

CSS - Test.



1) Types of CSS selector:

- * universal selector
- * ID selector
- * class selector
- * Group selector
- * Element selector.

1. universal selector

```
* {  
    margin: 0;
```

→ applies to every element in the document.

2. Element selector:

```
p {  
    color: red;  
}
```

→ applies to all paragraph elements.

3. ID selector:

```
#box {
```

background: yellow; → applies to only one element.

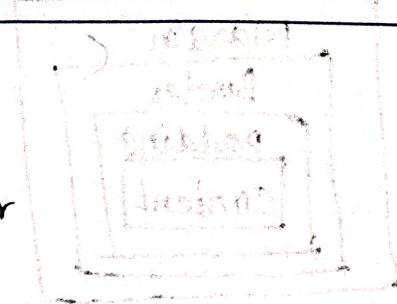
4. class selector:

```
.card {  
    border: 1px solid black;  
}
```

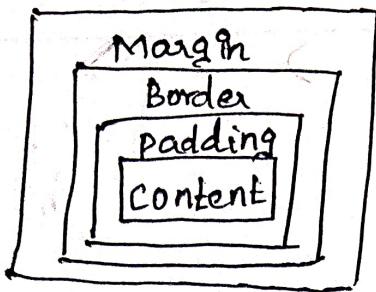
→ applies to all elements with class "card".

5. Group selector:

```
h1, p, div {  
    color: blue;  
}
```



2) CSS Box Model :



```
.box {  
    width: 200px;  
    padding: 20px;  
    border: 5px solid black;  
    margin: 10px;  
}
```

* Content means actual text or image

* Padding means space inside the box.

* Border = box outline

* Margin = space outside box

3) display : inline vs block vs inline-block.

* display : inline means contents are shown in same line.

Example

```
span {  
    display: inline;  
}
```

* display : block

* It will take full line. It can be

set width and height.

Ex:

```
div {
```

display : block;

```
}
```

color : black; font-size : 16px;

background-color : white;

* display : inline-block;

* Same line + can set size.

Example :

```
button {
```

display : inline-block;

```
}
```

font-size : 20px;

4) Background Properties:

```
body {
```

background-color : white;

background-color : red;

background-image : url("image.jpeg");

background-repeat : no-repeat;

background-position : center;

`background-size: cover;`

g.

Properties:

- * `background-color`

- * `background-image`

- * `background-repeat`

- * `background-position`

- * `background-size`

5) position property (with Example)-

Types:

- * static = default

- * relative : move from own place.

- * absolute : move from parent

- * fixed = stick to screen

- * sticky : mix of relative + fixed

Example:

```
.box {  
    position: relative;  
    top: 20px;  
    left: 50px;}
```



b) ~~positio~~ float & clear:

* float means push the element

left or right

* mostly used for image + text layout.

Syntax:

.image {

float: left;

}

* float values are

float: left;

float: right;

float: none;

Example:

img {

float: left;

margin-right: 10px;

}

etc

clear:-

* clear property is used to stop floating elements and move element to next line.

syntax:-

• footer {

 clear: both;

}

clear values:-

clear: left;

{left: none}

clear: right;

{right: none}

clear: both;

{left: none; right: none}

clear: none;

{left: auto; right: auto}

Example:-

<div class="footer"> Footer </div>

• footer {

 clear: both;

{left: none; right: none}

 background: black;

{color: white;}

 color: white;

}

7) Flex direction :

* it decides how items are arranged
row wise or column wise.

Syntax:

- container

```
display: flex;
```

```
flex-direction: row;
```

g

flex-direction values:

```
flex-direction: row;
```

```
flex-direction: row-reverse;
```

```
flex-direction: column;
```

```
flex-direction: column-reverse;
```

Example:

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

```
<div class = "box"> One </div>
```

```
<div class = "box"> Two </div>
```

```
</div>
```

css

.container {

display: flex; width: 100%; height: 100%;

flex-direction: column;

}

.box {

padding: 20px;

border: 1px black;

}

Output :

One

Two

8] Justify content property:

* Justify content used for align-items

items horizontally inside the flex container.

Syntax:

.container { justify-content: center; width: 100%; height: 100%; }

display: flex;

justify-content: center;

}

Justify Content values:

- * justify-content: flex-start; left margin more space
- * justify-content: flex-end; right margin more space
- * justify-content: center; middle alignment
- * justify-content: space-between; align-items: space-around
- * justify-content: space-around; align-items: space-between
- * justify-content: space-evenly; width: 111px

Example:

```
<div class="container"><p>Home</p>
  <p>about</p>
</div>
```

CSS:

```
.container {
```

```
  display: flex;
```

```
  justify-content: space-between;
```

```
  gap: 10px;
```

```
  width: 100px;
```

```
  justify-content: space-around;
```

```
  align-items: center;
```

```
  text-align: center;
```

```
  font-size: 1.2em;
```

```
  font-weight: bold;
```

```
  color: red;
```

```
  border: 1px solid black;
```

```
  padding: 5px;
```

a) Navigation Bar :-

* A navigation bar is a menu used to move from one page to another.

Types of navbar:

* Horizontal Navbar

* Vertical Navbar.

Example :-

```
<ul class="nav">  
    <li> Home </li>  
    <li> About </li>  
    <li> Contact </li>
```

css:-

```
.nav {  
    display: flex;  
    background: black;  
    padding: 0;  
}  
.nav li{  
    list-style: none;  
    color: white;  
    padding: 15px 20px;  
}
```



10) Create a Flex layout:

HTML:

```
<div class="header">  
    <div class="logo"> Logo </div>  
    <div class="title"> Website Title </div>  
</div>  
  
<div class="content">  
    <div class="sidebar"> Sidebar </div>  
    <div class="main"> Main Area </div>  
</div>  
  
<div class="footer"> Footer </div>
```

css:

```
.header {  
    display: flex;  
    justify-content: space-between;  
    background: lightblue;  
    padding: 10px;
```

3

.content {

display: flex;

}

.sidebar {

width: 30%;

background: lightgray;

"about""help"

padding: 20px;

"contact""news"

}

.main {

width: 70%;

background: lightgreen;

<h1>"news"</h1>

padding: 20px;

}

.footer {

<div><h3>"Footer"</h3>

<div> background: black;

color: white;

text-align: center;

padding: 10px;

<div><h3>"Footer"</h3>

}

background-color:

background-color:

background-color:

background-color:

background-color:

background-color:

background-color:

background-color: