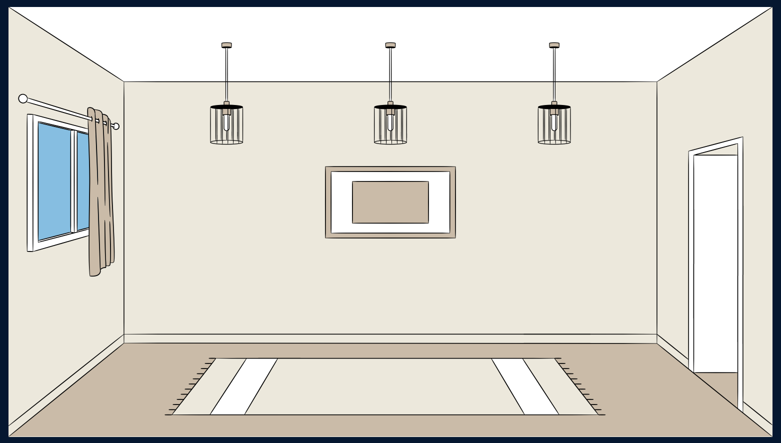
HTML

Structure/layout

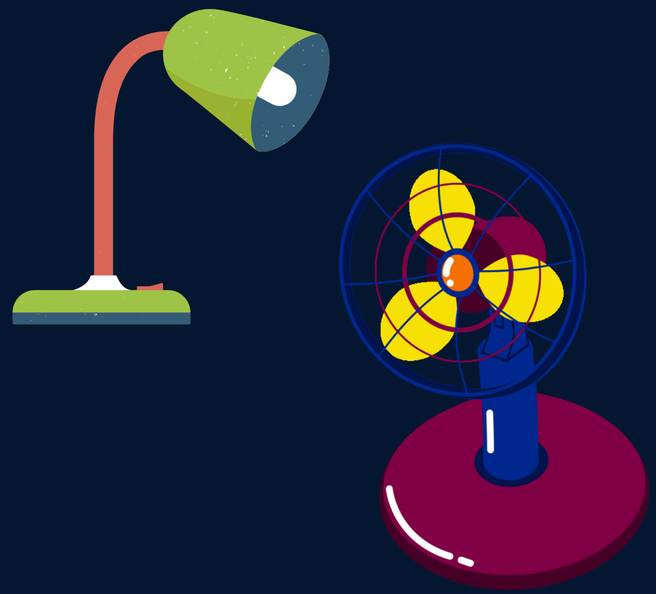


CSS

Style

JS

Logic



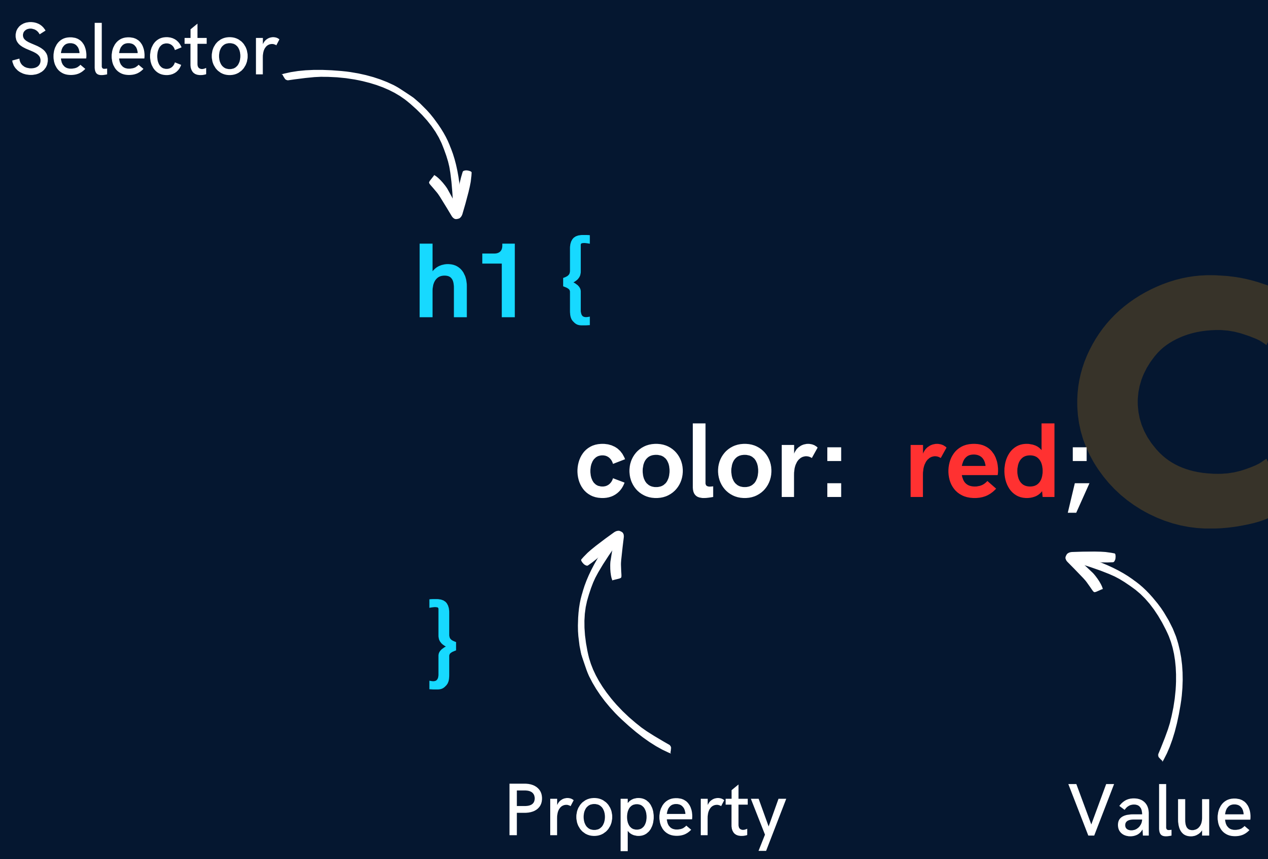
Level 1

CSS

Cascading Style Sheet

It is a language that is used to describe the style of a document.

Basic Syntax



Selector

```
h1 {
    color:  red;
}
```

Property

Value

Including Style

- Inline

```
<h1 style="color: red"> Apna College </h1>
```

- <style> tag

```
<style>
        h1 {
            color : red;
        }
</style>
```

Including Style

- External Stylesheet

Writing CSS in a separate document & linking it with HTML file

Color Property

Used to set the color of foreground

```
color:  red;

color:  pink;

color:  blue;

color:  green;
```

Background Color Property

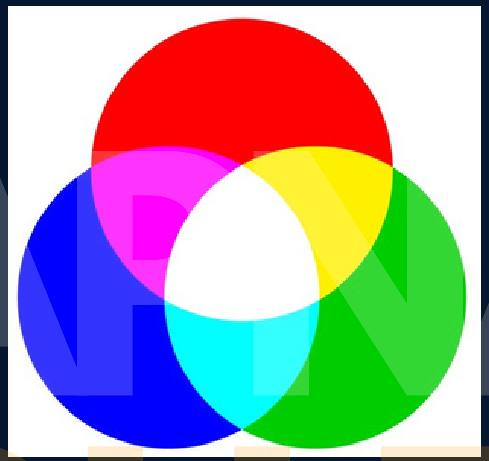
Used to set the color of background

```
background-color: red;

background-color: pink;

background-color: blue;

background-color: green;
```

Color Systems

- RGB

```
color:  rgb(255, 0, 0);

color:  rgb(0, 255, 0);
```



Color Systems

- Hex (Hexadecimal)

```
color: #ff0000;

color: #00ff00;
```

Selectors

- Universal Selector

```
* { }
```

- Element Selector

```
h1 { }
```

- Id Selector

```
#myId { }
```

- Class Selector

```
.myClass { }
```

Practice Set 1

Q1: Create a simple div with an id "box".
Add some text content inside the div.
Set its background color to blue.

Q2: Create 3 headings with h1, h2 & h3.
Give them all a class "heading" & set color of "heading" to red.

Q3: Create a button & set its background color to :

- green using css stylesheet
- blue using <style> tag
- pink using inline style

Text Properties

text-align

text-align : left / right / center

Text Properties

text-decoration

text-decoration : underline / overline / line-through

Text Properties

font-weight

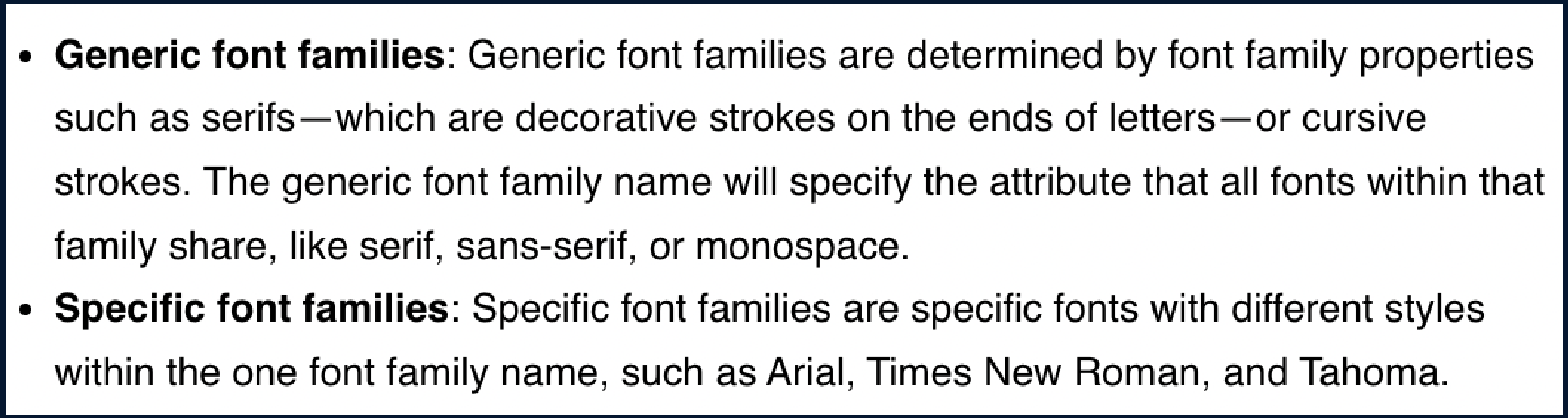
font-weight : normal / bold / bolder / lighter

font-weight : 100-900

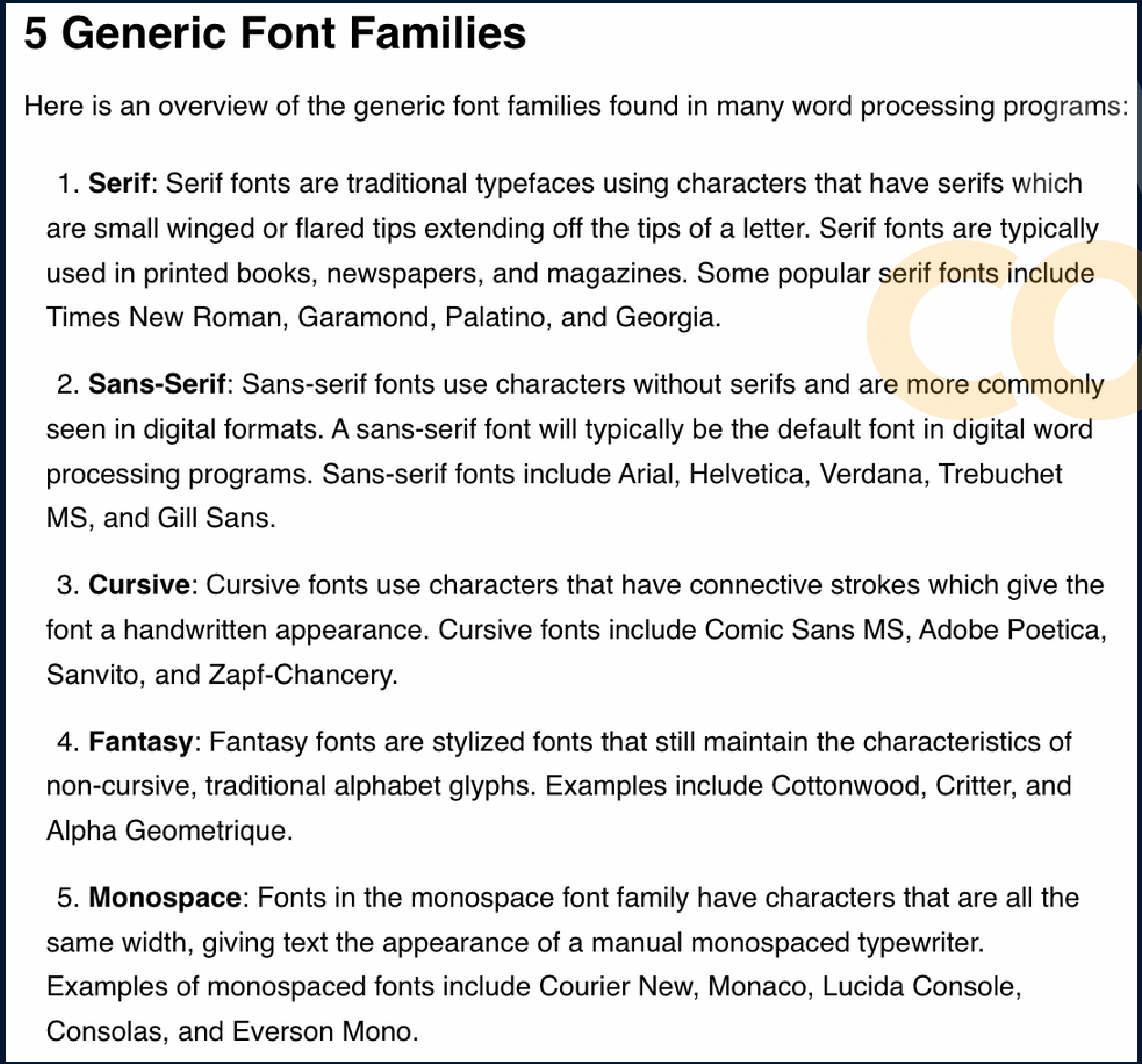
Text Properties

font-family

```
font-family : arial

font-family : arial, roboto
```



- **Generic font families**: Generic font families are determined by font family properties such as serifs—which are decorative strokes on the ends of letters—or cursive strokes. The generic font family name will specify the attribute that all fonts within that family share, like serif, sans-serif, or monospace.
- **Specific font families**: Specific font families are specific fonts with different styles within the one font family name, such as Arial, Times New Roman, and Tahoma.



5 Generic Font Families

Here is an overview of the generic font families found in many word processing programs:

1. **Serif**: Serif fonts are traditional typefaces using characters that have serifs which are small winged or flared tips extending off the tips of a letter. Serif fonts are typically used in printed books, newspapers, and magazines. Some popular serif fonts include Times New Roman, Garamond, Palatino, and Georgia.

2. **Sans-Serif**: Sans-serif fonts use characters without serifs and are more commonly seen in digital formats. A sans-serif font will typically be the default font in digital word processing programs. Sans-serif fonts include Arial, Helvetica, Verdana, Trebuchet MS, and Gill Sans.

3. **Cursive**: Cursive fonts use characters that have connective strokes which give the font a handwritten appearance. Cursive fonts include Comic Sans MS, Adobe Poetica, Sanvito, and Zapf-Chancery.

4. **Fantasy**: Fantasy fonts are stylized fonts that still maintain the characteristics of non-cursive, traditional alphabet glyphs. Examples include Cottonwood, Critter, and Alpha Geometrique.

5. **Monospace**: Fonts in the monospace font family have characters that are all the same width, giving text the appearance of a manual monospaced typewriter. Examples of monospaced fonts include Courier New, Monaco, Lucida Console, Consolas, and Everson Mono.

Units in CSS

Absolute

pixels (px)

96px = 1 inch

font-size: 2px;

Text Properties

line-height

line-height : 2px

line-height : 3

line-height : normal

Text Properties

text-transform

text-tranform : uppercase / lowercase / capitalize / none

Practice Set 2

Q1: Create a heading centred on the page with all of its text capitalized by default.

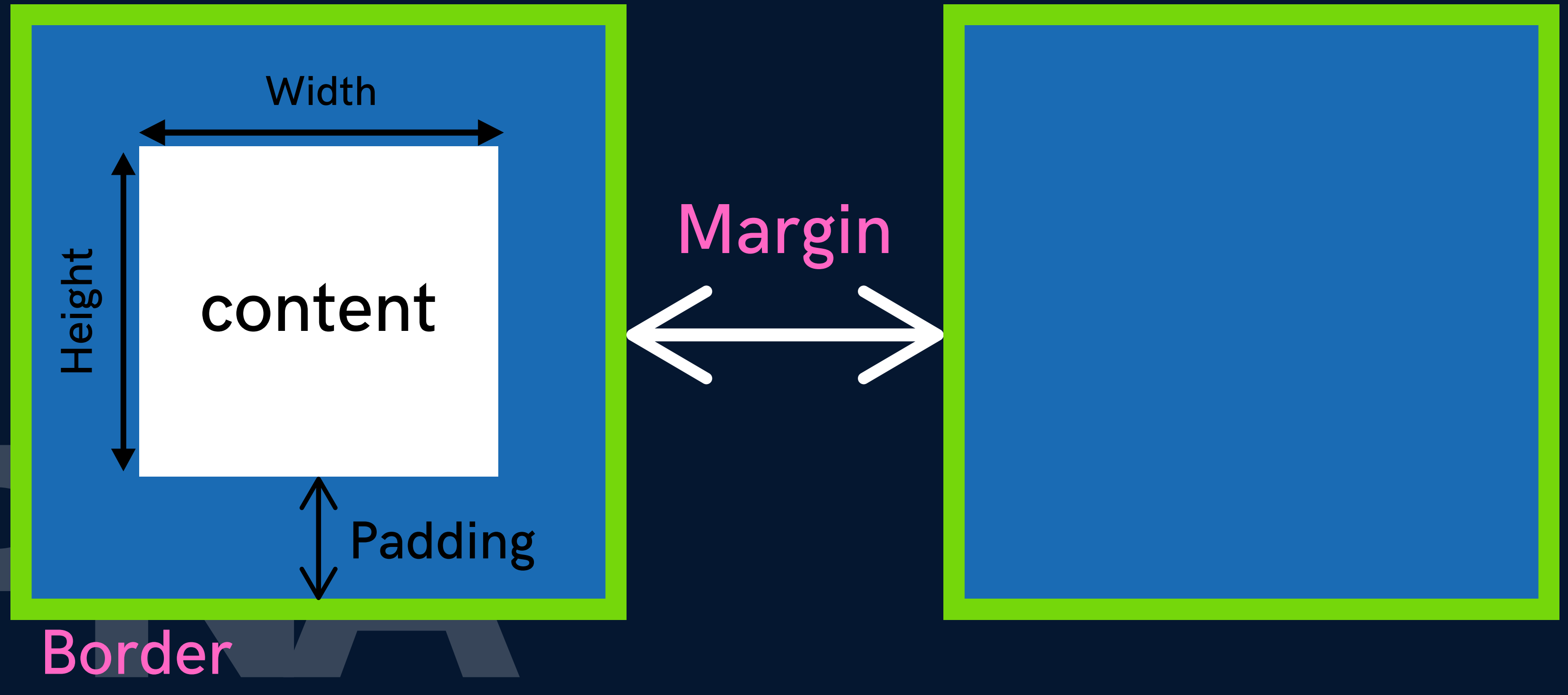
Q2: Set the font family of all the content in the document to "Times New Roman".

Q3: Create one div inside another div.
Set id & text "outer" for the first one & "inner" for the second one.
Set the outer div text size to 25px & inner div text size to 10px.

Level 2

Box Model in CSS

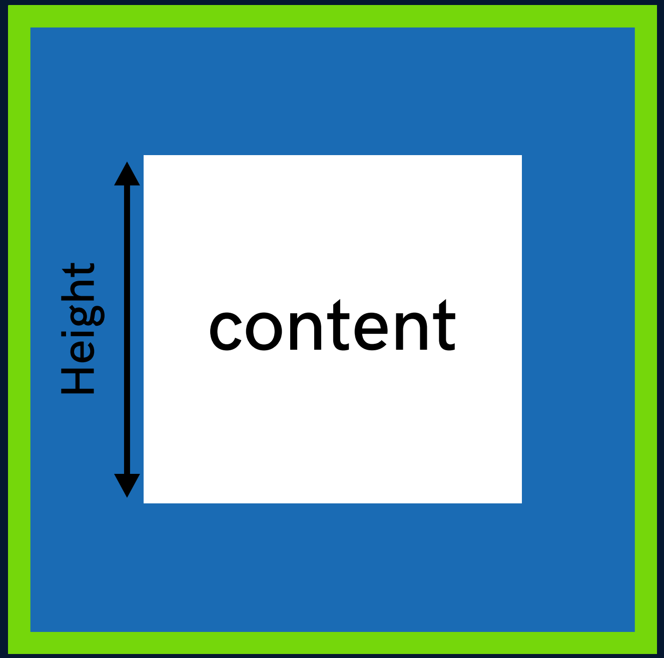
- Height
- Width
- Border
- Padding
- Margin



Height

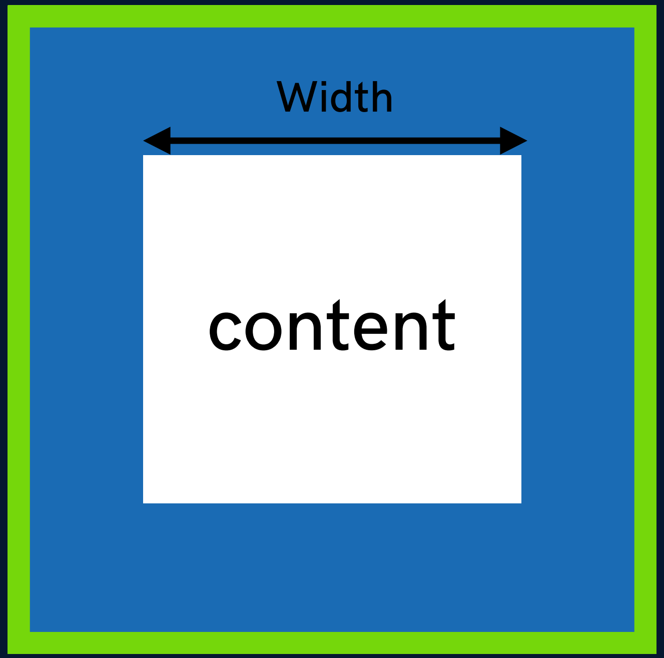
By default, it sets the content area height of the element

```
div {
    height:  50px;
  }
```



Width

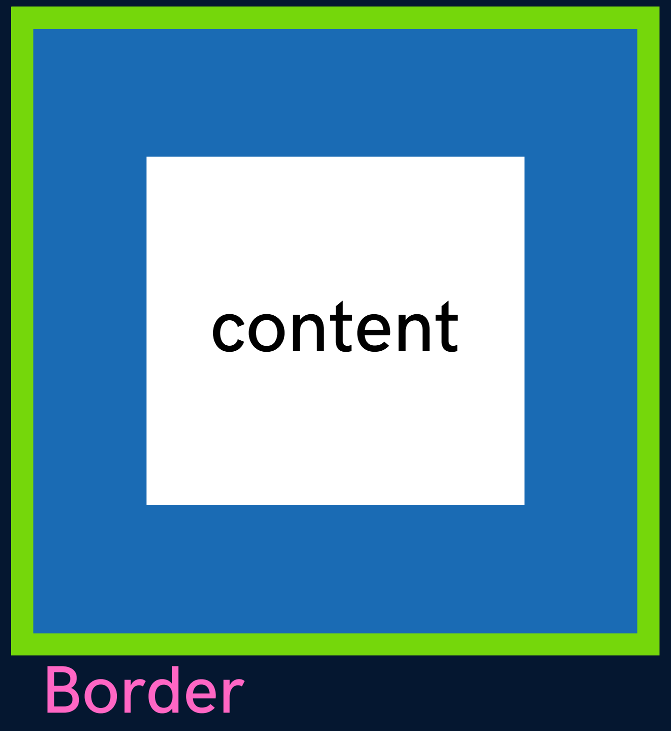
By default, it sets the content area width of the element



```
div {
        width:  50px;
  }
```

Border

Used to set an element's border



border-width : 2px;

border-style : solid / dotted / dashed

border-color : black;

Border

Shorthand

```
border : 2px solid black;
```

Border

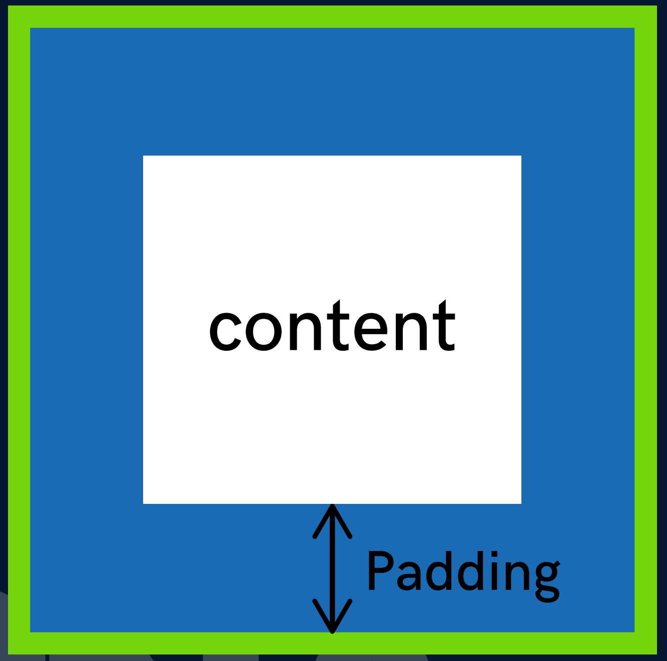
Used to round the corners of an element's outer border edge

border-radius : 10px;

border-radius : 50%;

Padding

- padding-left
- padding-right
- padding-top
- padding-bottom



Padding

Shorthand

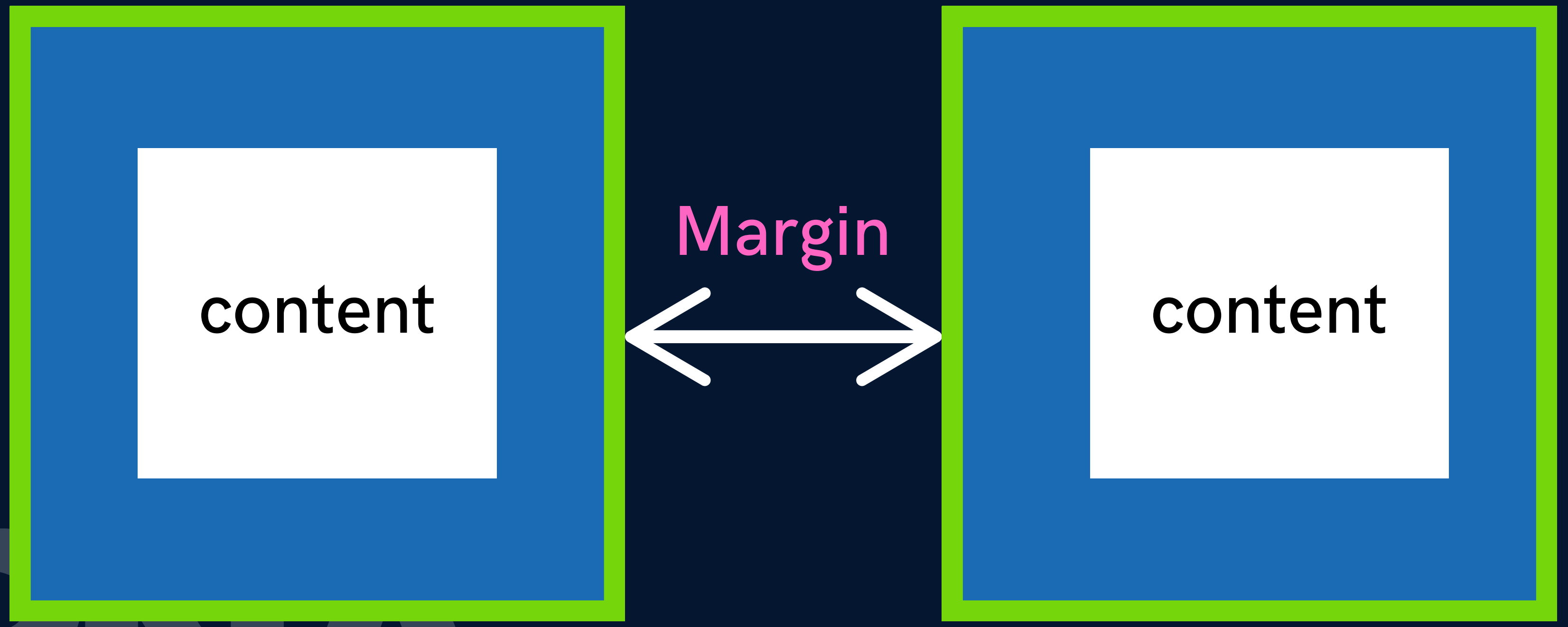
padding: 50px;

padding: 1px 2px 3px 4px;

top | right | bottom | left -> clockwise

Margin

- margin-right
- margin-left
- margin-top
- margin-bottom



Margin

Shorthand

margin: 50px;

margin: 1px 2px 3px 4px;

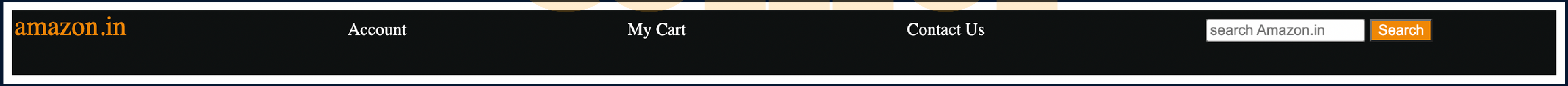
top | right | bottom | left -> clockwise

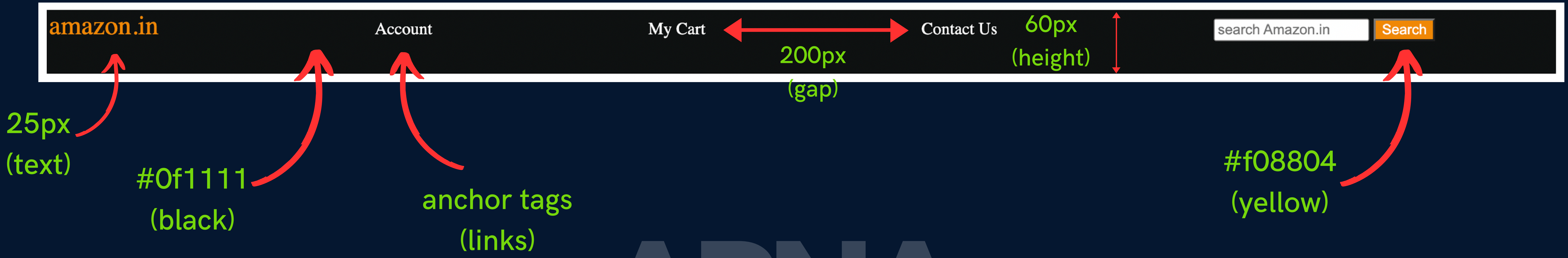
Practice Set 3

Q1: Create a div with height & width of 100px.

Set its background color to green & the border radius to 50%.

Q2: Create the following navbar.





25px (text)

#0f1111 (black)

anchor tags (links)

200px (gap)

60px (height)

#f08804 (yellow)

Display Property

display: inline / block / inline-block / none

- inline - Takes only the space required by the element. (no margin/ padding)
- block - Takes full space available in width.
- inline-block - Similar to inline but we can set margin & padding.
- none - To remove element from document flow.

Visibility

visibility: hidden;

Note : When visibility is set to none, space for the element is reserved.

But for display set to none, no space is reserved or blocked for the element.

Alpha Channel

opacity (0 to 1)

- RGBA

```
color:  rgba(255, 0, 0, 0.5);


color:  rgba(255, 0, 0, 1);
```

Practice Set 4

Q1: Create a webpage layout with a header, a footer & a content area containing 3 divs.
Set the height & width of divs to 100px.
(add the previous navbar in the header)

Q2: Add borders to all the divs.

Q3: Add a different background color to each div with an opacity of 0.5

Q4: Give the content area an appropriate height.

Level 3

position

Units in CSS

Relative

%

em

rem

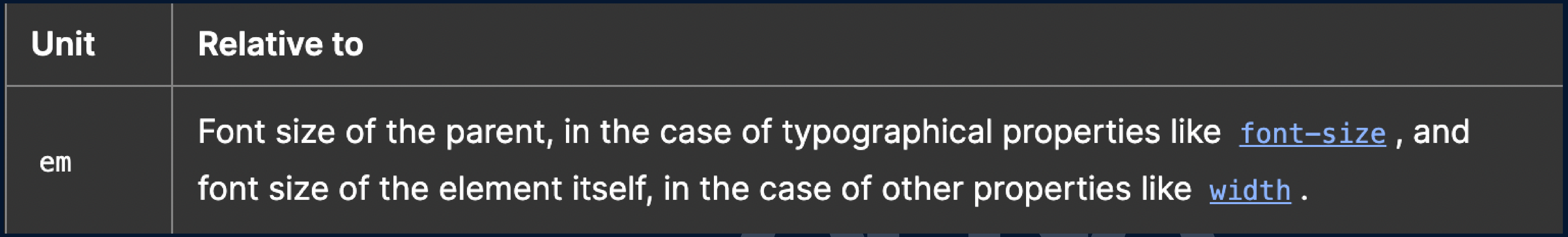
Percentage (%)

It is often used to define a size as relative to an element's parent object.

width : 33% ;

margin-left : 50% ;

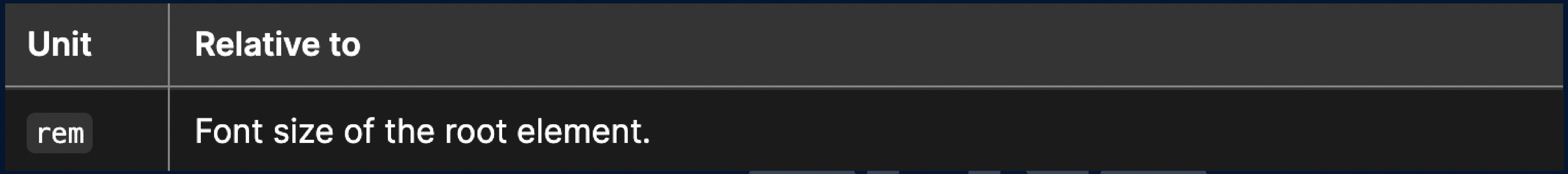
Em



Unit	Relative to
`em`	Font size of the parent, in the case of typographical properties like `font-size`, and font size of the element itself, in the case of other properties like `width`.

```
#box1 {
height: 100px;
width:200px;
background-color: green;
font-size: 10px
}

#box2 {
height: 50px;
width: 30%;
background-color: yellow;
margin-left: 10%;
font-size: 2em;           2 em means ki parent font size ka 2x ho jayega
width: 5em;               here 5em means ki jahan likha hai ussi ke font size ka  5x
```

Rem (Root Em)



Unit	Relative to
`rem`	Font size of the root element.

```
#box1 {
height: 100px;
width:200px;
background-color: green;
font-size: 10px
}

#box2 {
height: 50px;
width: 30%;
background-color: yellow;
margin-left: 10%;
font-size: 2em;
width: 5rem;
```

here 5rem means ki jo font size root element ka hoga uska 5x
toh by default 16 hota hai toh 16x5 = 80 px ho jayega

Others

vh: relative to 1% viewport height

screen ki height ke according hota hai
50vh means screen ki height 50% cover karega

vw : relative to 1% viewport width

screen ki width ke according hota hai
50vw means screen ka width 50% cover karega

Position

The position CSS property sets how an element is positioned in a document.

position : static / relative / absolute / fixed

Position

- static - default position (The top, right, bottom, left, and z-index properties have no effect)

- relative - element is relative to itself. (The top, right, bottom, left, and z-index will work)

  relative mtlb apne hi boundary se top, right, bottom, left hoga

- absolute - positioned relative to its closest positioned ancestor. (removed from the flow)

- fixed - positioned relative to browser. (removed from flow)

pure doc ke relative aur wo wahin fix ho jayega

- sticky - positioned based on user's scroll position

```
#box1 {
position: static;
}
#box2 {
position: relative;
top: 20px;          apni boundary se hi 20 px niche
left: 10px;          apni boundary se hi 10px right
}
#box3 {
position: absolute;   body tag (ancestor) ke hisab
                       se hoga
}
```

z-index

It decides the stack level of elements

Overlapping elements with a larger z-index cover those with a smaller one.

z-index : auto (0)

z-index : 1 / 2 / ...

z-index : -1 / -2 / ...

yeh batata hai ki konsa element upar aayega aur konsa niche agar overlap kr rhe hai toh .

Toh by default jo pahale likha hai wo niche aayega aur jo baad mein likha hai wo upar (z-index=0)

z-index ranges from -infifnity to + infinity
z-index >1 toh uski priority bhad jayega (jitna positive hoga) -- upar rahega
z-index <1 toh uski priority kum ho jayega (jitna neagtive hoga) -- niche chala jayega

Agar position static set hai toh z -index ki value set nhi hoskti

Background Image

Used to set an image as background

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

Background Size

background-size : cover / contain / auto

a area cover kr legi image jo lagayi hai

contain - yeh bhi cover hi karega puri jagah lekin puri image usme fit ho jayegi bhale hi wo badi hi kyon na ho

auto - jitni aa rahi utni hi aayegi

Practice Set 5

Qs: Create the following layout using the given html.

- Give the div a height, width & some background image.
- Use the appropriate position property for the div element to place it at the right end of the page. (The div should not move even on scroll)
- Use z-index to place the div on top of page.

```
<p> lorem*5 </p>
<div> Love Nature </div>
<p> lorem*5 </p>
```

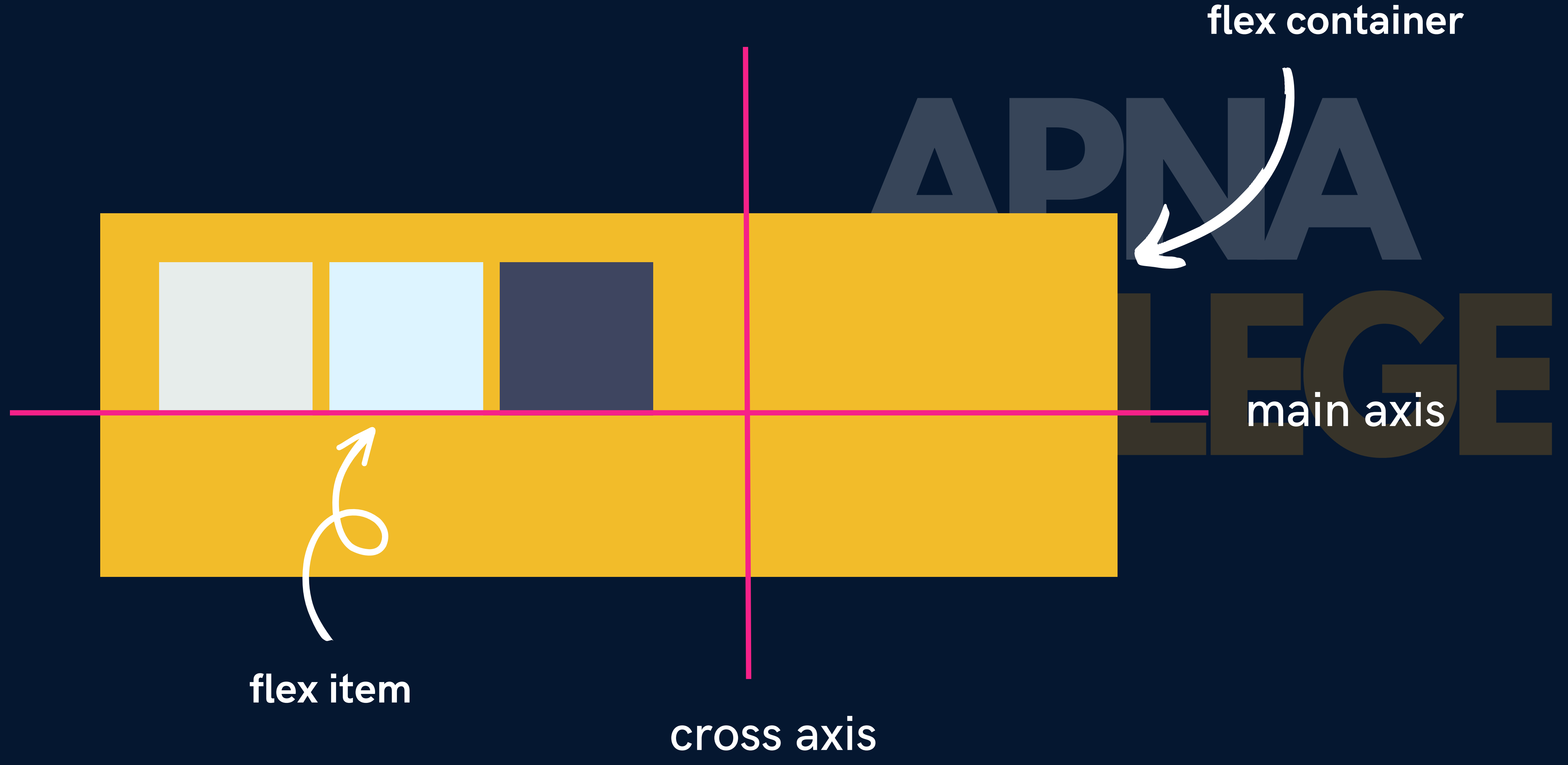
Level 4

Flexbox

Flexible Box Layout

It is a one-dimensional layout method for arranging items in rows or columns.

The Flex Model



Flexbox Direction

yeh items/elements ki direction badal denge

It sets how flex items are placed in the flex container, along which axis and direction.

here we talk about the main axis

- flex-direction : row; (default)

saare elements row mein arrange ho jayenge (left to right)
first element at leftmost position

- flex-direction : row-reverse;

saare elements row mein arrange ho jayenge (right to left)
first element at rightmost position

- flex-direction : column;

top se bottom

- flex-direction : column-reverse;

bottom se top

Note - If we made a flexbox container then all the flex items will be inside that box only

Flex Properties

for Flex Container

yeh direction toh same rakhenge but alignment wagera change ho jayegi

- justify-content : alignment along the main axis.

main axis ke end se 1st element aayega

flex-start / flex-end / centre / space-evenly /

main axis ke start se 1st element aayega

space saari evenly divide ho jayegi

space around -- aas pas space aa jayegi elements ke aur in-between space is 2x of the starting and ending space

space between -- saari spaces beech mein aayegi (same spaces)

- flex-wrap : nowrap / wrap / wrap-reverse

- align-items : alignment along the cross axis.

- align-content : alignment of space between & around the content along cross-axis

Flex Properties

for Flex Item

- align-self : alignment of individual along the cross axis.

it has more priority as compared to container properties

- flex-grow : how much a flex item will grow relative to the rest of the flex items if space is available

- flex-shrink : how much a flex item will shrink relative to the rest of the flex items if space is available

For more shrink make it more than 1 for less shrink make it less than 1
whatever factor u will write it will shrink accordingly
if 0 then no shrink will happen
if 1 then it will shrink normally

Practice Set 6

Qs: Create a navbar with 4 options in the form of anchor tags inside list items.
Now, use flexbox to place them all spaced equally in a single line.

Qs: Use flexbox to center one div inside another div.

```
.outer {
height: 400px;
width: 400px;
background-color: cornflowerblue;
display:flex;
justify-content: center; //for horizontally
                               align in centre
align-items: center;  // for vertically
                              align in centre
}

.inner {
height: 100px;
width: 100px;
background-color: darkorchid;
}
```

Qs: Which has higher priority - align-items or align-self?

it applies on the container so has less priority then the align-self

it applies on individual flex item so have high priority

Media Queries

Help create a responsive website

max mtlb usse niche
min mtlb usse upar

```
@media (width : 600px) {
    div {
        background-color : red;
    }
}
```

max-width likha ho toh mtlb 600px ya usse niche pr color green ho jayega

```
@media (min-width : 600px) {
    div {
        background-color : red;
    }
}
```

600 px ya uske upar width ho toh color green ho jayega

Media Queries

color red ho jaye when width is between 200px and 300 px

min nhi max

```
@media (min-width : 200px) and (min-width : 300px) {
    div {
        background-color : red;
    }
}
```

Practice Set 7

Qs: Add a media query to implement the following:

- the color of a div changes to green for viewport width less than 300px
- the color of a div changes to pink for width between 300px & 400px
- the color of a div changes to red for width between 400px & 600px
- the color of a div changes to blue for width above 600px

Level 5

Transitions

Transitions enable you to define the transition between two states of an element.

- transition-property : property you want to transition (font-size, width etc.)
- transition-duration : 2s / 4ms ..
- transition-timing-function : ease-in / ease-out / linear / steps ..
- transition-delay : 2s / 4ms ..

Transition Shorthand

property name | duration | timing-function | delay

```
transition: font-size 2s ease-in-out 0.2s;
```

CSS Transform

Used to apply 2D & 3D transformations to an element

- rotate

```
transform: rotate(45deg);
```

CSS Transform

- scale

```
transform: scale(2);

transform: scale(0.5);

transform: scale(1, 2);


transform: scaleX(0.5);

transform: scaleY(0.5);
```

CSS Transform

- translate

```
transform: translate(20px);

transform: translate(20px, 50px);


transform: translateX(20px);

transform: translateY(20px);
```

CSS Transform

- skew

```
transform: skew (30deg);
```

Animation

To animate CSS elements

```
@keyframe myName {
    from { font-size : 20px; }
    to { font-size : 40px; }
}
```

Animation Properties

- animation-name
- animation-duration
- animation-timing-function
- animation-delay
- animation-iteration-count
- animation-direction

Animation Shorthand

```
animation : myName 2s linear 3s infinite normal
```

% in Animation

```
@keyframe myName {
    0% { font-size : 20px; }
    50% { font-size : 30px; }
    100% { font-size : 40px; }
}
```

Practice Set 8

Qs: Create a simple loader using CSS

Step1 : create a div with circular shape & a thick border from one end (top/bottom/left/right)

Step2 : To make it spin create an animation which transforms it from 0deg to 360deg

Step3 : Add the animation property to the loader with infinite duration