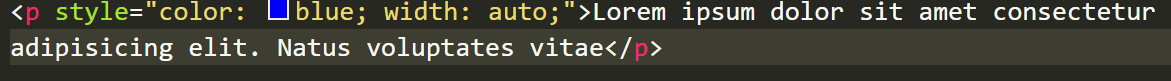
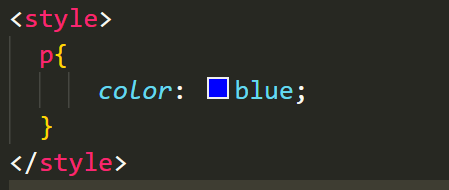
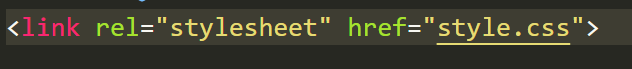
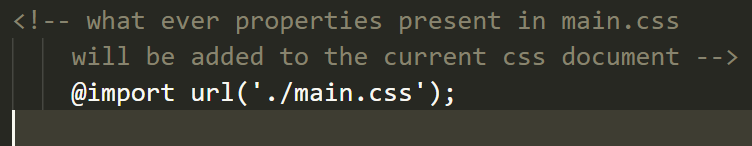
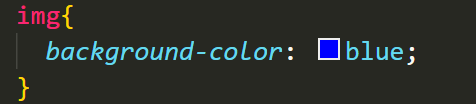
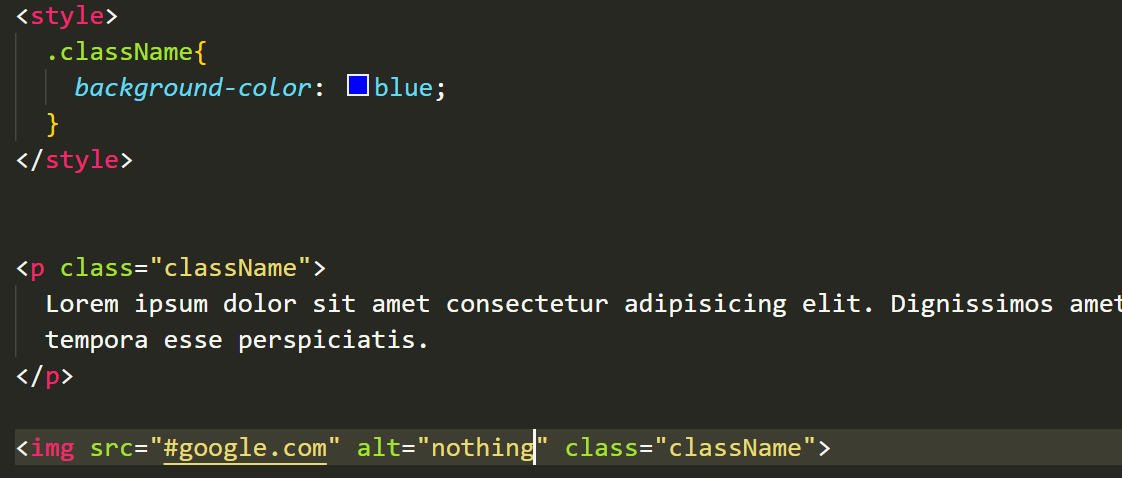
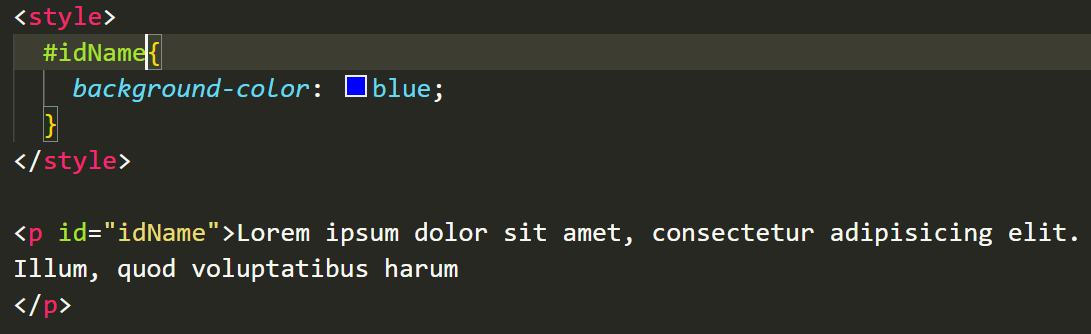
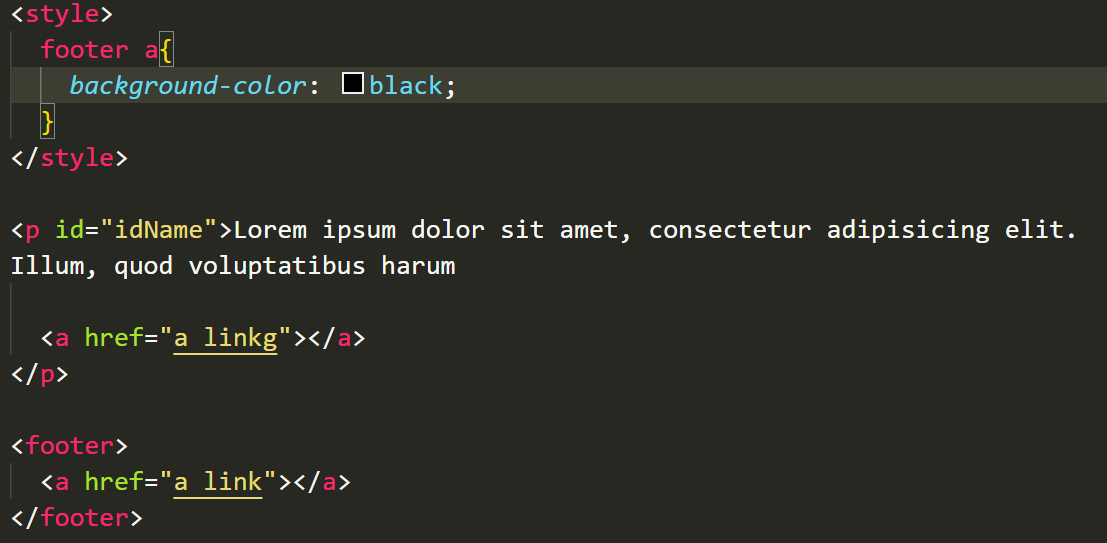
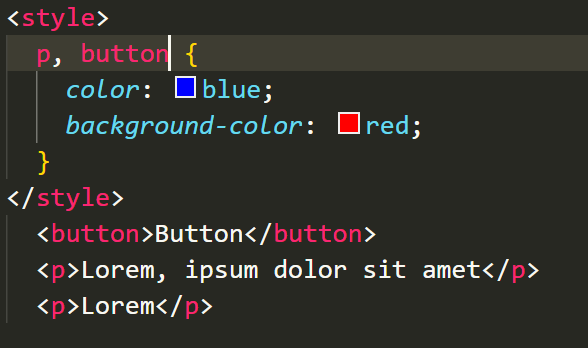
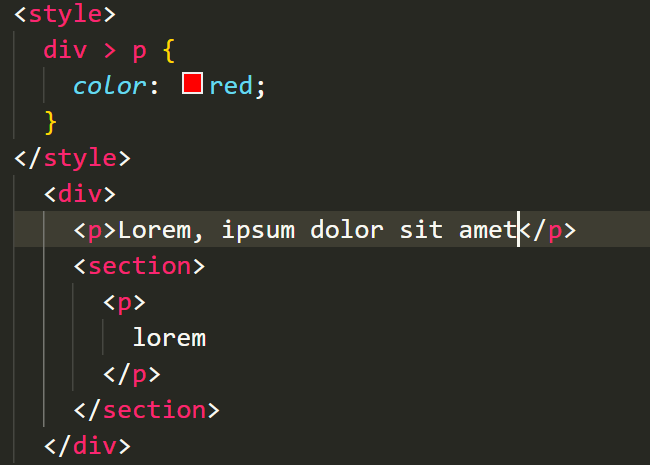
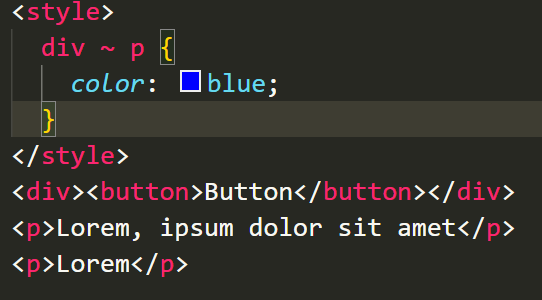
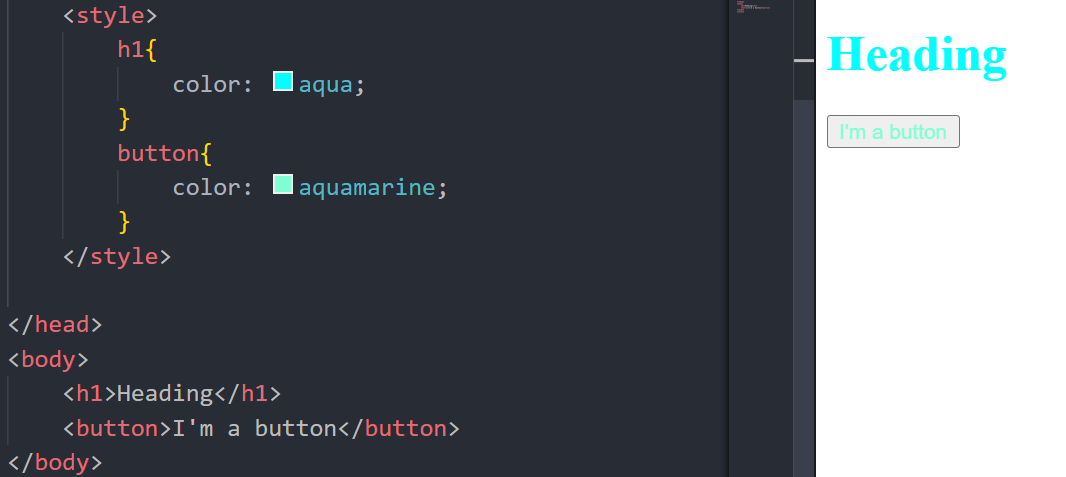
# Cascading Style sheet

* **Inline Css** is used for styling in the tag where we style elements in the tag
  + 
* Internal css where we give css inside html doc
* It is not recommended as we need to change all docs if any change to navigation
* **External Css:** Where we link css document using relation of stylesheets
  + 
  + 
* **Importing other css in our css**. We add import url which imports css form other css doc to our file
  + 

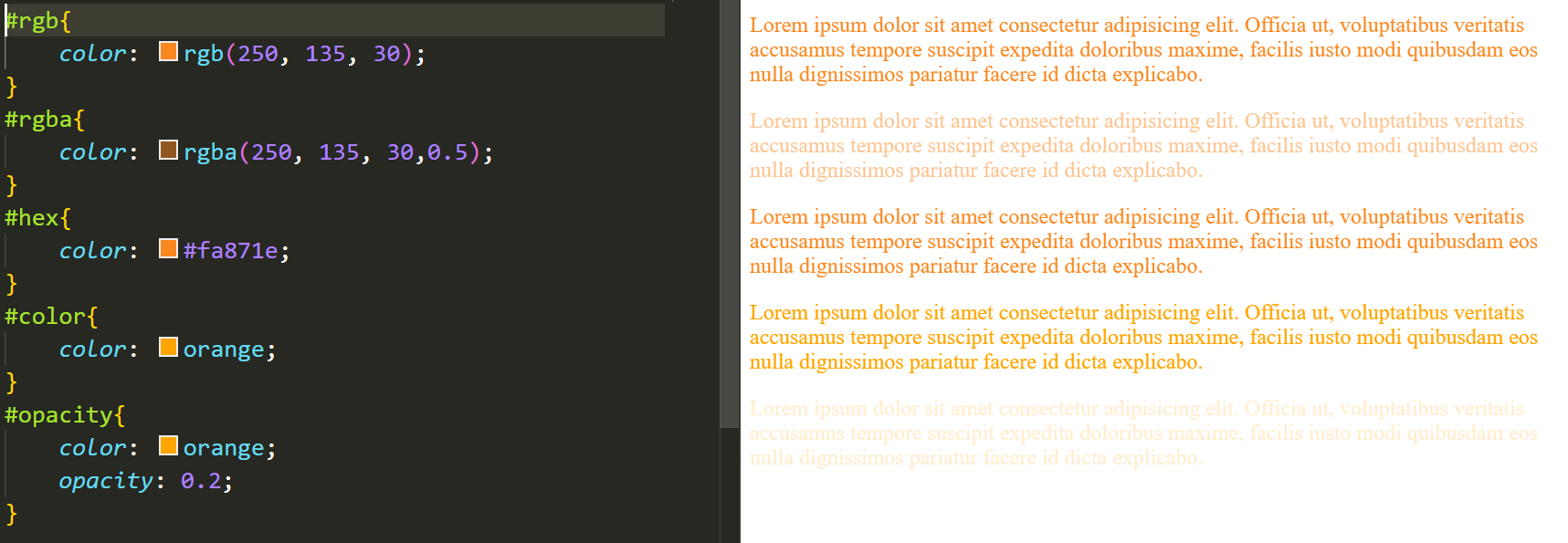
## **Selection:**

* Selectors are used for selecting the particular elements to style them
* **Element Selectors:** 
  + These are used to style the elements based on the tags
  + 
* **Class Selector:**
  + Styling is applied on a group of elements which is having same class name
  + 
* **ID Selector:**
  + When we need to style a particular element we give ID to that tag using which we can style that element
  + 
* **Contextual Selector:**
  + It is used for selecting tags which are inside other tags it will select all the p tags that are inside main tag even if it is nested
  + In Below scenario if we need to style only anchor tags which are in footer region then we can use this
  + 
* **Attribute Selector:**
  + Is used for selecting the tags which are having that attribute
  + 
* **Pseudo Classes:**
  + These classes work based on an action like hover active
  + So when we hover that style will apply
  + 
* **Pseudo Elements:**
  + These use two colons and is used for styling the individual elements instead of a whole tag
    - 
* **Group Selectors:**
  + Is used for styling multiple tags in one go which are having same styles
  + 
* **Relational Selectors:**
  + **Child Selector:**
    - Will select all the p tags which are direct child of div(main tag)
    - 
  + **Adjacent Sibling:**
    - Will only select the adjacent element the one that is next to dvi =>p. If the p is second element like div=>secton=>p all the in same level but as it is not right next to the div so it won’t select
    - 
  + **General Sibling:**
    - Will select all after the sibling tag means all p tags in below example
    - 
  + **Descendant Selector:**
    - Will select all elements which are below parent tag
    - 

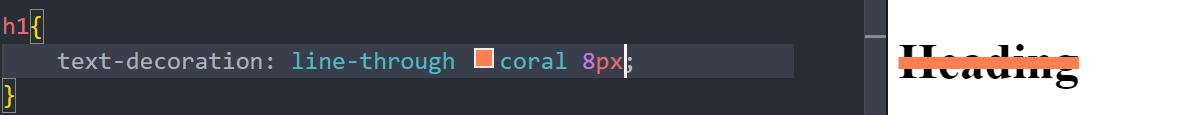
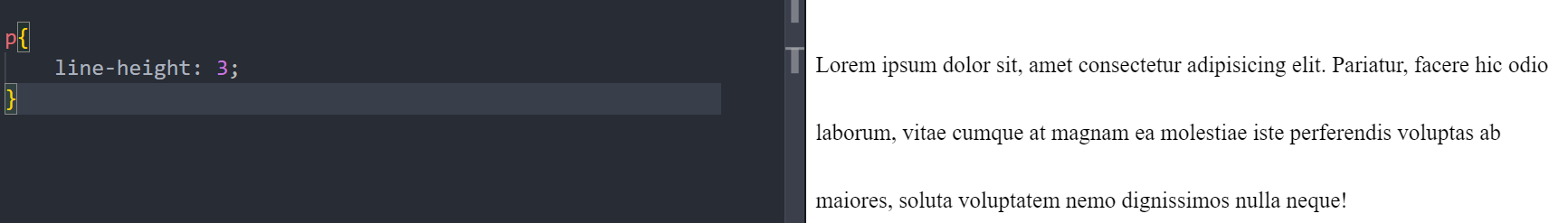
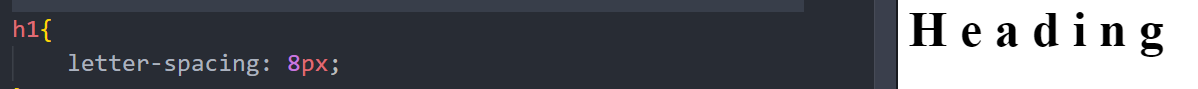
## Styling:

* Can do two types of styling one is internal where we use style tag inside head tag of html tag
* 
* External css is where we create a css file with all the required prop and it will be linked to html doc using link tag
  + <link rel="stylesheet" href="style.css">
  + </head>

## Color:

* color and background-color tag are used to color texts
  + RGB
  + Hex
  + Name
  + RGBA
    - A for alpha which can be used for opacity
    - 1 no transparency 0=> transparent
    - rgba(0,0,0,0)
    - This one doesn’t go with text where it is applied then only it will be applied
  + Opacity:
    - Opacity:1; => this is applied to the whole element
* 

## Text Properties:

* text-align -
  + Alignment of text horizontally within an element. Center, Right, Left and justify
  + Justify will spread out elements to fill the box
  + text-align: right;
* font-weight -
  + Used for thickness of font. Can use number and keywords like bold, normal, lighter, bolder
  + font-weight: 1000;
* text-decoration -
  + Used for underline or color of underline or thickness
  + Underline, over line, strikethrough, dotted, wavy double
  + 
* line-height -
  + Used for height of cursor line that it takes
  + 
  + 

## Font:

### size:

* + Px/pixels:
    - Absolute won’t change with respective to website so can’t be used with responsive websites
  + Percent/%
    - It will take the half the percentage value of the parent element width:50%
  + EM:
    - It means if 1em it will take same font size as it’s parent
    - 2em means twice the font size of its’ parent
    - Used for relativity with respective to another parameter to make it dependent on that parameter
    - Con: as ems are dependent on parents font size so for nested list it will take prev ones em and gradually reduce
    - 1em => 0.5em => 0.25em
  + Rem:
    - Root em => so if root is having a size then it’s nested elem will also have same size

### font-family:

* + We can include a set of fonts so that it will take the font that we have in our browser cause all fonts are not installed in our browser
  + Safe font browser website can help you choose

## Selector:

* Pseudo elements:
  + We need to use two semicolons to use them instead of one
  + These are used to select letters inside an element
  + a::first-element {.........} => all anchor tags first element behavior can be modified by this
  + Selection => when selecting elements text we get this behavior

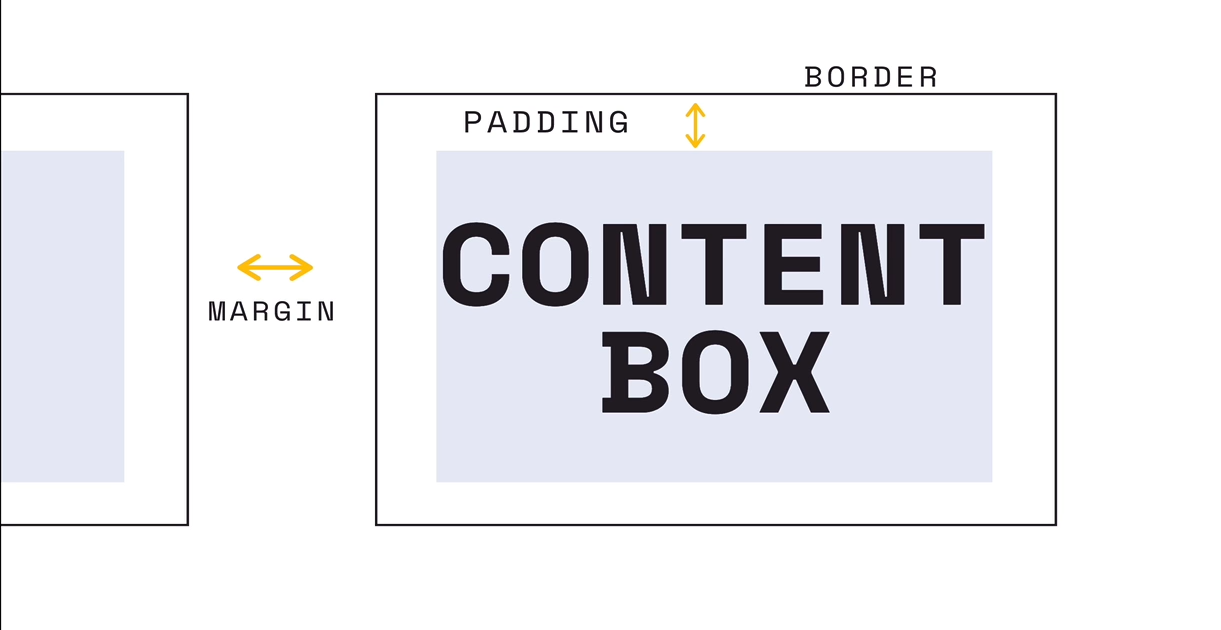
## Width Height:

* Width and height property are used for setting or defining our background box
* Width also defines our text spread width
  + Width: 250px;
  + Heighth: 250px:

## Background:

* **Background-image:** 
  + Property is used for setting up an image as a background
* **Background-repeat:** 
  + As we have a bigger box so image might repeat we can change its behavior using this
* **Background-position:** 
  + top/bottom/center/…. Can be used to set the image form where it needs to start
* **Background-siz**e:
  + Is used for sizing our image to fit to background box

## Box Model:

* Width and height controls the content box
* 
* **Border:**
  + We can also set each line style like border-left-color/width/style… or we can use a normal border where we can define all prop
    - border: red dashed 2px //width style color
  + border-radius:
  + border-width:
  + border-color:
  + border-style:
    - We can set two parameter at single time so it will take for 2sides 1 parameter
    - border-style: dashed solid
  + Box-sizing:border-box
    - This property is used to make the content box+border size = content width
    - If this property is not used then content width=content box and border size will be extra
* **Padding:**
  + This can also be applied shorthand like
  + padding: 1px 3px 6px/ 1px 8px => top horizontal bottom/top bottom => clock
* **Margin:**
  + Just like padding we have shorthand of it
* By default the width and height that we give is considered only for content box but when we add padding or border then those widths will be added to the actual width that we give so we need to adjust them accordingly
  + **Box-sizing**: content-box
    - Will set the content to the width and any padding or border will be added extra
  + **Box-sizing**: border-box
    - This will include padding and border inside the width that we specified

## Display:

* It is used to make make tags inline and block with it’s property
  + display: inline/block/inline-block/None
* In general we can’t give inline properties width cause it takes only the space as it consumes so for that reason we use inline-block
* Inline-block we can set a inline prop width
* None display prp is used to hide elements which makes it not to take more space

## Float:

* These are used to float our tags to left or right side and we can also give the width to them based on the parent width
  + float: right;
  + width: 30%;
* If anything here are overlapping then we can use
  + clear:right; clear:left; clear: both

## Position:

* Used for positioning elements with top/right/bottom/left properties
* Need to use negative to shift to negative side
  + Top: 100px => This needs to be used along with a position prop
* **Static:**
  + Won’t change anything will be as is the element
* **Relative:**
  + Will change the position of element relative to it’s actual position
    - Position: relative;
    - Top: 100px;
* **Absolute:**
  + It will place the position based on the nearest ancestor or based on it’s parent tag
* **Fixed:**
  + It will be placed relative to the initial containing block
  + It will be started their even we scroll down
* **Sticky:**
  + It will be sticked to the screen and by default will start from actual position

## **Transition:**

* Transitions are used to smoothly change style by giving time
  + transition: 1s;
* We can give more parameters
  + transition: propName/time/timing fuc/delay
* transition: border 2s;

## **Transform:**

* Used for transforming(rotating or moving elements) using different properties
* **Roate:**
  + transform: rotate(45deg)
  + transform-orgin: center/bottom right/ top /…..;
* **Scale:**
  + transform: scale(0.5);
  + transform: scale(x,y);
  + transform: scalex();
* **Translate:**
  + transform: translate(x, y);
  + transform: translatex();
* **Skew:**
  + Slightly changing it’s angle
  + transform: skew(30deg, y);
* transform: rotate(45deg) scale(1.2) translate(-200px)

## **Box-shadow:**

* Is used to give a shadow effect around the body
* box-shadow: x, y, y => look internet
* **Cursor**
  + **Cursor: pointer;**

## **Flex-Box:**

* Flex property is used to fit in all the elements inside the main container
* If the main container shrinks then the elements inside it will also shrink
* Even if elements are larger still they will be shortened to fit inside the main container
* Consider main container and elem as div below structure
  + Container
    - div\*4
  + display: flex => prop is used on container so that all div can apply
* **Flex-direction:**
  + row/row-reverse column/column-reverse;
  + Are the properties of in which the div flow
* **Justify-content:**
  + Is used for adjusting content on main-axis x
  + flex-start/ flex-end/ center
  + **Space-between**
    - Will take all the space and add it between elements
  + **Space-around**
    - Will create space around and between elements just adds at start and end
  + **Space-evenly**
    - Will space evenly along all places
* **Flex-wrap:**
  + Will wrap in next line if space is not enough in the main section
  + Flex-warp: wrap/wrap-reverse/nowrap
* **Align-items:**
  + It uses the cross axis to align items does have same property as justify-content but it moves only vertically
  + **Align-self:** 
    - Is used for aligning single elements
* **Flex-sizing:**
  + **Flex-basis:**
  + **Flex-grow:**
    - It will grow if empty space is available to fill the empty spaces
    - We can set max-width:300px so that it won’t grow more than that width
      * flex-grow: 2 => means it will take 2 times of the spaces compared to other
  + **Flex-shrink:**
    - Makes the elements shrink if we resize window and the numbers will make it shrink faster with respective to small changes
      * flex-shrink: 5/0 means no shrink will happen

## **Media Queries:**

* Media queries are used for changing styles of our display based on different properties like width, height and screen type etc
  + @media (min-width: 800px) and (max-width: 1000px){ range=800-1000
    - H1{
      * Background-color: red:
    - }
  + }