Lecture: Responsive design and Sass

design website / pages so that it renders well in any type of device

① Media query

assign CSS to particular type of html element on a particular type of media

→ computer
printer
phone
⋮

```
< p class = " screen-only"> For comp
                            screens
                            </p>
```

CSS

create media query

```
@ media  print {
    . screen-only {
        display : none;
        }
}
```

which media I want to apply to

apply to all classes called "screen-only"

apply this to print media only.

make invisible.

another media query

```
@ media ( min -width : 500px) {
```

```
@ media (min-width: 500px) {

        Apply this if media has
                    width ≥ 500px
    }
```

```
@ media   ( max -width: 499 px) {

        Apply this if media has width
                         ≤ 499 px
    }
```

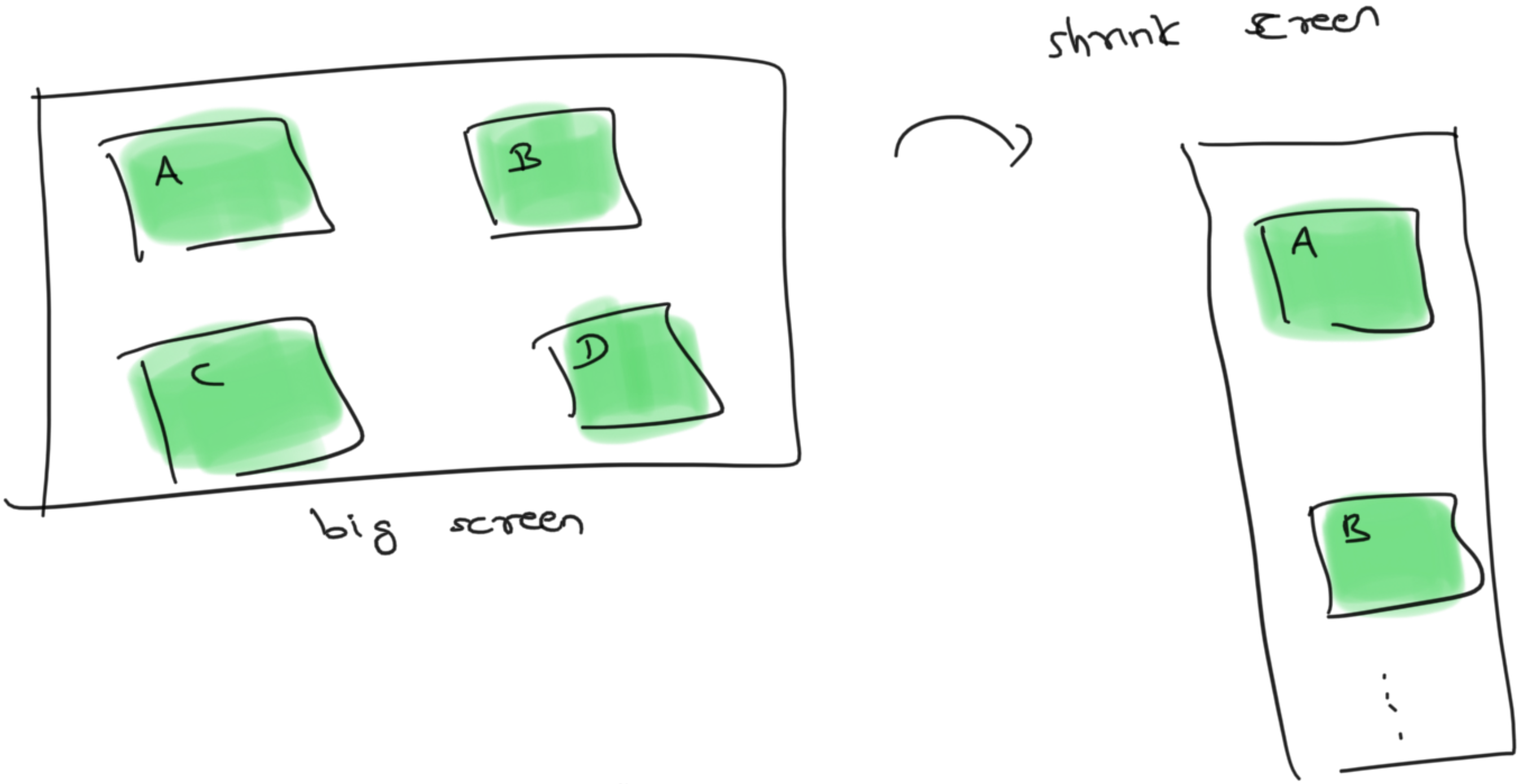
```
<meta     name = "viewport"

     content = " width = device -width ,

           initial -scale = 1.0" >
```

- viewport = visible area of your page
- set width of viewport as width of the device being used

Flex box



```
< div    class = "container>
```

```
    <div>        A        </div>

    <div>        B        </div>
                 ;
</div>
```

CSS

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

flexible container

allow content to wrap around.

```
.container > div {
    background-color: springgreen;
}
```

immediate div child to class containers...

apply this to

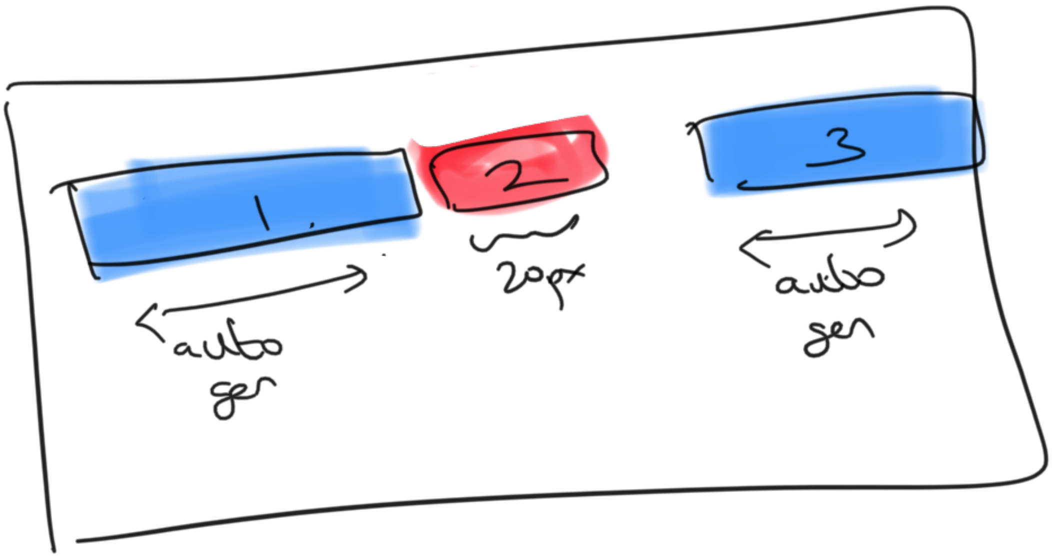
grid layout (grid box)

```
<div class = " x">
    <div class = "y"> 1 </div>
              ;
    <div class ="y"> n </div>
```

```
</div>
```

CSS

```
.x {
        display : grid;
         padding : 20 px
         grid-column-gap : 20 px;
         grid-row-gap : 20px;
          grid-template-columns :
                                      auto 20px auto;
     }
```



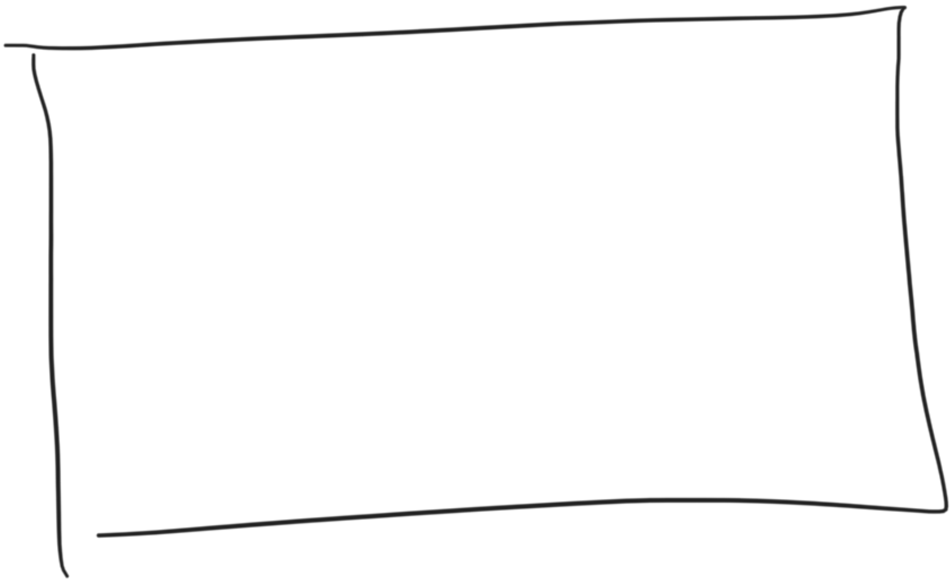
if screen shrunk, $\boxed{1}$ $\boxed{3}$ widths get scaled...

CSS Libraries

Bootstrap (flex, grid, and other things already implemented is CSS style sheet)

< link rel = " style sheet",
href = " link to bootstrap style sheet " >

Boot strap: each row in website → 12 cols



Syntax

< div class= " container">

<div class = " row" > ⟵ row..

< div class = " col - 3" > </div>

this takes up 3 / 12 cols

< div Class= " col- lg - 3 col - sm - 6" > </div>

on a large screen, on a small

</div>

takes up 3/12 cols   screen, takes up 6/12 cols

< /div>

Alerts In bootstrap

```
<div class = "alert alert-primary"
        role = "alert">

<div class = "alert alert-success"
        role = "alert">

<div class = "alert alert-danger"
        role = "alert">
```

→ getbootstrap.com

→ Documentation

→ Components

Sass

(language built on top of css)

- programatically generate css style sheets

- .scss (extension for css style sheets) variables in sass begin with $

```
$color : red ;
   ⋮
   CSS code where you can use $color ...
```

.scss $\xrightarrow{\text{sass}}$ .css

```
sass input.scss output.css
```

→ generates output.css from input.scss

* .css.map ↼ stores the way output.css is generated from input.scss.

Sass can "watch" a file / a directory of files. Any time any of them changes, Sass automatically recompiles the associated CSS files.

```
sass --watch input.scss:output.css
```

Github, other website deployment pages have built in support for sass

.scss → Github

(Github automatically compiles .scss → .css)

Install sass (free software)

Nesting css using sass

.scss file

```
div {
    font-size : 18px;
     p { color: blue;
       }
     ul { color: green;
        }
}
```

for any p – descendant of div, apply blue

for any ul – descendant of div, apply green.

etc.

same as

div p {color: blue;} in usual css.

Inheritance in sass

general styling

specialize to

more specific styles

- - - - - - -

message template

```
%message {

    font-family : sans-serif;

    padding : 20px

    font-size : 18 px

}
```

extend message templates styling to success class

```
.success {

    @extend %message;
    background-color: green;

}
```