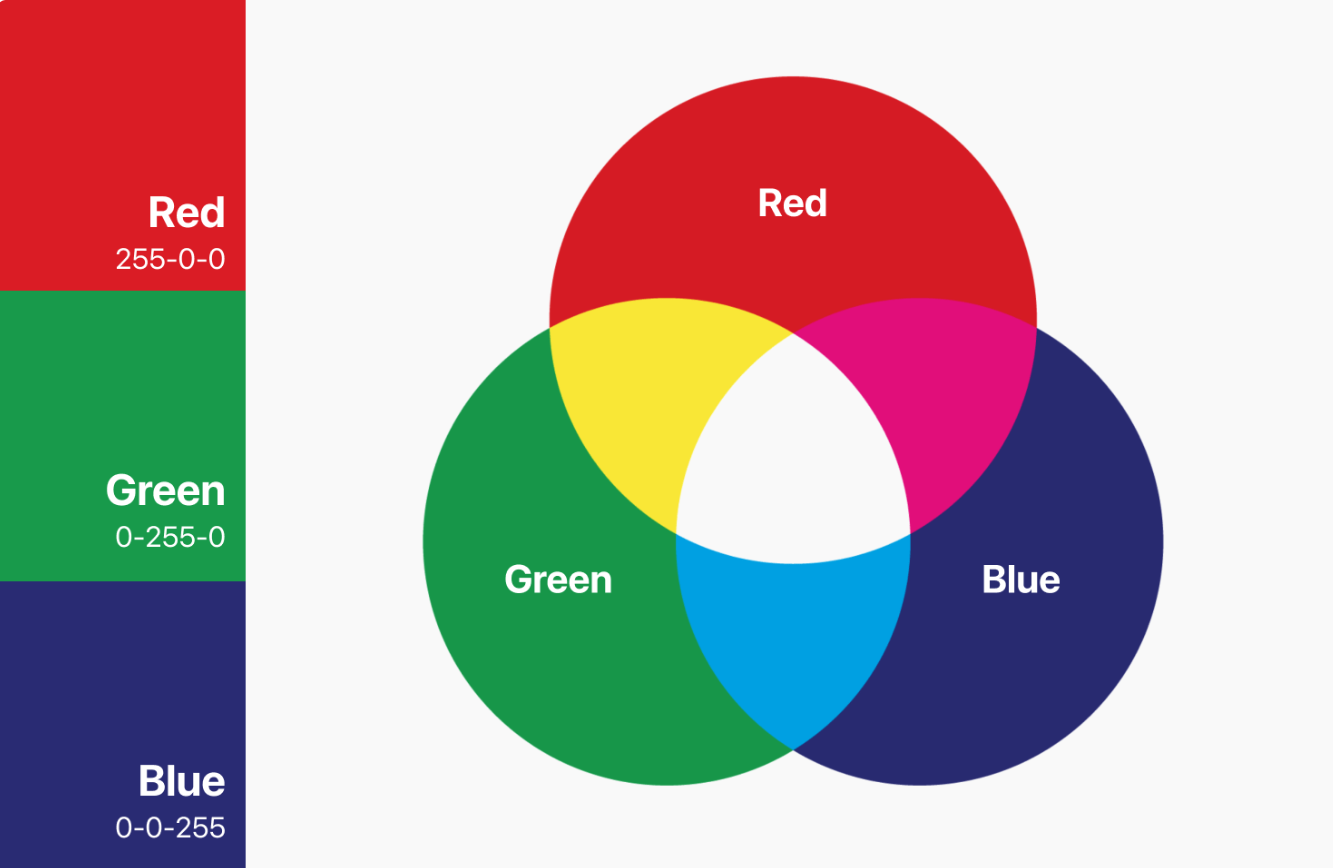
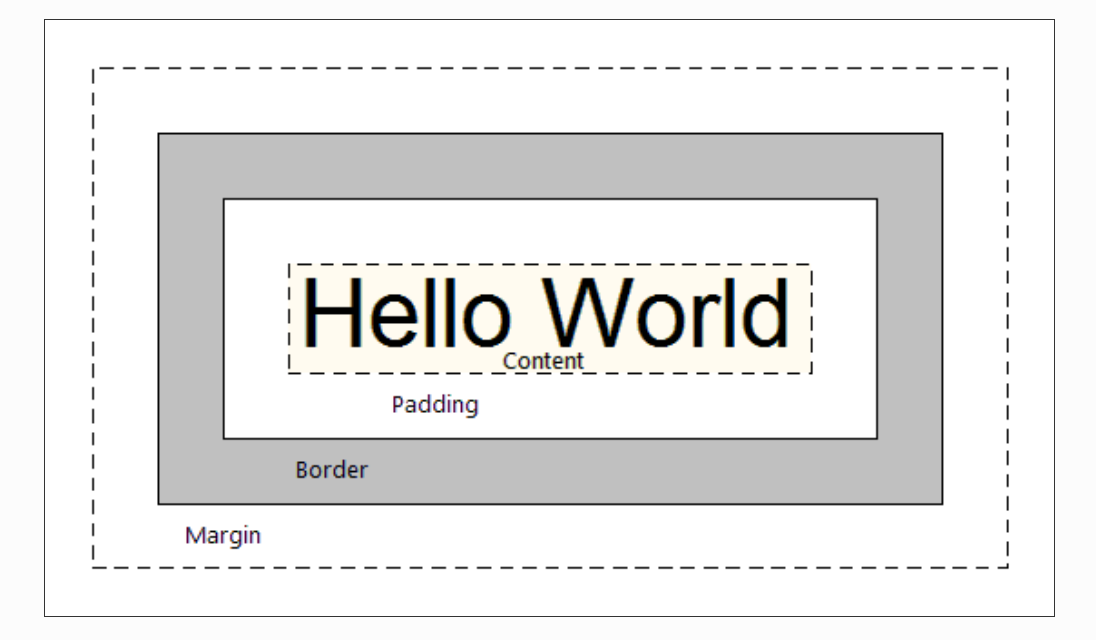
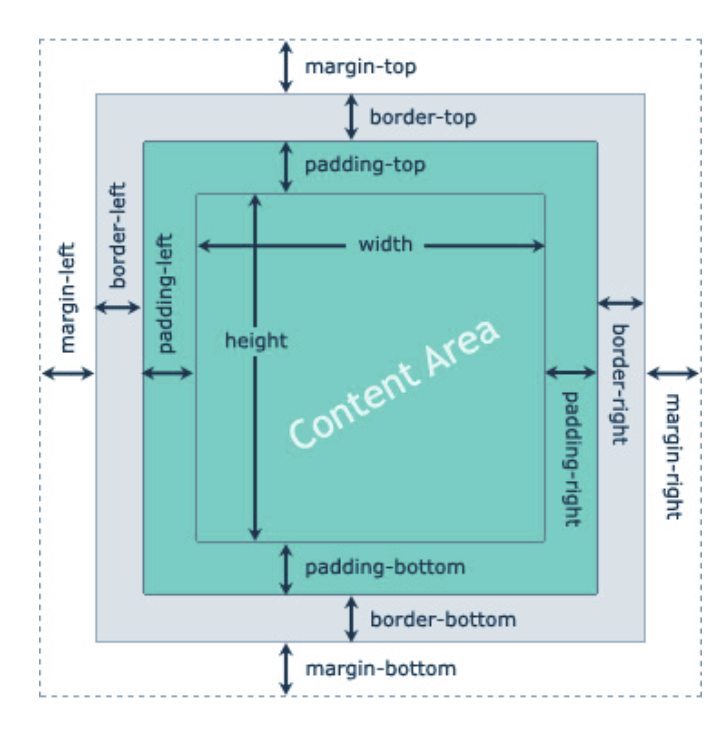
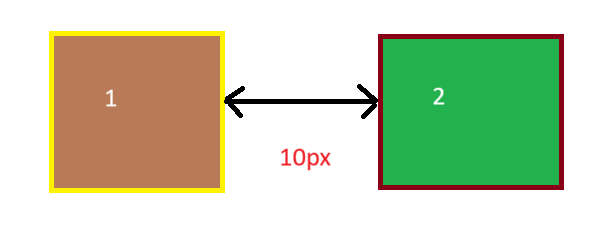
1. HTML: Structure/Skeleton  
   CSS: Style  
   JS: Logic
2. CSS:  
   - Cascading Style Sheet  
   - A language that describes style of a document.
3. 3 ways to add style to a document:  
   (i) External Stylesheet  
   (ii) <style> tag  
   (iii) Inline style.
4. **External stylesheet:**  
   - write CSS in a separate document and link it in HTML file via <link> tag  
   - <link> is defined inside <head> of HTML document  
   - <link rel =”stylesheet“ href=”<RELATIVE PATH TO CSS FILE>”>
5. **<style> tag:**  
   - write CSS inside <style> tag which is defined inside <head> of HTML.
6. **Inline style:**  
   - attached directly to HTML element.  
   - Ex: <div style = “width: 40px; height: 100px;”>…</div>
7. Property : **color**- color: green  
   - used to set the color of foreground.
8. Property : **background-color**   
   - background-color: green  
   - used to set the color of background.
9. **RGB color system**:  
   - rgb(a,b,c)  
   - Ex: color: rgb(200,120,200)  
   - Range of a,b,c = [0,255]  
   - Lowest value = 0; Highest value = 255  
   - a = content of red in resulting color  
    b = content of green in resulting color  
    c = content of blue in resulting color  
   - we can make any color by a certain combination of red, green, and blue.  
   - to make standard red color: rgb(255,0,0) //means only red color and no green, blue  
    to make standard green color: rgb(0,255,0) //means only green color and no red, blue  
    to make standard blue color: rgb(0,0,255) //means only blue color and no green, red  
    to make standard yellow color: rgb(255,255,0) // means only green and red, no blue   
    
10. **RGBA color system (Alpha Channel)** :  
    - color: rgba(a,b,c,d)  
    - everything same as RGB, only parameter d is additional.  
    - param d represents opacity of color(alpha). Value ranges from [0,1]  
    - more the opacity, darker the shade of color and vice versa.
11. **Hexadecimal** **color system**:  
    - #aabbcc  
    - very similar to rgb, with a different way of writing  
    - Ex: color: #e1ff00  
    - aa = content of red in resulting color  
     bb = content of red in resulting color  
     cc = content of red in resulting color  
    - Range of a,b,c = [0,9], a,b,c,d,e,f  
     0 is lowest and f is highest value   
    - Lowest value of aa/bb/cc => 00 (0 in RGB)  
     Highest value of aa/bb/cc => ff (255 in RGB)  
    - we can make any color with certain combination of aa,bb,cc or RGB  
    - to make standard red color: #ff0000 //means only red color and no green, blue  
     to make standard green color: #00ff00//means only green color and no red, blue  
     to make standard blue color: #0000ff //means only blue color and no green, red  
     to make standard yellow color: #ffff00 // means only green and red, no blue
12. Selectors:  
    - universal selector = \*{…}  
    - element selector = h1{…}  
    - id selector = #myid {…}  
    - class selector = .class1 {}
13. Property : **text-align**- text-align: center/left/right  
    - aligns an HTML element left/center/right of parent container
14. Property : **font-weight**- font-weight: normal/bold/bolder/lighter  
    - font-weight: [100,900] // 100 means lightest and 900 means boldest.
15. Property : **font-family**- font-family: ‘Segoe UI’, Tahoma, sans-serif   
    - it means set Segoe UI as font-family  
    - if browser does not support Segoe UI, set font family as Tahoma.  
    - if nothing supported on a certain browser, set it as sans-serif  
    - this L->R type of process is called as fallback mechanism
16. Measuring Units in CSS:  
     a. Absolute Units  
     b. Relative Units
17. **Pixels**:  
    - a type of absolute unit  
    - similar to other measuring units like cm,inches, mm, kms etc.  
    - 96 pixels = 1 inch  
    - used to measure size of HTML elements, screen sizes etc.  
    - Ex: font-size: 10px
18. Popular relative units in CSS  
    - percentage (%)  
    - em  
    -rem  
    - vh,vw  
    - relative units mean they vary based on screen window size, parent container config etc.
19. **Percentage**:  
    - width: 33%  
    - this unit is calculated relative to parent  
    - used to define size(width or height) of an element relative to parent container  
    - example:   
     width of parent = 100px  
     width of child defined as = 20%  
     Calculated width of child = 20px.
20. **em**:  
    - font-size: 0.5em  
    - if used for font-size, this unit is calculated relative to font-size of parent container  
    - if used for width, this unit is calculated relative to font-size of element itself  
    - example:  
     font-size of parent container = 20px  
     font-size of child = 2em (means 20\*2 = 40px)  
     width of child = 3em (means 40\*3 = 120px)
21. **viewports: vh and vw**  
    - viewport means visible area of screen (scroll area not included)  
    - 1vh means = 1% of viewport height (relative)  
     1vw means = 1% of viewport width (relative)  
    - 100vh means 100% of visible screen, same for 100vw.
22. Property: **line-height**  
    - line-height: 5px  
    - used when we have multiple lines of text  
    - represents how much vertical space each line will take
23. **Box model in CSS:**  
       
    - 6 important concepts in box model:  
     a. width  
     b. height  
     c. padding  
     d. margin  
     e. border  
     f. content-area  
       
    - Content can be any HTML element – a heading, a div, an image etc.  
    - By default, an HTML element occupies a certain width and height based on the content inside them. An image may occupy more space than a paragraph or a heading.  
    - We can explicitly specify the height and width of an element too  
        
    - **Width** and **Height** set width and height of content-area  
    - **Border** means boundary of content area. If padding present – border represent boundary of padding area.  
    - **Padding** means space b/w content area and border.  
    - **Margin** means space outside of element border (usually we set margin to have some space between two or more elements)
24. Property: **border**  
    - it means boundary of content area. If padding present – border represent boundary of padding area.  
    - border-width: 2px  
     border-style: solid/dashed/dotted  
     border-color: black  
    - shorthand => border: 2px solid black
25. Property: **border-radius**  
    - border-radius: 10px  
     border-radius: 50%  
    - round the corners of element’s border
26. Property: **padding**  
    - space b/w content area and border  
    - padding-top: 1px, padding-right: 2px, padding-bottom: 3px, padding-left: 4px  
    - shorthand => padding: 1px 2px 3px 4px (same as above, TRBL clockwise)  
    - shorthand => padding: 25px 50px (top, bottom = 25px & left, right = 50px)
27. Property: **margin**  
      
    - used to add space b/w two immediate elements – horizontally or vertically or both.  
    - margin-top: 1px, margin-right: 2px, margin-bottom: 3px, margin-left: 4px

- shorthand => margin: 1px 2px 3px 4px (same as above, TRBL clockwise)

- shorthand => margin: 25px 50px (top, bottom = 25px & left, right = 50px)  
- Note: If element1 has margin-right: 10px and element2 has margin-left: 10px, then distance b/w element1 and element2 is 10px and not 20px (see above pic)

1. Property: **display**  
   - display: inline/block/inline-block/none  
   - a property that every HTML element has by default and that affects the layout of box model for that element
2. **display: inline**  
   - takes only the space required by the element  
   - basically it represents only the content-area of element  
   - padding/margin doesn’t work properly with them  
   - width and height also don’t work properly  
   - don’t take 100% width by default, hence multiple inline elements tend to come in a single line  
   - example: span, link etc.
3. **display: block**  
   - takes full space available in width (of parent container)  
   - all box-model properties work properly  
   - block level elements start from a new line as they take full parent container width by default.  
   - example: div, paragraphs etc.
4. **display: inline-block**  
   - all props similar to inline elements except that padding,margin,width and height properties can be applied on them and work properly.
5. **display: none**  
   - element is removed from HTML document flow, like deleted.  
   - subsequent elements take up its space.
6. **visibility: hidden**  
   - element is still present in HTML document flow but becomes invisible.  
   - subsequent elements cannot take up its space.
7. Revisit:   
   - relative units