MODULE-2

ANATOMY OF CSS RULE...

```
          ↙ Value
 ↗ P {
selector   color: blue ;
 }        ↑
          Property
```

styles within ⇒ <style></style>
Every browser has a default styles.

ELEMENT, CLASS, ID SELECTOR.

Element selector, which selects element.
class selector.
accessed using . then classname

```
<P class="blue"></P>
.blue {
    color: blue;
}
```

id selector.
accessed with # the idname

```
#name {              <P id="name"></P>
    color: blue;
}
```

We can select with multiple selector. with help of COMMA, Eg: div, .blue {

}

COMBINING SELECTORS.

* Element with class selector.

```
P.red {              <P class="red"></P>
}                    <div class="red"></P>
```

* child selector.

```
article > p {        Every p that is a DIRECT CHILD
}                    of article is affected.
<article>
    <P></P>
    <P></P>
</article>
<article><div>
    <P></P>          } NOT AFFECTED.
</div>
</article>
```

*DESCENDANT SELECTOR:

```
article p {
}
```
Here any p that is inside article is affected.

Every p that is (at any level) of article is selected.

all the element with class selector, child, descendant selector. can have elemental, class, id selectors.

```
.coloured p {
}
```

```
#coloured > p {
}
```

PSEUDO-CLASS SELECTORS:

- :link
- :visited
- :hover
- :active
- :nth-child() ↳ No, odd, even

they are grouped together conventionally.

display: block [change default behavior of element from inline to block level]

It will take full width, take a newline by default.

target = " _blank" → new page on click.

element selectors : pseudo selectors.

```
div : nth-child (odd){
}
```

```
div :nth-child(4) {
}
```

CONFLICT RESOLUTION:

Last declaration of css wins.

When no conflict, declaration merge

Inheritance – some property are inherited

↳ cursor, font, font-family, font-size, font-style, font-wt, text-align, visiblity.

BOX MODEL

content	padding	border	margin	height

width.

Sideways cumulatively adds for margins.

Vertically larger one wins by collapse.

```
*{
    box-sizing : border-box. } entire box within limits "WIDTH"
}
```
all elements.

overflow: auto. } scroll bar

↳ Helps in reducing overflow.

Scroll : } both ways...

<link rel="stylesheet" href=" path">

EXTERNAL ↑

INTERNAL <style></style>

INLINE

↳ within elements.

BACKGROUND PROPERTY

background - image: url()
- repeat:
- position:
- color:

} we can just write background:

ie) backgroung: url(' ') no-repeat right centre blue;

POSITIONING ELEMENTS BY FLOAT

Float ⇒ It takes element out of regular documentation flow.

float: left ← everything floats the left out of normal documentation flow

clear: left; This makes that element such that there is no element is floating towards its left.

clear: both.
↳ left + right.

we will do this with 2 column layout.

if we have border, default: content-boxsizing. ↙ Than flex won't work properly.

box-sizing: border-box;

we should % for width, height wrt their viewports!

* floats don't have vertical margin collapse.

* Flexible layouts with float

RELATIVE & ABSOLUTE POSITIONING

static, relative, absolute.

offsets ⇒ top: , bottom: , left: , right:

Static ⇒ It is NOT affected by offset properties.
It is the default for all elements except html

Relative ⇒ element is NOT takken out of normal documentation flow.
Their position is preserved, even if it is moved.

Absolute ⇒ html is relative
All offsets are relative to the position of nearest ancessetor which is NOT static.

Position: absolute.

takken out of normal documentation flow.

* offset the relative container element affects its Contents aswell.

MEDIA QUERIES:

different styles for different screen size.

@media (______)
media features.

We can have more media feature if we Can use logical operators
↳ It can be true or false.

media feature ⇒ max - width: px ✓ Common
min - width: px ✓
orientation: potrait.
Screen
print

@media (min - width: 768px) and (max - width: 991px){ }
→ Common.
, = OR

* Don't overlap range boundaries

@media (min - width: 1200px){ }
@media (min - width: 992 px) and (max - width: 1199px){ },
↑
Breakpoints.

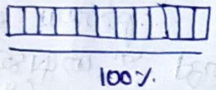
RESPONSIVE DESIGN

↳ site designed to adapt its layout to Viewing envi. by using fluid, Page-based grid, flex images & CSSS media queries.

%

size of device = width = size layout should adopt.

Common layout is 12 colum. grid responsive layout.

12 columns 

$\frac{100}{12} = 8.33\%$

100%

We can have nested grid layout

25% ⇒ 3
50% ⇒ 6
33.3% ⇒ 4

Metatag "viewport"

```
<meta name="viewport" content="width=daice-width, initial-scale=1">
```

↓

this tells the browser the size of device is the size of the browser.

↓ width of viewport

* It tells there is no necessary to zoom.

INTRO TO TWITTER BOOTSTRAP FRAMEWORK

Most popular HTML, css + JS framework.
It is also mobile first project.
Plan the mobile from start.
css framework is mobile ready.

Too big, too bloated. → lot of feature we will NOT use.
So, we can selectively DOWNLOAD..

.min
↳ minified Version

jquery 1.x
jquery 2.x → Internet ex 6, 7, 8.

WKt, how overriding works.

Thus,

```
~~style~~ <link rel="stylesheet" href="css/bootstrap.min.css">
          <link rel = "stylesheet" href ="css/styles.css">
```

our overrides bootstrap.

```
~ <script src=" JS/ Jquery -1. 11.3. min"> </script>  ← DEPENDS ON JQUERY.
<Script src = " bootstrap.min.js"> </script>
<Script src = " JS/ ourScript. JS"> </script>
```

* Console of shows errors.

BOOTSTRAP GRID SYSTEM.

This makes us to create responsive layout.

outter container can be of class.

Wrapper

class = "Container" → breakpoints, responsive.

class = "Container-fluid" → flex, stretches fully, consistent padding.

↳ 15px padding.

class="row" Creates horizontal groups of Columns & applies -ve layout for aligning with the contexts of the container class.

↱ bootstrap → gridsystem

Col - size - span ← How many colms it should span in 12col layout.

↓ size: Screen width identifier. Beyond that size they will collapse and other class may apply

spans above 12 will automatically wraps to next line.

→ Col - md - 4

Col - sm - 6

Col - xs - 6 } Never collapses . . .