CHAPTER # 2

CSS3

- **Introduction to CSS3**
- 2.1. Architecture of CSS
- 2.2. CSS Modules
- 2.3. CSS Framework
- 2.4. Selectors and Pseudo Classes
- 2.5. Fonts and Text Effects
- 2.6. Colors, Background Images, and Masks

What is CSS?

- CSS stands for Cascading Style Sheets. It is the language for describing the presentation of Web pages, including colours, layout, and fonts, thus making our web pages presentable to the users.
- CSS is designed to make style sheets for the web.
- CSS is not a programming language.

Methods for Adding CSS

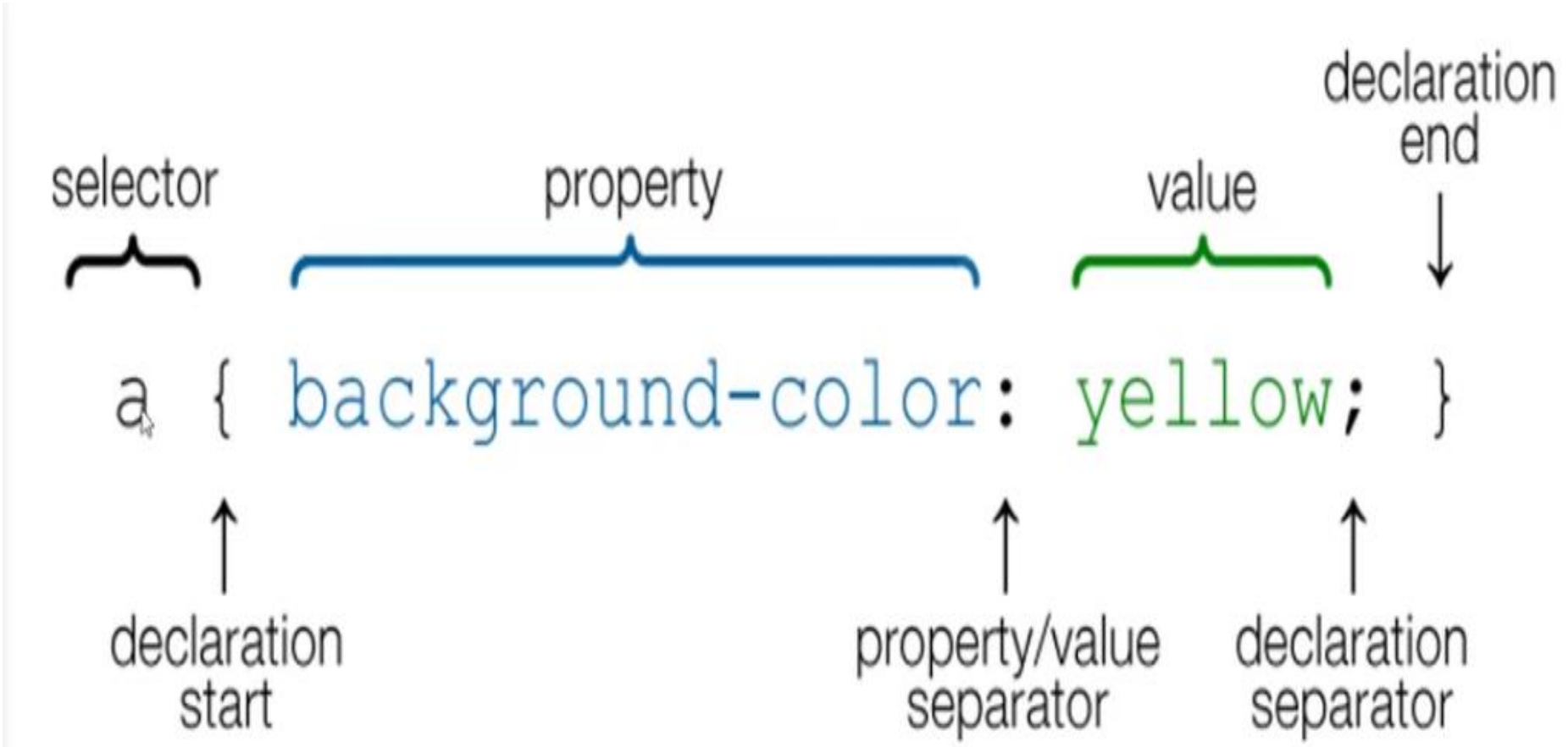
- **1. Inline CSS (Direct in HTML element)**
- Example:
- <h3 style="color:red;" > Have a great day </h3>
- Hello World!
- **2.Internal CSS (Using style tag within a single document)**
- <head>
- < style>
- h1{
- color:red;
- }
- </style>
- <head>
- <body>
- <h1> Have a great day </h1>
- </body>

- **3. External CSS (Linking an external CSS file).**
- **style.css**
- h1{
- color:red;
- }

- **Demo.html**
- <html>
- <head>
- <link rel="stylesheet" type="text/css" href="css/style.css">
- </head>
- <body>
- <h1> Have a great day </h1>
- </body>
- </html>

Selectors

- The selector is used to target elements and apply CSS
- Types of Selector
  - CSS Element Selector
  - CSS Id Selector
  - CSS Class Selector
  - CSS Universal Selector
  - CSS Group Selector
- **Priority of Selectors**
- Id > Class > Element


selector
property
value
declaration end
a { background-color: yellow; }
declaration start
property/value separator
declaration separator

- **1. The CSS element Selector**
- The element selector selects HTML elements based on the element name.
- **Example:**
- **style.css**
- body{
- background-color: aqua;
- color:brown;
- font-family:Arial, Helvetica, sans-serif;
- font-size: 16px;
- font-weight:normal;
- /*Same as above
- font:normal 16px Arial, Helvetica, sans-serif; */
- }

- demo.html
- <html>
- <head>
- <link rel="stylesheet" type="text/css" href="style.css">
- </head>
- <body>
- <h1> Have a great day </h1>
- </body>
- </html>
-

- **2. The CSS class selector**
- It is used with a period character **.** (full stop symbol) followed by the class name.
- Example:
- **style.css**
- .wrapper{
- background-color: blue;
- }
- **demo.html**
- <html>
- <head>
- <link rel="stylesheet" type="text/css" href="style.css">
- </head>
- <body>
-
- <h1> Have a great day </h1>
-
- </body>
- </html>

- **CSS Class Selector for specific element**
- **<html>**
- **<head>**
- **<style>**
- p.center {
- text-align: center;
- color: blue;
- }
- **</style>**
- **</head>**
- **<body>**
- **<h1** class="center"**>**This heading is not affected**</h1>**
- **<p** class="center"**>**This paragraph is blue and center-aligned.**</p>**
- **</body>**
- **</html>**

- **3. CSS Id Selector**
- It is used to select an element with given id.
- It is written with the hash character (#), followed by the id of the element.
- **Example:**
- #first{
- color:white;
- background: black;
- }
- We can add multiple classes to an element.
-
-

- **4) CSS Universal Selector**
- The universal selector is used as a wildcard character. It selects all the elements on the pages.
- **<html>**
- **<head>**
- **<style>**
- * {
- color: green;
- font-size: 20px;
- }
- **</style>**
- **</head>**
- **<body>**
- **<h2>**This is heading**</h2>**
- ** **This style will be applied on every paragraph.** **
- **<p** id="p1"**>**Me too!** **
- ** **And me!** **
- **</body>**
- **</html>**

- **5) CSS Group Selector**
- The grouping selector is used to select all the elements with the same style definitions.
- **Example:**
- **<html>**
- **<head>**
- **<style>**
- h1, h2, p {
- text-align: center;
- color: blue;
- }
- **</style>**
- **</head>**
- **<body>**
- **<h1>**Hello Javatpoint.com**</h1>**
- **<h2>**Hello Javatpoint.com (In smaller font)**</h2>**
- ** **This is a paragraph.** **
- **</body>**
- **</html>**

- **CSS Background**
- **background-color:** background-color: red;
- **Background:** rgba(red, green, blue, *alpha*).
- Background(0,255,0,0.5)
- **background-image**: background-image: url("paper1.gif");
- **background-repeat:** By default, the background-image property repeats the background image horizontally and vertically. Some images are repeated only horizontally or vertically.
  - repeat-x;
  - repeat-y;
  - no-repeat;

- **background-attachment:**
- The background-attachment property is used to specify if the background image is **fixed or scroll** with the rest of the page in browser window. If you set fixed the background image then the image will not move during scrolling in the browser.
- **Eg. background-attachment: fixed;**
- **background-position:**
- define the initial position of the background image. By default, the background image is placed on the top-left of the webpage.
- You can set the following positions:
- center,top,bottom,left,right
- **Eg. background-position: center;**

- **CSS Borders**
- The CSS border properties are given below
- border-style
- border-color
- border-width
- border-radius

- **CSS border-style:**
- eg. border-style: solid;

Value	Description
none	It doesn't define any border.
dotted	It is used to define a dotted border.
dashed	It is used to define a dashed border.
solid	It is used to define a solid border.
double	It defines two borders wlth the same border-width value.
groove	It defines a 3d grooved border. effect is generated according to border-color value.
ridge	It defines a 3d ridged border. effect is generated according to border-color value.
inset	It defines a 3d inset border. effect is generated according to border-color value.
outset	It defines a 3d outset border. effect is generated according to border-color value.

- **CSS border-width**
- p.one {
- border-style: solid;
- border-width: 5px;
- }
- **<body>**
- **<p** class="one"**>**Write your text here.**</p>**
- </body>

- **CSS border-color**
- Eg.
- p.one {
- border-style: solid;
- border-color: red;
- }
- **Border short hand:**
- border property is a shorthand property
  - border-width
  - border-style (required)
  - border-color
- **Eg.**
- p {
  border: 5px solid red;
  }

- **Border Radius:**
- p.round1 {
- border: 2px solid red;
- border-radius: 5px;
- padding: 5px;
- }
- **Eg.**
- Round border

- **Margins:**
- Margins are used to create space around elements, outside of any defined borders.
- **Margin Properties:**
- margin-top,margin-right,margin-bottom
- margin-left
- **Margin - Shorthand Property:**
- **margin: 25px 50px 75px 100px;**
  - top margin is 25px
  - right margin is 50px
  - bottom margin is 75px
  - left margin is 100px

```
<html>
<head>
<style>
div {
  border: 1px solid black;
  margin: 25px 50px 75px;
  background-color: lightblue;
}
</style>
</head>
<body>
<h2>The margin shorthand property - 3 values</h2>
<div>This div element has a top margin of 25px, a right and left margin of 50px, and a bottom margin of 75px.</div>
</body>
</html>
```

- **CSS Padding**
- Padding is used to create space around an element's content, inside of any defined borders.
- CSS has properties for specifying the padding for each side of an element:
- padding-top,padding-right,padding-bottom
- padding-left

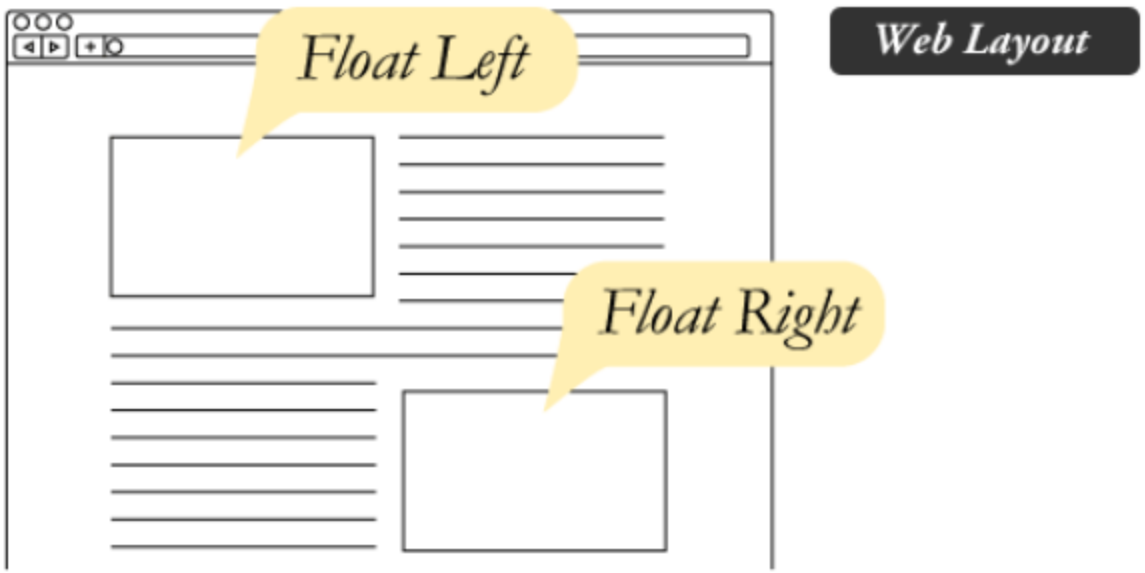
- Padding - Shorthand Property
- **padding: 25px 50px 75px 100px;**
  - top padding is 25px
  - right padding is 50px
  - bottom padding is 75px
  - left padding is 100px
- Eg.
- div {
  padding: 25px 50px 75px 100px;
  }

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  border: 1px solid black;
  background-color: lightblue;
  padding-top: 50px;
  padding-right: 30px;
  padding-bottom: 50px;
  padding-left: 80px;
}
</style>
</head>
<body>
<h2>Using individual padding properties</h2>
<div>This div element has a top padding of 50px, a right padding of 30px, a bottom padding of 50px, and a left padding of 80px.</div>
</body>
</html>
```

- **CSS Setting height and width**
- <html>
- <head>
- <style>
- div {
- height: 200px;
- width: 50%;
- background-color: powderblue;
- }
- </style>
- </head>
- <body>
- <h2>Set the height and width of an element</h2>
- This div element has a height of 200px and a width of 50%.

- </body>
- </html>

- CSS Float
- The **CSS float property** is *a positioning property*. It is used *to push an element to the left or right*, allowing other element to wrap around it.
- It is generally used with images and layouts.



- The float property can have one of the following values:
- left - The element floats to the left of its container
- right - The element floats to the right of its container
- none - The element does not float (will be displayed just where it occurs in the text). This is default
- inherit - The element inherits the float value of its parent

```
<html>
<head>
<style>
img {
        float: right;
}
</style>
</head>
<body>

<img src="good-morning.jpg" alt="Good Morning Friends"/>

This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text.

</p>
</body>
</html>
```

- **CSS font-family**
- This CSS property is used to provide a comma-separated list of font families.
- Ex.
- h1.a {
- font-family: "Times New Roman", Times, serif; color:Red;
- }

- **display: inline;**
- **Exmple:**
- <html>
- <head>
- <style>
- p {
- display: inline-block;
- }
- </style>
- </head>
- <body>
- Hello Javatpoint.com
- Java Tutorial.
- SQL Tutorial.
- HTML Tutorial.
- CSS Tutorial.
- </body>
- </html>

- **Navigation Bar:**
- **<html><head>**
- **<style>**
- **ul {**
- **list-style-type: none;**
- **margin: 0;**
- **padding: 0px;**
- **overflow: hidden;**
- **background-color: lightgray;**
- **}**
- **li {**
- **float: left;**
- **}**

- **li a {**
- **display: block;**
- **color: blue;**
- **font-size:20px;**
- **text-align: center;**
- **padding: 10px 20px;**
- **text-decoration: none;**
- **}**
- **.active{**
- **background-color: gray;**
- **color: white;**
- **}**

```
li a:hover {
  background-color: orange;
  color: white;
}
</style>
</head>

<body>

<ul>
  <li><a class="active" href="#home">Home</a></li>
  <li><a href="#">Java</a></li>
  <li><a href="#">HTML</a></li>
  <li><a href="#">CSS</a></li>
</ul>

</body>
</html>
```

CSS Text

- **Text Color and Background Color**
- h1 {
  color: green;
  background-color: lightgrey;
}
- **Text Alignment and Text Direction**
- **text-align** (left, right, centered, or justified.)
- **text-align-last**: align the last line of a text.
- Eg. text-align-last: right;
- **text-direction and unicode-bidi**:
- change the text direction of an element:

```
<html>
<head>
<style>
p.ex1 {
  direction: rtl;
  unicode-bidi: bidi-override;
}
</style>
</head>
<body>
<p>This is the default text direction.</p>
<p class="ex1">This is right-to-left text direction.</p>
</body>
</html>
```

- **vertical-align:** sets the vertical alignment of an element.
- Example:

- **Text Decoration: properties**
- text-decoration-line
- text-decoration-color
- text-decoration-style
- text-decoration-thickness
- text-decoration

- **Text-decoration-line: (to add decoration to text) and Text-decoration-color**
- **Example:**
- ```
  h1 {
    text-decoration-line: overline;
    text-decoration-color: red;
  }
  h2 {
    text-decoration-line: line-through;
    text-decoration-color: blue;
  }
  ```
  ```

- **Style for the Decoration Line**
- **Example:**
- ```
  h1 {
    text-decoration-line: underline;
    text-decoration-style: solid;
  }
  h2 {
    text-decoration-line: underline;
    text-decoration-style: double;
  }
  h3 {
    text-decoration-line: underline;
    text-decoration-style: dotted;
  }
  p.ex1 {
    text-decoration-line: underline;
    text-decoration-style: dashed;
  }
  ```
  ```

- Thickness for the Decoration Line
- **Example:**
- h2 {
  text-decoration-line: underline;
  text-decoration-thickness: 5px;
  }
  h3 {
  text-decoration-line: underline;
  text-decoration-thickness: 25%;
  }

- **Important:** All **links** in HTML are underlined by default. Sometimes you see that links are styled with no underline. The text-decoration: none; is used to remove the underline from links, like this:
- a {
  text-decoration: none;
  }

- <html>
- <head>
- <style>
- a {
- text-decoration: none;
- }
- </style>
- </head>
- <body>
- <h1>Using text-decoration: none</h1>
- A link with no underline: HOME
- </body>
- </html>

- **Text Transformation**
- used to specify uppercase and lowercase letters in a text.
- Example:
- p.uppercase {
  text-transform: uppercase;
  }
  p.lowercase {
  text-transform: lowercase;
  }
  p.capitalize {
  text-transform: capitalize;
  }

- **Text Spacing**
- text-indent
- letter-spacing
- line-height
- word-spacing
- white-space

- **text-indent**: specify the indentation of the first line of a text:
- p {
  text-indent: 50px;
  }
- **Letter Spacing:** specify the space between the characters in a text.
- h2 {
- letter-spacing: 5px;
- }
- h3 {
- letter-spacing: -2px;
- }

- **Line Height:** Specify the space between lines:
- Example:
- p.small {
  line-height: 0.8;
  }
  p.big {
  line-height: 1.8;
  } Word Spacing

- Word Spacing: specify the space between the words in a text.
- **Example:**
- p.one {
  word-spacing: 10px;
  }
  p.two {
  word-spacing: -2px;
  }

- **Text Shadow:** adds shadow to text.
- you only specify the horizontal shadow (2px) and the vertical shadow (2px):
- h1 {
  text-shadow: 2px 2px;
  }
- add a color (red) to the shadow:
- h1 {
  text-shadow: 2px 2px red;
  }

- add a blur effect (5px) to the shadow
- h1 {
  text-shadow: 2px 2px 5px red;
  }

- **CSS Lists**
- List Item Markers
- The **list-style-type** property specifies the type of list item marker.
- ul.a {
  list-style-type: circle;
  }
  ul.b {
  list-style-type: square;
  }
  ol.c {
  list-style-type: upper-roman;
  }
  ol.d {
  list-style-type: lower-alpha;
  }

- **Styling List With Colors**
- ol {
  background: #ff9999;
  padding: 20px;
  }
-
  ul {
  background: #3399ff;
  padding: 20px;
  }

  ol li {
  background: #ffe5e5;
  color: darkred;
  padding: 5px;
  margin-left: 35px;
  }

  ul li {
  background: #cce5ff;
  color: darkblue;
  margin: 5px;
  }

- **Styling Links:**
- Links can be styled with any CSS property (e.g. color, font-family, background, etc.).
- a {
  color: hotpink;
  }
- The four links states are:
- a:link - a normal, unvisited link
- a:visited - a link the user has visited
- a:hover - a link when the user mouses over it
- a:active - a link the moment it is clicked
-

- ```
  /* unvisited link */
  a:link {
    color: red;
  }

  /* visited link */
  a:visited {
    color: green;
  }

  /* mouse over link */
  a:hover {
    color: hotpink;
  }

  /* selected link */
  a:active {
    color: blue;
  }
  ```
  ```

- When setting the style for several link states, there are some order rules:
- a:hover MUST come after a:link and a:visited
- a:active MUST come after a:hover

- **The display Property**
- The display property specifies if/how an element is displayed.
- **Block-level Elements**
- A block-level element always starts on a new line and takes up the full width available (stretches out to the left and right as far as it can).
- Eg. Examples of block-level elements:
-
- <h1> - <h6>
- <p>
- <form>
- <header>
- <footer>
- <section>

- Inline Elements
- An inline element does not start on a new line and only takes up as much width as necessary.
- Examples of inline elements:
-
- <a>
- <img>
-

- **Example:**
- li {
  display: inline;
  }
- **Note:** Setting the display property of an element only changes **how the element is displayed**, NOT what kind of element it is. So, an inline element with display: block; is not allowed to have other block elements inside it.

- Navigation bar
- Navigation Bar = List of Links
- <!DOCTYPE html>
- <html>
- <body>

-
- Home
- News
- Contact
- About
-

- </body>
- </html>

- Remove the bullets and the margins and padding from the list:
- ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
  }

- **Horizontal Navigation Bar**
- There are two ways to create a horizontal navigation bar. Using **inline** or **floating** list items.
- **Inline List Items**
- Example:
- li {
  display: inline;
  }

```
<!DOCTYPE html>
<html>
<head>
<style>
ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
}

li {
  display: inline;
}
</style>
</head>
<body>

<ul>
  <li><a href="#home">Home</a></li>
  <li><a href="#news">News</a></li>
  <li><a href="#contact">Contact</a></li>
  <li><a href="#about">About</a></li>
</ul>

</body>
</html>
```

- **Floating List Items**
- float the <li> elements, and specify a layout for the navigation links:
- Example:
- li {
  float: left;
  }

  a {
  display: block;
  padding: 8px;
  background-color: #dddddd;
  }

- **CSS Fonts**
- **Font Family**
- **Eg.**
- body { font-family: Arial, Helvetica, sans-serif; }
- **Font Style**
- p.normal { font-style: normal; }
- p.italic { font-style: italic; }
- p.oblique { font-style: oblique; }

- **Font Size:**
- **Size with pixel:**
- h1 { font-size: 24px; }
- **Size with EM:**
- if you haven't set the font size anywhere on the page, then it is the browser default, which is normally **16px**. Therefore, by default **1em = 16px**, and **2em = 32px.**
- **h1** { font-size: 2em; /* 32px/16px=2em */ }

- **Font Size with Root EM**
- CSS3 has introduced rem unit (short for "root em") which is always relative to the font-size of the root element (html).
- This means that 1rem is equivalent to the font size of the html element, which is 16px by default.
- p span
- { font-size: 2rem; /* 2rem = 20px (not 28px)
- }

- **Font Size with Viewport Units**
- he font sizes can be specified using viewport units such as vw or vh.
- Viewport units refer to a percentage of the browser's viewport dimensions, where 1vw = 1% of viewport width, and 1vh = 1% of viewport height. Hence, if the viewport is 1600px wide, 1vw is 16px.
- body { font-size: 1vw; }
- h1 { font-size: 3vw; }

- **Font Weight**
- The font-weight property specifies the weight or boldness of the font.
- This property can take one of the following values: normal, bold, bolder, lighter, 100, 200, 300, 400, 500, 600, 700, 800, 900 and inherit.
- p { font-weight: bold; }

Pseudo-class

- The CSS pseudo-classes allow you to style the dynamic states of an element such as hover, active and focus state, as well as elements that are existing in the document tree but can't be targeted via the use of other selectors without adding any IDs or classes to them, for example, targeting the first or last child elements.

- Syntax:
- selector:pseudo-class { property: value; }
- **Anchor Pseudo-classes**
- Using anchor pseudo-classes links can be displayed in different ways
- a:link { color: blue; }
- a:visited { text-decoration: none; }

- These pseudo-classes change how the links are rendered in response to user actions.
- **:hover** applies when a user places cursor over the element, but does not select it.
- **:active** applies when the element is activated or clicked.
- **:focus** applies when the element has keyboard focus.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Example of Dynamic Anchor Pseudo-classes</title>
<style>
  a:link {
    color: blue;
  }
  a:visited {
    text-decoration: none;
  }
  a:hover {
    color: red;
  }
  a:active {
    color: gray;
  }
  a:focus {
    color: yellow;
  }
</style>
</head>
<body>
  <p>Visit <a href="#" target="_blank">HOME</a></p>
</body>
</html>
```

- **The :first-child Pseudo-class**
- The :first-child pseudo-class matches an element that is the first child element of some other element.
- The selector ol li:first-child select the first list item of an ordered list and removes the top border form it.
- Eg. ol li:first-child { border-top: none; }
- **The :last-child Pseudo-class**
- The :last-child pseudo-class matches an element that is the last child element of some other element.
- The selector ul li:last-child , select the last list item from an unordered list and removes the right border from it.
- Eg. ul li:last-child { border-right: none; }

- **The :nth-child Pseudo-class**
- a new :nth-child pseudo-class that allows you to target one or more specific children of a given parent element.
- The basic syntax of this selector can be given with :nth-child(N), where N is an argument, which can be a number, a keyword (even or odd), or an expression of the form xn+y where x and y are integers (e.g. 1n, 2n, 2n+1, …).

- **CSS3 Text Overflow**
- Text can overflow, when it is prevented from wrapping.
- use the CSS3 text-overflow property to determine how the overflowed text content will be displayed.
- property are: clip and ellipsis
- Syntax:

- p.clipped { text-overflow: clip;
- /* clipped the overflowed text */ }
- p.ellipses { text-overflow: ellipsis;
- /* display '…' to represent clipped text */ }

- **CSS column-count:**
- The column-count property partitions an element into columns.
- This property is used to create columns for long text content.
- Examples include articles, news, blogs, essays, and more.
- Syntax:
- column-count: *number* | auto | initial | inherit;

VALUE	DESCRIPTION
number	Number of columns to create
auto	Default. Column count is determined by column-width and others
initial	Sets the value to its default value.
inherit	Inherits the value from its parent element.

Example:

```
div
{
        column-count: 3;
}
```

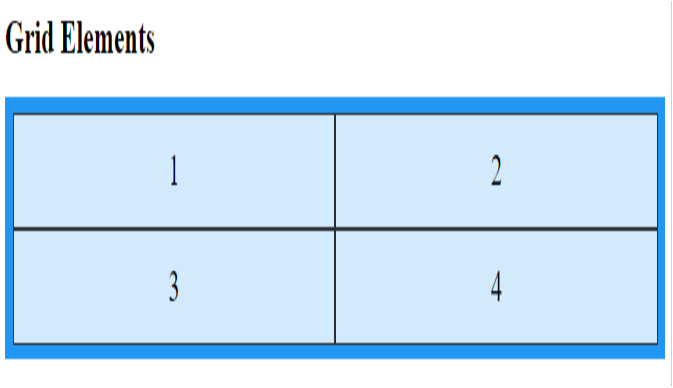
- **CSS column-span Property**
- The column-span property specifies how many columns an element should span across.
- **Syntax:**
- column-span: 1|all|initial|inherit;

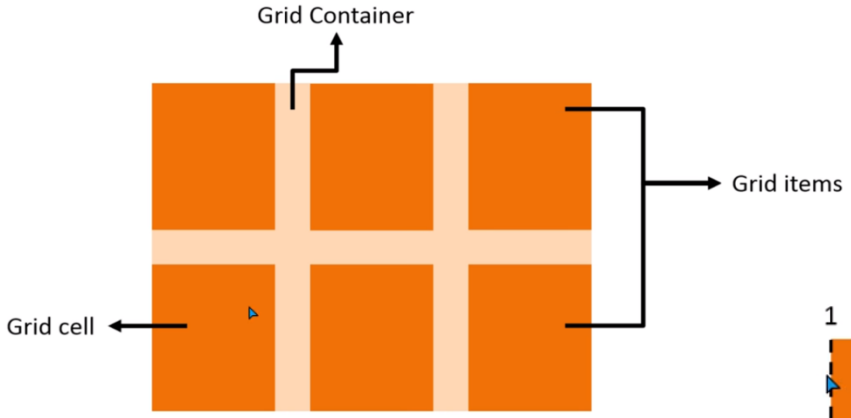
Value	Description
1	Default value. The element should span across one column
all	The element should span across all columns
initial	Sets this property to its default value.
inherit	Inherits this property from its parent element.

- **CSS3 column-rule Property:**
- Specify the width, style and color of the rule between columns.
- **Syntax:**
- **column-rule:** *column-rule-width column-rule-style column-rule-color*|initial|inherit;
- Eg. column-rule: 4px solid red;

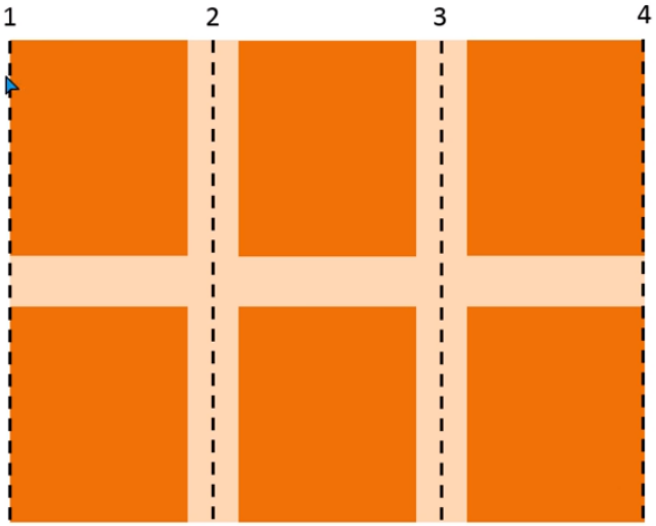
CSS Grid Layout Module

- The CSS Grid Layout Module offers a grid-based layout system, with rows and columns, making it easier to design web pages without having to use floats and positioning.
- **Grid Elements**
- A grid layout consists of a parent element, with one or more child elements.
- **Example:**
-
  1
  2
  3
  4
-

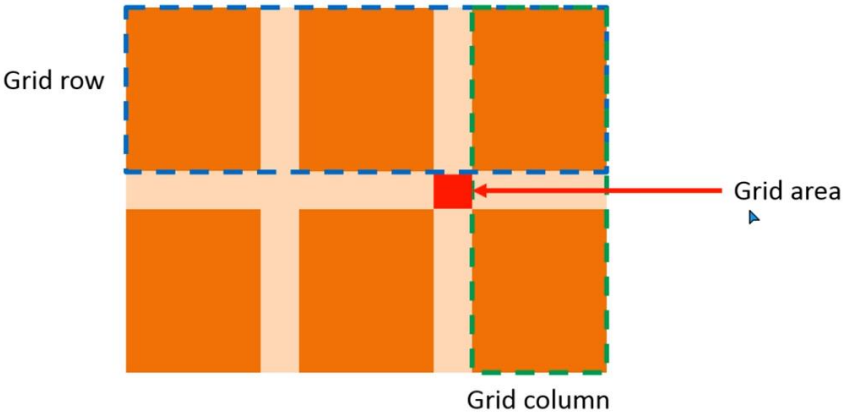




display : grid



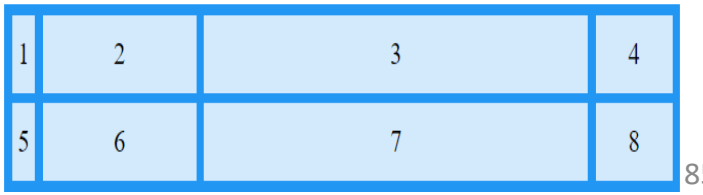
Grid line



- CSS Grid Properties:
  - grid-template-rows
  - grid-template-columns
  - grid-template-areas
  - grid-template
  - grid-row-gap
  - grid-column-gap
  - grid-gap
  - justify-items
  - align-items
  - justify-content
  - align-content
  - grid-auto-rows
  - grid-auto-columns
  - grid-auto-flow
  - grid-row-start
  - grid-row-end
  - grid-row
  - grid-column-start
  - grid-column-end
  - grid-column
  - grid-area
  - justify-self
  - align-self
  - order

- **Display Property:**
- An HTML element becomes a grid container when its display property is set to grid or inline-grid.
- Eg.
- .grid-container {
  display: grid;
  }

- **Grid Gaps**
- You can adjust the gap size by using one of the following properties:
- column-gap: **column-gap: 50px;**
- row-gap: **row-gap: 50px;**
- Gap: **gap: 50px 100px; (row-gap column-gap shorthand)**
- **CSS grid-template-columns Property**
- The grid-template-columns property specifies the number (and the widths) of columns in a grid layout.
- Make a 4 columns grid container:
- **grid-template-columns: auto auto auto auto; or**
- **grid-template-columns: 30px 200px auto 100px;**



- **CSS grid MinMax() function**
- To change the height and width of rows and columns dynamically.
- **Function:**
- Max-content:
- grid-template-columns: max-content 1fr 1fr;
- Min-content:
- grid-template-columns:max-content 1fr min-content;
- Minmax:
- grid-template-row:repeat(2,minmax(150,mincontent));