## Selectors:

1. Type selector

There are the html elements that can be used for styling

Ex:

h1 {

color: green;

background-color: white;

font-size: 20px;

}

1. Class selector

In type selectors the keyword we use for styles is the element name which is default html keywords

If we want more specific style to few elements keeping the common style as-is, we can go for class, class is an attribute and value can be customised and this custom value will be used while giving styles. We use dot to represent class

Html -- <h1 class=”mystyle”> <h1>

Css

.mystyle {

/\* styles \*/

}

1. Id selector

For Html elements, there is another attribute called id, we can provide styling based on these. called id selectors. There are other uses of id apart from providing styles.

1. Universal selector

-- irrespective of the specific code, if we want to apply styles to entire page/body, we can use these universal selectors.

-- we can use \*

\* {

}

Colors: (used for color and background-color styles)

We have 140+ colors name

1. Hex values

Instead of using three numbers between 0 and 255 , you use six hexadecimal numbers. Hex numbers can be **0-9 and A-F** . Hex values are always prefixed with a # symbol.

Ex: **#rrggbb**

Where rr (red), gg (green) and bb (blue) are hexadecimal values between 00 and ff (same as decimal 0-255).

For example, #ff0000 is displayed as red, because red is set to its highest value (ff) and the others are set to the lowest value (00).

To display black, set all values to 00, like this: #000000.

To display white, set all values to ff, like this: #ffffff.

3 Digit HEX Value

Sometimes you will see a 3-digit hex code in the CSS source.

The 3-digit hex code is a shorthand for some 6-digit hex codes.

The 3-digit hex code has the following form:

**#*rgb***

So, if we have #ff00cc, it can be written like this: #f0c.

1. Rgb values – red green blue

**The max value of each of the colors is 255.** **The minimum value is 0**. Colors are almost always written with the Red value first, the Green value second, and the Blue value third. Memorize "RGB" and you will remember the ordering.

1. RGBA values – extra alpha

An RGBA color value is specified with: rgba(red, green, blue, alpha). The alpha parameter is a number between **0.0 (fully transparent) and 1.0 (fully opaque)**.

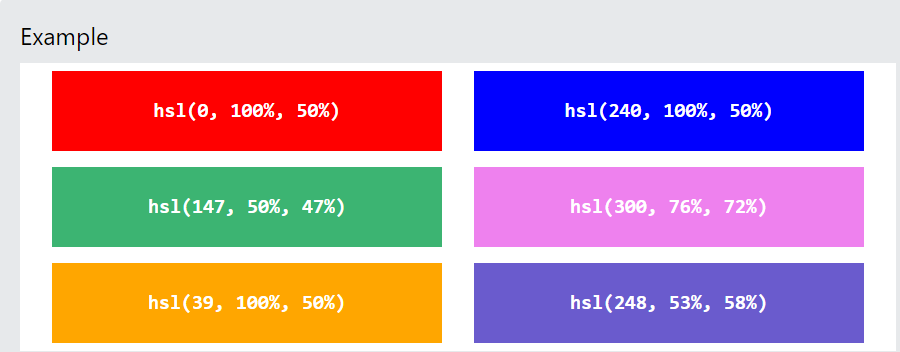
1. HSL Values - **Hue**-**Saturation**-**Lightness**

Hue is a degree on the color wheel from 0 to 360. Hue literally means colour.

0 is red, 120 is green, and 240 is blue.

Saturation is a percentage value. 0% means a shade of gray, and 100% is the full color.

Lightness is also a percentage. 0% is black, 50% is neither light or dark, 100% is white



1. HSLA values – extra alpha

The alpha parameter is a number between **0.0 (fully transparent) and 1.0 (fully opaque)**.

Ex: hsla(360, 50%, 25%, 0.5)

1. Opacity – transparent and opaque ness

Instead of giving alpha along with hue, saturation and lightness, we can give it as additional property as below

Ex: background-color: **hsla**(360, 50%, 25%); **opacity:** 0.5;

## Gradients:

This ain’t applicable for background-color.

1. Linear (down, up, left, right, diagonal)

Linear-gradient(<*arguments*>)

For diagonal – we can use ‘to bottom right’, ‘to top left’, etc.,

1. Radial (defined by their center)
2. Conic (rotated around center point)

Conic-gradient(<*arguments*>)

## Units & Sizes:

Expressing the length in

1. Absolute – cm, mm, in, px, pt (points), pc (picas)

px is mostly used

1 in = 96 px = 2.54 cm

Ex: height: 250px; -- indicating the height absolutely

1. Relative

Based on another element

Keywords used:

1. ‘em’ relative to font-size

ex – relative to hex , ch, rem, vw, vh, etc.,

Ex: height: 4em; -- indicates that it multiplies current height with 4.

Current height means whatever is applicable to that element without applying style.

## CSS Images:

1. Thumbnail

Properties:

Border

Padding

width

1. Transparent image

Opacity: 0.0 to 1.0

1. Rounded-image

border-radius, border-top-right-radius, etc.,

1. Responsive image

max-width, height

1. Center an image

Margin-left, margin-right

display: block; -- layout property