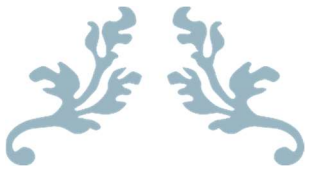


CSS

Cascading Style Sheet

What is Css :-

- Cascading Style Sheets is a stylesheet language used to describe the look & presentation of a document written in Html.
- It defines how elements are displayed on a web page.
- CSS is used to provide styling to HTML elements, making web pages visually appealing and user-friendly.

Syntax :- selector { property : value ; }

- The **selector** is used to target the specific HTML element you want to apply styles to, such as an element (h1), a class (.class), or an ID (#id).
- The **property** defines the specific style aspect you want to change, like the text color, font size, or background.
- The **value** specifies the exact setting for the chosen property, such as red for color or 16px for font size.

Three Ways To Use CSS :-

- There are three ways of inserting a style sheet :
    1. External CSS
    2. Internal CSS
    3. Inline CSS

1. External CSS :-

- External CSS is a separate CSS file (with a .css extension) that contains all the styling rules and is linked to an HTML file using the <link> tag.
- It is used to apply the same styles across multiple HTML pages.

Example :-

index.html

```
<html>
<head>
  <title>External CSS Example</title>
  <link rel="stylesheet" href="style.css"> <!-- Linking external CSS -->
</head>
<body>
  <h1>Hello World</h1>
  <p>This is a paragraph.</p>
</body>
</html>
```

style.css

```
h1 {
  color : green;
}
p {
  font-size : 18px;
  color : gray;
}
```

2. Inline CSS :-

- An inline style may be used to apply a unique style for a single element.
- Inline CSS is used to apply styles directly to a specific HTML element using the style attribute inside the opening tag.
- It is useful for quick, one-time styling but is not recommended for large projects.

Example :-

```
<html>
<body>
  <p style="color : red ;">Hello</p>
</body> </html>
```

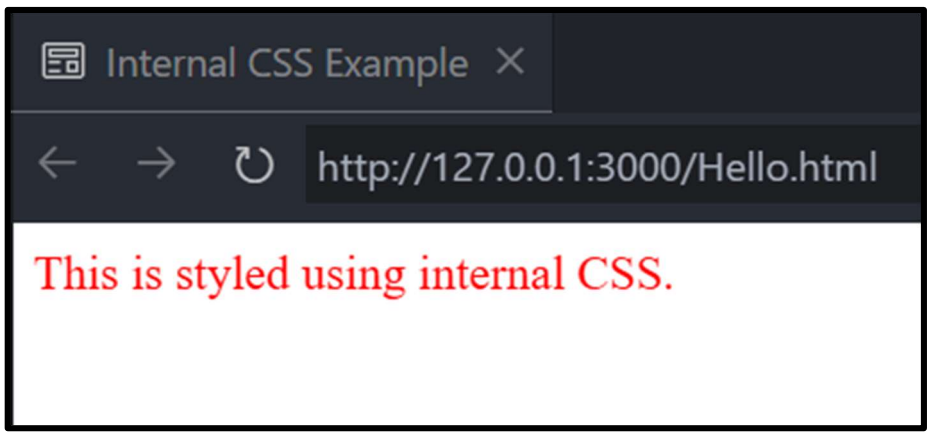
3. Internal CSS :-

- An Internal CSS is a method of adding styles within the <style> tag placed inside the <head> section of an HTML file.
- It is used when you want to apply styles to a single HTML page only.

Example :-

```
<html>
<head>
 <title>Internal CSS Example</title>
 <style>
   p {
    font-size : 16px ;
    color : red ;
   }
 </style>
</head>
<body>
 <p>This is styled using internal CSS.</p>
</body>
</html>
```

Output :-



CSS Comments :-

- Comments are used to explain the code, and may help when you edit the source code at a later date.
- Comments are ignored by browsers.
- A CSS comment is placed inside the <style> element, and starts with /* and ends with */.

Css Selector : -

- A CSS Selector is a pattern used in CSS to select and target HTML elements on a web page so that specific styles can be applied to them.

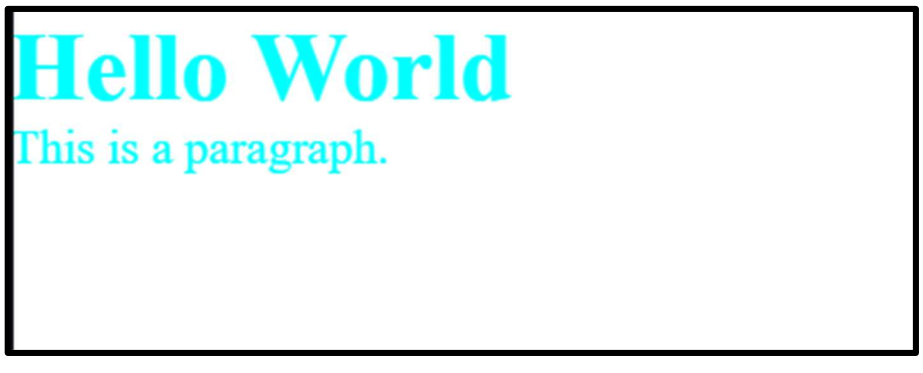
1. Universal Selector (*)

- The Universal Selector (*) in CSS is used to select all elements on a web page, regardless of their type.

Example :-

```
 <style>
   * {
    margin: 0;
    padding: 0;
   }
 </style>
<body> <h1>Hello World</h1>  <p>This is a paragraph.</p> </body>
```

Output :-



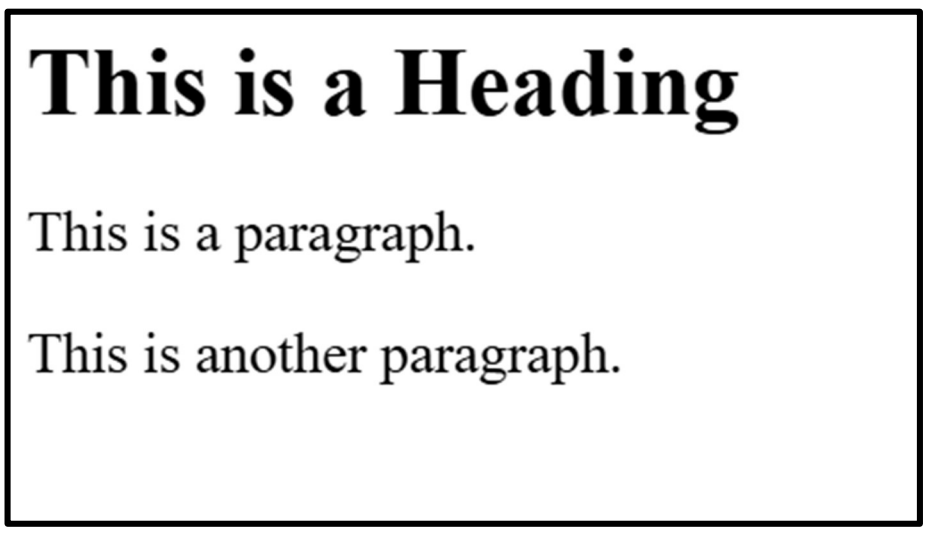
2. Element Selector (Tag Selector)

- An Element Selector is used in CSS to select all HTML elements of a specific type or tag name (like p, h1, div) and apply styles to them.

Example :-

```
<html>
<body>
  <style>
  p {
    font-size: 19px;
  }
 </style>
 <p>This is a paragraph.</p> <p>This is another paragraph.</p>
</body> </html>
```

Output :-



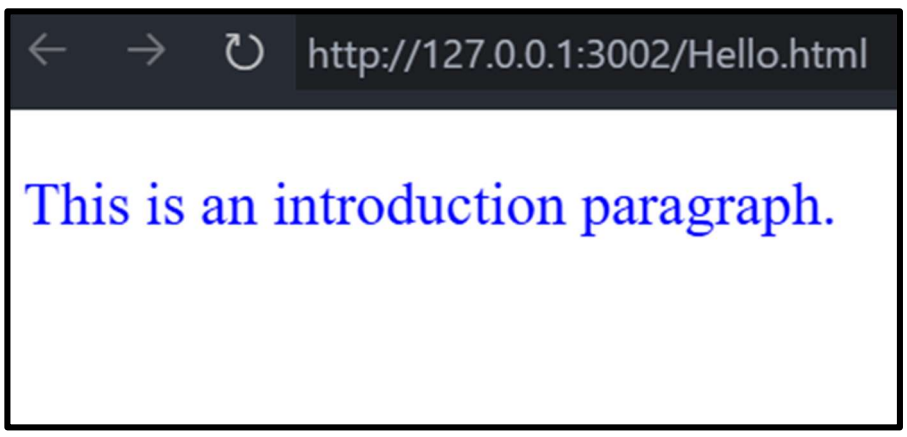
3. ID Selector (#)

- An ID Selector is used in CSS to select and style an HTML element with a specific id attribute. It is written using a # symbol followed by the ID name.
- The id of an element is unique within a page, so the id selector is used to select one unique element.

Example :-

```
<!DOCTYPE html>
<html> <body>
<style>
 #intro {
   color : blue;
   font-size : 20px;
 }
</style>
<p id="intro">This is an introduction paragraph.</p>
</body> </html>
```

Output :-



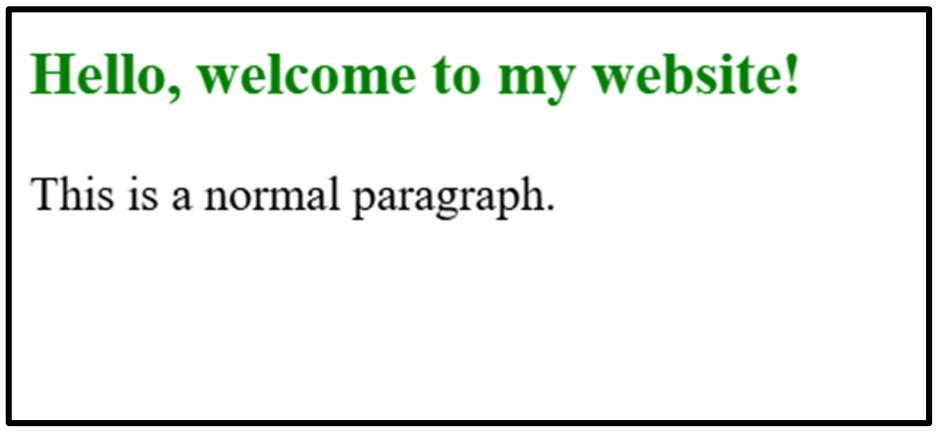
4. Class Selector (.)

- A class selector is used in CSS to select HTML elements that have a specific class attribute. It allows you to apply the same style to multiple elements.

Example :-

```
<!DOCTYPE html> <html> <body>
<style>
 .greeting {
   color: green;
   font-weight: bold;
 }
</style>
<p class="greeting">Hello, welcome to my website!</p>  </body>  </html>
```

Output :-



5. Grouping Selector

- A Grouping Selector in CSS is used to apply the same style to multiple elements by separating their selectors with a comma (,).
- Instead of writing separate style rules for each element, grouping lets you combine them to avoid repetition.

Example :-

```
<html> <body>
<style>
   h1, h2, p {
      color: darkblue;
      font-family: Arial;
      text-align: center;
   }
</style>
<h1>This is Heading 1</h1>  <h2>This is Heading 2</h2>  <p>This is a paragraph styled using grouping selector.</p>
</body>
</html>
```

Output :-

6. Pseudo Class Selector

- A pseudo-class selector is used to define a special state of an HTML element, such as when it is hovered, visited, or the first child of its parent.
- **Syntax :-** selector : pseudo-class { property : value ; }

Example :-

```
<!DOCTYPE html>
<html> <body>
  <style>
    /* Hover :- This pseudo-class is used to apply a style when the user moves the mouse pointer over an element. */
    a:hover {  color : red;   }
    /* Visited :- This pseudo-class targets hyperlinks that the user has already clicked (visited), allowing different styles. */
    a:visited {  color : purple;  }
    /* Active :- This pseudo-class applies a style to an element at the moment it is being clicked or activated. */
    button:active {
      background-color : green;
      color : white;
    }
    /* Focus :- This pseudo-class is triggered when an element ( like an input field ) is selected or focused by the user. */
    input:focus {   border : 5px solid rgb(116, 241, 7);   }
  </style>
  <h2>Pseudo-class Example</h2>
  <a href="#">This is a link (hover and visited)</a><br><br>
  <button>Click Me (active)</button><br><br>
  <input type="text" placeholder="Focus me"><br><br>
</body>
</html>
```

Output :-

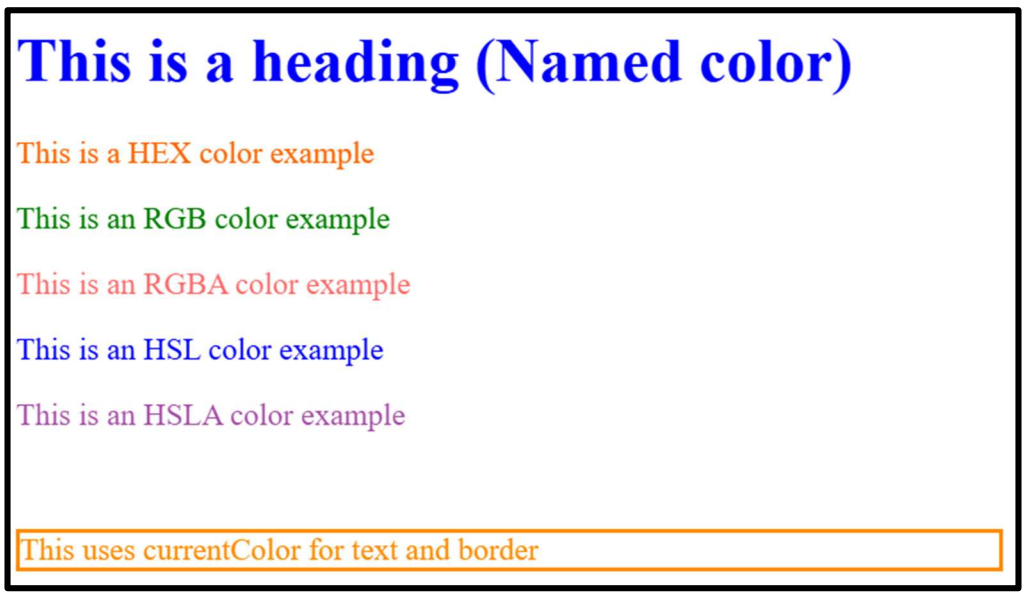
Css Color : -

- CSS Color is a property in CSS used to set the color of text, borders, backgrounds, and other elements in a webpage using different color values like names, hex codes, RGB, etc.

Example :-

```
<!DOCTYPE html>
<html>
<body>
 <style>
   h1 {  color : blue; /* Named color */  }
   .hex {  color : #ff6600; /* HEX color - orange */  }
   .rgb {  color : rgb(0, 128, 0); /* RGB color - green */  }
   .rgba { color : rgba(255, 0, 0, 0.6); /* RGBA color - red with 60% opacity */  }
   .hsl {  color : hsl(240, 100%, 50%); /* HSL color - blue */  }
  .hsla {  color : hsla(300, 100%, 25%, 0.7); /* HSLA color - purple with 70% opacity */  }
  .transparent {  color : transparent; /* Fully transparent text */  }
  .currentColorExample {
    color : darkorange; /* Base text color */
    border : 2px solid currentColor; /* Border matches text color */
  }
</style>
<h1>This is a heading (Named color)</h1>
<p class="hex">This is a HEX color example</p>
<p class="rgb">This is an RGB color example</p>
<p class="rgba">This is an RGBA color example</p>
<p class="hsl">This is an HSL color example</p>
<p class="hsla">This is an HSLA color example</p>
<p class="transparent">This text is invisible using transparent color</p>
 <p class="currentColorExample">This uses currentColor for text and border</p>
</body>
</html>
```

Output :-



Interview Question :-

1. what is difference between rgb ( ) and rgba ( ) color notations in css ?

- RGB defines color using Red, Green, and Blue values, while RGBA adds an extra Alpha value for transparency (opacity).

2. How can you make the text color of an element fully transparent using css ?

- You can make the text color fully transparent using either of these CSS options :- color : transparent ; **or** color : rgba(0, 0, 0, 0) ;

3. Can we apply color gradients using the color property ?

- No. Gradients are applied using background-image, not the color property.

Background Properties : -

- CSS background properties are used to control the background effects (color, images, position, size, etc.) of HTML elements.

CSS Background Properties :-

1. **background-color** sets the background color of an element. Example :- background-color : lightblue;
2. **background-image** sets an image as the background of an element. Example :- background-image : url("bg.jpg");
3. **background-repeat** controls whether and how the background image repeats. Example :- background-repeat : no-repeat;
4. **background-position** sets the position of the background image inside the element. Example :- background-position : center top;
5. **background-size** sets the size of the background image (like cover or contain). Example :- background-size : cover;
6. **background-attachment** sets whether the background image scrolls with the page or stays fixed. Example :- background-attachment : fixed;

Example :-

```
<!DOCTYPE html>
<html>
<body>
 <style>
  .background-example {
   /* 1. background-color : sets background color */
   background-color : lightblue;
   /* Values : -
      - color names (e.g., red, blue)
      - hex (#ff0000)
      - rgb (rgb(255, 0, 0))
      - rgba (rgba(255, 0, 0, 0.5))
      - hsl (hsl(0, 100%, 50%))
   */
   /* 2. background-image : sets background image */
   background-image: url('https://via.placeholder.com/150');
   /* Values:
      - url("path/image.jpg")
      - none
      - multiple images : url("1.jpg"), url("2.jpg")
   */
   /* 3. background-repeat : repeats image */
   background-repeat : no-repeat;
   /* Values :-
      - repeat (default) : repeats both directions
      - no-repeat : no repeat
      - repeat-x : repeat horizontally
      - repeat-y : repeat vertically
      - space : repeats with spacing
   */
   /* 4. background-position: position of image */
   background-position: center top;
   /* Values :-
      - left, right, top, bottom, center
      - x y values: 50% 50%, 100px 200px
      - keywords or combinations: right bottom, left center, etc.
   */
   /* 5. background-size: size of background image */
   background-size: cover;
   /* Values :-
      - auto (default)
      - cover: cover entire area
      - contain: fit without cropping
      - specific size: 100px 200px, 50% 50%
   */
   /* 6. background-attachment: scroll behavior */
   background-attachment : fixed;
   /* Values :-
      - scroll (default) : scrolls with content
      - fixed : stays fixed on screen
      - local : scrolls with the element's scroll
    */
  }
 </style>
 <h2>CSS Background Example with All Values in Comments</h2>
 <div class="background-example">
  <p>This box uses 6 background properties. Check the CSS comments for all values.</p>
 </div>
</body>
</html>
```

CSS Borders :-

- The CSS border properties allow you to specify the style, width, and color of an element's border.
- The border-style property specifies what kind of border to display.

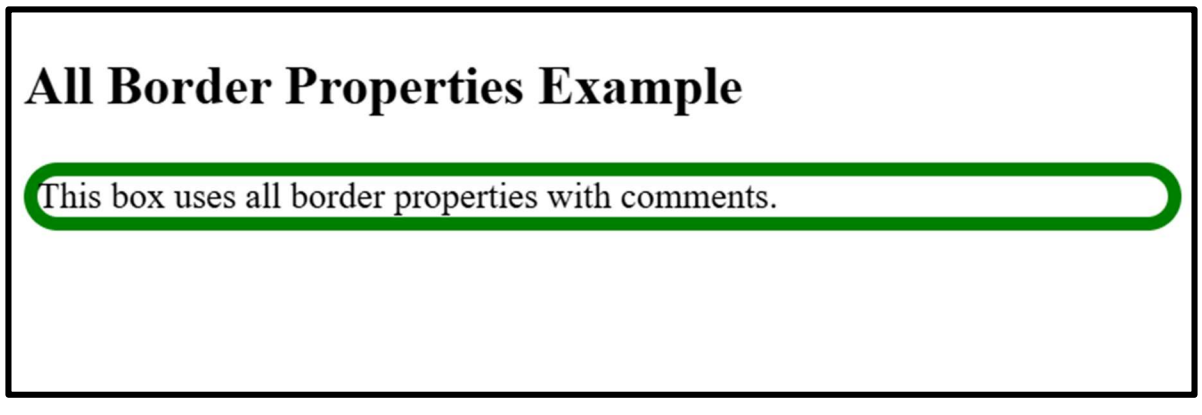
CSS Border Properties :-

- **border** – The border property is a shorthand that sets the width, style, and color of all four sides of an element's border in a single line.
- **border-width** – The border-width property controls the thickness of the border and can be set using values like pixels or keywords such as thin, medium, or thick.
- **border-style** – The border-style property defines the line pattern of the border such as solid, dashed, dotted, double, groove, ridge, inset, or outset.
- **border-color** – The border-color property sets the color of the border and supports named colors, hex codes, RGB, RGBA, and HSL formats.
- **border-radius** – The border-radius property rounds the corners of an element's border by applying a radius value to the corners.

Example :-

```
<html>
<body>
<style>
.border-example {
  /* Shorthand property : sets border-width, border-style, and border-color for all sides */
  Border : 4px solid green;
  /* Individual side borders */
  border-top : 5px dashed red;       /* Top border : 5px wide, dashed, red */
  border-right : 6px dotted blue;    /* Right border : 6px wide, dotted, blue */
  border-bottom : 4px double orange; /* Bottom border : 4px wide, double line, orange */
  border-left : 3px solid purple;    /* Left border : 3px wide, solid, purple */
  /* Widths of all borders */
  border-width : 6px;  /* Width of all sides values :- thin , medium , thick , 3px */
  /* Border style for all sides */
  border-style : solid; /* Style of border :- solid, dashed, dotted, double, groove, ridge, inset, outset border*/
  /* Border color for all sides :- */
  border-color : green; /* Color Values :- #ff0000 ,  rgb(0,255,0) , rgba(0,0,255,0.5) */
  /* Rounds the corners of the box */
  border-radius : 15px;  /* Rounded corners :- 15px radius */
}
</style>
<h2>All Border Properties Example</h2>
<div class="border-example">
  This box uses all border properties with comments.
</div>
</body>
</html>
```

Output :-



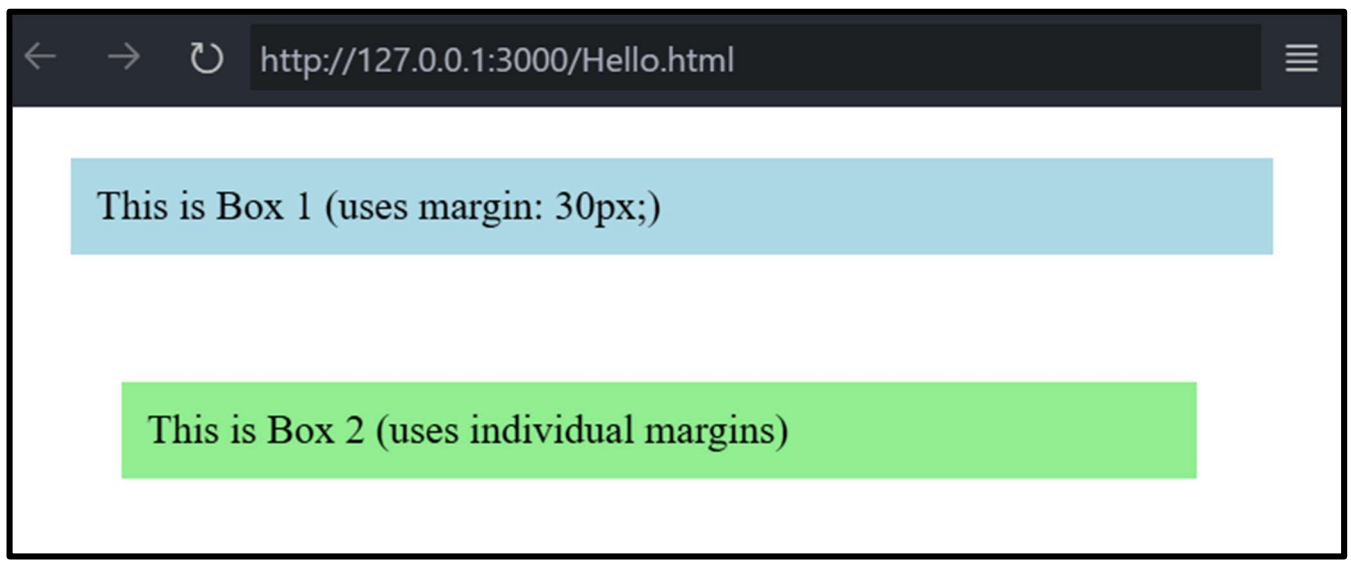
CSS Margins :-

- CSS Margin is a layout property that defines the space outside the border of an HTML element.
- It defines the outer spacing between the element and its neighbouring elements.

Example :-

```
<!DOCTYPE html>
<html>
<body>
  <style>
    .box1 {
      background-color : lightblue;
      margin : 30px; /* adds space around the box */
      padding : 10px;
    }
    .box2 {
      background-color: lightgreen;
      margin-top : 50px;
      margin-right : 20px;
      margin-bottom : 30px;
      margin-left : 10px;
      padding :  10px;
    }
  </style>
  <div class="box1">This is Box 1 (uses margin : 30px;)</div>
  <div class="box2">This is Box 2 (uses individual margins)</div>
</body> </html>
```

Output :-



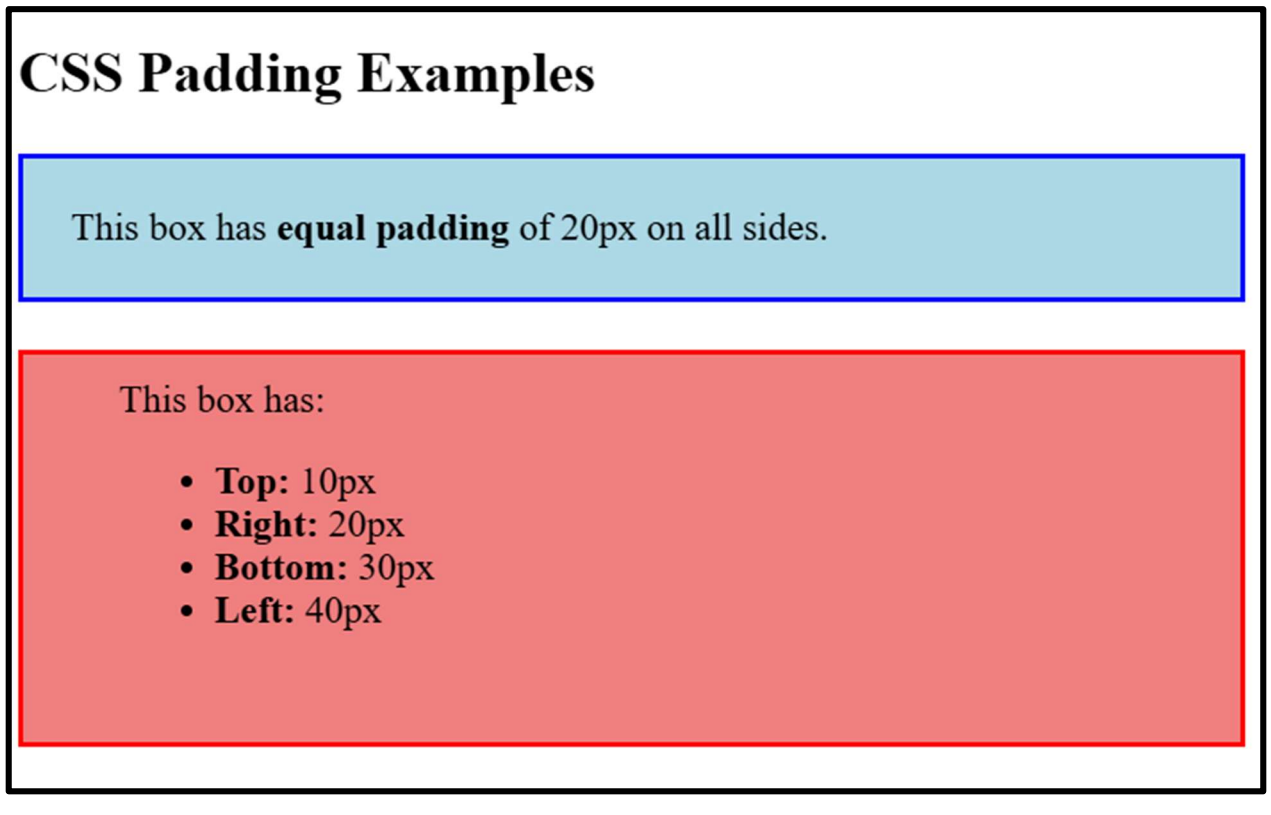
CSS Padding :-

- Padding in CSS is used to create space inside an element between its content and its border.
- It creates inner spacing inside the element, pushing the content away from the edges of the element's border.

Example :-

```
<!DOCTYPE html>
<html>
<body>
  <style>
    .box1 {
      background-color : lightblue;
      padding : 20px; /* Same padding on all sides */
      border : 2px solid blue;
      margin-bottom : 20px;
    }
```

Output :-



```
  .box3 {
    background-color : lightcoral;
    padding : 10px 20px 30px 40px; /* Top, Right, Bottom, Left */
    border : 2px solid red;
  }
</style>
<h2>CSS Padding Examples</h2>
<div class="box1">This box has <strong>equal padding</strong> of 20px on all sides.</div>
<div class="box3">  This box has :-
  <ul>
    <li><strong>Top :</strong> 10px</li>
    <li><strong>Right :</strong> 20px</li>
    <li><strong>Bottom :</strong> 30px</li>
    <li><strong>Left :</strong> 40px</li>
   </ul>
  </div>
</body> </html>
```

UNITS in Css :-

- Units in CSS are used to define measurements for various properties such as width, height, margin, padding, font-size, etc.
- They determine the size or length of an element on a webpage. CSS units are mainly divided into two categories :

1. Absolute Units :- Absolute units are fixed and do not change depending on screen size or resolution.

Unit	Meaning	Example
px	Pixels	width : 100px;

Pixel :- px is an absolute unit of measurement in css, representing the smallest unit of screen space. It provide fixed and consistent size on all screens.

2. Relative Units :- Relative units are based on other values, such as parent elements or root font size.

Unit	Meaning	Example
%	Percentage of parent element (16px = 100%) (1px = 6.25%) (54px X 6.25% = 337.5 %)	width : 50%;
em	Font size of the current element (1em = 16 px) (54px / 16 = 3.374 em)	padding : 2em;
rem	Font size of the root (<html>) element (1rem = 10px) (54px = 5.4 rem)	font-size : 1.5rem;

- **Percentage :- percentage are relative units based on the parent element's size or the containing block.**
- **Em :- em is a relative unit that is calculated based on the font size of parent elements.**
- vw (viewport width) :- vw is a responsive CSS unit where 1vw equals 1% of the browser's visible width.
- vh (viewport height) :- vh is a responsive CSS unit where 1vh equals 1% of the browser's visible height.
- **Root em :- rem is also a relative unit like em but it's based on the font size of root (html) element.**

1. **how does using rem units in your css benefits a responsive web design ?**

- rem units in CSS benefits responsive web design by allowing all elements to scale consistently relative to the root font size, making it easier to create flexible, accessible, and easily adjustable layouts across different screen sizes.

2. **what is the main diffrence between em and rem units in css ?**

- The main difference between em and rem units in CSS is em is relative to the font size of its parent element. rem is relative to the root element's (html) font size. This makes rem more predictable for consistent sizing across a site.

Fonts in Css :-

- In CSS, the font property is used to set the appearance of text in terms of its style, weight, size, and family (like Arial, Verdana, etc.).

CSS Font Properties :-

1. **The font-family property** specifies the font type (like Arial or Times New Roman) that should be used to display the text on a web page.
2. **The font-size property** defines the size of the font and can be set in various units such as pixels (px), ems (em), rems (rem), or percentages (%).
3. **The font-weight property** in CSS is used to set the thickness or boldness of the text, allowing you to control how light or heavy the font appears on the screen.
4. **The font-style property** is used to make text appear in a normal style, italic style, or an oblique (slanted) version of the font.
5. **The font property** is a shorthand that allows you to set multiple font-related properties like font-style, font-variant, font-weight, font-size and font-family all in one line.
6. **The font-variant property** is used to display text in small-caps format, where lowercase letters appear as smaller versions of uppercase letters.

Example :-

```
<!DOCTYPE html>
<html>
<body>
 <style>
  /* Style for paragraph using all font-related properties individually */
  p {
   /* font-style : normal | italic | oblique */
   font-style : italic;
   /* font-variant : normal | small-caps */
   font-variant : small-caps;
   /* font-weight : normal | bold | bolder | lighter | 100 to 900 */
   font-weight : 700;
   /* font-size : length (px, em, rem) | keyword (small, medium, large) */
   font-size : 20px;
   /* font-family : specific font names + generic font family at the end */
   font-family : "Georgia", "Times New Roman", serif ;
  }
  /* Same properties written using font shorthand */
  h1 {
   /* Syntax :- font : font-style font-variant font-weight font-size/line-height font-family; */
   font : oblique small-caps bold 24px/1.5 "Georgia", serif;
  }
 </style>
 <h1>This is a heading styled with font shorthand</h1>
 <p>This is a paragraph styled with all individual font properties in CSS.</p>
</body>
</html>
```

Output :-

Text in Css :-

- **Text properties** are used to control the appearance, layout, spacing, alignment, transformation, and decoration of text elements on a webpage.
- These properties help enhance readability and provide visual styling to textual content.

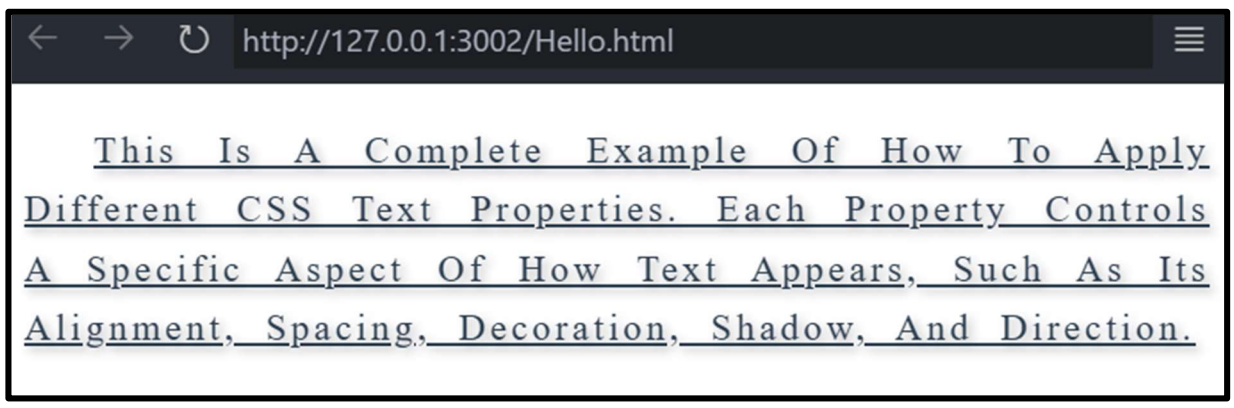
CSS Text Properties :-

- **The color property** sets the color of the text using color names (like red), hex values (#ff0000), RGB, or HSL formats.
- **The text-align property** controls the horizontal alignment of text inside a block-level element and can be set to left, right, center, or justify.
- **The text-decoration property** adds or removes decorative lines on text, such as underline, overline, line-through, or removes them using none.
- **The text-transform property** changes the case of the text, such as converting it to uppercase, lowercase, or capitalize (first letter each capitalized).
- **The letter-spacing property** specifies the amount of space between individual letters in the text.
- **The word-spacing property** controls the spacing between words in the text, increasing or decreasing the gap between them.
- **The vertical-align property** aligns text vertically relative to the surrounding content or its line, with values like top, middle, bottom, or baseline.
- **The text-shadow property** adds shadow effects to the text using values for horizontal and vertical offset, blur, and shadow color.

Example :-

```
<!DOCTYPE html>
<html> <body>
 <style>
  p {
    /* Sets the text color */
    color : #2c3e50;
    /* Aligns text horizontally : left | right | center | justify */
    text-align : justify;
    /* Adds decoration to text : underline | overline | line-through | none */
    text-decoration : underline;
    /* Changes text case : uppercase | lowercase | capitalize | none */
    text-transform : capitalize;
    /* Sets spacing between characters ( letter spacing ) */
    letter-spacing : 2px;
    /* Sets spacing between words */
    word-spacing : 6px;
    /* Adds shadow to the text Syntax :-  horizontal-offset   vertical-offset   blur-radius   color */
    text-shadow : 2px 2px 4px rgba(0, 0, 0, 0.3);
    /* Vertically aligns text relative to line : baseline | top | middle | bottom */
    vertical-align: baseline;
  }
 </style>
 <p> This is a complete example of how to apply different CSS text properties. Each property controls a specific aspect of how text appears, such as its
    alignment, spacing, decoration, shadow, and direction. </p>  </body> </html>
```

Output :-



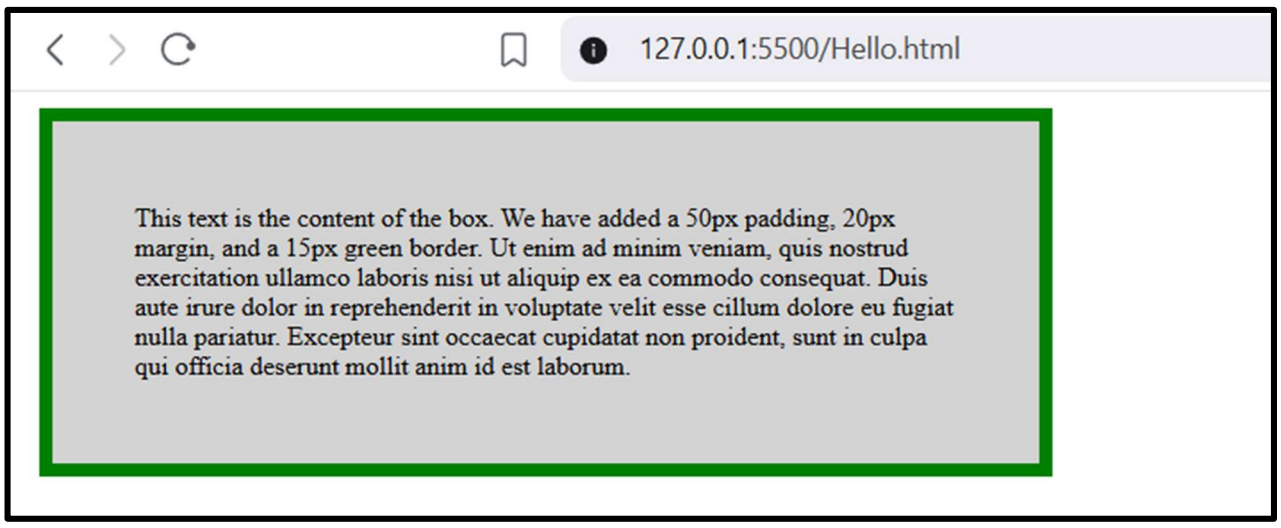
1. How can you control the spacing between line of text in css ?

- The line-height property is used to control the spacing between lines of text. It can set to a specific unit like px or a relative value like 1.5, which multiplies the font sizes.

Box Model in Css :-

- The CSS Box Model is a fundamental concept that describes how every HTML element is rendered as a rectangular box.
- It consists of four parts :- Content, Padding, Border, and Margin.

Example :-




```
<!DOCTYPE html>
<html>
<head>
 <style>
  /* CSS for the box model example */
  div {
    background-color : lightgrey ;       /* Sets background color for visibility */
    width : 300px;                  /* Content width (excluding padding & border) */
    padding : 50px;                  /* Space inside the box, around the content [ syntax all side :- padding : top / right / bottom / left ] */
    border : 15px solid green;          /* Green border around the box */
    margin : 20px;                  /* Space outside the box */
  }
 </style>
</head> <body>
 <div>
  This text is the content of the box. We have added a 50px padding, 20px margin, and a 15px green border. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
 </div>
</body> </html>
```

Gradient in Css :-

- A gradient in CSS is a way to create a smooth transition between two or more colors without using an image. It allows backgrounds to blend colors in various directions.
- Gradients are set using the background-image property with special gradient functions.
- **Linear Gradient (linear-gradient)** :- A linear gradient creates a smooth transition between two or more colors along a straight line.
- The direction of a linear gradient can be set from top to bottom, left to right, or at an angle using degrees like 45deg.
- **Radial Gradient (radial-gradient)** :- A radial gradient creates a transition that radiates outward from a central point, forming a circular or elliptical color pattern.
- Radial gradients allow you to specify the shape (circle or ellipse), size, and position of the gradient's center.
- **Conic Gradient (conic-gradient)** :- A conic gradient creates a color transition that rotates around a center point, like the slices of a pie chart.
- Conic gradients are supported in all modern browsers and are useful for circular progress indicators or pie chart backgrounds.

Linear Gradient Syntax :- background-image : linear-gradient(direction, color1, color2, ...);

Ex :- background-image: linear-gradient(to right, red, orange, yellow);

Radial Gradient Syntax :- background-image : radial-gradient(shape size at position, color1, color2, ...);

Ex :- background-image : radial-gradient(ellipse at center, blue 20% , white 40%, green 40%);

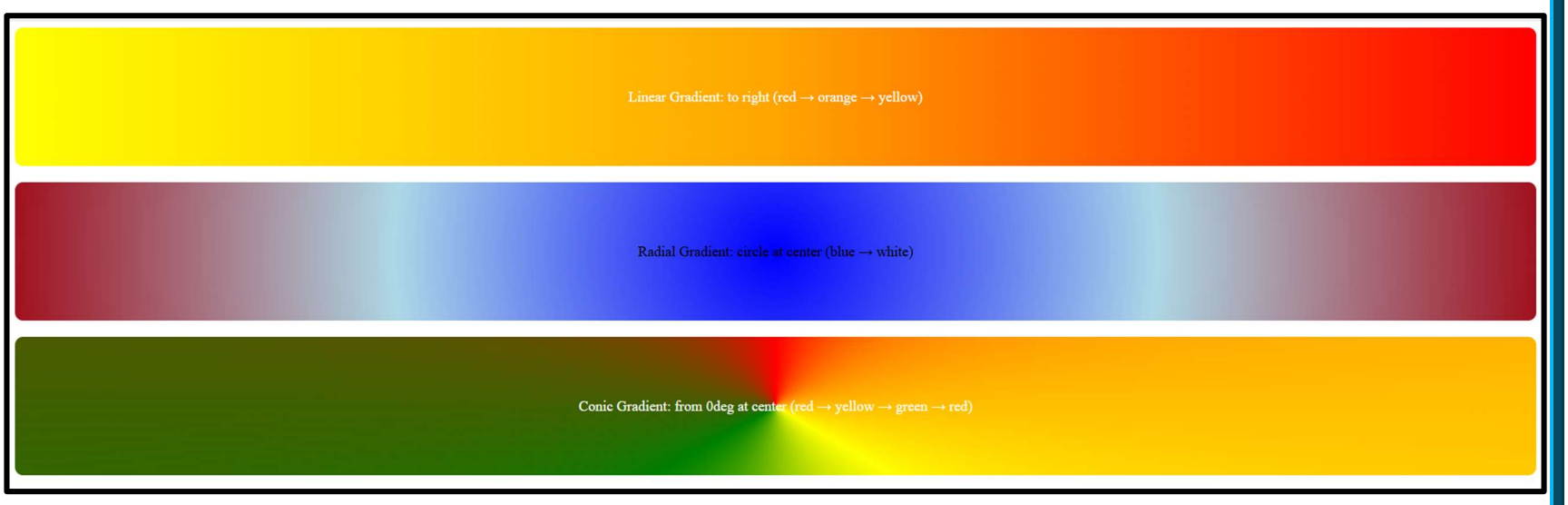
Conic Gradient Syntax :- background-image : conic-gradient(from angle at position, color1, color2, ...);

Ex :- background-image : conic-gradient(from 0 deg at center, red, yellow, green, red);

Example :-

```
<!DOCTYPE html>
<html>
<body>
<style>
  .box {
    height: 150px;
    color: white;
    font-size: 18px;
    display: flex;
    align-items: center;
    justify-content: center;
    border-radius: 10px;
    padding: 10px;
    margin-bottom: 20px;
}
/* Linear Gradient Example */
.linear {
  /* Syntax : linear-gradient(direction, color1, color2, ...) */
  background-image : linear-gradient(to left, red, orange, yellow);
}
/* Radial Gradient Example */
.radial {
  /* Syntax : radial-gradient(shape size at position, color1, color2, ...) */
  background-image : radial-gradient(circle at center, blue, lightblue, rgb(161, 16, 30));
  color: black;
}
/* Conic Gradient Example */
  .conic {
    /* Syntax : conic-gradient(from angle at position, color1, color2, ...) */
    background-image : conic-gradient(from 0deg at center, red, yellow, green, red);
  }
 </style>
 <div class="container">
  <div class="box linear">Linear Gradient: to right (red → orange → yellow)</div>
  <div class="box radial">Radial Gradient: circle at center (blue → white)</div>
  <div class="box conic">Conic Gradient: from 0deg at center (red → yellow → green → red)</div>
 </div>
</body>
</html>
```

Output :-



List Properties in Css :-

- In CSS, a list refers to a set of items defined using HTML list elements like <ul>, <ol>, and <li>, and CSS is used to style and control the appearance of these lists, including their bullets, numbering, spacing, and custom markers.

Common List Properties :-

- **The list-style-type property** specifies the kind of marker (such as disc, square, circle, decimal, or roman numerals) that is used for each list item.
- **The list-style-position property** determines whether the list marker appears inside the list item box or outside it.
- **The list-style-image property** allows you to set a custom image as the bullet or marker for list items instead of the default symbols.
- **The list-style property** is a shorthand that combines list-style-type, list-style-position, and list-style-image into a single line for cleaner code.

Example :-

```
<!DOCTYPE html>
<html>
<body>
  <style>
    ol.custom-list {
     /* list-style-type : defines the numbering or bullet style */
     /* Possible values for ordered list :  - decimal  ⟶ 1, 2, 3   - decimal-leading-zero ⟶ 01, 02, 03   - lower-roman ⟶ i, ii, iii   - upper-roman ⟶ I, II, III
       - lower-alpha  ⟶ a, b, c     - upper-alpha  ⟶ A, B, C    - lower-greek  ⟶ α, β, γ     */
     list-style-type : upper-roman;
     /* list-style-position : controls position of marker relative to text */
     /* Possible values :
       - outside ⟶ marker outside the text box (default)
       - inside  ⟶ marker inside the text box
     */
     list-style-position : inside;
     /* list-style-image : uses image as marker instead of bullets or numbers */
     /* Possible values :
       - url("image.png") ⟶ use a custom image as marker
       - none             ⟶ no image used (default)
     */
     list-style-image : url('https://cdn-icons-png.flaticon.com/16/25/25694.png');
     /* Optional : shorthand property ( combines all above )  Syntax :- list-style: <list-style-type> <list-style-position> <list-style-image>;  */
     /* list-style : square inside none; */
    }
</style>
<h2>Styled Ordered List with All List Properties</h2>
<ol class="custom-list">
  <li>Introduction</li>
  <li>Body</li>
  <li>Conclusion</li>
</ol> </body> </html>
```

Output :-

ANCHOR States in Css :-

- Anchor states refer to the different interactive states of a hyperlink (<a> tag), which allow you to change the appearance of links when users hover, visit, click, or focus on them.
- The : **link pseudo-class** is used to style an anchor tag that has not been visited by the user yet.
- The : **visited pseudo-class** is used to style links that the user has already clicked or visited.
- The : **hover pseudo-class** is triggered when the user moves the mouse pointer over the link.
- The : **active pseudo-class** is applied to a link at the moment it is being clicked.

Example :-

```
<!DOCTYPE html>
<html>
<head>
 <style>
  /* Style for unvisited link */
  a:link {
   color : blue; /* This style is applied when the link has not been visited */
   text-decoration : none; /* Removes the underline */
  }
  /* Style for visited link */
  a:visited {
   color : purple; /* This style is applied after the user clicks the link */
  }
  /* Style when mouse hovers over the link */
  a:hover {
   color : green; /* Changes the link color when hovered */
   text-decoration : underline; /* Underlines the link on hover */
  }
  /* Style when the link is actively being clicked */
  a:active {
    color : red; /* Temporary color when the link is being clicked */
   }
 </style>
</head>
<body>
 <h2>Anchor Tag States Example</h2>
 <!-- Anchor tag with a real link -->
 <a href="https://www.example.com" target="_blank">Visit Example.com</a>
</body>
</html>
```

Output :-

Combinators in Css :-

- combinators are special symbols used to define the relationship between two or more selectors, allowing you to apply styles to elements based on their position in the HTML structure relative to other elements.

Combinators Property :-

1. **The descendant combinator (space)** selects all elements that are nested within another element, regardless of how deeply they are nested.

Example :- div p selects all <p> elements inside a , even if they are not direct children.

2. **The child combinator (>)** selects an element that is a direct child or another elements.

Example :- div > p selects only <p> elements that are directly inside a , not nested deeper.

3. **The adjacent sibling combinator (+)** selects an element that is the immediate next sibling of a specified element.

Example :- h1 + p selects the first <p> element that comes immediately after an <h1>.

4. **The general sibling combinator (~)** selects all sibling elements that follow a specified element, sharing the same parent.

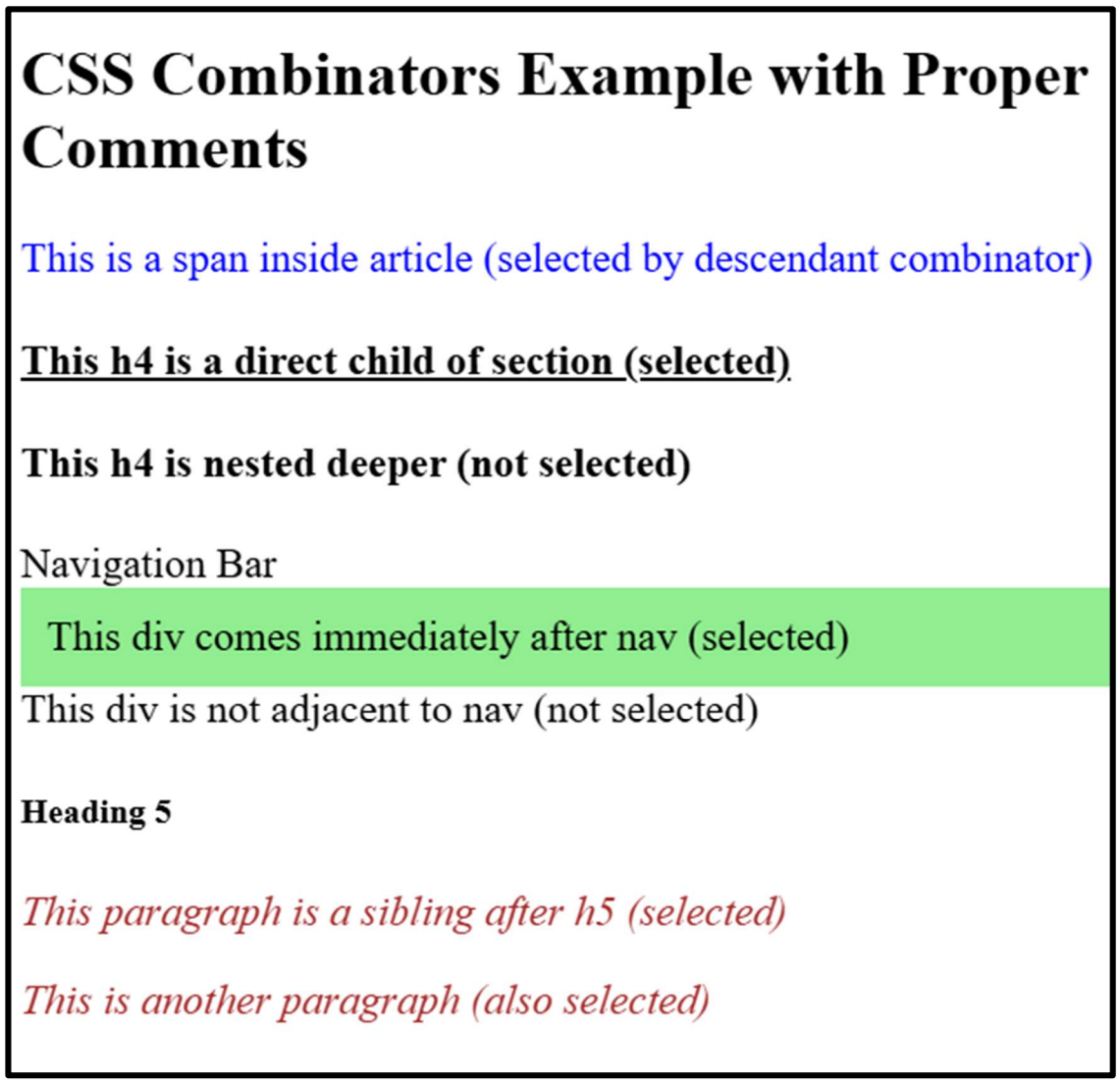
Example :- h1 ~ p selects all <p> elements that come after an <h1> as siblings.

Example :-

```
<!DOCTYPE html>
<html>
 <style>
   /*   1. Descendant Combinator ( space )
This rule selects every <span> element that is located anywhere inside an <article> element,  regardless of how deeply nested it is.  */
   article span {
     color : blue;
   }
   /*  2. Child Combinator ( > )
This rule selects only the <h4> elements that are direct children of a <section> element. It will NOT select <h4> elements that are nested deeper inside other elements within <section>. */
   section > h4 {
     text-decoration : underline;
   }
   /*  3. Adjacent Sibling Combinator ( + )
This rule selects the <div> element that comes immediately after a <nav> element. Only the first immediate sibling is selected. */
   nav + div {
     background-color : lightgreen;
     padding : 10px;
   }
   /*  4. General Sibling Combinator ( ~ )
This rule selects all <p> elements that are siblings of a <h5> element and appear after it. These elements must share the same parent as the <h5> element.
*/
   h5 ~ p {
     font-style : italic;
     color : brown;
   }
 </style>
```

```
<body>
 <h2>CSS Combinators Example with Proper Comments</h2>
 <!-- Example for Descendant Combinator -->
 <article>
  <div>
   <span>This is a span inside article (selected by descendant combinator)</span>
  </div>
 </article>

 <!-- Example for Child Combinator -->
 <section>
  <h4>This h4 is a direct child of section (selected)</h4>
  <div>
   <h4>This h4 is nested deeper (not selected)</h4>
  </div>
 </section>

 <!-- Example for Adjacent Sibling Combinator -->
 <nav>Navigation Bar</nav>
 <div>This div comes immediately after nav (selected)</div>
 <div>This div is not adjacent to nav (not selected)</div>

 <!-- Example for General Sibling Combinator -->
 <h5>Heading 5</h5>
 <p>This paragraph is a sibling after h5 (selected)</p>
 <p>This is another paragraph (also selected)</p>
</body>
</html>
```

Output :-



Displays Property in css :-

- The display property in CSS is used to define how an HTML element is displayed or behaves in the document layout.
- It determines the layout behavior (like block, inline, flex, grid, etc.) of an HTML element.

Common Properties of Display :-

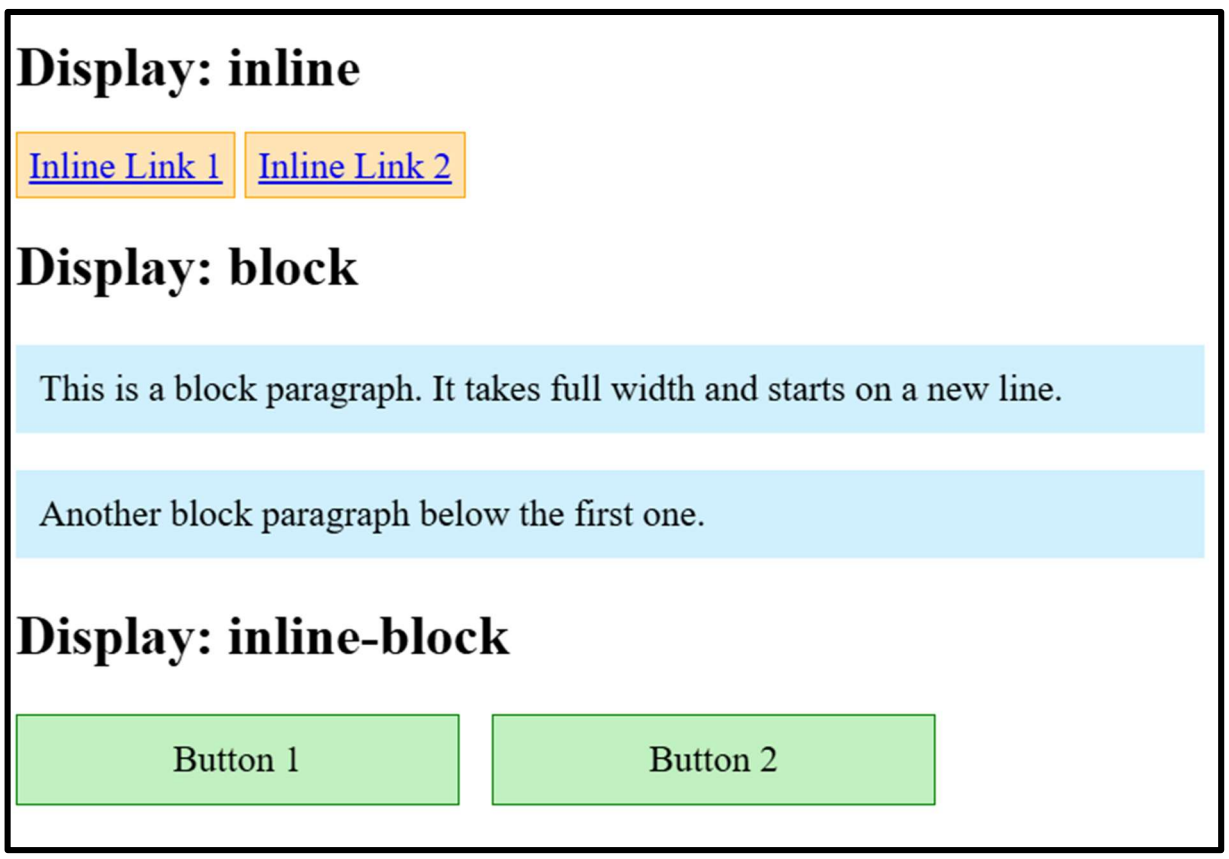
- **block** – This value makes the element behave like a block-level element, which means it takes up the full width available and starts on a new line.
- **inline** – This value makes the element behave like an inline element, meaning it only takes up as much width as needed and does not start on a new line.
- **inline-block** – This value makes the element flow inline like an inline element, but it also allows you to set width and height like a block element.
- **none** – This value hides the element completely from the page; it will not appear and does not take up any space in the layout.

Example :-

```
<!DOCTYPE html>
<html>
```

```
<style>
  /* Inline example : <a> tag behaves inline by default */
  .inline-link {
    display : inline;
    background-color : #ffe4b5; /* light orange */
    padding : 5px;
    border : 1px solid orange;
    /* width/height won't apply */
  }
  /* Block example : <p> tag turned into block (default) */
  .block-paragraph {
    display : block;
    background-color : #d0f0fd; /* light blue */
    padding : 10px;
    margin-bottom : 10px;
  }
  /* Inline-block example : styled buttons aligned side by side */
  .inline-block-button {
    display : inline-block;
    background-color : #c1f0c1; /* light green */
    padding : 10px 20px;
    width : 150px;
    text-align : center;
    margin-right : 10px;
    border : 1px solid green;
  }
 </style>
<body>
  <h2>Display : inline</h2>
  <a href="#" class="inline-link">Inline Link 1</a>
  <a href="#" class="inline-link">Inline Link 2</a>
  <!-- These links appear on the same line -->
  <h2>Display : block</h2>
  <p class="block-paragraph">This is a block paragraph. It takes full width and starts on a new line.</p>
  <p class="block-paragraph">Another block paragraph below the first one.</p>
  <h2>Display : inline-block</h2>
  <div class="inline-block-button">Button 1</div>
  <div class="inline-block-button">Button 2</div>
</body>
</html>
```

Output :-



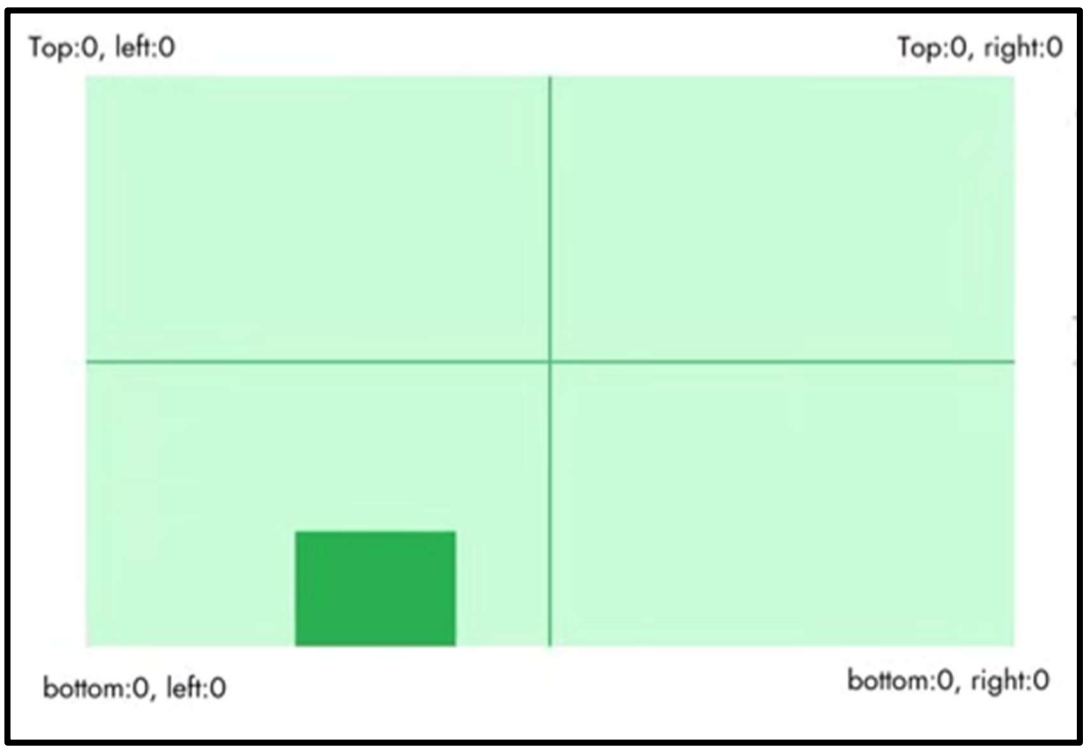
Position in css :-

- Position in CSS is a property used to control the placement of elements on a web page.
- It determines how an element is positioned in the document flow and in relation to its parent or the browser window.

Position Property :-

- **Static** – The element is positioned according to the normal document flow and cannot be moved using top, left, right, or bottom.
- **Relative** – The element is positioned relative to its normal position and can be moved using top, left, right, or bottom without affecting the layout space.
- **Absolute** – The element is positioned relative to the nearest positioned ancestor and is removed from the normal document flow.
- **Fixed** – The element is positioned relative to the browser window and stays in place even when the page is scrolled.
- **Sticky** – The element behaves like relative until it reaches a scroll threshold, after which it sticks in place like fixed.

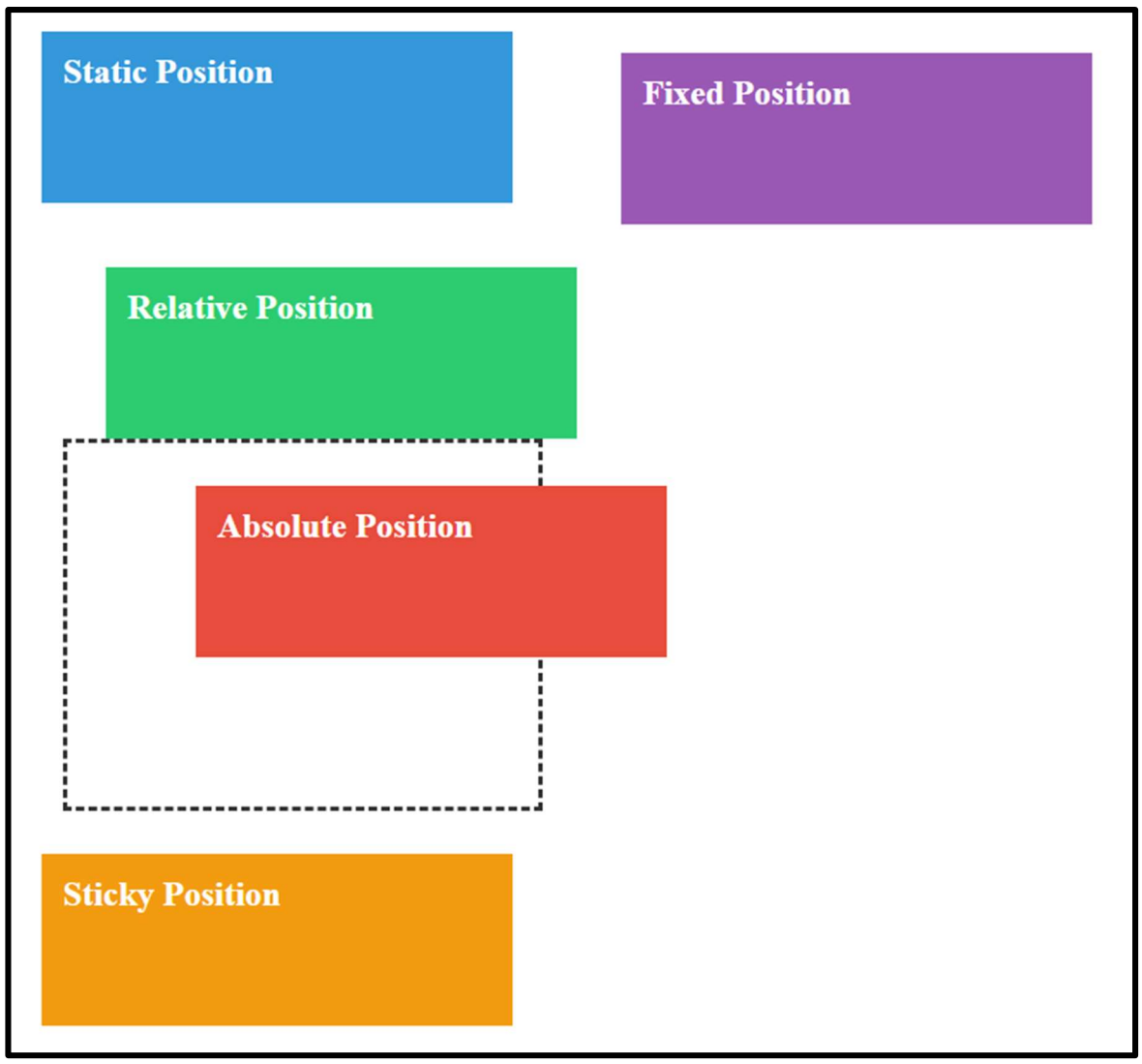
Example :-




```
<!DOCTYPE html>
<html>
 <style>
  div {
    width : 200px;
    height : 60px;
    padding : 10px;
    color : white;
    font-weight : bold;
  }
  /* 1. Static */
  .static-box {
    background-color : #3498db;
    position : static; /* Default - follows normal flow */
  }
  /* 2. Relative */
  .relative-box {
    background-color : #2ecc71;
    position : relative; /* Positioned relative to its original position */
    top : 20px;
    left : 30px;
  }
  /* 3. Absolute (inside a container) */
  .container {
    position : relative; /* Needed for absolute child */
    height : 150px;
    border : 2px dashed #333;
    margin : 20px;
  }
```

```
.absolute-box {
  background-color : #e74c3c;
  position : absolute; /* Positioned inside container */
  top : 10px;
  left : 50px;
}
/* 4. Fixed */
.fixed-box {
  background-color : #9b59b6;
  position : fixed; /* Always stays at top-right of the page */
  top : 10px;
  right : 10px;
}
/* 5. Sticky */
  .sticky-box {
    background-color : #f39c12;
    position : sticky; /* Sticks when scrolling reaches top */
  }
  .spacer {
    height : 10px; /* Extra space for scroll */
  }
 </style>
<body>
 <div class="static-box">Static Position</div>
 <div class="relative-box">Relative Position</div>
 <div class="container">
  <div class="absolute-box">Absolute Position</div>
 </div>
 <div class="fixed-box">Fixed Position</div>
  <div class="sticky-box">Sticky Position</div>
  <div class="spacer"></div>
</body>
</html>
```

Output :-



Z-Index in css :-

- Z-index is a CSS property used to control the stacking order of overlapping HTML elements.
- Higher z-index means the element will appear on top of those with a lower value.
- It Only works on positioned elements (position :- relative, absolute, fixed, or sticky).
- If two elements have the same z-index, the one written later in the HTML will appear on top.

Example :-

```
<!DOCTYPE html>
<html>
 <style>
   .container {
     position : relative;
   }
   .image {
     position : absolute;
     z-index : 1; /* Bottom layer - image */
   }
   .overlay-text {
     position : absolute;
     top : 120px;
     left : 160px;
     color : white;
     font-size : 24px;
     background-color : rgba(0, 0, 0, 0.5);
     padding : 10px;
     z-index : 2; /* Middle layer - text */
}
.top-box {
     position : absolute;
     top : 100px;
     left : 10px;
     width : 100px;
     height : 100px;
     background-color : yellow;
     border : 2px solid red;
     z-index : 3; /* Top layer - box */
   }
 </style>
<body>
<div class="container">
<img src="img.jpg" class="image">
 <div class="overlay-text">Hello, I'm on top!</div>
 <div class="top-box"></div>
</div>
</body>
</html>
```

Output :-

Overflow in Css :-

- The overflow property in CSS is used to control how content is handled when it overflows its container.
- It is most commonly applied to elements with a fixed width and height.
- If the content is too large to fit inside the box, the overflow property determines what will happen to the extra content.

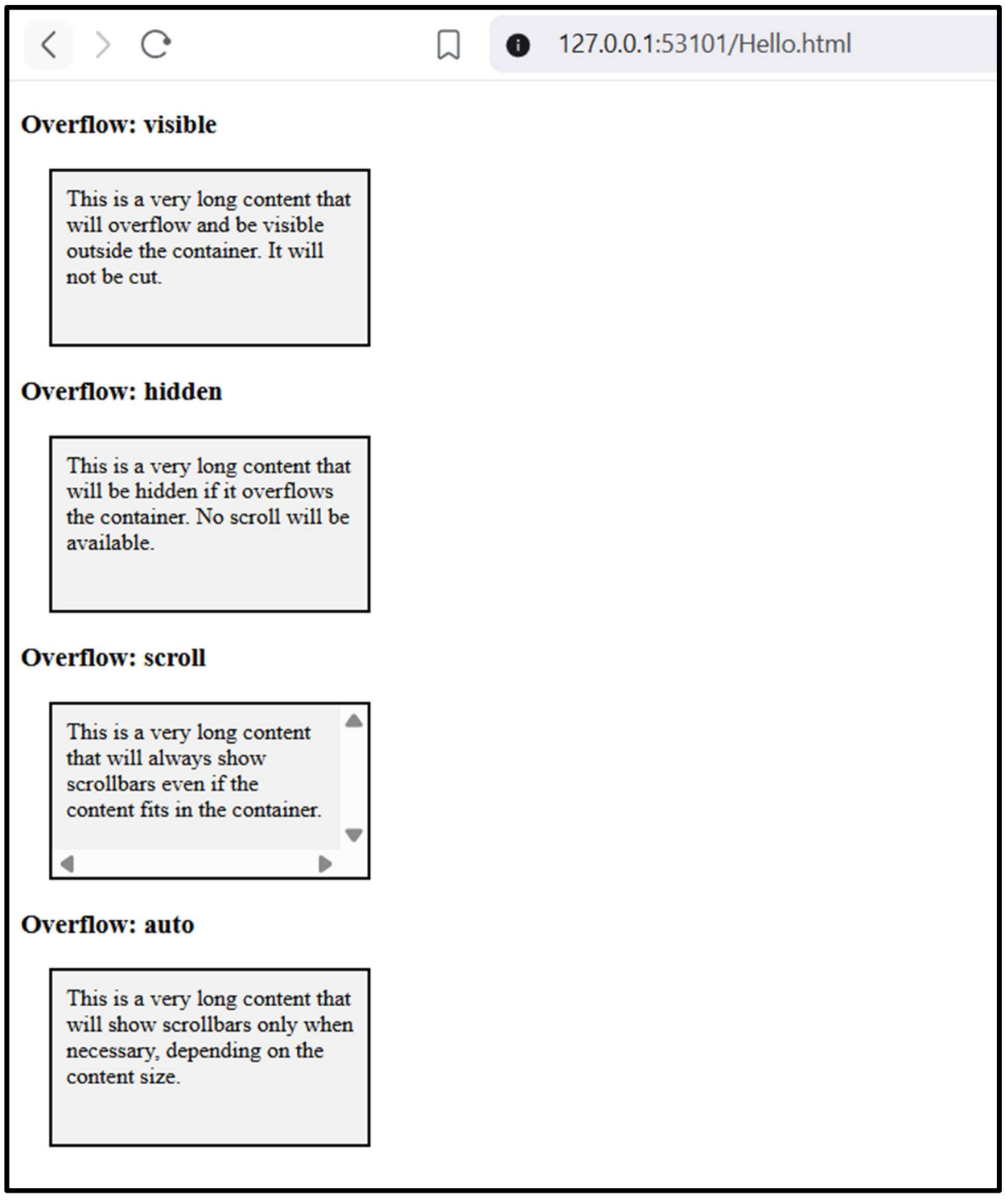
Overflow Property :-

- The default value of the overflow property is **visible**, meaning the overflowed content will be visible outside the element.
- The value **hidden** clips the content that exceeds the box and hides it from view.
- The value **scroll** adds scrollbars to the box, allowing users to scroll through the overflowed content.
- The value **auto** adds scrollbars only when necessary i.e., only if the content overflows.

Example :-

```
<html>
  <style>
   .container {
     Width : 200px;
     height : 100px;
     margin : 20px;
     padding : 10px;
     border : 2px solid black;
     background-color : #f2f2f2;
   }
   /* overflow : visible */
   .overflow-visible {
     overflow : visible;
     /* Extra content is visible outside the box */
   }
   /* overflow : hidden */
   .overflow-hidden {
     overflow : hidden;
     /* Extra content is cut off and hidden */
   }
   /* overflow : scroll */
   .overflow-scroll {
     overflow : scroll;
     /* Scrollbars are always shown */
   }
   /* overflow : auto */
   .overflow-auto {
     overflow : auto;
     /* Scrollbars appear only when needed */
   }
</style>
```

Output :-



```
<body>
  <h3>Overflow : visible</h3>
  <div class="container overflow-visible"> This is a very long content that will overflow and be visible outside the container. It will not be cut </div>
  <h3>Overflow : hidden</h3>
  <div class="container overflow-hidden"> This is a very long content that will be hidden if it overflows the container. No scroll will be available. </div>
  <h3>Overflow : scroll</h3>
  <div class="container overflow-scroll"> This is a very long content that will always show scrollbars even if the content fits in the container. </div>
  <h3>Overflow : auto</h3>
  <div class="container overflow-auto"> This is a very long content that will show scrollbars only when necessary, depending on the content size. </div>
</body>
</html>
```

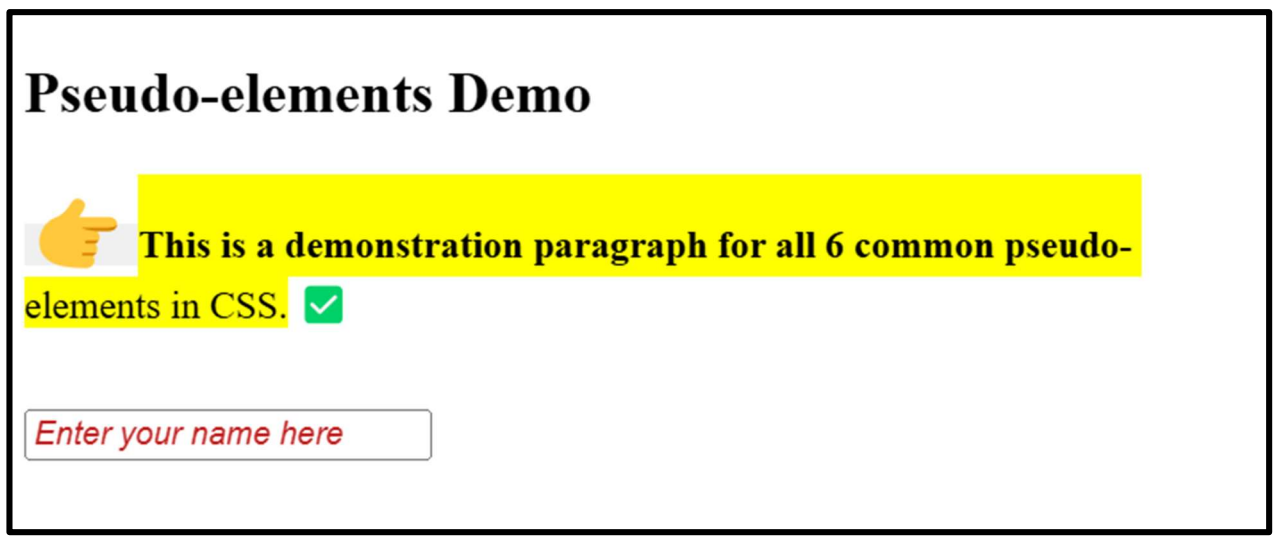
Pseudo Elements in Css :-

- A pseudo-element allows you to style specific parts of an element without adding extra HTML.
- Pseudo-elements are written using two colons (::) followed by the name of the pseudo-element.
- The **::before** pseudo-element inserts content before the content of an element.
- The **::after** pseudo-element inserts content after the content of an element.
- The **::first-letter** pseudo-element is used to style the first letter of a text block.
- The **::first-line** pseudo-element styles only the first line of a text block.
- The **::selection** pseudo-element styles the portion of text that a user highlights or selects.
- The **::placeholder** pseudo-element is used to style the placeholder text inside input and textarea fields.
- The **::marker** pseudo-element is used to style the bullet or number in list items.

Example :-

```
<!DOCTYPE html>
<html>
  <style>
    /* 1.  ::before - adds content before paragraph */
    p::before {
      content : "👉 ";
      color : blue;
    }
    /* 2.  ::after - adds content after paragraph */
    p::after {
      content : " ☑";
      color : green;
    }
    /* 3.  ::first-letter - styles the first letter of the paragraph */
    p::first-letter {
      font-size : 200%;
      color : red;
    }
```

Output :-



```
/* 4.  ::first-line - styles only the first line of the paragraph */
p::first-line {
  font-weight : bold;
  background-color : #f0f0f0;
}
/* 5.  ::selection - styles the selected text by the user */
p::selection {
  background-color : yellow;
  color : black;
}
/* 6.  ::placeholder - styles input placeholder text */
input::placeholder {
  color : rgb(194, 35, 35);
  font-style : italic;
   }
  </style>
<body>
  <h2>Pseudo-elements Demo</h2>
  <p>This is a demonstration paragraph for all 6 common pseudo-elements in CSS.</p>
  <br>
  <input type="text" placeholder="Enter your name here">
</body>
</html>
```

Pseudo Class in Css :-

- A pseudo-class in CSS is used to apply styles to an HTML element based on its special state or position, without needing to add a class or ID in the HTML.
- For example, when a user hovers over a button or when a checkbox is checked, you can style it using a pseudo-class.

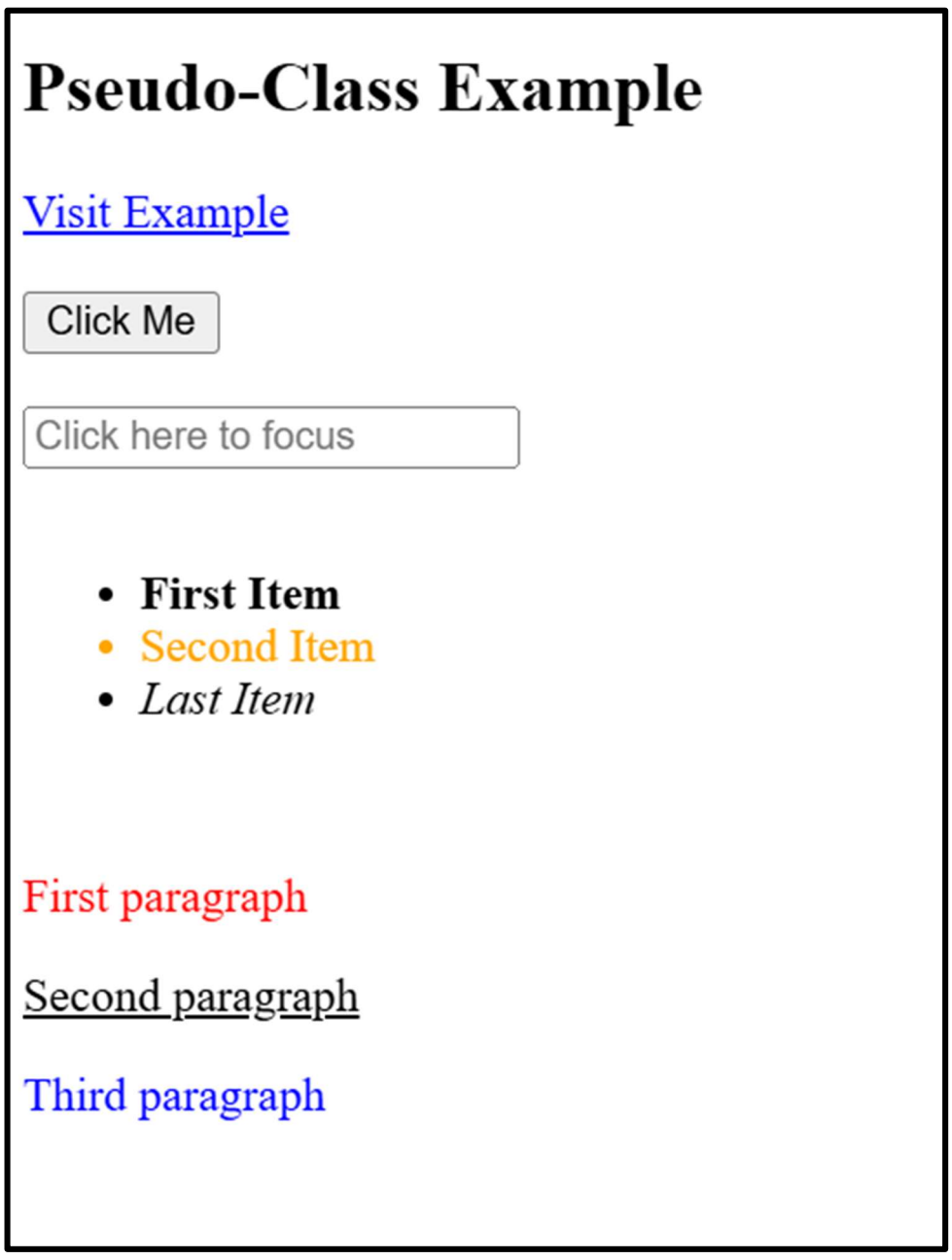
Pseudo Properties :-

- **:hover** applies when the user places their mouse pointer over an element, such as a link or button.
- **:active** applies when an element is being clicked or activated by the user.
- **:focus** applies when an element, such as an input box or button, gains focus via keyboard or mouse.
- **:visited** applies styles to links that the user has already visited.
- **:link** applies styles to hyperlinks that have not yet been visited.
- **:first-child** selects an element that is the first child inside its parent.
- **:last-child** selects an element that is the last child inside its parent.
- **:nth-child(n)** selects the element that is the nth child in its parent, where n can be a number, keyword, or formula.
- **:first-of-type** selects the first element of a specific type within its parent container.
- **:last-of-type** selects the last element of a specific type within its parent container.
- **:nth-of-type(n)** selects the nth element of a specific type among its siblings.

Example :-

```
<!DOCTYPE html>
<html>
<style>
/* :hover - changes link color when hovered */
a:hover {
  text-decoration : none;
  color : rgb(255, 202, 8);
}
/* :active - changes button background when clicked */
button:active {
  background-color : green;
}
/* :focus - highlights input when focused */
input:focus {
  border : 5px solid rgb(255, 202, 8);
}
/* :visited - changes color of visited links */
a:visited {
  color : purple;
}
/* :link - changes color of unvisited links */
a:link {
  color : blue;
}
/* :first-child - makes first list item bold */
ul li:first-child {
  font-weight : bold;
}
/* :last-child - makes last list item italic */
ul li:last-child {
  font-style : italic;
}
/* :nth-child(n) - colors every second list item */
ul li:nth-child(2) {
  color : orange;
}
/* :first-of-type - makes first paragraph red */
div p:first-of-type {
  color : red;
}
/* :last-of-type - makes last paragraph blue */
div p:last-of-type {
  color : blue;
}
/* :nth-of-type(n) - underlines the second paragraph */
div p:nth-of-type(2) {
  text-decoration : underline;
}
</style>
```

Output :-



```
<body>
<h2>Pseudo-Class Example</h2>
<!-- : hover, : visited, : link -->
<a href="https://www.example.com" target="_blank">Visit Example</a>   <br><br>
<!-- : active -->
<button>Click Me</button>  <br><br>
<!-- : focus -->
<input type="text" placeholder="Click here to focus">   <br><br>
<!-- : first-child, : last-child, : nth-child -->
<ul>
  <li>First Item</li>
  <li>Second Item</li>
  <li>Last Item</li>
</ul>  <br>
<!-- : first-of-type, : last-of-type, : nth-of-type -->
<div>
  <p>First paragraph</p>
  <p>Second paragraph</p>
  <p>Third paragraph</p>
</div>
</body>  </html>
```

Column Layout in Css :-

- The column layout in CSS allows you to split content into multiple vertical columns, similar to newspaper-style layouts.

Column Layout Property :-

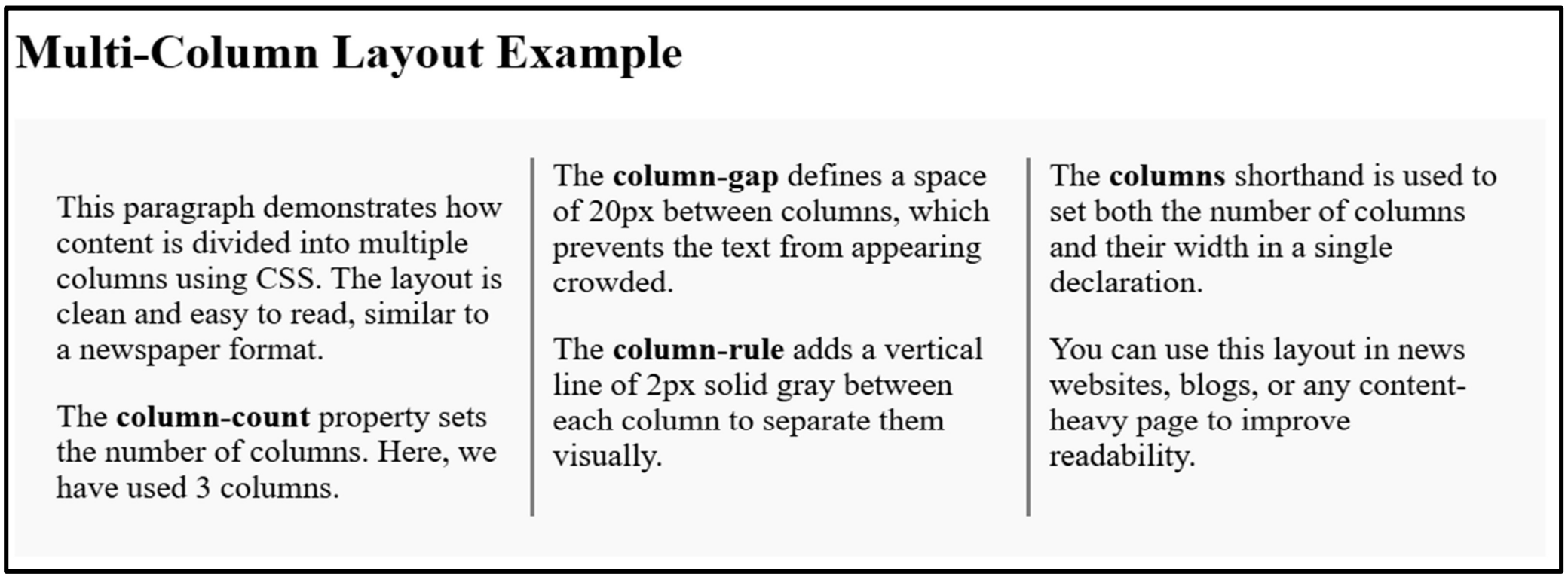
- The **column-count** property specifies the exact number of columns into which the content should be divided.
- The **column-gap** property defines the space between the columns to prevent the content from being too close.
- The **column-rule** property adds a vertical line between the columns, just like a border, for visual separation.
- The **columns** property is a shorthand that combines both column-count and column-width in one line.
- The **column-width** property sets the ideal width of each column, and the browser automatically decides how many columns can fit.

Example :-

```
 <html>
<style>
   .column-box {
     columns : 3 200px;           /* Shorthand for 3 columns with 200px width each */
     column-gap : 20px;           /* Space between columns */
     column-rule : 2px solid gray; /* Vertical line between columns */
     padding : 20px;
     background-color : #f9f9f9;
   }
 </style>
```

```
<body>
<h2>Multi-Column Layout Example</h2>
  <div class="column-box">
<p> This paragraph demonstrates how content is divided into multiple columns using CSS. The layout is clean and easy to read, similar to a newspaper format. </p>
    <p> The <strong>column-count</strong> property sets the number of columns. Here, we have used 3 columns.  </p>
    <p> The <strong>column-gap</strong> defines a space of 20px between columns, which prevents the text from appearing crowded.  </p>
    <p> The <strong>column-rule</strong> adds a vertical line of 2px solid gray between each column to separate them visually.  </p>
    <p> The <strong>columns</strong> shorthand is used to set both the number of columns and their width in a single declaration.  </p>
    <p> You can use this layout in news websites, blogs, or any content-heavy page to improve readability.  </p>
  </div>
</body>
</html>
```

Output :-



CSS Layout - Float and Clear :-

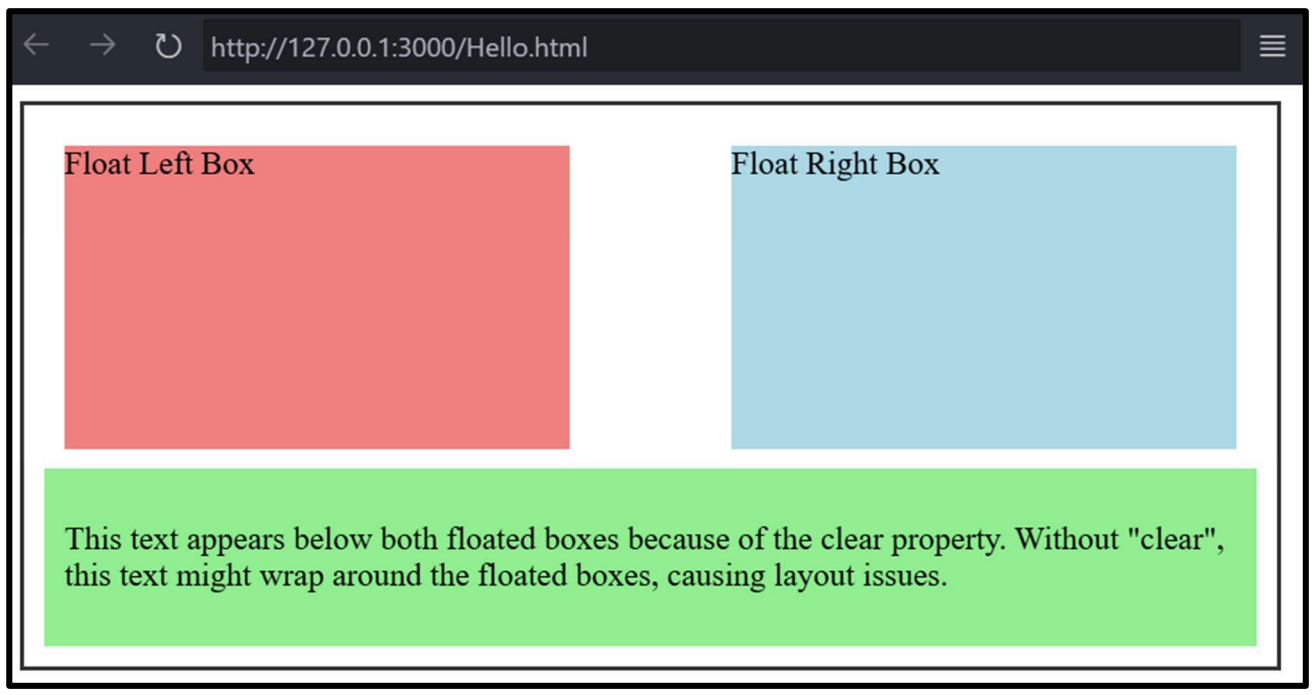
- The float property is used to place an element to the left or right of its container, allowing text or inline elements to wrap around it.
- The clear property is used to prevent an element from wrapping around a floated element. It is commonly used to fix layout issues when floats are used.

Example :-

```
<!DOCTYPE html>
<html>
  <style>
    .container {
      width : 600px;            /* width of container */
      border : 2px solid #333;   /* solid border with dark color */
      padding : 10px;           /* space inside container */
    }
```

```
.box1 {
  float : left;            /* float element to the left */
  width : 250px;           /* width of box */
  height : 150px;          /* height of box */
  background-color : lightcoral; /* background color */
  margin : 10px;           /* margin around box */
}
.box2 {
  float : right;           /* float element to the right */
  width : 250px;           /* width of box */
  height : 150px;          /* height of box */
  background-color : lightblue;  /* background color */
  margin : 10px;           /* margin around box */
}
  .clearfix {   clear : both;    }          /* clear both left and right floats */
  .text {
    background-color : lightgreen; /* background color for text box */
    padding : 10px;          /* space inside the text box */
  }
</style>
<body>
 <div class="container">
  <div class="box1">Float Left Box</div>  <!—float : left -->
  <div class="box2">Float Right Box</div> <!—float : right -->
  <div class="clearfix"></div> <!—clear : both -->
  <div class="text">
   <p> This text appears below both floated boxes because of the clear property.  Without "clear", this text might wrap around the floated boxes, causing
     layout issues.  </p>
  </div>
 </div>
</body> </html>
```

Output :-



CSS Opacity :-

- The opacity property in CSS is used to control the transparency level of an element.
- The opacity property can take a value from 0.0 to - 1.0. The lower the value, the more transparent.

Example :-

```
.box2 {
        opacity : 0.5;  /* 50% transparent */
        width : 200px;
        height : 100px;
       background-color : green;   }
```

CSS Table :-

- In CSS, a table refers to the styling of HTML table elements (like <table>, <tr>, <td>, <th>) to control the layout, spacing, borders, and appearance of tabular data on a web page.

Example :-

```
<!DOCTYPE html>
<html>
 <style>
   table {
     border-collapse : collapse; /* Merges borders */
     width : 80%; /* Table width */
     height : 200px; /* Table height */
     margin : 20px auto; /* Center table */
   }
   th, td {
     border : 2px solid black; /* Table border */
     padding : 12px;
     text-align : center;
   }
   tr:hover {
     background-color : #f2f2f2; /* Hover effect */
   }
   th {
     background-color : #4CAF50;
     color : white;
   }
 </style>
<body>
<h2 style="text-align : center;">CSS Table Example</h2>
<table>
 <tr> <th>Roll No</th>  <th>Name</th>  <th>Marks</th>  </tr>
 <tr> <td>01</td>   <td>Zeel</td>  <td>95</td>  </tr>
 <tr>  <td>02</td>  <td>Raj</td>   <td>89</td>  </tr>
 <tr>  <td>03</td>  <td>Priya</td> <td>92</td>  </tr>
</table>
</body>
</html>
```

Output :-

CSS Table Example

Roll No	Name	Marks
01	Zeel	95
02	Raj	89
03	Priya	92

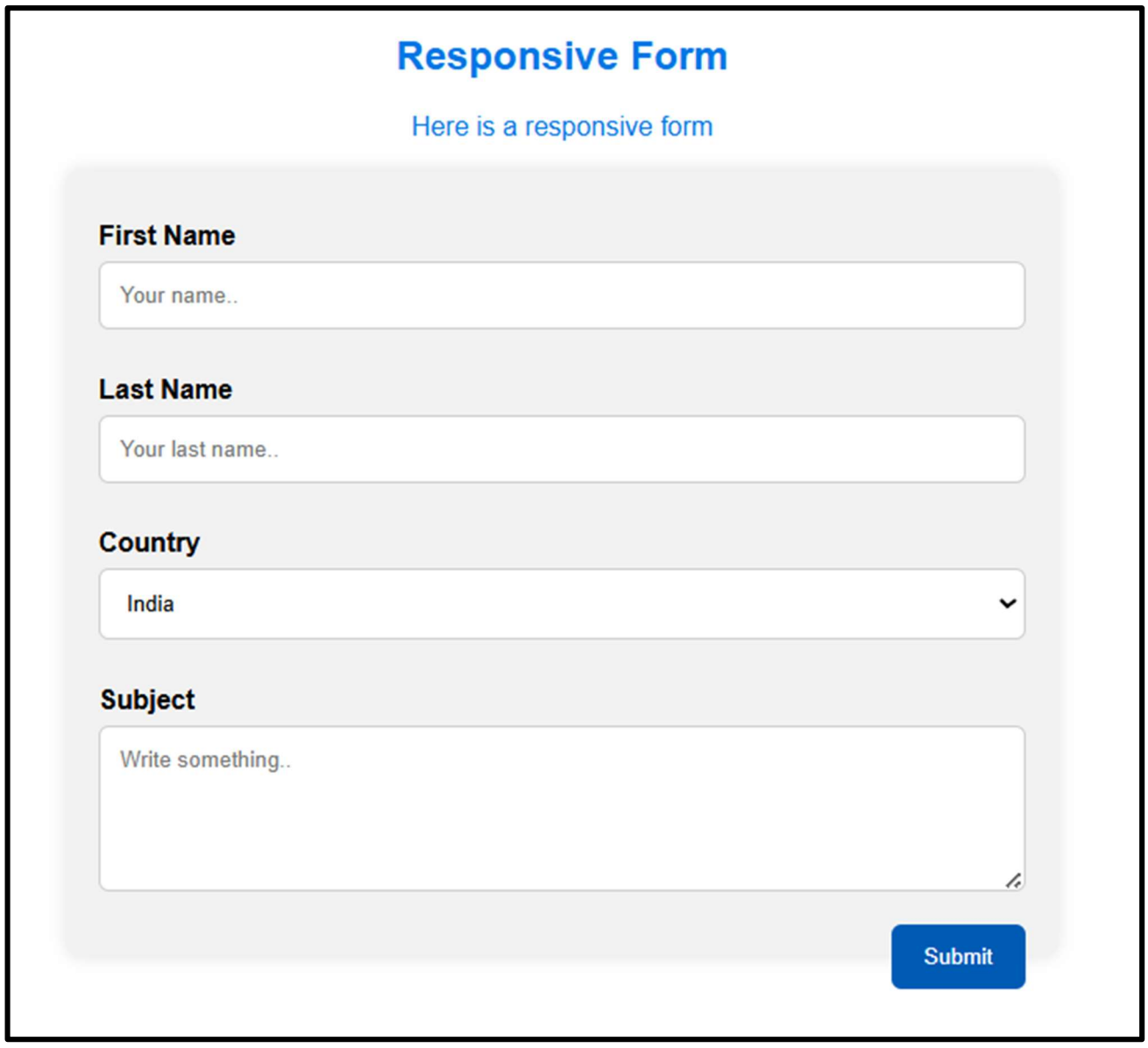
Form in CSS :-

Example :-

```
<html> <style>
   * {   box-sizing : border-box;
         font-family : Arial, sans-serif;
     }
```

```
body {
  margin : 0;
  padding : 0;
  background-color : #ffffff;
}
h2, p {
  text-align : center;
  color : #0077e6;
}
.container {
  border-radius : 10px;
  background-color : #f2f2f2;
  padding : 20px;
  max-width : 600px;
  margin : auto;
  box-shadow : 0 0 10px rgba( 0, 0, 0, 0.1 );
}
label {
  padding : 12px 0 6px 0;
  display : block;
  font-weight : bold;
}
input[type=text], select, textarea {
  width : 100%;
  padding : 12px;
  border : 1px solid #ccc;
  border-radius : 6px;
  resize : vertical;
  margin-bottom : 16px;
}
  input[type=submit] {
    background-color : #0077e6;
    color : white;
    padding : 12px 20px;
    border-radius : 6px;
    cursor : pointer;
    float : right;
  }
  input[type=submit]:hover {
    background-color : #005bb5;
  }
 </style>
<body>
```

Output :-



```
<h2>Responsive Form</h2>
<p>Here is a responsive form</p>
<div class="container">
  <form>
    <label for="fname">First Name</label>
    <input type="text" id="fname" name="firstname" placeholder="Your name..">
    <label for="lname">Last Name</label>
    <input type="text" id="lname" name="lastname" placeholder="Your last name..">
    <label for="country">Country</label>
    <select id="country" name="country">
      <option value="india">India</option>
      <option value="usa">USA</option>
      <option value="uk">UK</option>
    </select>
    <label for="subject">Subject</label>
    <textarea id="subject" name="subject" placeholder="Write something.." style="height:100px"></textarea>
    <input type="submit" value="Submit">
  </form>
 </div>
</body>
</html>
```

Flexbox in CSS :-

- Flexbox is a CSS layout model used to design responsive and efficient layouts by distributing space and aligning items within a container, even when their sizes are unknown or dynamic.
- It allows elements to be arranged in rows or columns and provides powerful alignment and spacing capabilities.
- To activate Flexbox, we set the container's CSS property to display : flex.

Flexbox Parent Property :-

- The **flex-direction** property defines the direction of the items (e.g., row | column | row-reverse | column-reverse).
- **justify-content** property aligns items horizontally(e.g. left | right | flex-start | flex-end | center | space-between | space-around | space-evenly)
- The **align-items** property aligns items vertically within the container. (e.g. stretch | flex-start | flex-end | center | baseline)
- The **flex-wrap** property used with flex containers to control whether the flex items should stay in one line or wrap onto multiple lines if there isn't enough space.

Example of Parent Property :-

```
<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="UTF-8">
 <title>Flexbox Examples</title>
 <style>
   body {
     font-family : Arial;
     margin : 20px;
   }
```

```
.section-title {
  margin-top : 40px;
  font-size : 20px;
  font-weight : bold;
  color : #333;
}
.container {
  display : flex;
  flex-wrap : wrap;
  border : 2px solid #333;
  padding : 10px;
  margin-bottom : 20px;
  min-height : 120px;
  gap : 10px;
  background : #f4f4f4;
}
.item {
  width : 80px;
  height : 40px;
  background-color : teal;
  color : white;
  text-align : center;
  line-height : 40px;
}
.label {
  margin-top : 5px;
  font-size : 14px;
  color : #555;
}
</style>
</head>
<body>
<!-- FLEX-DIRECTION -->
<div class="section-title">flex-direction</div>
<div class="label">row (default)</div>
<div class="container" style="flex-direction : row;">
<div class="item">1</div>
<div class="item">2</div>
<div class="item">3</div>
</div>
<div class="label">row-reverse</div>
<div class="container" style="flex-direction : row-reverse;">
<div class="item">1</div>  <div class="item">2</div>  <div class="item">3</div> </div>
```

```
<div class="label">column</div>
<div class="container" style="flex-direction : column; height : 200px;">
  <div class="item">1</div>
  <div class="item">2</div>
  <div class="item">3</div>
</div>
<div class="label">column-reverse</div>
<div class="container" style="flex-direction : column-reverse; height : 200px;">
  <div class="item">1</div>
  <div class="item">2</div>
  <div class="item">3</div>
</div>
<!-- JUSTIFY-CONTENT -->
<div class="section-title">justify-content</div>
<div class="label">flex-start</div>
<div class="container" style="justify-content : flex-start;">
  <div class="item">A</div><div class="item">B</div><div class="item">C</div>
</div>
<div class="label">flex-end</div>
<div class="container" style="justify-content : flex-end;">
  <div class="item">A</div><div class="item">B</div><div class="item">C</div>
</div>
<div class="label">center</div>
<div class="container" style="justify-content : center;">
  <div class="item">A</div><div class="item">B</div><div class="item">C</div>
</div>
<div class="label">space-between</div>
<div class="container" style="justify-content : space-between;">
  <div class="item">A</div><div class="item">B</div><div class="item">C</div>
</div>
<div class="label">space-around</div>
<div class="container" style="justify-content : space-around;">
  <div class="item">A</div><div class="item">B</div><div class="item">C</div>
</div>
<div class="label">space-evenly</div>
<div class="container" style="justify-content : space-evenly;">
<div class="item">A</div><div class="item">B</div><div class="item">C</div>
</div>
<!-- ALIGN-ITEMS -->
<div class="section-title">align-items</div>
<div class="label">stretch (default)</div>
<div class="container" style="align-items : stretch; height: 100px;">
  <div class="item">A</div><div class="item">B</div><div class="item">C</div>    </div>
```

```
<div class="label">flex-start</div>
<div class="container" style="align-items : flex-start; height: 100px;">
  <div class="item">A</div><div class="item" style="height: 60px;">B</div><div class="item">C</div>
</div>
<div class="label">flex-end</div>
<div class="container" style="align-items : flex-end; height: 100px;">
  <div class="item">A</div><div class="item" style="height : 60px;">B</div><div class="item">C</div>
</div>
<div class="label">center</div>
<div class="container" style="align-items : center; height: 100px;">
  <div class="item">A</div><div class="item" style="height : 60px;">B</div><div class="item">C</div>
</div>
<div class="label">baseline</div>
<div class="container" style="align-items : baseline; height: 100px;">
   <div class="item" style="font-size : 12px;">A</div>
   <div class="item" style="font-size : 24px;">B</div>
   <div class="item" style="font-size : 18px;">C</div>
 </div>
</body>
</html>
```

Flexbox Child Property :-

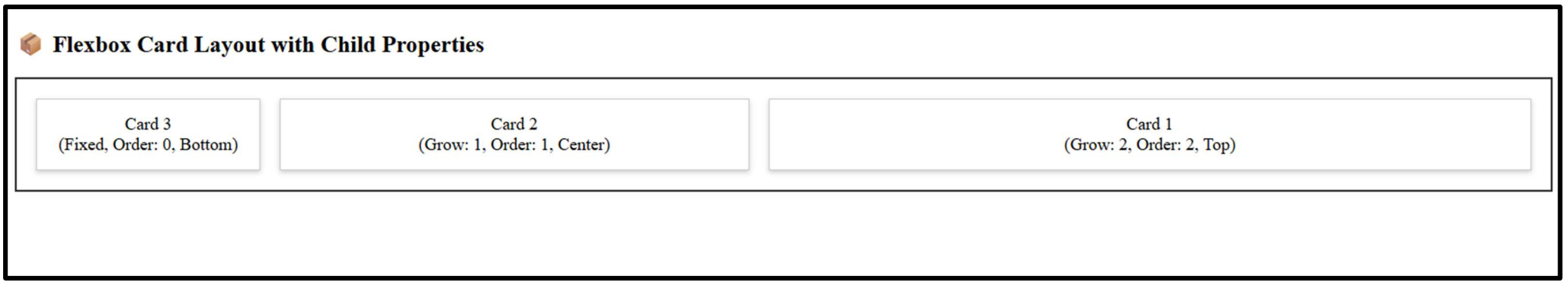
- The **order** property defines the order in which flex items appear within the flex container.
- The **flex-grow** property defines how much a flex item should grow relative to the other items when extra space is available in the container.
- The **flex-shrink** property determines how much a flex item should shrink if the container is too small to fit all items.
- The **flex-basis** property sets the initial **main size** of a flex item before any remaining space is distributed.
- The **flex** property is a shorthand for setting all three properties: flex-grow, flex-shrink, and flex-basis in a single line.
- The **align-self** property overrides the flex containers align-items property for an individual item. It controls the alignment of a specific item along the cross-axis.

Example :-

```
<!DOCTYPE html>
<html lang="en">
  <style>
    .card-container {
      display : flex;              /* Flexbox container */
      gap : 20px;                  /* Space between cards */
      border : 2px solid #333;
      padding : 20px;
      align-items : stretch;       /* Stretch all cards to equal height */
    }
```

```
.card {
  background-color : white;
  border : 1px solid #ccc;
  padding : 15px;
  text-align : center;
  font-size : 18px;
  box-shadow : 0 2px 5px rgba(0,0,0,0.2);
}
.card1 {
  order : 2;                  /* Appears third in row */
  flex-grow : 2;              /* Grows twice as much as others */
  flex-shrink : 1;            /* Shrinks normally if space is tight */
  flex-basis : 150px;         /* Starts at 150px */
  align-self : flex-start;    /* Align to top of container */
}
.card2 {
  order : 1;                  /* Appears second in row */
  flex-grow : 1;              /* Normal grow */
  flex-shrink : 1;            /* Normal shrink */
  flex-basis : 150px;         /* Starts at 150px */
  align-self : center;        /* Align to center vertically */
}
.card3 {
  order : 0;                  /* Appears first in row */
  flex : 0 0 200px;           /* No grow, no shrink, fixed width */  /* Syntax :- flex : <flex-grow> <flex-shrink> <flex-basis>; */
  align-self : flex-end;      /* Align to bottom of container */
}
</style>
<body>
<h2> 📦 Flexbox Card Layout with Child Properties</h2>
<div class="card-container">
  <div class="card card1">Card 1<br>(Grow : 2, Order : 2, Top)</div>
  <div class="card card2">Card 2<br>(Grow : 1, Order : 1, Center)</div>
  <div class="card card3">Card 3<br>(Fixed, Order : 0, Bottom)</div>
</div>
</body> </html>
```

Output :-



GRID in CSS :-

- CSS Grid is a layout system in CSS that allows you to create complex, responsive web designs by arranging elements into rows and columns.
- It is a two-dimensional layout system, meaning it can handle both horizontal (rows) and vertical (columns) alignment of content.

Grid Container Properties :-

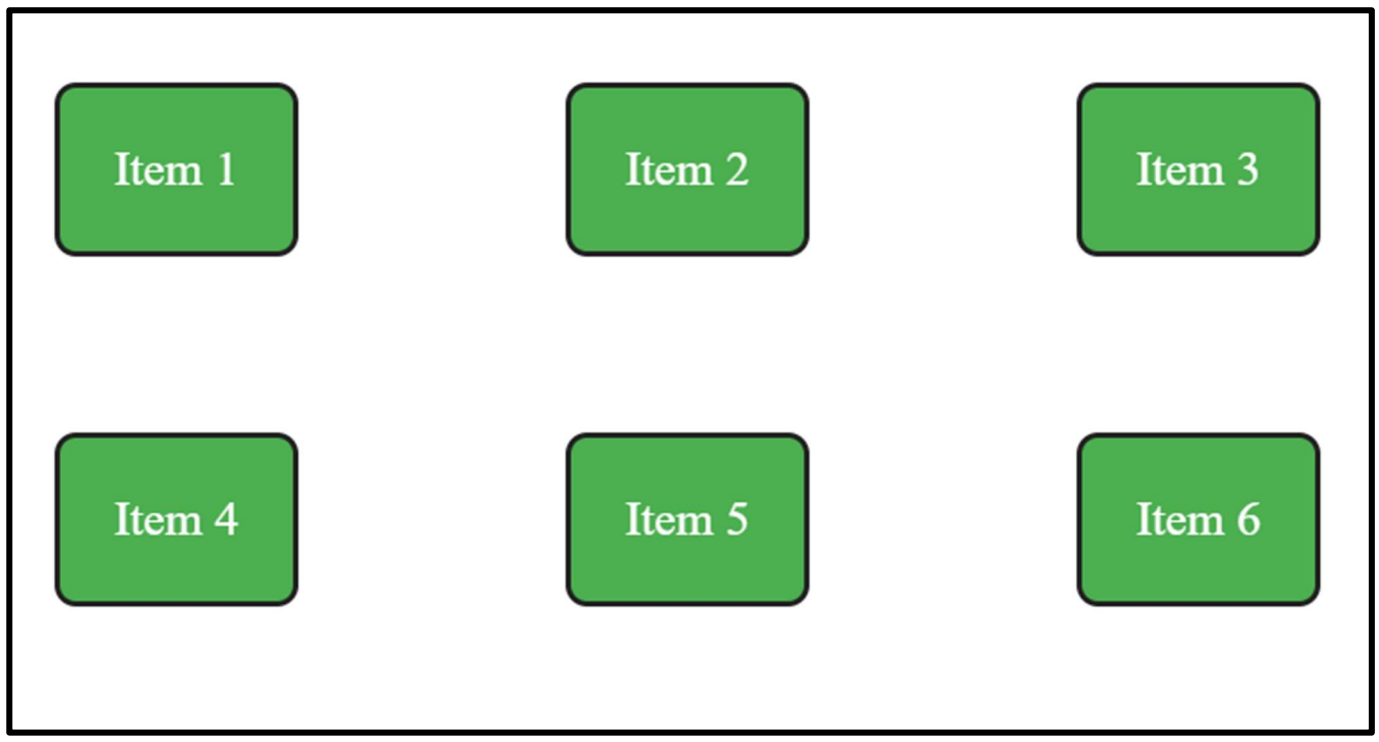
- The **grid-template-columns** property defines the number of columns in the grid and the width of each column.
- The **grid-template-rows** property sets the number of rows in the grid and the height of each row.
- The **gap** property sets the space (gap) between rows and columns in the grid layout.
- The **row-gap** property specifically sets the vertical space between the grid rows.
- The **column-gap** property specifically sets the horizontal space between the grid columns.
- The **justify-content** property aligns the entire grid structure horizontally inside the container.
- The **align-content** property aligns the entire grid structure vertically inside the container.
- The **place-content** property is shorthand for setting both align-content and justify-content in one line.
- The **justify-items** property aligns grid items horizontally within their cells inside the grid container.
- The **align-items** property aligns grid items vertically within their cells inside the grid container.
- The **place-items** property is shorthand for setting both align-items and justify-items in one line.

Example Container (Parent Property) :-

```
<!DOCTYPE html>
<html lang="en">
  <style>
    /* Grid container */
    .grid-container {
      Display : grid; /* Enables CSS Grid layout */
      grid-template-columns : 1fr 2fr 1fr; /* Creates 3 columns: left and right are 1fr, middle is 2fr wide */
      grid-template-rows : 100px 100px; /* Creates 2 rows of 100px height */

      gap : 10px; /* Sets space between both rows and columns */
      row-gap : 30px; /* Sets vertical gap between rows */
      column-gap : 20px; /* Sets horizontal gap between columns */

      justify-content : center; /* Aligns the entire grid horizontally inside the container */
      align-content : center; /* Aligns the entire grid vertically inside the container */
      place-content : center; /* Shorthand for justify-content and align-content */
      justify-items : center; /* Aligns items horizontally inside each grid cell */
      align-items : center; /* Aligns items vertically inside each grid cell */
      place-items : center; /* Shorthand for justify-items and align-items */
    }
    /* Grid items */
    .grid-item {
      background-color : #4CAF50;
      color : white;
      padding : 20px;
      font-size : 18px;
      text-align : center;
```

```
    border-radius : 8px;
    border : 2px solid #222;
  }
 </style>
</head>
<body>
 <div class="grid-container">
  <div class="grid-item">Item 1</div>
  <div class="grid-item">Item 2</div>
  <div class="grid-item">Item 3</div>
  <div class="grid-item">Item 4</div>
  <div class="grid-item">Item 5</div>
  <div class="grid-item">Item 6</div>
 </div>
</body>
</html>
```

Output :-



Grid Item Properties :-

- The **grid-column-start** property sets the starting column line where the grid item will begin.
- The **grid-column-end** property sets the ending column line where the grid item will stop.
- The **grid-row-start** property sets the starting row line where the grid item will begin.
- The **grid-row-end** property sets the ending row line where the grid item will stop.
- The **grid-column** is a shorthand property for setting both grid-column-start and grid-column-end.
- The **grid-row** is a shorthand property for setting both grid-row-start and grid-row-end.
- The **grid-area** property defines a grid item's size and position, or places it in a named area.
- The **justify-self** property aligns the grid item horizontally inside its individual grid cell.
- The **align-self** property aligns the grid item vertically inside its individual grid cell.
- The **place-self** property is shorthand for setting both align-self and justify-self.

Example Item (Child Property) :-

```
<!DOCTYPE html>
<html lang="en">
<style>
.grid-container {
  display : grid;
  grid-template-columns : repeat(4, 100px); /* 4 columns of 100px each */
  grid-template-rows : repeat(4, 100px);    /* 4 rows of 100px each */
  gap : 10px;
  background-color : #f0f0f0;
  padding : 20px;
}
/* Common style for all items */
.item {
  background-color : #4CAF50;
  color : white;
```

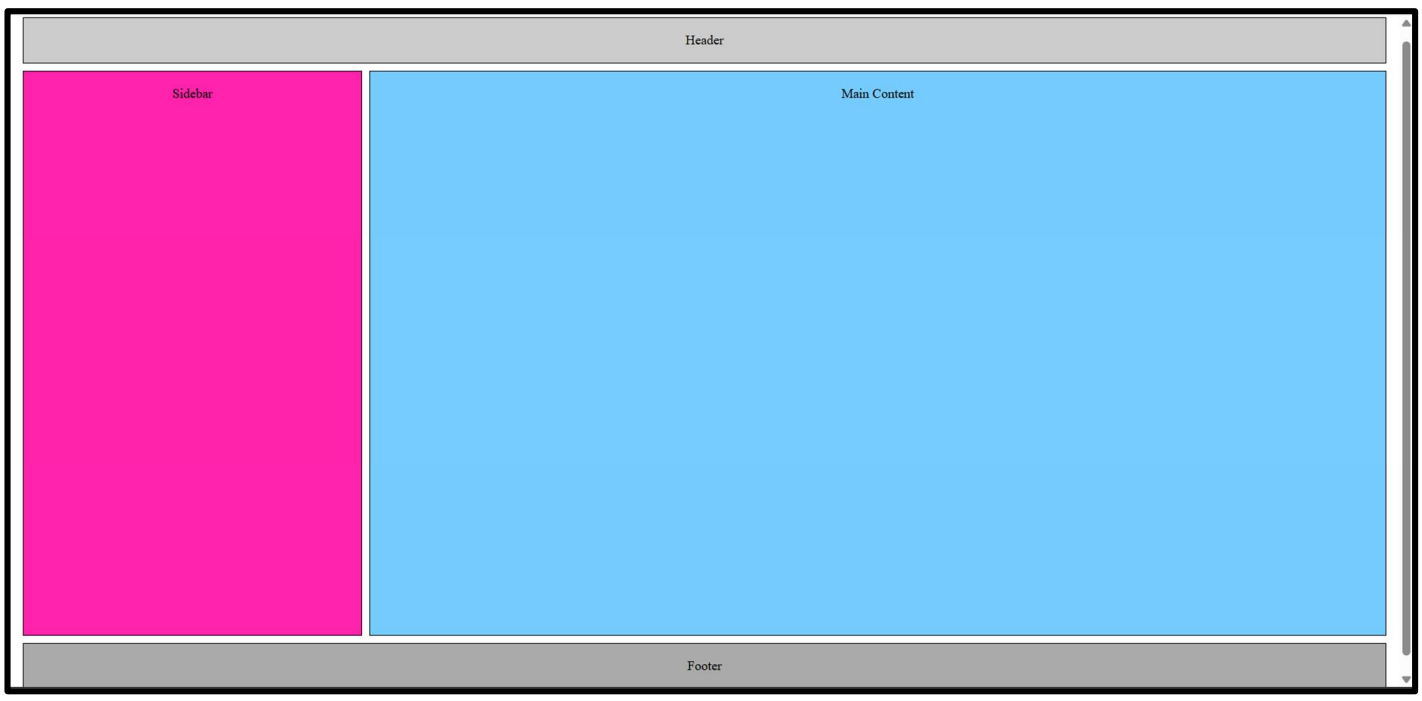
```
  font-size : 18px;
  display : flex;
  align-items : center;
  justify-content : center;
}
/* Using grid-column-start and grid-column-end */
.item1 {
  grid-column-start : 1;  /* Start at column 1 */
  grid-column-end : 3;    /* End at column 3 */
  grid-row-start : 1;     /* Start at row 1 */
  grid-row-end : 2;       /* End at row 2 */
}
/* Using shorthand grid-column and grid-row */
.item2 {
  grid-column : 3 / 5;     /* Same as start 3, end 5 */
  grid-row : 1 / 2;        /* Same as start 1, end 2 */
}
/* Using grid-area to define exact placement */
.item3 {
  grid-area : 2 / 1 / 4 / 3;  /* row-start / col-start / row-end / col-end */
}
/* Using justify-self, align-self, and place-self */
.item4 {
  grid-column : 3 / 5;
  grid-row : 2 / 4;
  background-color : #FF5722;
}
  </style>
  <body>
  <div class="grid-container">
    <div class="item item1">Item 1</div>
    <div class="item item2">Item 2</div>
    <div class="item item3">Item 3</div>
    <div class="item item4">Item 4</div>
  </div>
</body>
</html>
```

Output :-

Example :-

```
<!DOCTYPE html>
<html lang="en">
<style>
/* Container as Grid */
.grid-container {
  display : grid; /* Enables grid layout */
  grid-template-columns : 1fr 3fr; /* 2 columns: 1 part sidebar, 3 parts content */
  grid-template-rows : auto 1fr auto; /* 3 rows: header, main, footer */
  gap : 10px; /* Space between items */
  height : 100vh; /* Full viewport height */
  padding : 10px;
}
/* Grid Items Placement */
.item1 {
  grid-column : 1 / 3; /* Span header across both columns */
  background-color : #ccc;
}
.item2 {
  grid-row : 2 / 3; /* Sidebar on left */
  background-color : #f2a;
}
.item3 {
  grid-row : 2 / 3;
  grid-column : 2 / 3; /* Main content on right */
  background-color : #7cf;
}
.item4 {
  grid-column : 1 / 3; /* Footer spans both columns */
  background-color : #aaa;
}
/* All Items Common Style */
.item {
  Padding : 20px;
  color : #000;
  font-size : 18px;
  text-align : center;
  border : 1px solid #000;
}
  </style>
```

Output :-




```
<body>
  <div class="grid-container">
    <div class="item item1">Header</div>
    <div class="item item2">Sidebar</div>
    <div class="item item3">Main Content</div>
    <div class="item item4">Footer</div>
  </div>
</body>
</html>
```

Transitions in CSS :-

- The CSS Transitions allow you to smoothly change property values over a given duration, instead of the change happening instantly.
- The shorthand syntax follows the format :- transition : property name | duration | timing-function | delay ;
- Example :- transition : background-color 5s ease 2s, width 5s ease 2s;

Transition Property :-

- The transition-property specifies which CSS property should be transitioned.
- The transition-duration defines how long the transition takes to complete.
- The transition-timing-function determines the speed curve of the transition (e.g., linear or ease-in).
- The transition-delay sets the amount of time to wait before the transition starts.

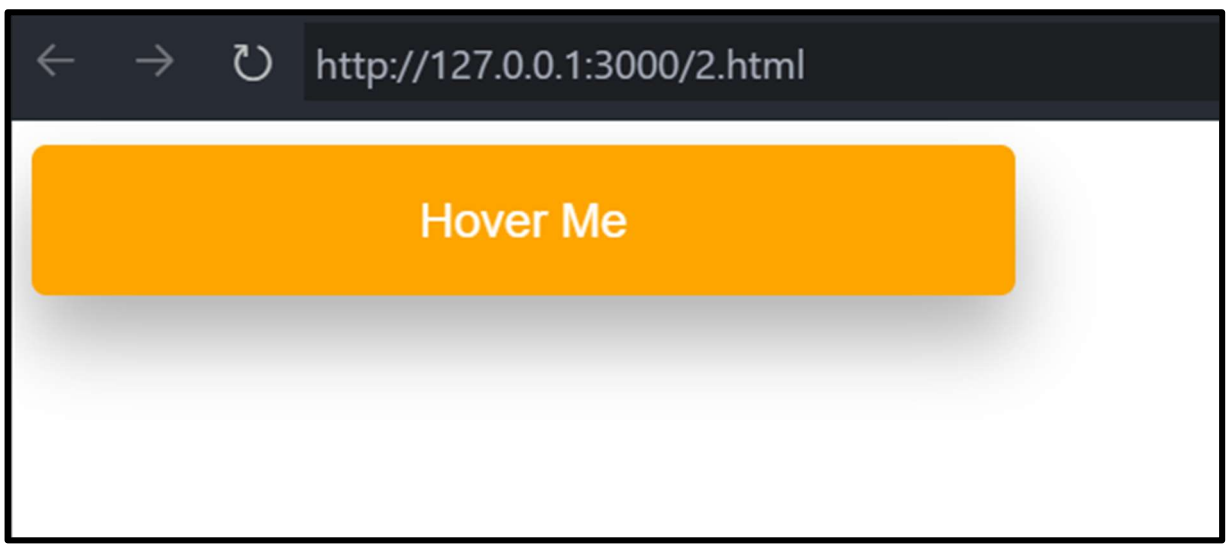
Common Timing Functions :-

- ease starts slow, becomes faster, and ends slow (default).
- linear maintains the same speed throughout the transition.
- ease-in starts slow and ends fast.
- ease-out starts fast and ends slow.
- ease-in-out starts and ends slow.

Example :-

```
<!DOCTYPE html>
<html>
<style>
button {
  width : 150px;
  height : 50px;
  font-size : 16px;
  background-color : green;
  color : white;
  border : none;
  border-radius : 5px;
  transition-property : background-color, width;
  transition-duration : 5s, 5s;
  transition-timing-function : ease-in-out;
  transition-delay : 2s, 2s;
}
button:hover {
  background-color : orange;
  width : 20.5rem;
  box-shadow : 0px 15px 25px rgba(0, 0, 0, 0.3);
}
</style>
<body>
<button>Hover Me</button>
</body>
</html>
```

Output :-



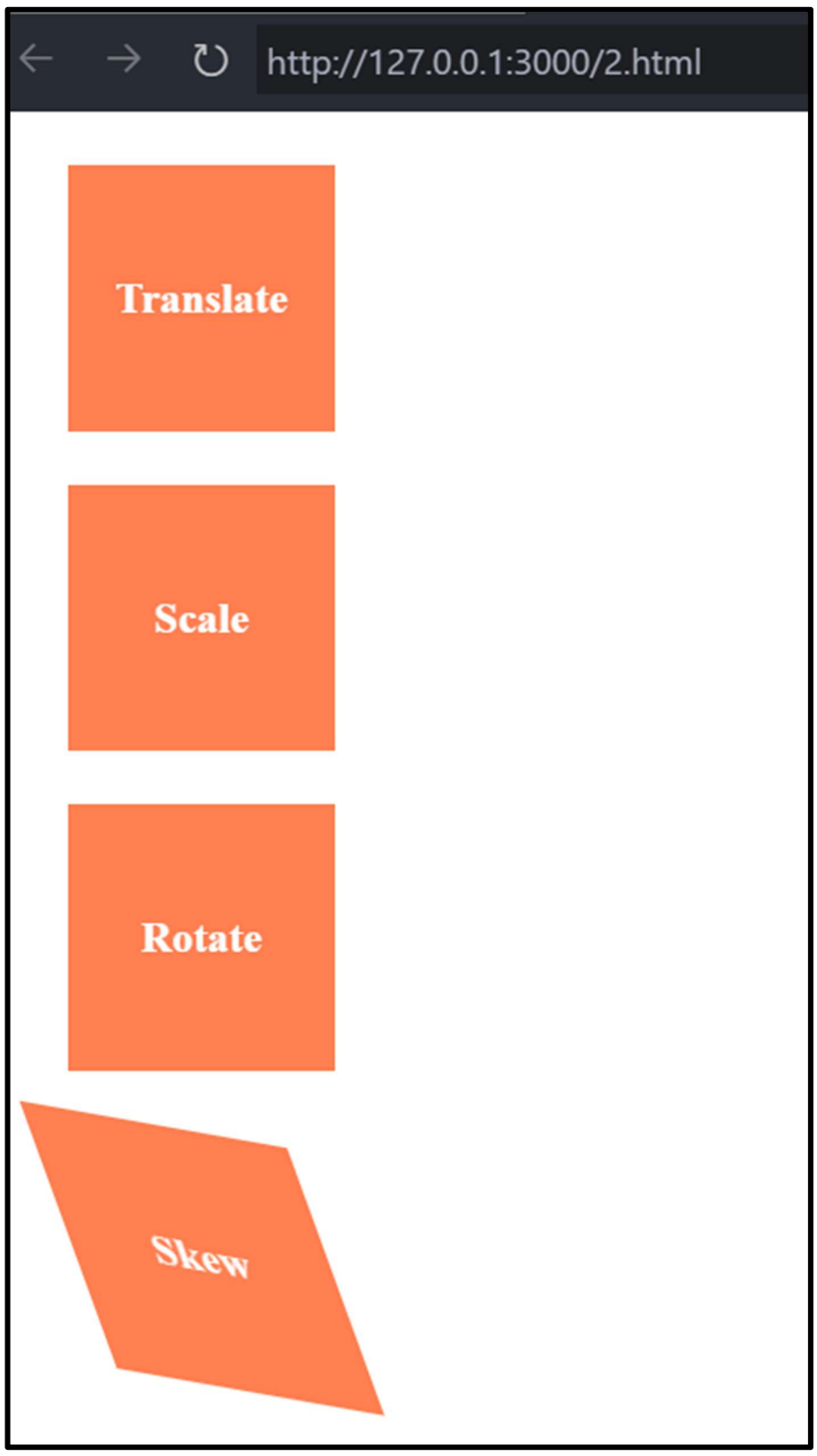
Transform in CSS :-

- The transform property in CSS allows you to visually change (or "transform") an element's position, size, shape, and orientation without affecting the surrounding elements or layout.

Example :-

```
<!DOCTYPE html>
<html>
  <style>
    .box {
      width : 100px;
      height : 100px;
      background-color : coral;
      color : white;
      font-weight : bold;
      text-align : center;
      line-height : 100px;
      margin : 20px;
      transition : transform 0.5s;
    }
    /* Translate example */
    .translate-box:hover {
      transform : translate(50px, 30px); /* Move right 50px and down 30px */

    }
    /* Scale example */
    .scale-box:hover {
      transform : scale(1.5); /* 1.5 times bigger */
    }
    /* Rotate example */
    .rotate-box:hover {
      transform : translateX(100px)  rotate(45deg); /* Rotate 45 degrees */
    }
    /* Skew example */
    .skew-box:hover {
      transform : skew(20deg, 10deg); /* Skew X : 20deg, Y : 10deg */
    }
  </style>
<body>
  <div class="box translate-box">Translate</div>
  <div class="box scale-box">Scale</div>
  <div class="box rotate-box">Rotate</div>
  <div class="box skew-box">Skew</div>
</body>
</html>
```

Output :-



➤ **Values and Examples :-**

1. translate – Moves the element :-

Function	Description	Example
translate(x, y)	Move in 2D (X and Y)	translate(50px, 30px)
translateX(x)	Move only along X	translateX(100px)
translateY(y)	Move only along Y	translateY(-50px)
translateZ(z)	Move in 3D (needs perspective)	translateZ(100px)
translate3d(x, y, z)	Move in 3D	translate3d(50px, 30px, 40px)

2. scale – Resize the element :-

Function	Description	Example
scale(x, y)	Scale in 2D	scale(1.5, 2)
scaleX(x)	Scale width only	scaleX(2)
scaleY(y)	Scale height only	scaleY(0.5)
scaleZ(z)	Scale in 3D (needs perspective)	scaleZ(1.2)
scale3d(x, y, z)	Scale in 3D	scale3d(1.5, 2, 1.2)

3. rotate – Rotate the element :-

Function	Description	Example
rotate(angle)	Rotate in 2D	rotate(45deg)
rotateX(angle)	Rotate around X axis (3D)	rotateX(60deg)
rotateY(angle)	Rotate around Y axis (3D)	rotateY(90deg)
rotateZ(angle)	Rotate around Z axis (3D)	rotateZ(30deg)
rotate3d(x, y, z, angle)	Rotate in 3D space	rotate3d(1, 1, 0, 45deg)

4. skew – Skew (slant) the element :-

Function	Description	Example
skew(x, y)	Skew in both X and Y	skew(20deg, 10deg)
skewX(x)	Skew only horizontally	skewX(25deg)
skewY(y)	Skew only vertically	skewY(15deg)

Animation in CSS :-

- Animation in CSS is a technique that allows you to gradually change the style or appearance of an HTML element over time without using JavaScript.
- CSS Animation is a way to smoothly transition between different CSS styles over a period of time using @keyframes and animation properties.

Animation Properties :-

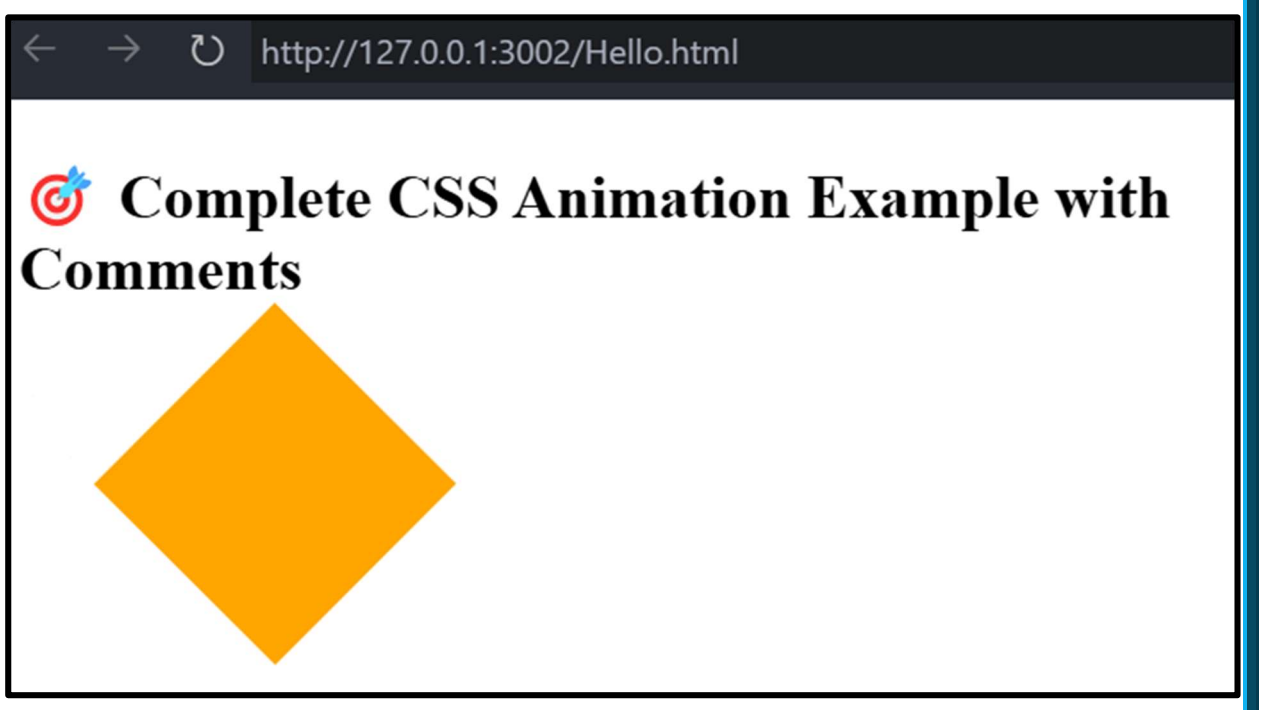
- The **animation-name** specify the name of the @keyframes animation to apply to the element.
- The **animation-duration** defines how long the animation takes to complete one full cycle, usually written in seconds (s) or milliseconds (ms).
- The **animation-timing-function** sets the speed curve of the animation, controlling how the animation progresses over time (e.g., linear or ease-in).
- The **animation-delay** determines the amount of time to wait before the animation starts after being applied to the element.
- The **animation-iteration-count** defines how many times the animation should repeat, with options like a specific number or infinite.
- The **animation-direction** controls whether the animation should run forward, backward, or alternate between directions in each cycle.
- The **animation-play-state** property allows you to control whether the animation is running or paused.
- The **animation-fill-mode** property defines how styles are applied to an element before the animation starts and after it ends.
- The **animation** shorthand property lets you write all animation-related properties in one line instead of writing each one separately.

Syntax Shorthand :- animation : name | duration | timing-function | delay | iteration-count | direction | fill-mode | play-state;

Example :-

```
<!DOCTYPE html>
<html>
<style>
/* Keyframes define what happens during animation */
@keyframes myAnimation {
  0%   { background-color : red; transform : translateX(0px) rotate(0deg); }
  25%  { background-color : orange; transform : translateX(50px) rotate(45deg); }
  50%  { background-color : yellow; transform : translateX(100px) rotate(90deg); }
  75%  { background-color : green; transform : translateX(50px) rotate(135deg); }
  100% { background-color : blue; transform : translateX(0px) rotate(180deg); }
}
.box {
  width : 100px;
  height : 100px;
  background-color : red;
  /* === Animation Properties === */
  animation-name : myAnimation;
  /* Value :- any name defined by @keyframes */

  animation-duration : 4s;
  /* Values :- time in s or ms (e.g., 2s, 500ms) */

  animation-timing-function : ease-in-out;
  /* Values :-
    linear          - same speed from start to end
    ease            - slow start, fast middle, slow end (default)
    ease-in         - slow start
    ease-out        - slow end
    ease-in-out     - slow start and end
  */

  animation-delay : 1s;
  /* Values :- time before animation starts (e.g., 0s, 3s) */

  animation-iteration-count : infinite;
  /* Values :-
    1 (default)
    any number (e.g., 2, 3.5)
    infinite - repeats forever
  */
```

Output :-

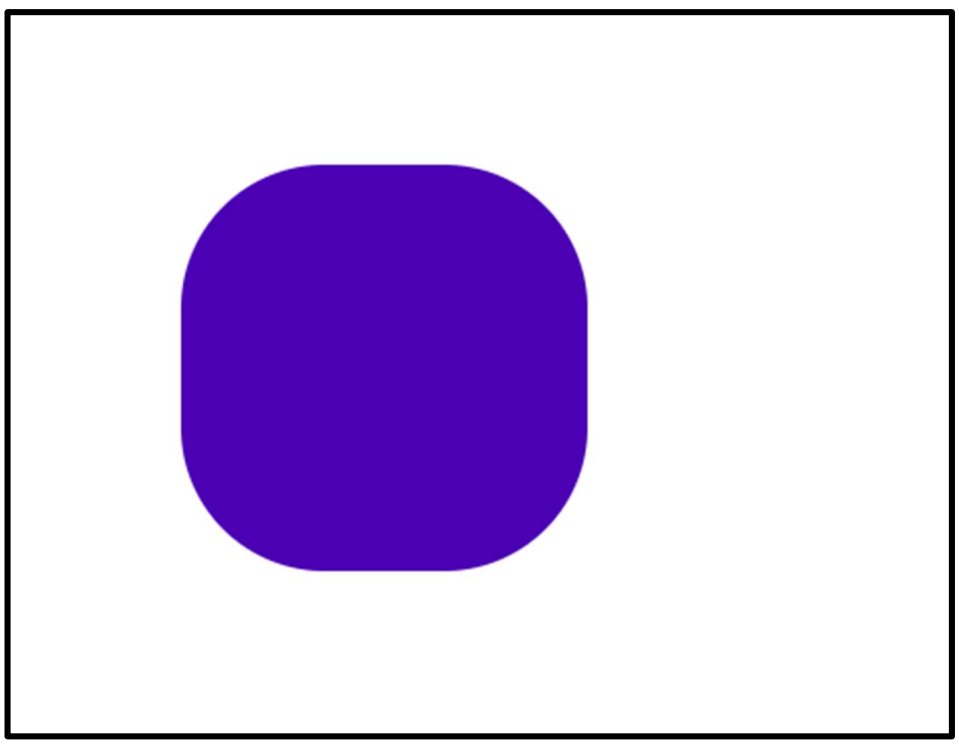


```
animation-direction : alternate;
/* Values :-
   normal     - default ( forward )
   reverse    - backward direction
   alternate  - forward then backward
   alternate-reverse - backward then forward
*/
animation-fill-mode : both;
/* Values :-
   none       - does not apply styles before/after animation
   forwards   - keeps final style after animation
   backwards  - applies first keyframe style during delay
   both       - applies both forwards and backwards
*/
 animation-play-state : running;
 /* Values :-
    running  - plays animation
    paused   - pauses animation
 */
}
</style>
<body>
<h2> 🎯 Complete CSS Animation Example with Comments</h2>
<div class="box"></div>
</body>
</html>
```

Example :-

```
<html> <style>
   /* Animation to turn a square into a circle and back */
   @keyframes circleEffect {
    from {
      width : 100px;
      height : 100px;
      border-radius : 0%;
      background-color : red;
    }
    to {
      width : 150px;
      height : 150px;
      border-radius : 50%;
      background-color : blue;
    }
}
```

Output :-



```
   .circle-box {
     margin : 50px;
     animation-name : circleEffect;           /* Animation name */
     animation-duration : 2s;                 /* Duration */
     animation-iteration-count : infinite;    /* Run forever */
     animation-direction : alternate;         /* Reverse each time */
     animation-timing-function : ease-in-out;  /* Smooth in and out */
   }
  </style>
<body>
<div class="circle-box"></div>
</body>
</html>
```

CSS Variables :-

- A CSS Variables allow you to store reusable values like colors, font sizes, spacing, etc., in one place, and use them throughout your styles.

Example :-

```
<!DOCTYPE html>
<html>
<head>
  <style>
    /* Define CSS Variables */
    :root {
      -- back : #811515;
      --text-color : #f0e8e8;
      --padding-size : 20px;
      --border-radius : 10px;
      --font-size : 18px;
      --text-align : center;
    }
    .box {
      background-color : var(--back);
      color : var(--text-color);
      padding : var(--padding-size);
      border-radius : var(--border-radius);
      font-size : var(--font-size);
      text-align : var(--text-align);
    }
  </style> <body>
  <div class="box"> <p>This is a box using CSS variables.</p> </div>
</body>
 </html>
```

Output :-

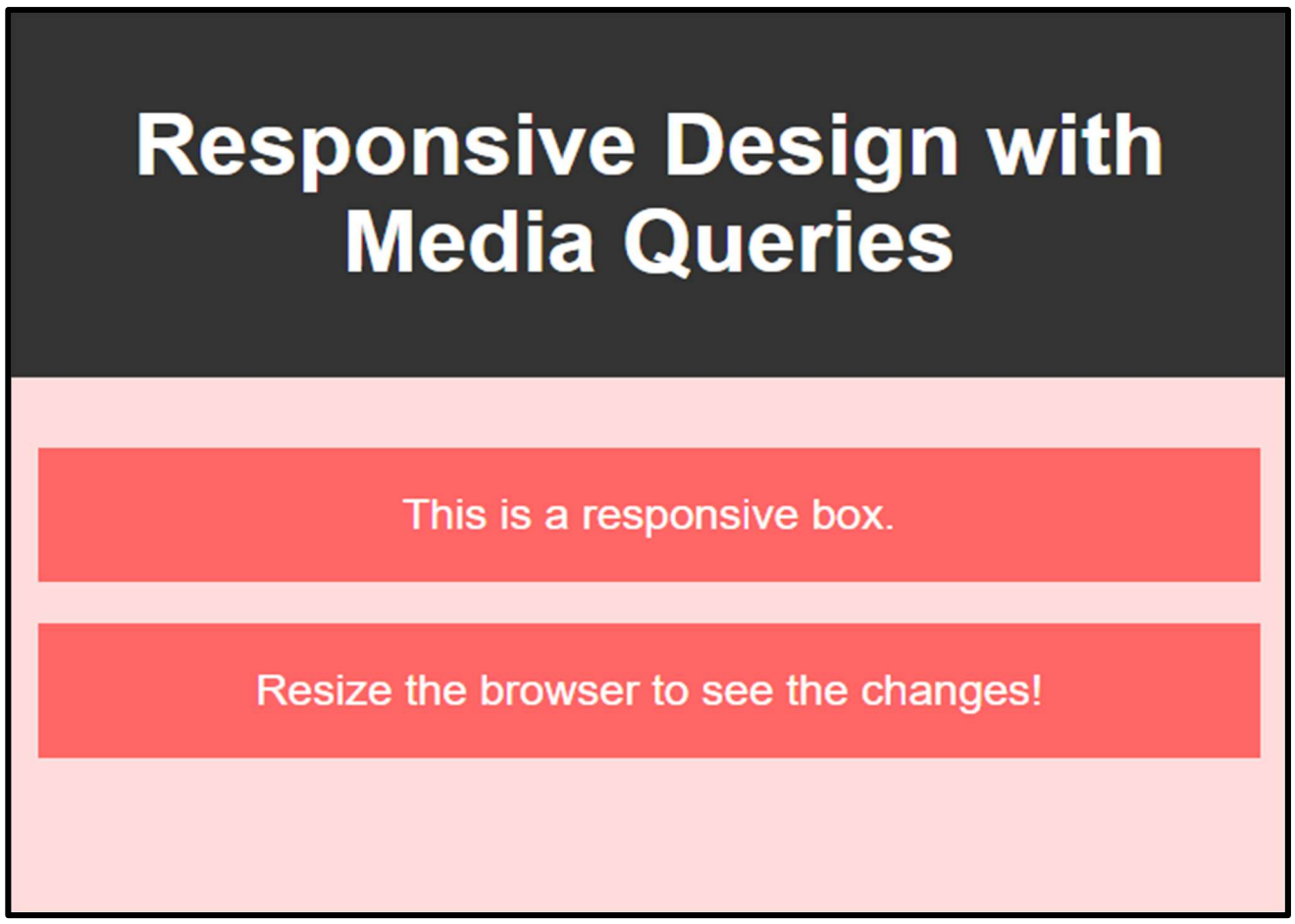
Media Queries :-

- A Media Queries allow you to apply CSS styles based on the device's screen size, resolution, orientation, etc.
- It helps in making your website responsive, meaning it looks good on mobile, tablet, and desktop.
- **Syntax :-** @media (condition) { /* CSS rules here */ }.

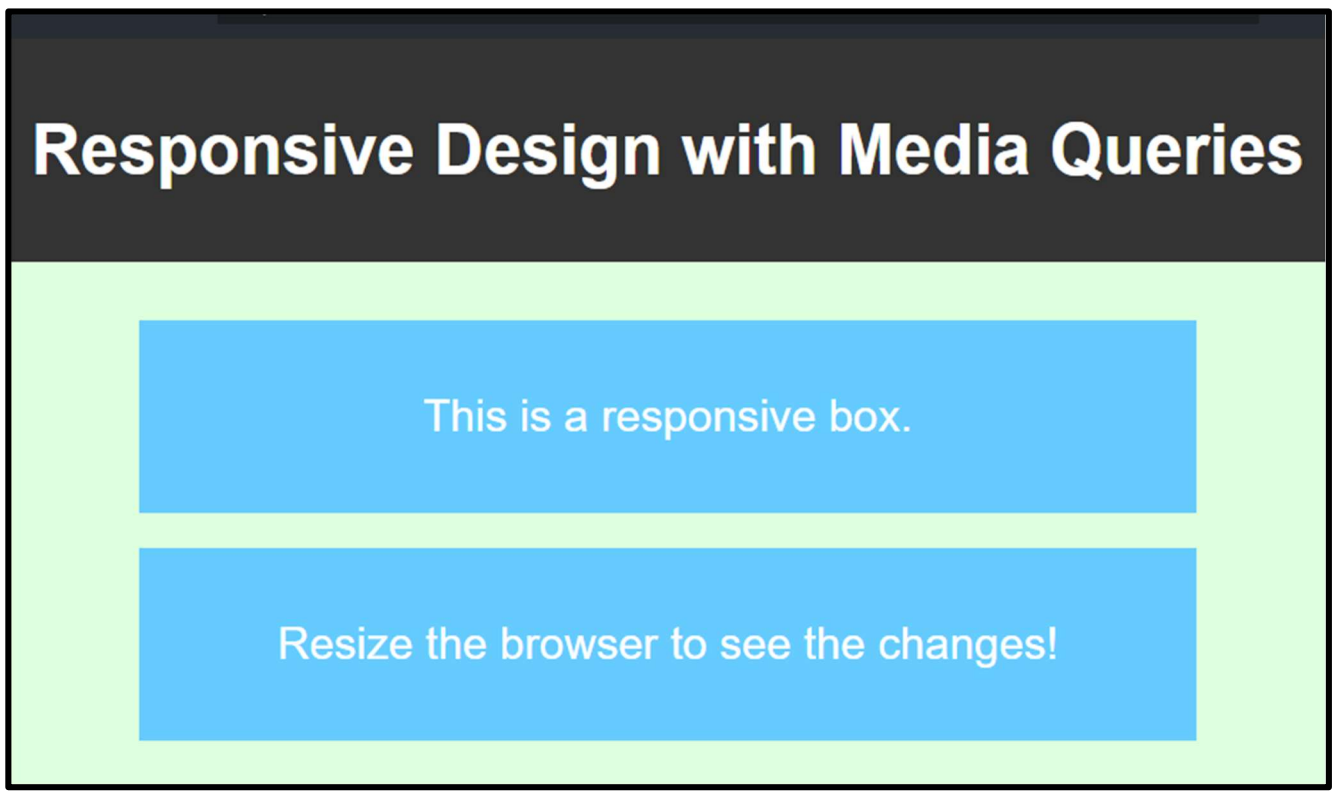
Example :-

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Media Query Example</title>
  <style>
    /* Base Styles (Common to all devices) */
    body {
      margin : 2vw;
      font-family : Arial, sans-serif;
      font-size : 1rem;
      background-color : #f2f2f2;
    }
    header {
      background-color : #333;
      color : white;
      text-align : center;
      padding : 2vh 0;
    }
    .container {
      width : 80%;
      margin : auto;
      padding : 2vh 0;
    }
    .box {
      background-color : #4CAF50;
      color : white;
      padding : 2rem;
      margin : 1rem 0;
      text-align : center;
      font-size : 1.2rem;
    }
    /* Mobile Styles */
    @media (max-width: 600px) {
      body {
        background-color : #ffdddd;
      }
```

Mobile Output :-

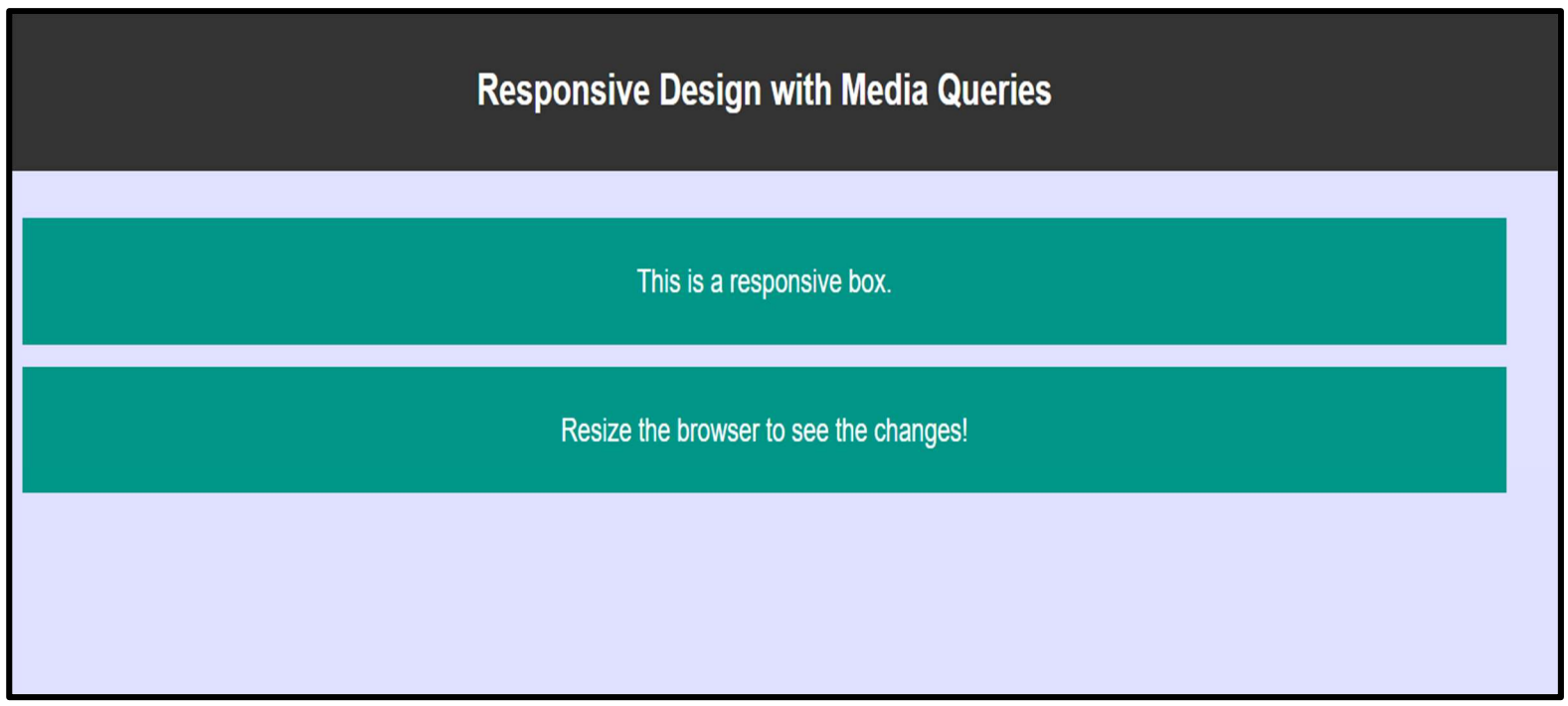


Tablet Output :-



```
  .container {
    width : 95%;
  }
  .box {
    font-size : 1rem;
    padding : 1rem;
    background-color : #ff6666;
  }
}
/* Tablet Styles */
@media (min-width: 601px) and (max-width: 1024px) {
  body {
    background-color : #ddffdd;
  }
  .box {
    background-color : #66ccff;
    font-size : 1.3rem;
  }
}
/* Desktop Styles */
@media (min-width: 1025px) {
  body {
    background-color : #e0e0ff;
  }
  .box {
    background-color : #009688;
    font-size : 1.5rem;
  }
}
 </style>
</head>
<body>
 <header>
  <h1>Responsive Design with Media Queries</h1>
 </header>
 <div class="container">
  <div class="box">This is a responsive box.</div>
  <div class="box">Resize the browser to see the changes!</div>
 </div>
</body>
</html>
```

Desktop Output :-



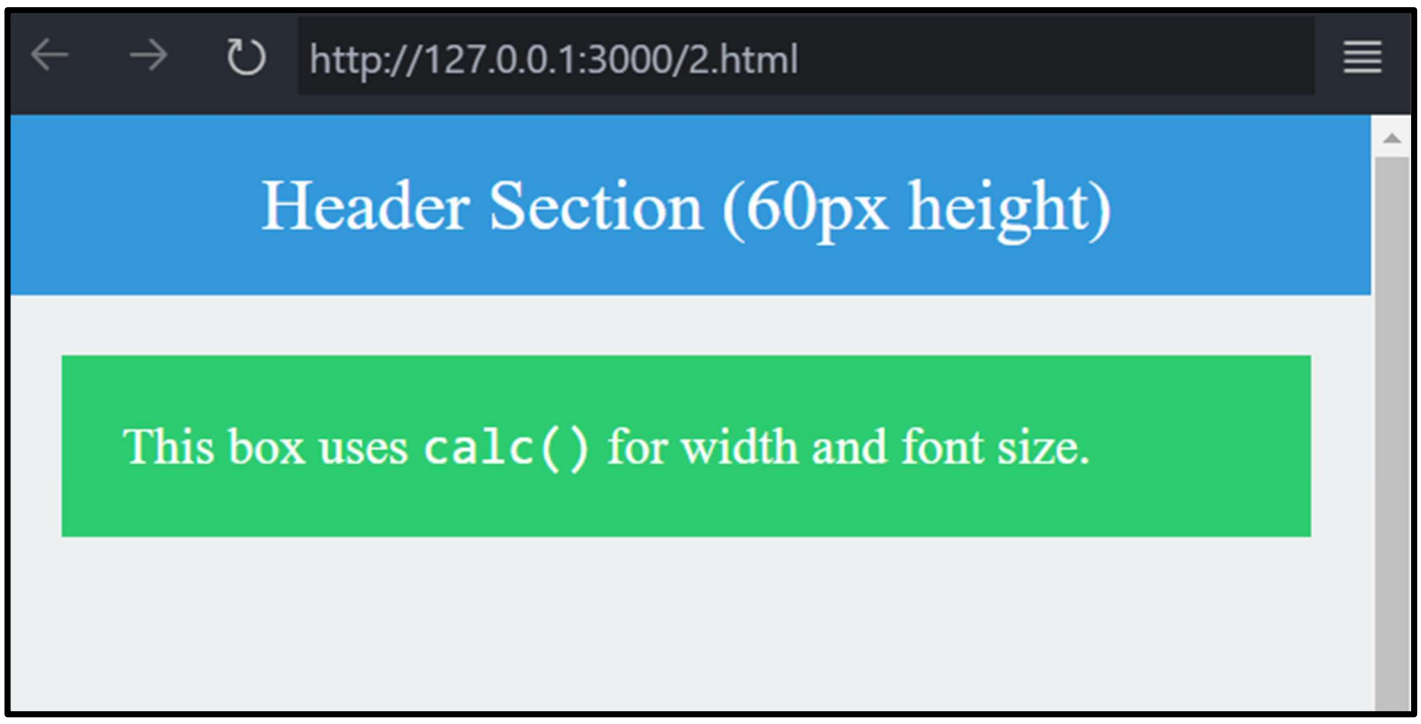
Calc( ) Function :-

- The calc( ) function in CSS allows you to perform mathematical calculations directly in your CSS code.
- It allows you to combine different units like %, px, em, etc.
- calc( ) is most commonly used with layout-related properties like width, height, margin, padding, top, and left.
- The syntax for calc( ) is property :- calc(expression);

Example :-

```
<!DOCTYPE html>
<html>
<head>
 <style>
   header {
     height : 60px;
     background-color : #3498db;
     color : white;
     text-align : center;
     line-height : 60px;
     font-size : 24px;
   }
  .content {
    height : calc(100vh - 60px); /* Full height minus header */
    background-color : #ecf0f1;
    padding : 20px;
  }
  .box {
    width : calc(100vw - 40px); /* Full width minus padding */
    background-color : #2ecc71;
    color : white;
    padding : 20px;
    font-size : calc(15px + 0.5vw); /* Responsive text size */
   }
 </style>
<body>
 <header> Header Section (60px height) </header>
 <div class="content">
   <div class="box"> This box uses <code>calc( )</code> for width and font size. </div>
 </div>
</body>
</html>
```

Output :-



Custom Fonts in CSS using @font-face :-

- The @font-face rule in CSS allows you to use custom fonts on your website, even if those fonts are not installed on the user's device.

Example :-

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Lobster Font with @font-face</title>
  <style>
    /* Load custom font using @font-face */
    @font-face {
      font-family : 'LobsterCustom';
      src : url('Lobster-Regular.woff2') format('woff2');
      font-weight : normal;
      font-style : normal;
    }
  h1 {
    font-family : 'LobsterCustom', cursive;
    font-size : 48px;
    color : #c0392b;
  }
   body {
    background : #fff8f0;
    padding : 40px;
   }
  p {
    font-size : 18px;
   font-family : Arial, sans-serif;
    color : #555;
  }
    }
  </style>
</head>
<body>
  <h1>Stylish Heading</h1>
  <p>This paragraph is using system font, but the heading uses Lobster font via @font-face.</p>
</body>
</html>
```

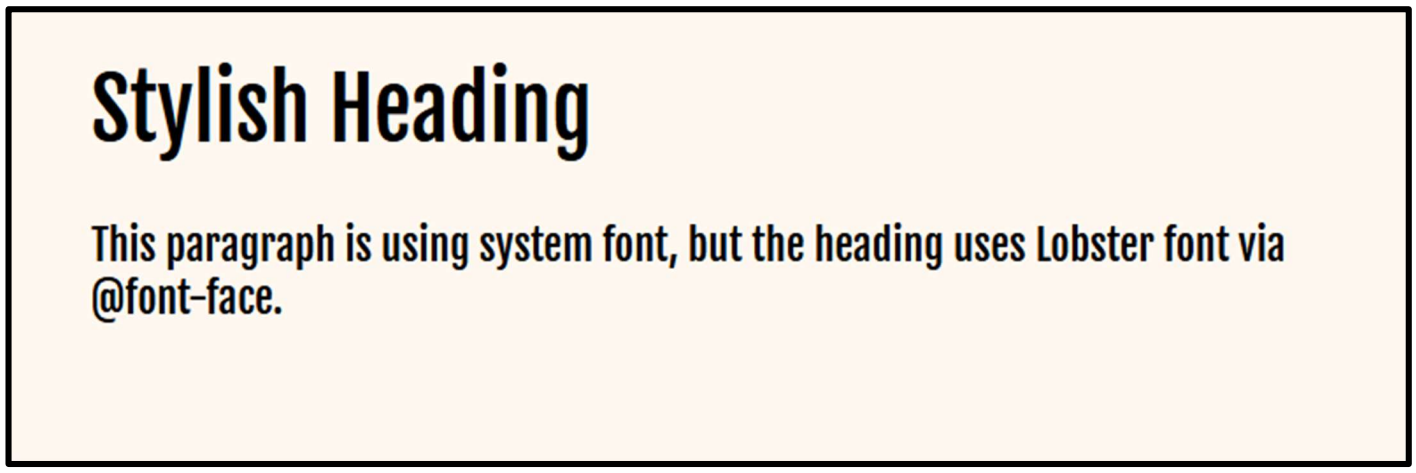
Output :-

Other Example using Google Font :-

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Google Font Example</title>
```

```
<link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link href="https://fonts.googleapis.com/css2?family=Fjalla+One&display=swap" rel="stylesheet">
<style>
  h1 {
  font-family : "Fjalla One", sans-serif;
  font-weight : 400;
  font-style : normal;
}
  body {
   background : #fff8f0;
    padding : 40px;
    }
  p {
  font-family : "Fjalla One", sans-serif;
  font-weight : 400;
  font-style : normal;
    }
</style>
</head>
<body>
  <h1>Stylish Heading</h1>
  <p>This paragraph is using system font, but the heading uses Lobster font via @font-face.</p>
</body>
</html>
```

Output :-



CSS Nesting :-

- The CSS Nesting is a way to write CSS rules inside other rules to reflect the HTML hierarchy and reduce repetition.

Example :-

```
<!DOCTYPE html>
<html>
  <style>
    nav {
      background-color : #333;
      padding : 10px;
      ul {
        list-style: none;
        padding : 0;
        margin : 0;
        display : flex;
        li {
          margin-right : 20px;
```

Output :-

```
        a {
          color : white;
          text-decoration : none;
          font-weight : bold;
          &:hover {
            color : yellow;
          }
        }
      }
    }
  }
 </style>
<body>
 <nav>
  <ul>
    <li><a href="#">Home</a></li>
    <li><a href="#">About</a></li>
    <li><a href="#">Contact</a></li>
  </ul>
 </nav>
</body>
</html>
```

---

accent-color Property :-

- The accent-color property in CSS is used to change the highlight or accent color of form controls and interactive elements like :

  Checkboxes

  Radio buttons

  Range sliders

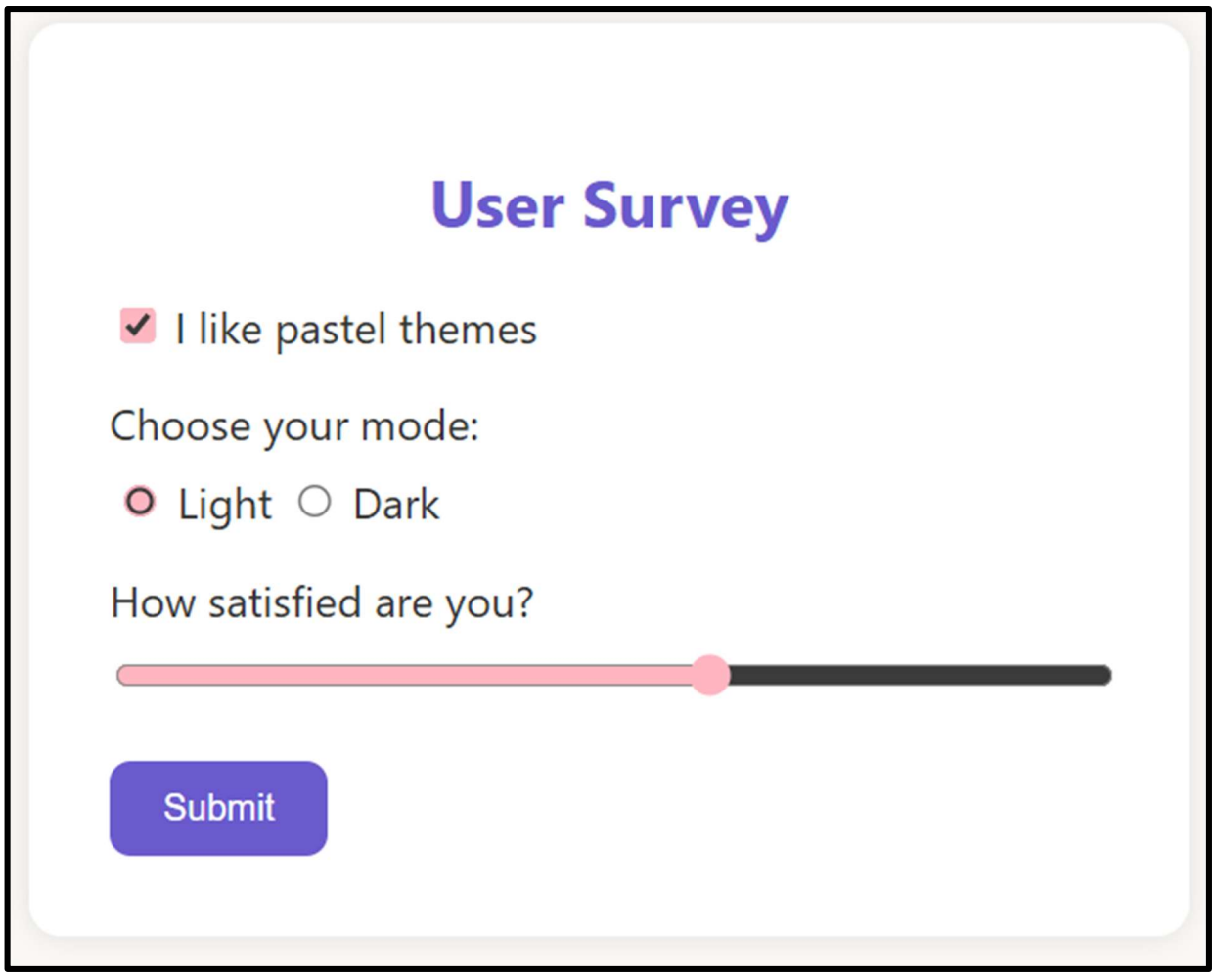
  Progress bars
- This allows you to style native HTML controls without removing their native functionality or accessibility.

Example :-

```
<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="UTF-8">
 <title>Pastel Accent Color Form</title>
 <style>
  body {
    background-color : #faf9f6;
    font-family : "Segoe UI", sans-serif;
    padding : 40px;
    color : #333;
  }
```

```
form {
  background : #fff;
  padding : 30px;
  border-radius : 12px;
  max-width : 500px;
  margin : auto;
  box-shadow : 0 0 12px rgba(0, 0, 0, 0.1);
}
h2 {
  text-align : center;
  color : #6a5acd;
}
label {
  display : block;
  margin : 15px 0 8px;
  font-size : 16px;
}
input[type="checkbox"],
input[type="radio"],
input[type="range"] {
  accent-color : #ffb6c1; /* Light Pink */
}
input[type="range"] {
  width : 100%;
}
button {
  margin-top : 20px;
  padding : 10px 20px;
  background : #6a5acd;
  color : white;
  border : none;
  border-radius : 8px;
  cursor : pointer;
 }
 button:hover {
  background : #5a4cc0;
 }
 </style>
</head>
<body>
 <form> <h2>User Survey</h2>
  <label><input type="checkbox" checked> I like pastel themes</label>
  <label>Choose your mode :</label>
  <input type="radio" name="theme" checked> Light
  <input type="radio" name="theme"> Dark
  <label>How satisfied are you?</label>
  <input type="range" min="0" max="100" value="60">
  <button type="submit">Submit</button>
 </form>
</body>
</html>
```

Output :-



Project 1 : How To Centre A Div Inside Div :-

Method 1 :- Flexbox (Center Horizontally & Vertically)

```
<style>
.outer {
  Display : flex;
  justify-content : center; /* horizontal center */
  align-items : center;     /* vertical center */
  height : 300px;
  background-color : lightgray;
}
.inner {
  width : 100px;
  height : 100px;
  background-color : steelblue;
}
</style>
<div class="outer">
  <div class="inner">Centered</div>
</div>
```

Method 2 :- Using Absolute Positioning

```
<style>
.outer {
  position : relative;
  height : 300px;
  background-color : lightgray;
}
.inner {
  position: absolute;
  top : 50%;
  left : 50%;
  transform : translate(-50%, -50%);
  width : 100px;
  height : 100px;
  background-color : steelblue;
}
</style>
<div class="outer">
  <div class="inner">Centered</div>
</div>
```

Output :-

Method 3 : Using CSS Grid

```
<style>
.outer {
  display : grid;
  place-items : center; /* centers both horizontally and vertically */
  height : 300px;
  background-color : lightgray;
}
.inner {
  width : 100px;
  height : 100px;
  background-color : steelblue;
}
</style>
<div class="outer">
  <div class="inner">Centered</div>
</div>
```

Project 2 : Neon Loader :-

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Document</title>
    <style>
      html {
        font-size : 62.5%;
      }
      body {
        height : 100vh;
        display : grid;
        place-items : center;
        background-color : #191825;
}
.loader {
  width : 20rem;
  aspect-ratio : 1;
  border-radius : 50%;
  background-color : #865dff;
  position : relative;
  animation : circleloader 2s linear infinite reverse;
}
@keyframes circleloader {
  100% {
    rotate : 360deg;
  }
}
.loader::before {
  content: "";
  position : absolute;
  top : 0;
  left : 0;
  width : 17rem;
  aspect-ratio : 1;
  border-radius : 50%;
  background-color : #191825;
}
```

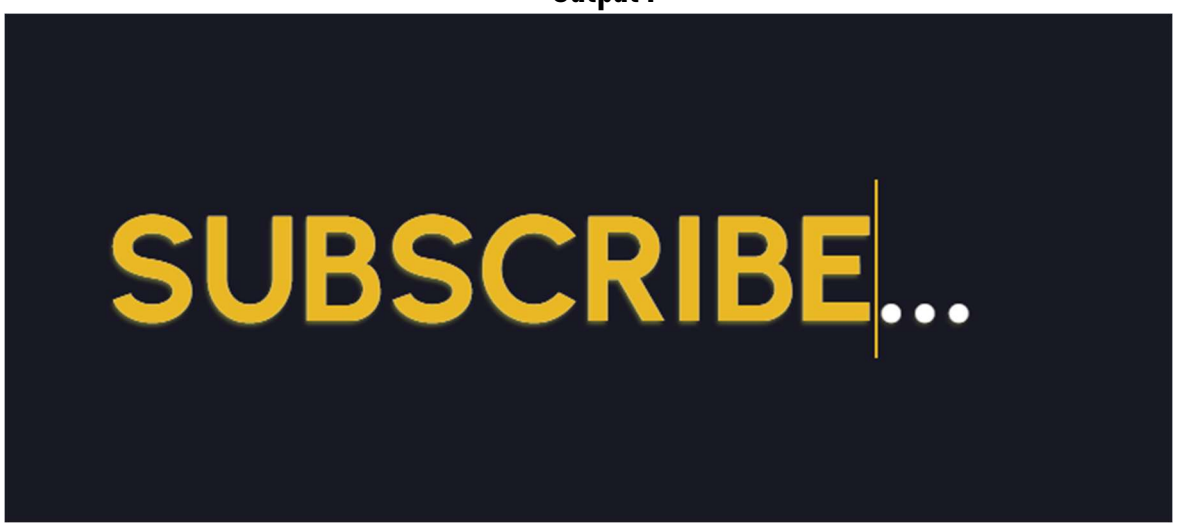
Output :-

```
.loader::after {
  content: "";
  position : absolute;
  top : -2.5rem;
  left : 8.5rem;
  width : 5rem;
  aspect-ratio : 1;
  border-radius : 50%;
  background-color : #865dff;
  box-shadow : 0 0 1rem #865dff, 0 0 1.5rem #865dff, 0 0 2rem #865dff, 0 0 2.5rem #865dff, 0 0 3rem #865dff;
}
</style>
</head>
<body>
  <div class="loader"></div>
</body>
</html>
```

Mini Project 3 : text reveal :-

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <link rel="preconnect" href="https://fonts.googleapis.com" />
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin />
    <link
href="https://fonts.googleapis.com/css2?family=Bungee+Spice&family=Jost:wght@300;400;500;700&family=Poppins:wght@200;300;400;600&family=Quicksand:wght@300;400;500;600;700&family=Urbanist:wght@300;400;500;600;700;800;900&display=swap" rel="stylesheet" />
    <style>
      * {
        margin : 0;
        padding : 0;
        box-sizing : border-box;
        font-family : "Urbanist";
        /* font-family : "Bungee Spice", cursive; */
      }
      body {
        height : 100vh;
        display : grid;
        place-items : center;
        background-color : #191825;
      }
      h1 {
        text-align : center;
        color : #fff;
        font-weight : bold;
        font-size : 96px;
        text-shadow : -1px 2px 3px rgba(236, 247, 76, 0.2);
        position : relative;
      }
      h1::before {
        content : "SUBSCRIBE...";
        position : absolute;
        top : 0;
        left : 0;
        width : 80%;
        color : #e9b824;
        text-transform : uppercase;
        border-right : 2px solid #e9b824;
        overflow : hidden;
        animation : textreveal 2s linear infinite;
      }
```

Output :-




```
      @keyframes textreveal {
        0% {
          width : 0;
        }
        50% {
          width : 100%;
        }
        100% {
          width : 0;
        }
      }
    </style>
  </head>
  <body>
    <h1>SUBSCRIBE...</h1>
  </body>
</html>
```

Vendor Prefixes in css :- (css-auto-prefix Extension)

- Vendor prefixes are special prefixes added before CSS properties to ensure compatibility with specific browsers that haven't fully adopted a CSS feature yet.

Common Vendor Prefixes :-

- WebKit uses the -webkit- prefix, which is supported by browsers like Chrome, Safari, and the newer versions of Microsoft Edge.
- Mozilla uses the -moz- prefix, which is supported by the Firefox browser.
- Microsoft uses the -ms- prefix, which is supported by Internet Explorer and older versions of Microsoft Edge.
- Opera uses the -o- prefix, which was supported by older versions of the Opera browser before it switched to the WebKit engine.

Example :-

```
<!DOCTYPE html>
<html>
<head>
  <style>
    .rotate-box {
      width : 150px;
      height : 150px;
      background-color : orange;

      /* Vendor Prefixes */
      -webkit-transform : rotate(5deg); /* Chrome, Safari */
      -moz-transform : rotate(5deg);    /* Firefox */
      -ms-transform : rotate(5deg);     /* Internet Explorer */
      -o-transform : rotate(5deg);      /* Old Opera */

      /* Standard property */
      Transform : rotate(5deg);
    }
  </style>
</head>
<body>
  <div class="rotate-box">Rotated Box</div>
</body>
</html>
```

---

1. Css Project Link :- https://cssprojectwebsite.netlify.app/

2. GitHub Project Link :- https://github.com/Vaghnai123/CssProject

---

Final Project :-

Style.css :-

```
@import
url("https://fonts.googleapis.com/css2?family=Jost:wght@300;400;500;700;900&family=Poppins:wght@200;300;400;600&family=Quicksand:wght@300;400;500;600;700&family=Urbanist:wght@300;400;500;600;700;800;900&display=swap");
/* base rule  */
* {
  margin : 0;
  padding : 0;
  box-sizing : border-box;
  font-family : "Urbanist", sans-serif;
}
html {
  font-size : 62.5%;
  /* 1rem = 10px  */
  /* scroll-behavior : smooth; */
  scroll-snap-type : Y proximity;
}
h1,h2,h3,h4,h5,h6 {  font-family : "Poppins", sans-serif;  }
p, li, a, label {
  font-family : "Urbanist", sans-serif;
  font-size : 1.8rem;
  letter-spacing : 0.1rem;
  font-weight : 400;
  color : var(--para-color);
  line-height : 1.5;
}
a {  text-decoration : none;  }
li {  list-style : none;  }
/* theme */
:root {
  --main-color : #0062ff;
  --supporting-color : #ebf3fe;
  --font-color : #424242;
  --bg-color : #f7fcff;
  --heading-color : #000a19;
  --hero-heading-color : #003b99;
  --white-color : #ffffff;
  --para-color : #504e4d;
  --bnt-hover-bg-color : #003b99;
  --btn-box-shadow : rgba(100, 100, 111, 0.2) 0px 7px 29px 0px;
  --footer-bg-color : #040d12;
}
```

```
/* layout */
section {
  scroll-snap-align : center;
}
.container {
  max-width : 142rem;
  margin : 0 auto;
  padding : 9.6rem  2.4rem;
}
:is(
    .section-about,
    .section-blog,
    .section-contact,
    .section-course,
    .section-why--choose,
    .section-contact--homepage
  )
  .container:first-child {
  padding : 6.4rem 2.4rem 2.4rem 2.4rem;
}
.grid {
  display : grid;
}
.grid-two--cols {
  grid-template-columns : repeat(2, 1fr);
}
.grid-three--cols {
  grid-template-columns : repeat(3, 1fr);
}
.grid-four--cols {
  grid-template-columns : repeat(4, 1fr);
}
/* module / reusable  */
.section-common-heading {
  font-size : 3.2rem;
  font-weight : 700;
  text-transform : capitalize;
}
.section-common-subheading {
  color : var(--heading-color);
}
```

```
.section-common--title {
  font-size : 2rem;
  margin : 2.4rem 0 1.2rem 0;
}
.fa-solid {
  Padding : 2.4rem;
  background-color : var(--supporting-color);
  font-size : 2.4rem;
  border-radius : 50%;
  -webkit-border-radius : 50%;
  -moz-border-radius : 50%;
  -ms-border-radius : 50%;
  -o-border-radius : 50%;
  color : var(--hero-heading-color);
}
.btn {
  display : inline-block;
  padding : 1.2rem 3.2rem;
  background-color : var(--main-color);
  color : var(--white-color);
  border-radius : 0.6rem;
  -webkit-border-radius : 0.6rem;
  -moz-border-radius : 0.6rem;
  -ms-border-radius : 0.6rem;
  -o-border-radius : 0.6rem;
}
/* states  */
.btn:hover {
  background-color : var(--bnt-hover-bg-color);
  cursor : pointer;
  box-shadow : var(--btn-box-shadow);
}
/***** navbar Section *****/
.section-navbar {
  Width : 100%;
  box-shadow : rgba(0, 0, 0, 0.05) 0px 6px 24px 0px, rgba(0, 0, 0, 0.08) 0px 0px 0px 1px;
}
.section-navbar .container {
  display : flex;
  justify-content : space-between;
  align-items : center;
  padding : 1.8rem 2.4rem;
}
```

```
.section-navbar .navbar ul {
  display : flex;
  gap : 3.2rem;
  & li a {
    color : var(--heading-color);
    text-transform : uppercase;
    font-weight : 600;
    font-size : 1.6rem;
    display : inline-block;
    position : relative;
    &::after {
      content : "";
      position : absolute;
      bottom : -0.3rem;
      left : 0;
      width : 0%;
      border-bottom : 0.2rem solid var(--main-color);
      transition : all 0.3s linear;
      -webkit-transition : all 0.3s linear;
      -moz-transition : all 0.3s linear;
      -ms-transition : all 0.3s linear;
      -o-transition : all 0.3s linear;
    }
  }
  & li a:hover:after {
    width : 100%;
  }
} /***** End navbar Section *****/

/*****  hero Section *****/
main {
  position : relative;
  background-image : linear-gradient( to top right, #3d86fa, #4484fb, #679eff, #b3d2ff, #ebf3fe );
}
.custom-shape-divider-bottom-1696162272 {
  position : absolute;
  bottom : 0;
  left : 0;
  width : 100%;
  overflow : hidden;
  line-height : 0;
  transform : rotate(180deg);
}
```

```
.custom-shape-divider-bottom-1696162272 svg {
  position : relative;
  display : block;
  width : calc(100% + 1.3px);
  height : 12rem;
}
.custom-shape-divider-bottom-1696162272 .shape-fill {
  fill : #ffffff;
}

.section-hero .grid {
  align-items : center;
  gap : 6.4rem;
  & .hero-subheading {
    text-transform : uppercase;
    letter-spacing : 0.2rem;
    font-size : 1.7rem;
    word-spacing : 0.2rem;
    color : var(--hero-heading-color);
    font-weight : 700;
  }
  & .hero-heading {
    font-size : 5.8rem;
    line-height : 1.5;
    font-family : "Jost";
    color : var(--hero-heading-color);
    font-weight : 900;
  }
  & .hero-para {
    color : var(--white-color);
    margin : 2rem 0 4.2rem 0;
  }
}

.section-hero--image img {
  width : 100%;
  height : auto;
  transform : scaleX(-1);
  -webkit-transform : scaleX(-1);
  -moz-transform : scaleX(-1);
  -ms-transform : scaleX(-1);
  -o-transform : scaleX(-1);
}  /***** End  hero Section *****/
```

```
/***** about Section *****/
.section-about .grid {
  gap : 6.4rem;
}
.section-about .about-div {
  text-align : center;
}
.section-about img {
  padding : 2.4rem;
  background-color : var(--supporting-color);
  width : 15rem;
  height : auto;
  border-radius : 50%;
  -webkit-border-radius : 50%;
  -moz-border-radius : 50%;
  -ms-border-radius : 50%;
  -o-border-radius : 50%;
  transition : all 0.3s linear;
  -webkit-transition : all 0.3s linear;
  -moz-transition : all 0.3s linear;
  -ms-transition : all 0.3s linear;
  -o-transition : all 0.3s linear;
}

.icon:hover > img {
  transform : rotate(360deg);
  -webkit-transform : rotate(360deg);
  -moz-transform : rotate(360deg);
  -ms-transform : rotate(360deg);
  -o-transform : rotate(360deg);
  /* rotate : 360deg; */
  background : linear-gradient(to right, #0575e6, #021b79);
}
/***** End about Section *****/

/***** course Section *****/
.section-course {
  background-color : var(--bg-color);
}
.section-course .grid {
  gap : 3.2rem;
}
```

```
.section-course .course-div {
  padding : 3.2rem;
}
.section-course .course-div : hover {
  box-shadow : var(--btn-box-shadow);
}
.course-div : nth-child(2) .icon .fa-solid {
  color : #68bf9b;
  background-color : #e7f6ef;
}
.course-div : nth-child(3) .icon .fa-solid {
  color : #ff8b52;
  background-color : #fbebe8;
}
.course-div : nth-child(4) .icon .fa-solid {
  color : #183d3d;
  background-color : #ccf7f7;
}
.course-div : nth-child(5) .icon .fa-solid {
  color : #d988b9;
  background-color : rgb(247, 223, 238);
}
.course-div : nth-child(6) .icon .fa-solid {
  color : #ff9b50;
  background-color : #f6dfce;
}
.course-div : nth-child(7) .icon .fa-solid {
  color : #1450a3;
  background-color : #dce9fa;
}
/***** End course Section *****/

/*****  why choose  Section *****/
.section-why--choose .grid {  gap : 9.6rem;  }
.why-choose--div {  margin-top : 3.2rem;  }
.section-why--choose .text-align--right .why-choose--div {
  display : flex;
  flex-direction : column;
  justify-content : start;
  align-items : end;
  text-align : right;
}
```

```
.common-text--highlight {
  width : 6rem;
  aspect-ratio : 1;
  background-color : var(--supporting-color);
  color : var(--main-color);
  display : flex;
  justify-content : center;
  align-items : center;
  font-size : 2.4rem;
  font-weight : 700;
  border-radius : 50%;
  -webkit-border-radius : 50%;
  -moz-border-radius : 50%;
  -ms-border-radius : 50%;
  -o-border-radius : 50%;
}
.choose-center--div img {
  width : 90%;
  height : auto;
}
.choose-center--div, figure {
  display : flex;
  justify-content : center;
  align-items : center;
  position : relative;
}
.choose-center--div figure:before,
.choose-center--div figure::after {
  content : " ";
  position : absolute;
  top : 50%;
  left : 50%;
  transform : translate(-50%, -50%);
  -webkit-transform : translate(-50%, -50%);
  -moz-transform : translate(-50%, -50%);
  -ms-transform : translate(-50%, -50%);
  -o-transform : translate(-50%, -50%);
  background-color : var(--main-color);
  width : 45rem;
  height : 45rem;
  border-radius : 50%;
  z-index : -1;
}
```

```
.choose-center--div figure::before {
  animation : circle 5s linear infinite;
  -webkit-animation : circle 5s linear infinite;
}
@keyframes circle {
  0% {
    background-color : #b3d0ff;
  }
  25% {
    background-color : #80b1ff;
  }
  50% {
    background-color : #4d91ff;
  }
  75% {
    background-color : #99c0ff;
  }
  100% {
    background-color : #3381ff;
  }
}

.choose-center--div figure::after {
  width : 50rem;
  height : 50rem;
  background-color : transparent;
  z-index : -2;
  border : 0.5rem solid var(--supporting-color);
} /***** End  why choose  Section *****/
/***** blog Section *****/
.section-blog {
  background-color : var(--bg-color);
  & .grid {  gap : 6.4rem;  }
  & .blog {
    box-shadow : var(--btn-box-shadow);
    transition : scale 0.3s linear;
    -webkit-transition: scale 0.3s linear;
    -moz-transition: scale 0.3s linear;
    -ms-transition: scale 0.3s linear;
    -o-transition: scale 0.3s linear;
    &:hover {  scale : 1.1;   }
    & .blog-content {  padding : 1.4rem 2.4rem 2.4rem;  }
  }
```

```
& img {
  width : 100%;
  height : auto;
}
& .blog-date {
  display : flex;
  justify-content : space-between;
  margin-top : 1.6rem;
  font-size : 1.4rem;
  & .fa-solid {
    background-color : transparent;
    padding : 0;
  }
}
& .section-common--title {
   margin-top: 0.8rem;
  }
} /***** End blog Section *****/

/*****  contact home Section *****/
.section-contact--homepage {
  width : 60%;
  min-height : 30rem;
  margin : 0 auto;
  box-shadow : rgba(60, 64, 67, 0.3) 0px 1px 2px 0px, rgba(60, 64, 67, 0.15) 0px 2px 6px 2px;
  border-radius : 0.2rem;
  -webkit-border-radius : 0.2rem;
  -moz-border-radius : 0.2rem;
  -ms-border-radius : 0.2rem;
  -o-border-radius : 0.2rem;
  padding : 0  3.2rem;
  position : relative;
  bottom : -15rem;
  z-index : 1;
  background-color : var(--white-color);
  & .grid {
    align-items : center;
    gap : 6.4rem;
  } }
.contact-title {
  font-size : 3.8rem;
  line-height : 1.2;
}
```

```
.contact-content p {
  margin : 1.2rem 0 2.4rem 0;
}
.section-contact--homepage a {
  color : var(--white-color);
  text-transform : capitalize;
}
.section-contact--homepage .fa-solid {
  padding : 0;
  color : var(--white-color);
  background-color : var(--main-color);
}
.section-contact--homepage img {
  width : 90%;
  height : auto;
}
/***** End  contact home Section *****/

/***** actual contact page Section *****/
.section-contact {
  & .grid {
    align-items : center;
    gap : 6.4rem;
  }
}
.mb-3 {
  margin-bottom : 3.2rem;
}
label {
  display : block;
  text-transform : capitalize;
}
.contact-content .grid {
  gap : 6.4rem;
}
input, textarea {
  width : 100%;
  padding : 1.4rem 2.4rem;
  font-size : 1.7rem;
  letter-spacing : 0.1rem;
}
```

```
::placeholder {
  font-size : 1.6rem;
  letter-spacing : 0.1rem;
}
.btn-submit {
  font-size : 1.8rem;
  border : none;
  text-transform : capitalize;
}
input:focus-visible, textarea:focus-visible {
  outline : 0.1rem solid var(--bnt-hover-bg-color);
}
/***** End actual contact page Section *****/

/***** footer Section *****/
footer {
  background-color : var(--footer-bg-color);
  padding-top : 15rem;
}
footer * {
  color : var(--white-color);
}
footer .grid {
  text-align : right;
}
.footer-subheading {
  font-size : 2.2rem;
  font-weight : 700;
}
.footer-1--div p {
  margin : 1rem 0 3.2rem 0;
}

.footer-1--div {
  text-align : left;
  & .social-footer--icons {
    display : flex;
    gap : 2.4rem;
    & .fa-brands {
      width : 5rem;
      aspect-ratio : 1;
      background-color : var(--supporting-color);
      color : var(--bnt-hover-bg-color);
```

```
      border-radius : 50%;
      -webkit-border-radius : 50%;
      -moz-border-radius : 50%;
      -ms-border-radius : 50%;
      -o-border-radius : 50%;
      /*display : grid;
      place-items : center;
      */
      display : flex;
      justify-content : center;
      align-items : center;
      transition : all 0.3s linear;
      -webkit-transition : all 0.3s linear;
      -moz-transition : all 0.3s linear;
      -ms-transition : all 0.3s linear;
      -o-transition : all 0.3s linear;
      &:hover {
        color : var(--supporting-color);
        background-color : var(--bnt-hover-bg-color);
        rotate : 360deg;
      }
    }
  }
}
/***** End footer Section *****/

/***** scrollbar Section *****/
/* width */
::-webkit-scrollbar {
  width : 1rem;
}

/* Track */
::-webkit-scrollbar-track {
  background : var(--supporting-color);
}

/* Handle */
::-webkit-scrollbar-thumb {
  Background : var(--main-color);
  border-radius : 50px;
}
```

```
/* Handle on hover */
::-webkit-scrollbar-thumb:hover {
  background : var(--heading-color);
}   /***** End scrollbar Section *****/

/***** media queries Section *****/
/* @media (max-width: 1380px) { */
@media (width <= 1400px) {
  html {
    font-size : 56.25%;
    /* 1rem = 10px  */
  }
  .section-hero img {
    width : 90%;
  }
}

@media (width <= 1220px) {
  html {
    font-size : 54%;
    /* 1rem = 9px  */
  }
  :is(
      .section-about,
      .section-blog,
      .section-course,
      .section-contact--homepage,
      .section-why--choose
    )
    .grid {  gap : 6.4rem;  }
}

@media (width <= 1100px) {
  :is(.section-course, .section-blog) .grid-four--cols {
    grid-template-columns : repeat(auto-fit, minmax(250px, 1fr));
  }
  .section-why--choose {
    & .choose-left--div {
      order : 2;
    }
    & .choose-right--div {
      order : 3;
    }
```

```
  & .choose-center--div {
    order : 1;
    & figure::before {
      width : 0;
      height : 0;
      background-color : transparent;
      padding : 0;
    }
    & figure::after {
      width : 0;
      height : 0;
      background-color : transparent;
      padding : 0;
    }
  }
}

@media (width <= 998px) {
  .section-hero {
    height : auto;
    padding-bottom : 5rem;
    & .grid-two--cols {
      grid-template-columns : 1fr;
      & .section-hero--content {
        order : 2;
      }
      & .section-hero--image {
        order : 1;
        & img {
          width : 50%;
        }
      }
    }
  }
}
.section-about .about-div {
  text-align : left;
}
footer .grid-four--cols {
  grid-template-columns : repeat(3, minmax(100px, 1fr));
  text-align : left;
  & .footer-1--div {
    grid-row : 2 / 3;
```

```
      grid-column : 1 / -1;
      margin-top : 6.4rem;
    }
  }
}

@media (width <= 768px) {
  .grid-two--cols,
  .grid-three--cols {
    grid-template-columns : 1fr;
  }
  .section-navbar .container {
    display : flex;
    flex-direction : column;
  & .navbar ul {
    gap : 2.4rem;
  }
  & .navbar-brand {
    text-align : center;
    margin-bottom : 2.4rem;
  }
}
.section-hero .grid .hero-heading {
  font-size: 3.8rem;
}
.section-why--choose {
  & img { width: 30%;  }
  & .grid {  gap: 1.4rem;  }
  & .text-align--right .why-choose--div {
    align-items : start;
    text-align : left;
  }
  }
  .section-contact--homepage .btn {
    display : block;
    text-align : center;
  }
  .contact-title {  font-size : 2.4rem;  }
  :is(.section-contact--homepage) .container:first-child {
    padding : 6.4rem 0rem 2.4rem 0rem;
  }
}
/***** End media queries Section *****/
```

Index.html :-

```
<!DOCTYPE html>
<html lang="en">
   <link  rel="stylesheet"  href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0/css/all.min.css"
     integrity="sha512-9usAa10IRO0HhonpyAIVpjrylPvoDwiPUiKdWk5t3PyolY1cOd4DSE0Ga+ri4AuTroPR5aQvXU9xC6qOPnzFeg=="
     crossorigin="anonymous"   referrerpolicy="no-referrer" />
   <link rel="stylesheet" href="css/style.css" />
 </head>
 <bod>
   <!-- ========== Start NAVBAR Section ========== -->
   <header class="section-navbar">
     <div class="container">
       <div class="navbar-brand">
         <a href="index.html"> <img src="./images/logo.png"   alt="my thapa eduhub logo"   width="80%"   height="auto" /> </a>
       </div>
     <nav class="navbar">
       <ul>
         <li class="nav-item"> <a class="nav-link" href="index.html"> home </a> </li>
         <li class="nav-item"> <a class="nav-link" href="about.html"> about </a> </li>
         <li class="nav-item"> <a class="nav-link" href="service.html"> services </a> </li>
         <li class="nav-item"> <a class="nav-link" href="blog.html"> blog </a> </li>
         <li class="nav-item"> <a class="nav-link" href="contact.html"> contact </a> </li>
       </ul>
     </nav>
   </div>
</header>
<!-- ========== End NAVBAR Section ========== -->
<!-- ========== Start HERO  Section ========== -->
<main>
  <div class="section-hero">
<div class="section-hero">
  <div class="container grid grid-two--cols">
    <div class="section-hero--content">
      <p class="hero-subheading">EMPOWERING LIFELONG LEARNING</p>
      <h1 class="hero-heading"> Unlock Your Potential with Thapa EduHub </h1>
      <p class="hero-para"> Welcome to EduHub, your gateway to knowledge and skill development. We are dedicated to providing high-quality,
        accessible education for learners of all ages and backgrounds </p>
      <div class="hero-btn"> <a href="services.html" class="btn btn-white">Learn with us</a> </div>
    </div>
    <div class="section-hero--image">
      <figure>
        <img   src="./images/main-hero1.webp"   alt="a 3d character doing coding stuff " />
      </figure>
    </div>  </div>  </div>
```

```
    <div class="custom-shape-divider-bottom-1696162272">
      <svg  data-name="Layer 1"  xmlns="http://www.w3.org/2000/svg"  viewBox="0 0 1200 120"  preserveAspectRatio="none" >
        <path
          d="M0,0V46.29c47.79,22.2,103.59,32.17,158,28,70.36-5.37,136.33-33.31,206.8-
37.5C438.64,32.43,512.34,53.67,583,72.05c69.27,18,138.3,24.88,209.4,13.08,36.15-6,69.85-17.84,104.45-29.34C989.49,25,1113-14.29,1200,52.47V0Z"
          opacity=".25"
          class="shape-fill"></path>
        <path
          d="M0,0V15.81C13,36.92,27.64,56.86,47.69,72.05,99.41,111.27,165,111,224.58,91.58c31.15-10.15,60.09-26.07,89.67-39.8,40.92-19,84.73-46,130.83-
49.67,36.26-2.85,70.9,9.42,98.6,31.56,31.77,25.39,62.32,62,103.63,73,40.44,10.79,81.35-6.69,119.13-24.28s75.16-39,116.92-43.05c59.73-
5.85,113.28,22.88,168.9,38.84,30.2,8.66,59,6.17,87.09-7.5,22.43-10.89,48-26.93,60.65-49.24V0Z"
          opacity=".5"
          class="shape-fill"></path>
        <path
          d="M0,0V5.63C149.93,59,314.09,71.32,475.83,42.57c43-7.64,84.23-20.12,127.61-26.46,59-
8.63,112.48,12.24,165.56,35.4C827.93,77.22,886,95.24,951.2,90c86.53-7,172.46-45.71,248.8-84.81V0Z"
          class="shape-fill"></path>
      </svg>
    </div>
  </main>
  <!-- ========== End HERO  Section ========== -->


  <!-- ========== Start about Section ========== -->
  <section class="section-about">
    <div class="container">
      <h2 class="section-common-heading">About EduHub</h2>
      <p class="section-common-subheading"> Guiding Your Learning Journey through Research, Design, and  Development Excellence. </p>  </div>


    <div class="container grid grid-three--cols">
      <div class="about-div"> <div class="icon"> <img src="images/research.png" alt="thapa eduhub research team" /> </div>
        <h3 class="section-common--title">Research</h3>
        <p> We start by learning what you need. We look at what's new and exciting in learning and choose topics that match what you want to learn. </p>
    </div>

    <div class="about-div">  <div class="icon"> <img src="images/design.png" alt="thapa eduhub research team" /> </div>
      <h3 class="section-common--title">Design</h3>
      <p> Next, we make the lessons. We make them fun and interesting for you. We make sure they work for different ways people like to learn. </p>
    </div>

    <div class="about-div"> <div class="icon"> <img src="images/Coding.png" alt="thapa eduhub research team" /> </div>
      <h3 class="section-common--title">Development</h3>
      <p> After that, we turn the lessons into things you can use online. We use the latest tools to make sure they work well and make you happy. </p>
    </div>  </div>  </section>
  <!-- ========== End about Section ========== -->
```

```
<!-- ========== Start courses Section ========== -->
<section class="section-course">
  <div class="container">
    <h2 class="section-common-heading">Explore Our Courses</h2>
    <p class="section-common-subheading"> Discover a variety of courses across different categories. </p>
  </div>

  <div class="container grid grid-four--cols">
    <div class="course-div">
      <div class="icon">
        <i class="fa-solid fa-code"></i>
      </div>
      <h3 class="section-common--title">JS Development</h3>
      <p> Creating dynamic, interactive web applications for user engagement. </p>
    </div>

<div class="course-div">
  <div class="icon"><i class="fa-solid fa-laptop-code"></i></div>
  <h3 class="section-common--title">Python</h3>
  <p> Versatile language for problem-solving and application development. </p>
</div>

<div class="course-div">
  <div class="icon"><i class="fa-solid fa-laptop-code"></i></div>
  <h3 class="section-common--title">c++ Development</h3>
  <p>Powerful, efficient coding language for software development.</p>
</div>

<div class="course-div">
  <div class="icon"><i class="fa-solid fa-laptop-code"></i></div>
  <h3 class="section-common--title">Java Development</h3>
  <p>Cross-platform application development with Java programming.</p>
</div>

<div class="course-div">
  <div class="icon"><i class="fa-solid fa-laptop-code"></i></div>
  <h3 class="section-common--title">SQL Development</h3> <p>Effective data management and retrieval using SQL database language. </p>
</div>

<div class="course-div">
  <div class="icon"><i class="fa-solid fa-laptop-code"></i></div>
  <h3 class="section-common--title">HTML Development</h3>   <p>Web content structure with essential markup language.</p>
</div>
```

```
    <div class="course-div">
      <div class="icon"><i class="fa-solid fa-laptop-code"></i></div>
      <h3 class="section-common--title">CSS Development</h3>
      <p>Styling web elements for visually appealing websites.</p>
    </div>
  </div>
</section>
<!-- ========== End courses Section ========== -->

<!-- ========== Start why choose Section ========== -->
<section class="section-why--choose">
  <div class="container">
    <h2 class="section-common-heading">Why Choose Thapa EduHub</h2>
  <p class="section-common-subheading">
    Choose Thapa EduHub for a holistic, enriching learning experience that empowers you to achieve your goals.
  </p>
</div>
<div class="container grid grid-three--cols">

    <div class="choose-left--div text-align--right">
      <div class="why-choose--div">
        <p class="common-text--highlight">1</p>
        <h3 class="section-common--title">Expert Instructors</h3>
       <p> Learn from experienced teachers who are passionate about sharing knowledge and guiding you toward success in your learning journey. </p>
      </div>

      <div  class="why-choose--div"   data-aos="zoom-in-up"   data-aos-delay="300">
      <p class="common-text--highlight">2</p>
      <h3 class="section-common--title">Interactive Lessons</h3>
        <p>
          Engage in hands-on learning experiences that make education fun
          and memorable. Interactive lessons encourage active participation
          and deeper understanding.
        </p>
       </div>

       <div  class="why-choose--div"    data-aos="zoom-in-up"   data-aos-delay="600">
        <p class="common-text--highlight">3</p>
        <h3 class="section-common--title">Lifelong Learning Support</h3>
    <p> Our commitment to your education doesn't end with a certificate. Receive ongoing support and resources for continuous learning and growth. </p>
       </div>
</div>
```

```
<!--? center part of image  -->
<div class="choose-center--div">
  <figure>
    <img src="./images/mobile.png" alt="thapa technical offical new website ">
  </figure>
</div>

<!--? right side the content  -->
<div class="choose-right--div text-align--left">

  <div class="why-choose--div">
    <p class="common-text--highlight">4</p>
    <h3 class="section-common--title">Expert Instructors</h3>
    <p> Learn from experienced teachers who are passionate about sharing knowledge and guiding you toward success in your learning journey. </p>
  </div>

    <div   class="why-choose--div text-align--left"  data-aos="zoom-in-up"   data-aos-delay="300">
    <p class="common-text--highlight">5</p>
    <h3 class="section-common--title">Lifelong Learning Support</h3>
<p> Our commitment to your education doesn't end with a certificate. Receive ongoing support and resources for continuous learning and growth. </p>
  </div>

  <div  class="why-choose--div text-align--left"   data-aos="zoom-in-up"  data-aos-delay="600">
    <p class="common-text--highlight">6</p>
    <h3 class="section-common--title">Lifelong Learning Support</h3>
    <p>
      Our commitment to your education doesn't end with a certificate.
      Receive ongoing support and resources for continuous learning and
      growth.
    </p>
  </div>

    </div>
  </div>
  </section>
<!-- ========== End why choose Section ========== -->

  <!-- ========== Start contact home Section ========== -->
<div class="section-contact--homepage" id="section-contact--homepage">
  <div class="container grid grid-two--cols">
    <div class="contact-content">
      <h2 class="contact-title">Let's revolutionize the way you study</h2>
```

```
      <p>Join our free bootcamps to know us better</p>
      <div class="btn"> <a href="contact.html"> start now <i class="fa-solid fa-arrow-circle-right"></i> </a> </div>
    </div>
    <div class="contact-image"> <img src="images/main.png" alt="a guy doing coding"> </div>
  </div>
</div>
<!-- ========== End contact home Section ========== -->
<!-- ========== Start  FOOTER Section ========== -->
<footer>
  <div class="container grid grid-four--cols">

    <div class="footer-1--div">
      <div class="logo-brand">
        <a href="index.html" class="footer-subheading">EduHUb</a>
      </div>
      <p>Let's revolutionize the way you study with EduHub</p>
    <div class="social-footer--icons">
      <a href="https://discord.gg/fDuFEXfyyH" target="_blank" alt="my discord server link" >  <i class="fa-brands fa-discord"></i> </a>
        <a href="https://www.youtube.com/thapatechnical" target="_blank" alt="my youtube channel link" > <i class="fa-brands fa-youtube"></i> </a>
        <a href="https://www.instagram.com/thapatechnical/" target="_blank" alt="my instagram profile link" > <i class="fa-brands fa-instagram"></i> </a>
    </div>
  </div>

  <div class="footer-2--div">
    <p class="footer-subheading">  Courses </p>
      <ul>
        <li> <a href="index.html"> CSS </a> </li>
        <li> <a href="index.html"> CSS3 </a> </li>
        <li> <a href="index.html"> CSS </a> </li>
        <li> <a href="index.html"> CSS </a> </li>
        <li> <a href="index.html"> CSS </a> </li>
        <li> <a href="index.html"> CSS </a> </li>
      </ul>
  </div>
  <div class="footer-3--div">
    <p class="footer-subheading"> Links  </p>
      <ul>
        <li> <a href="index.html"> home </a> </li>
        <li> <a href="index.html"> home </a> </li>
        <li> <a href="index.html"> home </a> </li>
        <li> <a href="index.html"> home </a> </li>
        <li> <a href="index.html"> home </a> </li>
      </ul>
  </div>
```

```
    <div class="footer-4--div">
      <p class="footer-subheading"> Blogs </p>
        <ul>
          <li> <a href="index.html"> HTML </a> </li>
          <li> <a href="index.html"> HTML </a> </li>
          <li> <a href="index.html"> HTML </a> </li>
          <li> <a href="index.html"> HTML </a> </li>
          <li> <a href="index.html"> HTML </a> </li>
        </ul>
    </div>

    </div>
  </footer>
  <!-- ========== End FOOTER Section ========== -->
 </body>
</html>
```

About.html :-

```
<!DOCTYPE html>
<html lang="en">
 <head>
   <link   rel="stylesheet"   href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0/css/all.min.css"
     integrity="sha512-9usAa10IRO0HhonpyAIVpjrylPvoDwiPUiKdWk5t3PyolY1cOd4DSE0Ga+ri4AuTroPR5aQvXU9xC6qOPnzFeg=="
     crossorigin="anonymous"   referrerpolicy="no-referrer" />
   <link rel="stylesheet" href="css/style.css" />
 </head>
 <bod>
   <!-- ========== Start NAVBAR Section ========== -->
   <header class="section-navbar">
     <div class="container">
       <div class="navbar-brand">
<div class="navbar-brand">
  <a href="index.html"> <img  src="./images/logo.png"  alt="my thapa eduhub logo"   width="80%"    height="auto" /> </a>
</div>
<nav class="navbar">
  <ul>
    <li class="nav-item">
      <a class="nav-link" href="index.html"> home </a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="about.html"> about </a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="service.html"> services </a>
    </li>
```

```
        <li class="nav-item">
          <a class="nav-link" href="blog.html"> blog </a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="contact.html"> contact </a>
        </li>
      </ul>
    </nav>
  </div>
</header>
<!-- ========== End NAVBAR Section ========== -->

<!-- ========== Start about Section ========== -->
<section class="section-about">
  <div class="container">
    <h2 class="section-common-heading">About Thapa EduHub</h2>
    <p class="section-common-subheading"> Guiding Your Learning Journey through Research, Design, and Development Excellence. </p>
</div>

<div class="container grid grid-three--cols">
    <div class="about-div">
      <div class="icon">
        <img src="images/research.png" alt="thapa eduhub research team" />
      </div>
      <h3 class="section-common--title">Research</h3>
      <p> We start by learning what you need. We look at what's new and exciting in learning and choose topics that match what you want to learn. </p>
    </div>

    <div class="about-div">
      <div class="icon"> <img src="images/design.png" alt="thapa eduhub research team" /> </div>
      <h3 class="section-common--title">Design</h3>
      <p> Next, we make the lessons. We make them fun and interesting for you. We make sure they work for different ways people like to learn. </p>
    </div>

    <div class="about-div">
      <div class="icon"> <img src="images/Coding.png" alt="thapa eduhub research team" /> </div>
      <h3 class="section-common--title">Development</h3>
      <p> After that, we turn the lessons into things you can use online. We use the latest tools to make sure they work well and make you happy. </p>
    </div>
  </div>
</section>
<!-- ========== End about Section ========== -->
```

```
<!-- ========== Start contact home Section ========== -->
<div class="section-contact--homepage" id="section-contact--homepage">
  <div class="container grid grid-two--cols">
    <div class="contact-content">
      <h2 class="contact-title">Let's revolutionize the way you study</h2>
      <p>Join our free bootcamps to know us better</p>
      <div class="btn">
        <a href="contact.html"> start now <i class="fa-solid fa-arrow-circle-right"></i> </a>
      </div>
    </div>
    <div class="contact-image">
      <img src="images/main.png" alt="a guy doing coding" />
    </div>
  </div>
</div>
<!-- ========== End contact home Section ========== -->

<!-- ========== Start   FOOTER Section ========== -->
<footer>
  <div class="container grid grid-four--cols">
    <div class="footer-1--div">
      <div class="logo-brand"> <a href="index.html" class="footer-subheading">EduHUb</a> </div>
      <p>Let's revolutionize the way you study with Thapa EduHub</p>

      <div class="social-footer--icons">
        <a href="https://discord.gg/fDuFEXfyyH" target="_blank"> <i class="fa-brands fa-discord"></i>  </a>
        <a href="https://www.youtube.com/thapatechnical" target="_blank"> <i class="fa-brands fa-youtube"></i>  </a>
        <a href="https://www.instagram.com/thapatechnical/" target="_blank"> <i class="fa-brands fa-instagram"></i>  </a>
      </div>
    </div>
  </div>

  <div class="footer-2--div">
    <p class="footer-subheading">Courses</p>
    <ul>
      <li><a href="index.html"> CSS </a></li>
      <li><a href="index.html"> CSS3 </a></li>
      <li><a href="index.html"> CSS </a></li>
      <li><a href="index.html"> CSS </a></li>
      <li><a href="index.html"> CSS </a></li>
    </ul>
  </div>
```

```
<div class="footer-3--div">
  <p class="footer-subheading">Links</p>
  <ul>
    <li><a href="index.html"> home </a></li>
    <li><a href="index.html"> home </a></li>
    <li><a href="index.html"> home </a></li>
    <li><a href="index.html"> home </a></li>
    <li><a href="index.html"> home </a></li>
  </ul>
</div>

<div class="footer-4--div">
  <p class="footer-subheading">Blogs</p>
  <ul>
    <li><a href="index.html"> HTML </a></li>
    <li><a href="index.html"> HTML </a></li>
    <li><a href="index.html"> HTML </a></li>
    <li><a href="index.html"> HTML </a></li>
    <li><a href="index.html"> HTML </a></li>
  </ul>
 </div>
</div>
</div>
</footer>
<!-- ========== End FOOTER Section ========== -->
</body>
</html>
```

Service.html

```
<!DOCTYPE html>
<html lang="en">
 <head>
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Services EDUHUB</title>
  <link  rel="stylesheet"  href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0/css/all.min.css"
    integrity="sha512-9usAa10IRO0HhonpyAIVpjrylPvoDwiPUiKdWk5t3PyolY1cOd4DSE0Ga+ri4AuTroPR5aQvXU9xC6qOPnzFeg=="
    crossorigin="anonymous"
    referrerpolicy="no-referrer" />

  <link rel="stylesheet" href="css/style.css" />
</head>
<bod>
```

```
<!-- ========== Start NAVBAR Section ========== -->
<header class="section-navbar">
  <div class="container">
    <div class="navbar-brand">
      <a href="index.html">
        <img  src="./images/logo.png"  alt="my thapa eduhub logo"  width="80%"   height="auto" />
      </a>
    </div>
    <nav class="navbar">
      <ul>
        <li class="nav-item">
          <a class="nav-link" href="index.html"> home </a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="about.html"> about </a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="service.html"> services </a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="blog.html"> blog </a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="contact.html"> contact </a>
        </li>
      </ul>
    </nav>
  </div>
</header>
<!-- ========== End NAVBAR Section ========== -->


<!-- ========== Start courses Section ========== -->
<section class="section-course">
  <div class="container">
    <h2 class="section-common-heading">Explore Our Courses</h2>
    <p class="section-common-subheading"> Discover a variety of courses across different categories. </p>
  </div>
  <div class="container grid grid-four--cols">
    <div class="course-div">
      <div class="icon">  <i class="fa-solid fa-code"></i> </div>
      <h3 class="section-common--title">JS Development</h3>
      <p> Creating dynamic, interactive web applications for user engagement. </p>
    </div>
```

```
<div class="course-div">
  <div class="icon"><i class="fa-solid fa-laptop-code"></i></div>
  <h3 class="section-common--title">Python</h3>
  <p> Versatile language for problem-solving and application development. </p>
</div>

<div class="course-div">
  <div class="icon"><i class="fa-solid fa-laptop-code"></i></div>
  <h3 class="section-common--title">c++ Development</h3>
  <p>Powerful, efficient coding language for software development.</p>
</div>

<div class="course-div">
  <div class="icon"><i class="fa-solid fa-laptop-code"></i></div>
  <h3 class="section-common--title">Java Development</h3>
  <p>Cross-platform application development with Java programming.</p>
</div>

<div class="course-div">
  <div class="icon"><i class="fa-solid fa-laptop-code"></i></div>
  <h3 class="section-common--title">SQL Development</h3>
  <p> Effective data management and retrieval using SQL database language. </p>
</div>

<div class="course-div">
  <div class="icon"><i class="fa-solid fa-laptop-code"></i></div>
  <h3 class="section-common--title">HTML Development</h3> <p>Web content structure with essential markup language.</p>
</div>

<div class="course-div">
    <div class="course-div">
      <div class="icon"><i class="fa-solid fa-laptop-code"></i></div>
      <h3 class="section-common--title">CSS Development</h3> <p>Styling web elements for visually appealing websites.</p>
    </div>
  </div>
</section>
<!-- ========== End courses Section ========== -->

<!-- ========== Start contact home Section ========== -->
<div class="section-contact--homepage" id="section-contact--homepage">
  <div class="container grid grid-two--cols">
    <div class="contact-content">
      <h2 class="contact-title">Let's revolutionize the way you study</h2>
      <p>Join our free bootcamps to know us better</p>
```

```
      <div class="btn">
        <a href="contact.html">
          start now <i class="fa-solid fa-arrow-circle-right"></i>
        </a>
      </div>
    </div>
    <div class="contact-image">
      <img src="images/main.png" alt="a guy doing coding" />
    </div>
  </div>
</div>
<!-- ========== End contact home Section ========== -->

<!-- ========== Start   FOOTER Section ========== -->
<footer>
  <div class="container grid grid-four--cols">
    <div class="footer-1--div">
      <div class="logo-brand">
        <a href="index.html" class="footer-subheading">Thapa EduHUb</a>
      </div>
      <p>Let's revolutionize the way you study with Thapa EduHub</p>

      <div class="social-footer--icons">
        <a href="https://discord.gg/fDuFEXfyyH" target="_blank">
          <i class="fa-brands fa-discord"></i>
        </a>
        <a href="https://www.youtube.com/thapatechnical" target="_blank">
          <i class="fa-brands fa-youtube"></i>
        </a>
        <a href="https://www.instagram.com/thapatechnical/" target="_blank">
          <i class="fa-brands fa-instagram"></i>
        </a>
      </div>
    </div>
  <div class="footer-2--div">
    <p class="footer-subheading">Courses</p>
    <ul>
      <li><a href="index.html"> CSS </a></li>
      <li><a href="index.html"> CSS3 </a></li>
      <li><a href="index.html"> CSS </a></li>
      <li><a href="index.html"> CSS </a></li>
      <li><a href="index.html"> CSS </a></li>
    </ul> </div>
```

```
<div class="footer-3--div">
  <p class="footer-subheading">Links</p>
  <ul>
    <li><a href="index.html"> home </a></li>
    <li><a href="index.html"> home </a></li>
    <li><a href="index.html"> home </a></li>
    <li><a href="index.html"> home </a></li>
    <li><a href="index.html"> home </a></li>
  </ul>
</div>

<div class="footer-4--div">
  <p class="footer-subheading">Blogs</p>
  <ul>
    <li><a href="index.html"> HTML </a></li>
    <li><a href="index.html"> HTML </a></li>
    <li><a href="index.html"> HTML </a></li>
    <li><a href="index.html"> HTML </a></li>
    <li><a href="index.html"> HTML </a></li>
  </ul>
 </div>
</div>
</footer>
<!-- ========== End   FOOTER Section ========== -->
</bod>
</html>
```

Contact.html :-

```
<!DOCTYPE html>
<html lang="en">
 <head>
<head>
 <link   rel="stylesheet"  href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0/css/all.min.css"
  integrity="sha512-9usAa10IRO0HhonpyAIVpjrylPvoDwiPUiKdWk5t3PyolY1cOd4DSE0Ga+ri4AuTroPR5aQvXU9xC6qOPnzFeg=="
  crossorigin="anonymous"
  referrerpolicy="no-referrer" />

 <link rel="stylesheet" href="css/style.css" />
 <style>
  footer {
    padding-top : 0;
  }
 </style>
</head>
<body>
```

```
<!-- ========== Start NAVBAR Section ========== -->
<header class="section-navbar">
  <div class="container">
    <div class="navbar-brand">
      <a href="index.html"> <img  src="./images/logo.png"   alt="my thapa eduhub logo"   width="80%"    height="auto" />  </a>
    </div>
    <nav class="navbar">
      <ul>
        <li class="nav-item">
          <a class="nav-link" href="index.html"> home </a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="about.html"> about </a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="service.html"> services </a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="blog.html"> blog </a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="contact.html"> contact </a>
        </li>
      </ul>
    </nav>
  </div>
</header>
<!-- ========== End NAVBAR Section ========== -->


<!-- ========== Start contact Section ========== -->
<div class="section-contact">
  <div class="container">
    <h2 class="section-common-heading">Contact Us</h2>
    <p class="section-common-subheading">
      Get in touch with us. We are always here to help you.
    </p>
  </div>

  <div class="container grid grid-two--cols">
    <div class="contact-content">
      <form action="https://formspree.io/f/xyyqbjej" method="POST">
     <div class="grid grid-two--cols mb-3">
```

```
        <div>
          <label for="username">username</label>
          <input  type="text"   name="username"   id="username"  required   autocomplete="off"   placeholder="enter your name" />
        </div>
        <div>
          <label for="email">enter your email</label>
          <input  type="email"   name="email"   id="email"   autocomplete="off"   required  placeholder="abc@thapa.com" />
        </div>
      </div>

      <div class="mb-3">
        <label for="subject">subject</label>
        <input   type="text"     name="subject"     id="subject"     placeholder="your main title" />
      </div>

      <div class="mb-3">
        <label for="message">message</label>
        <textarea name="message"  id="message"   cols="30"   rows="10"  placeholder="enter your message" ></textarea>
      </div>

      <div>
        <button type="submit" class="btn btn-submit">send message</button>
      </div>
    </form>
  </div>

  <div class="contact-map">
 <iframe
src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d121059.04711144073!2d73.7805582336797!3d18.524598599672156!2m3!1f0!2f0!3f0!3m2!1i1024!2i768!4f13.1!3m3!1m2!1s0x3bc2bf2e67461101%3A0x828d43bf9d9ee343!2sPune%2C%20Maharashtra!5e0!3m2!1sen!2sin!4v1696220815737!5m2!1sen!2sin"
        width="100%"
        height="500px"
        style="border : 0"
        allowfullscreen=" "
        loading="lazy"
        referrerpolicy="no-referrer-when-downgrade"
      ></iframe>
    </div>
  </div>
</div>
<!-- ========== End contact Section ========== -->
```

blog.html :-

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>THAPA EDUHUB</title>
    <link  rel="stylesheet"     href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0/css/all.min.css"
      integrity="sha512-9usAa10IRO0HhonpyAIVpjrylPvoDwiPUiKdWk5t3PyolY1cOd4DSE0Ga+ri4AuTroPR5aQvXU9xC6qOPnzFeg=="
      crossorigin="anonymous"  referrerpolicy="no-referrer" />

    <link rel="stylesheet" href="css/style.css" />
  </head>
  <body>
     <!-- ========== Start NAVBAR Section ========== -->
    <header class="section-navbar">
     <div class="container">
      <div class="navbar-brand">
        <a href="index.html"> <img  src="./images/logo.png"   alt="my thapa eduhub logo"   width="80%"    height="auto" />  </a>
      </div>
      <nav class="navbar">
        <ul>
          <li class="nav-item">
            <a class="nav-link" href="index.html"> home </a>
          </li>
          <li class="nav-item">
            <a class="nav-link" href="about.html"> about </a>
          </li>
          <li class="nav-item">
            <a class="nav-link" href="service.html"> services </a>
          </li>
          <li class="nav-item">
            <li class="nav-item">
              <a class="nav-link" href="blog.html"> blog </a>
            </li>
            <li class="nav-item">
              <a class="nav-link" href="contact.html"> contact </a>
            </li>
         </ul>
       </nav>
     </div>
    </header>
    <!-- ========== End NAVBAR Section ========== -->
```

```
<!-- ========== Start blog Section ========== -->
<div class="section-blog">
  <div class="container">
    <h2 class="section-common-heading">Explore Our Blog</h2>
    <p class="section-common-subheading"> Explore our blog for insightful articles, tips, and updates on the world of education, skill development, and
     personal growth. </p>
  </div>

  <div class="container grid grid-four--cols">
    <div class="blog">
      <figure>
        <img  src="https://images.pexels.com/photos/1181373/pexels-photo-1181373.jpeg?auto=compress&cs=tinysrgb&w=1260&h=750&dpr=1"
          alt="a guy writing code" />
      </figure>
    <div class="blog-content">
      <div class="blog-date">
        <p><i class="fa-calendar fa-solid"></i> Sept 22 2023</p>
        <p><i class="fa-solid fa-arrow-right"></i></p>
      </div>
      <h3 class="section-common--title"> Code your first react app with us </h3>
    </div>
  </div>

  <div class="blog">
    <figure>
      <img  src="https://images.pexels.com/photos/11035396/pexels-photo-11035396.jpeg?auto=compress&cs=tinysrgb&w=1260&h=750&dpr=1"  alt=" " />
    </figure>
    <div class="blog-content">
      <div class="blog-date"> <p><i class="fa-calendar fa-solid"></i> Sept 22 2023</p>  <p><i class="fa-solid fa-arrow-right"></i></p> </div>
      <h3 class="section-common--title"> Code your first react app with us </h3>
    </div>
  </div>

  <div class="blog">
    <figure>
      <img src="https://images.pexels.com/photos/4389463/pexels-photo-4389463.jpeg?auto=compress&cs=tinysrgb&w=1260&h=750&dpr=1"  alt=" " />
    </figure>
    <div class="blog-content">
      <div class="blog-date"> <p><i class="fa-calendar fa-solid"></i> Sept 22 2023</p>  <p><i class="fa-solid fa-arrow-right"></i></p> </div>
      <h3 class="section-common--title"> Code your first react app with us </h3>
    </div>
  </div>
```

```
    <div class="blog">
      <figure>
        <img  src="https://images.pexels.com/photos/3861961/pexels-photo-3861961.jpeg?auto=compress&cs=tinysrgb&w=1260&h=750&dpr=1"  alt=" " />
      </figure>
      <div class="blog-content">
        <div class="blog-date">
          <p><i class="fa-calendar fa-solid"></i> Sept 22 2023</p>
          <p><i class="fa-solid fa-arrow-right"></i></p>
        </div>
        <h3 class="section-common--title">
          Code your first react app with us
        </h3>
      </div>
    </div>
  </div>
<!-- ========== End blog Section ========== -->

<!-- ========== Start contact home Section ========== -->
<div class="section-contact--homepage" id="section-contact--homepage">
  <div class="container grid grid-two--cols">
    <div class="contact-content">
      <h2 class="contact-title">Let's revolutionize the way you study</h2>
      <p>Join our free bootcamps to know us better</p>
      <div class="btn">
        <a href="contact.html"> start now <i class="fa-solid fa-arrow-circle-right"></i> </a>
      </div>
    </div>
    <div class="contact-image">
      <img src="images/main.png" alt="a guy doing coding" />
    </div>
  </div>
</div>
<!-- ========== End contact home Section ========== -->
```

Output :-

THAPA EDUHUB

HOME ABOUT SERVICES BLOG CONTACT

EMPOWERING LIFELONG LEARNING

Unlock Your Potential with Thapa EduHub

Welcome to EduHub, your gateway to knowledge and skill development. We are dedicated to providing high-quality, accessible education for learners of all ages and backgrounds

Learn with us

THAPA EDUHUB

HOME ABOUT SERVICES BLOG CONTACT

About Thapa EduHub

Guiding Your Learning Journey through Research, Design, and Development Excellence.

Research

We start by learning what you need. We look at what's new and exciting in learning and choose topics that match what you want to learn.

Design

Next, we make the lessons. We make them fun and interesting for you. We make sure they work for different ways people like to learn.

Development

After that, we turn the lessons into things you can use online. We use the latest tools to make sure they work well and make you happy.

Explore Our Courses

Discover a variety of courses across different categories.

JS Development

Creating dynamic, interactive web applications for user engagement.

Python

Versatile language for problem-solving and application development.

c++ Development

Powerful, efficient coding language for software development.

Java Development

Cross-platform application development with Java programming.

SQL Development

Effective data management and retrieval using SQL database language.

HTML Development

Web content structure with essential markup language.

CSS Development

Styling web elements for visually appealing websites.