CSS Crash Course

What is CSS?

* CSS stands for cascading style sheets and is the language used to style an HTML document.
* It is a stylesheet language not a programming language.
* With CSS you can control the layout of the webpage, the spacing between elements, what color the text should be in and so on....
* Select HTML elements and apply styles.

How to add CSS?

1. Inline styles.
2. Internal stylesheet.
3. External stylesheet

1. **Inline styles:**

```
<h1 style="color: orange">CSS Crash Course</h1>
```

* Inline styles using is not best practice.
* Because it is difficult to maintain.
* Code changability is difficult.

2. **Internal stylesheet:**

* **CSS Rule syntax:**

```
selector   style declaration.
  h1       { colour: orange; }
```

* In internal stylesheet, we can write css rule set in the head tag.

```
<style>
    h1 {
        color: orange;
    }
</style>
```

* The internal stylesheet also not recommended.
* Because it is less efficient.
* Difficulty in code maintance.

3. External style sheet:

Step①: create a new file.

Step②: name the file with extension .css.

Step③: Write a link tag in head section. i.e.

<link rel="stylesheet" href="styles.css" />.

Step④: add the styles in file. and apply the HTML document.

* It is most useful and recommended method of applying styles to HTML document.
* we can use styles for different html documents.

CSS selectors:

1. type selector: Tag selector, apply same style with &dm tagname.
2. universal selector: Select all elements & apply style.

eg: * {
colour: blue;
}

3. class selector.

apply particular styles to the elements.

eg: .colour_text{
color: blue;
}

Id selector:

* In Id selector we can specify a unique name for the selector.

```
Eg: #test {
        color: blue;
    }
<p id = "test"> Id selector </p>.
```

* In advance we can learn about selectors i.e combination of selectors.

Hex codes:

#FFFFFF - white.
#000000 - ~~white~~ black.

color: rgba(0,0,0) - black.
rgba(0,0,0.5) - less intense of black color.

Background color:

```
background-color: orange;
```

fonts:

* font-family: roboto;
* font-style: italic;
* font-weight: bold;
* text-decoration: underline;
* font-size: 20px;

Text layout Properties:

* text-align: center; eg: right, left etc..
* line-height: 2;
* letter-spacing: 4px;
* word-spacing: 20px;

* style for list:

```
ul {
    list-style-type: circle; disc is default.
                     square.
                     None.
                     disc.

    margin: 0;
    Padding: 0;
}.
```

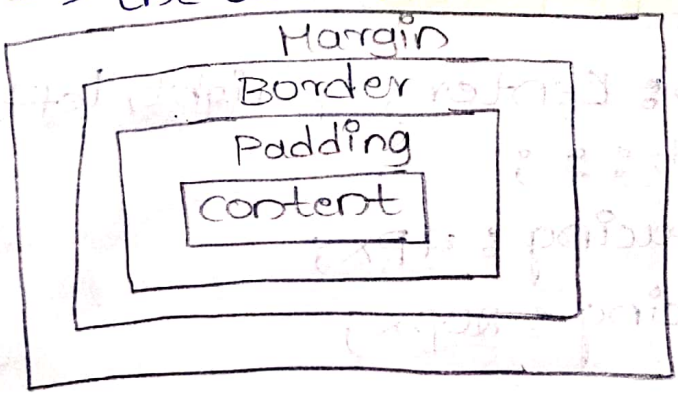
CSS Box model:

* Css Box model mainly revolves around 3 properties.
  1. Order
  2. Padding
  3. Margin.

* In web design, every element is considered a rectangular box

* The css box model is a sort of a standard by which brousers render HTML elements.

* the box model consist of 4 parts.
  - → Margins.
  - → Borders
  - → padding
  - → the actual content.



- box-model {

```
border : 1px solid red ;
height : 50px;
width : 300px;
Padding : 20px;
margin : 20px;
```

}

border

border-radius.
border-color
border-left.
border-right.
border-top
border-bottom.

Padding

Padding-left - Pl
Padding-right - Pr
Padding-top - Pt
Padding-bottom - Pb.

Margin

Margin-left - ml
margin-right - mr
margin-top - mt
margin-bottom - mb.

Cascade:

* When two rules apply that have equal specificity, the one that comes last in the css is the one that will be used.

Specificity:

* Specificity is how the browser decides which rule applies if multiple rules have different selectors but apply to the same element.

eg: element selector → least specificity.
class selector.
id selector.
inline style → more specificity.

exception:

"!important" is used to tell the style is important inorder the specificity.

Eg: h3 {
color: red;
color: blue !important;
}

The o/p should be blue.

Inheritance:

* Some css property values set on parent elements are inherited by their child elements, and some aren't

Eg:
 inheritance

css:

.inheritance {
color: orange;
font-family: 'Segoe UI';
font-size: 30px;
}

th, td {
border: 1px solid #ccc;
padding: 15px
}

border-bottom: 1px solid #ccc;

```
table {
    border-collapse: collapse;
    width: 100%;
}

td {
    text-align: center;
    height: 50px;
    vertical-align: bottom;
}
```

display property:

```
display: inline;
display: block;
```

which makes the element from block to inline and inline to block display.

```
display: none;
```

which hides the element.

Position property:

```
position: static;
```

which makes the element places normal Position.

```
position: Relative;
top: 50px;
left: 50px;
right: 50px;
bottom: 50px;
```

which makes the element places in a desired position.

```
position: fixed;
bottom: 0px;
right: 0px;
```

which the element fixed in same position even the web page scrolled.

Position : absolute ;

bottom : 0px ;

right : 0px ;

→ it can be placed fixed within the parent element.

Position : sticky ;

→ it can make the element sticky. (or) constant while scrolling the web page.

Css units

They are 2 types of css units they are.

1. absolute.
2. Relative.

* Absolute units are generally considered to always be the same size.

Unit	Name	Equivalent to.
cm	Centimeters	1cm = 37.8px
mm	millimeters	1mm = 1/10th of 1cm
Q	Quater-millimeters	1Q = 1/40th of 1cm
in	inches	1in = 2.54cm = 96px
Pc	Picas	1Pc = 1/6th of 1in
pt	Points	1pt = 1/72th of 1in
px	Pixels	1px = 1/96th of 1in.

* Relative units on the other hand are relative to something else, perhaps the size of the parent element's font, or the size of the viewport.

Unit	Relative to.
em	font size of the parent in the case of typographical properties like font-size. & width
ex	x-height of the element's font
ch	the advance measure (width) of the glyph "0" of the element's font.
rem	font size of the root element
lh	Line height of the element
vh	1% of the viewport height.
vw	" " " width
vmin	1% of the viewport smaller dimension
vmax	" " " longer dimension.

Math functions:

1. calc().

```
height : calc(50vh - 100px);
```

2. Max(), min().

```
height : max(50vh, 60px);
height : min(50vh, 60px);
```