

CSS - Cascading style sheet

CSS Attributes:

background-color

height

width

border-style

border-top-style

color

text-decoration

display

visibility

position

text-align

font-size

font-family

font-weight

line-height

float

Inline CSS - used beside the html tag, have to put a ; after line completes

Internal CSS - inside the html file itself

it is written in head tag

```
<style> Your CSS Code </style>
```

EX:

```
<style>
```

```
    body {
```

```
background-color: red;

}

</style>
```

All websites has some default css added by browser. So if something doesnt work as expected try to overwrite the particular attribute by some googling.

External CSS: - Will affect all the pages in your website

- Create a file called styles.css
- In html file in the head, use link element. use that css file in the href attribute inside link element.

Hierarchy of CSS is inline,internal and external. It means if the same property is given with the help of all 3 css, the priority would be given to the inline CSS, then internal and only after that external.

Selectors :- 3 types to select an element - tags, class, id

for **tag:**

```
element {

}
```

for **class:**

```
.class_name {

}
```

for **id:**

```
#id_name {

}
```

The difference between ids and class.

- multiple elements can have same class but only one element can has an id.

Pseudo class: It has the : as the prefix. we can use them to indicate different state of an HTML element.

Ex : Hover class :

```
a:hover {

    css code;
```

}

It will hover the a element.

div: - Its an html tag which works only with css. It is used to divide the content of the website.

Height, Width, Border, Padding and Margin:

Height and **width** are simply height and width of the box.

Border is a border which will not include the inside of the box. So if you have box of 300x300 and border of 15px, it will contain the space for 315x315.

Padding : Padding is the space between the border and the content of the box. (inside the box space).

Margin : Margin is the space between the border and any other element near the bordered box. (outside the box space)

span: - Span is html tag usually only used with css. it helps us target only particular part of text and style it differently than the rest of the text. we can use it with block elements like p and h1. If we want to target a particular part of the text and dont use span, then we would have to put that in another p tag. Which will lead to break the text in 2 lines since p is block element. thats when span is useful. Because span is an inline display element.

Display Property:

Block: - Block elements take the entire width of the page blocking for any other elements to be next to them. Ex: - h1-h6,p,div,ol,ul,li,form

- You can change width in block elements.
- You can change its display property to inline which will make them inline elements but after then you will not be able to change the width.

Inline: - Inline elements only take space which is required and can fit another elements next to them Ex: - span,img,a

- You can not change width in inline elements. the width is content based.
- You can change its display property to block which will make them block elements but after that the elements will not stay on the same line.

Inline-block: - Its basically the both block and inline. You will be able to place elements right beside each other and also be able to give them the width.

None: - It simply remove the element as well as the space it contains.

Using **visibility** property we can hide the element but its space will not be removed.

Position property:

In order to position the element on the screen the way I want.

static: - Just go along with default html rules.

relative: - it positions the element relative to the static position

- If we change an elements position property to relative, and describe left, right, top and bottom, then it would position the element the way we described.
- When we change the relative position of an element, it doesnt affect the position of anything else on the screen. Which means it would just get overlapped.

absolute: - It positions the element relative to the parent element.

- It will affect the position of other elements on screen. If there are three divs in stacked 100x100. and if we make the first div's position to absolute and do left: 200px; , In this case the 2nd div will come on top and the first div will move to the right of 2nd div.
- If we give an element position of absolute, if that element doesnt have any parent, it will take the body as parent. If it does have parent, we have to give that parent element the position of relative. if we dont, again it will consider the body as parent.

fixed: - It will stay at the fixed position that we mentioned even if our website is scrolled.

To center elements:

We can use a property called **text-align**. Have to give this to the parent container.

If we have given width to the block element, text-align wont work. So another way to center the element is **setting margin to auto**. It will center the element either vertically or horizontally.

Font styling:

There are two major font families. Serif and sans-serif. serif has the edges around letters while sans-serif doesn't.

If you want to see people seeing exact same font when they are visiting the website, you have to do font embedding which will embed your font into your website.

To do that, visit fonts.google.com

Select the fonts

copy the link

paste it in the index.html with all the other link elements.

Font sizing:

Should use % in font-size compared to px to make font size more dynamic.

Another way to make fonts dynamic is to use em.

16px = 100% = 1em

em and % are inherited from parent. For example if i have set the font-size to 2em in the body, and i set the font-size of 5em in h1, then total of 7em would be applied to the font-size. To prevent that you have to say 5rem which means 5 root em. then it will ignore the parent font size and only 5em will be applied.

Floating:

float property would make the element float right or left in the parent div or any parent element.

If we don't want to use float on any particular element, we can use clear property which can be set to left and right. it simply tells that clear the left or right part of this element.

Ideal case to use float is when you want to **wrap text** beside an element like image.

CSS button generator is a tool which will generate us a great button and its code. We just have to put that in our css file.