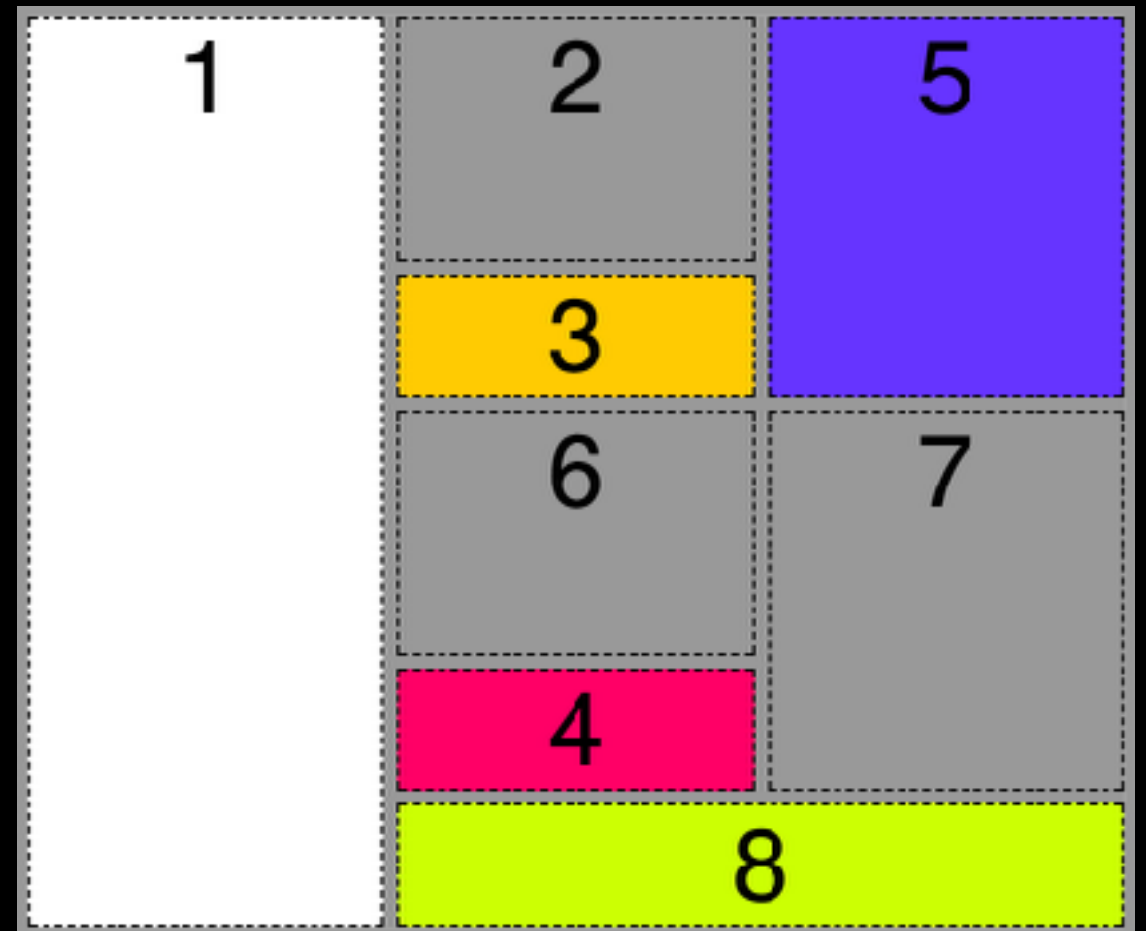
Øvelse 2

(Husk at gemme en CSS
for hver besvarelse)



Øvelse 3

(Husk at gemme en CSS
for hver besvarelse)



Øvelse 4

(Morsenøgle)

Her skal også laves HTML!

Tips:

livet (eller denne øvelse) kan
blive lettere hvis man læser

om:

“repeat” i “grid-template-
columns/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

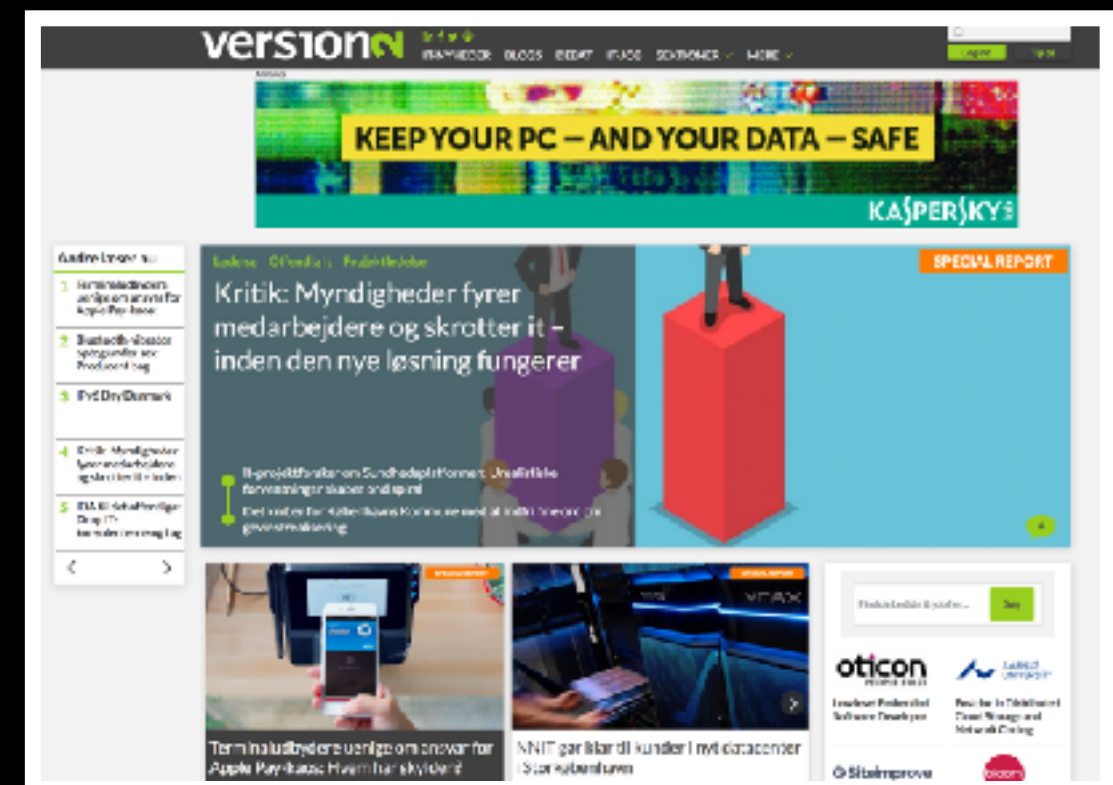
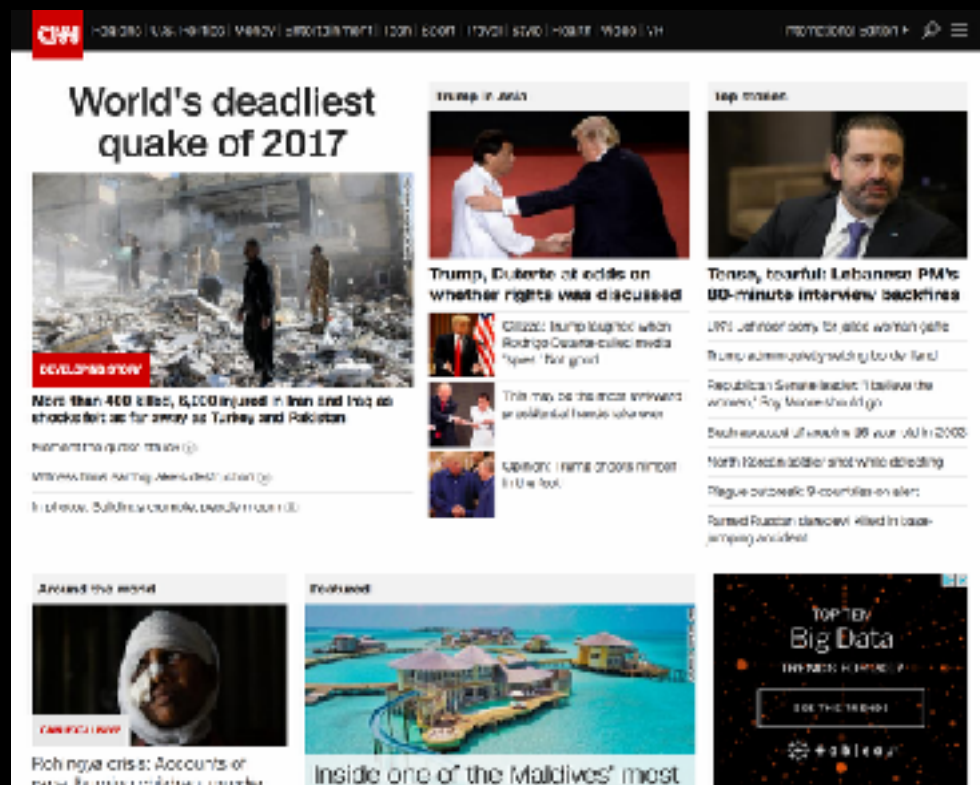
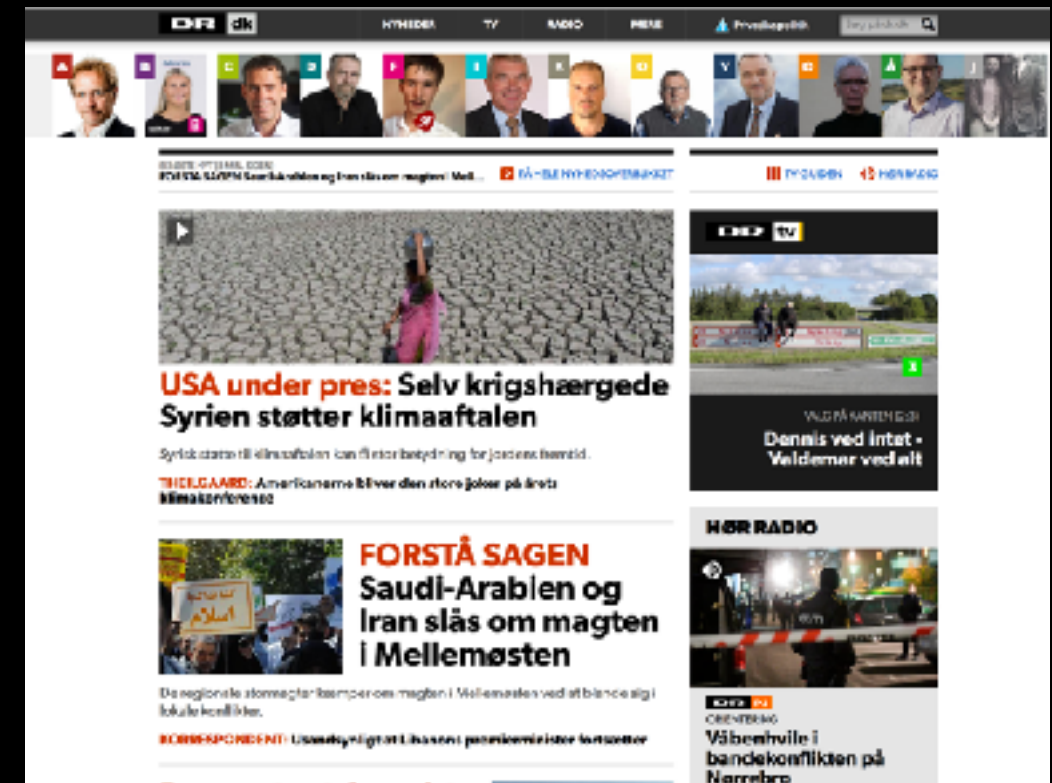
Øvelse 5

(Fri øvelse)

Skitsér en forside til et fiktivt nyhedssite
(papir eller Balsamiq)

Udfør en live prototype (HTML og CSS) med
brug af CSS Grid

Indsæt overskrifter, mumletekst (Lorem
ipsum..) og evt. billeder



Vejledende løsninger

Øvelse 1-4, findes på GitHub efter kl. 17



Litteratur

- Rachel Andrew: <https://alistapart.com/article/the-new-css-layout-excerpt>
- Supplerende litteratur:
 - <https://alistapart.com/article/the-story-of-css-grid-from-its-creators>
 - Rachel Andrew: <https://gridbyexample.com/>
- Anbefalede videoer:
 - Rachel Andrew: <https://youtu.be/N5Lt1SLqBmQ>
 - Daniel ?: <https://youtu.be/HgwCeNVPl00>