

Intro til CSS Grid

...

Hvad er Css Grid?

Et to-dimensionelt grid baseret layout system, der gør det muligt at opdele vores layout i fleksible rækker og kolonner.

Endelig er der fuld adskillelse mellem markup og layout!

Terminologi:

- Grid Container

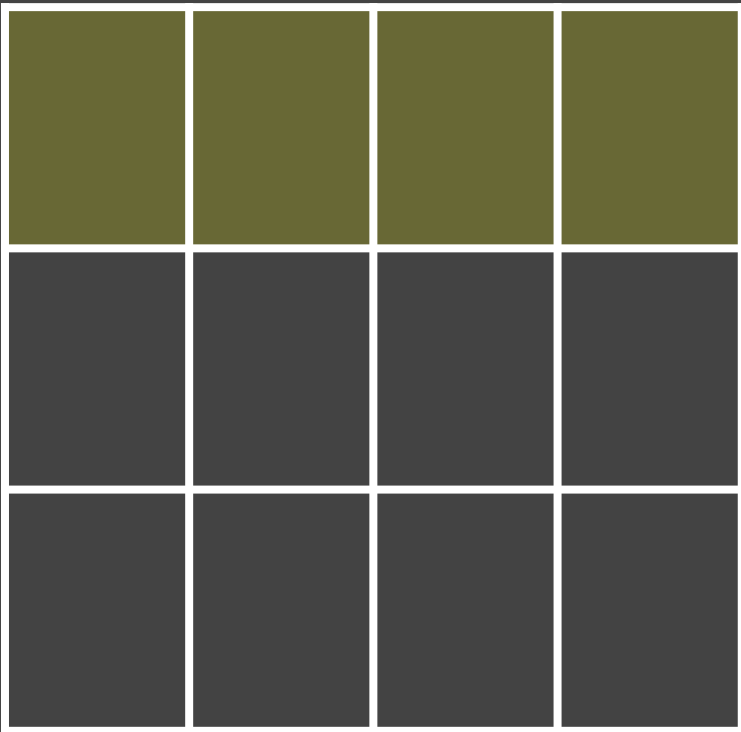
- Grid Items

- Grid Line

- Grid Track

- Grid Cell

- Grid Area

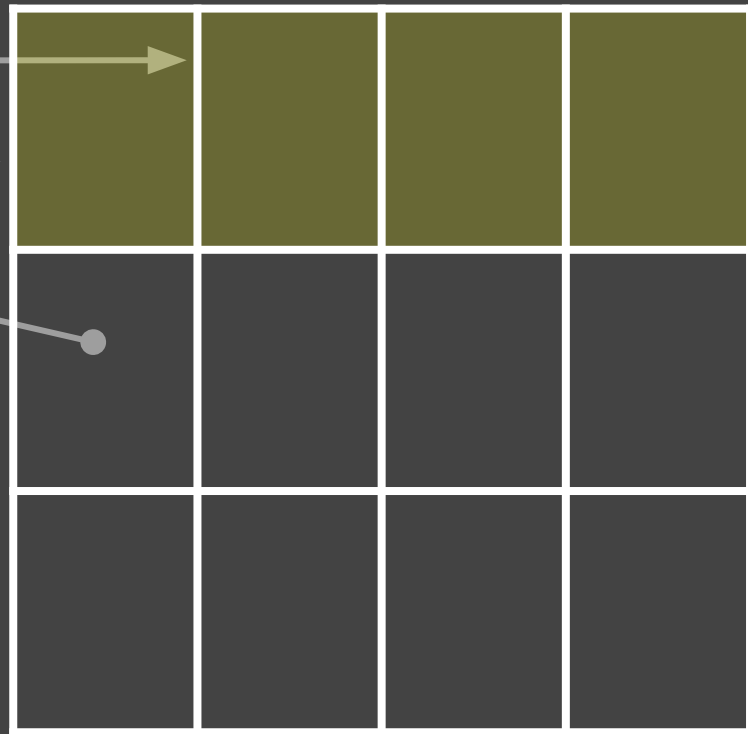


Css Grid

Grid Line

Grid Track

Grid Cell



Properties for Grid Container

display : grid - definerer et html element som grid-container

grid-template-columns : bredden og evt. navn på de enkelte kolonner

grid-template-rows : højden og evt. navn på de enkelte rækker

shortcode > grid-template: grid-template-rows / grid-template-columns

grid-column-gap : afstand mellem kolonner

grid-row-gap : afstand mellem rækker

shortcode > grid-gap: afstand mellem kolonner / afstand mellem rækker

basic properties

For at få elementer opsat i et basis grid, skal vi bruge to properties på vores grid container:

`display : grid;` - definerer et html element som grid-container

`grid-template-columns` : angiver bredden og antal af kolonner

bredden kan angives i px, % m.m (se dokumentationen)

bredden kan også angives som en relativ størrelse `fr` (fraction)

ex.

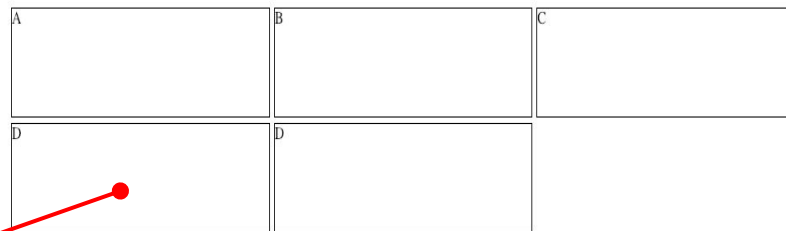
`grid-template-columns: 1fr 1fr 2fr ;`

Den sidste fylder dobbelt så meget som de to første.

Hvis alle kolonner er lige brede kan det med fordel skrives som:

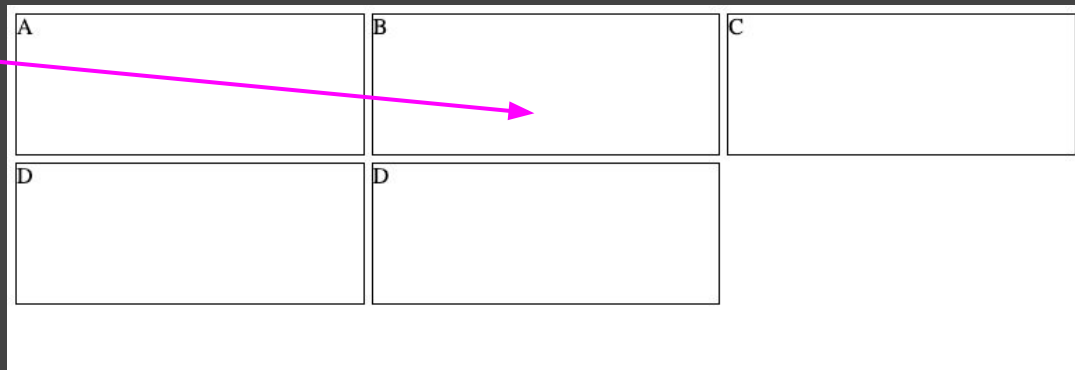
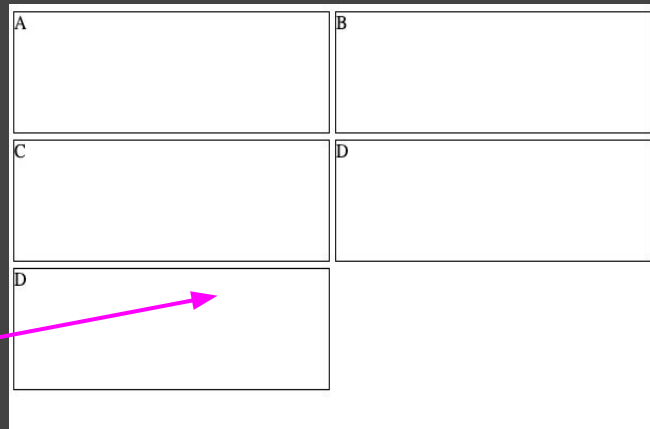
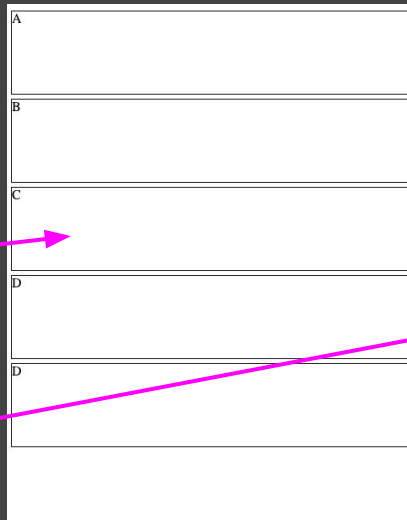
`grid-template-columns: repeat(3, 1fr);` (efter formlen: antal kolonner, bredde)

```
4 <head>
5   <meta charset="UTF-8">
6   <title>Css Grid</title>
7   <style>
8     .container {
9       display: grid;
10      grid-template-columns: repeat(3, 1fr);
11      grid-gap: 5px;
12      grid-template-columns: repeat(3, 1fr);
13      max-width: 960px;
14      margin: 0 auto;
15    }
16
17    .item {
18      border: 1px solid #000;
19      height: 100px;
20    }
21  </style>
22 </head>
23
24 <body>
25   <div class="container">
26     <div class="item item-a">A</div>
27     <div class="item item-b">B</div>
28     <div class="item item-c">C</div>
29     <div class="item item-d">D</div>
30     <div class="item item-d">D</div>
31   </div>
32 </body>
33
```



Mobile first

```
4 <head>
5   <meta charset="UTF-8">
6   <title>Css Grid</title>
7   <style>
8     .container {
9       display: grid;
10      grid-gap: 5px;
11    }
12    .item {
13      border: 1px solid #000;
14      height: 100px;
15    }
16    @media screen and (min-width:600px){
17      .container{
18        grid-template-columns: repeat(2, 1fr);
19      }
20    }
21    @media screen and (min-width:768px){
22      .container{
23        grid-template-columns: repeat(3, 1fr);
24        max-width: 960px;
25        margin: 0 auto;
26      }
27    }
28  |
29  </style>
30 </head>
31
32 <body>
33   <div class="container">
34     <div class="item item-a">A</div>
35     <div class="item item-b">B</div>
36     <div class="item item-c">C</div>
37     <div class="item item-d">D</div>
38     <div class="item item-d">D</div>
39   </div>
40 </body>
```



gridøvelse 1

opsæt den dynamiske dyreliste i [06-visMedTemplate.html](#), med css grid, så de 3 dyr står ved siden af hinanden på desktop, og over hinanden på mobil.

Hint: arbejd “mobile first”

REFERENCER

Eksempler:

<https://gridbyexample.com>

Guide:

<https://css-tricks.com/snippets/css/complete-guide-grid/>

Reference:

https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Grid_Layout

<https://developer.mozilla.org/en-US/docs/Web/CSS/:nth-child>