

What is IDE

1. IDE stands for Integrated Development Environment.
2. Software suite that consolidates basic tools required for software development.
3. Central hub for coding, finding problems, and testing.
4. Designed to improve developer efficiency.



Need of IDE

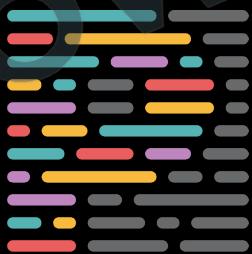
1. Streamlines development.
2. Increases productivity.
3. Simplifies complex tasks.
4. Offers a unified workspace.
5. IDE Features
 1. Code Autocomplete
 2. Syntax Highlighting
 3. Version Control
 4. Error Checking

```
@Composable
fun MessageCard(msg: Message) {
    Row(modifier = Modifier.padding(all = 8.dp)) {
        Image(
            painter = painterResource(R.drawable.android_studio_logo),
            contentDescription = "Profile Picture",
            modifier = Modifier
                .size(45.dp)
        )
        Spacer(modifier = Modifier.width(8.dp))
        Column (Modifier
            .background(color = Color.White)) {
            Text(text = msg.author, color = Color.Black)
            Spacer(modifier = Modifier.height(1.dp))
            Text(text = msg.body, color = Color.Black)
        }
    }
}
```



VsCode Extensions

1. Live Server
2. Prettier



KNOWLEDGE GATE¹

FrontEnd / BackEnd / FullStack



Client Side / Front-
End

Web Development



Server Side
Back-End

M E R N

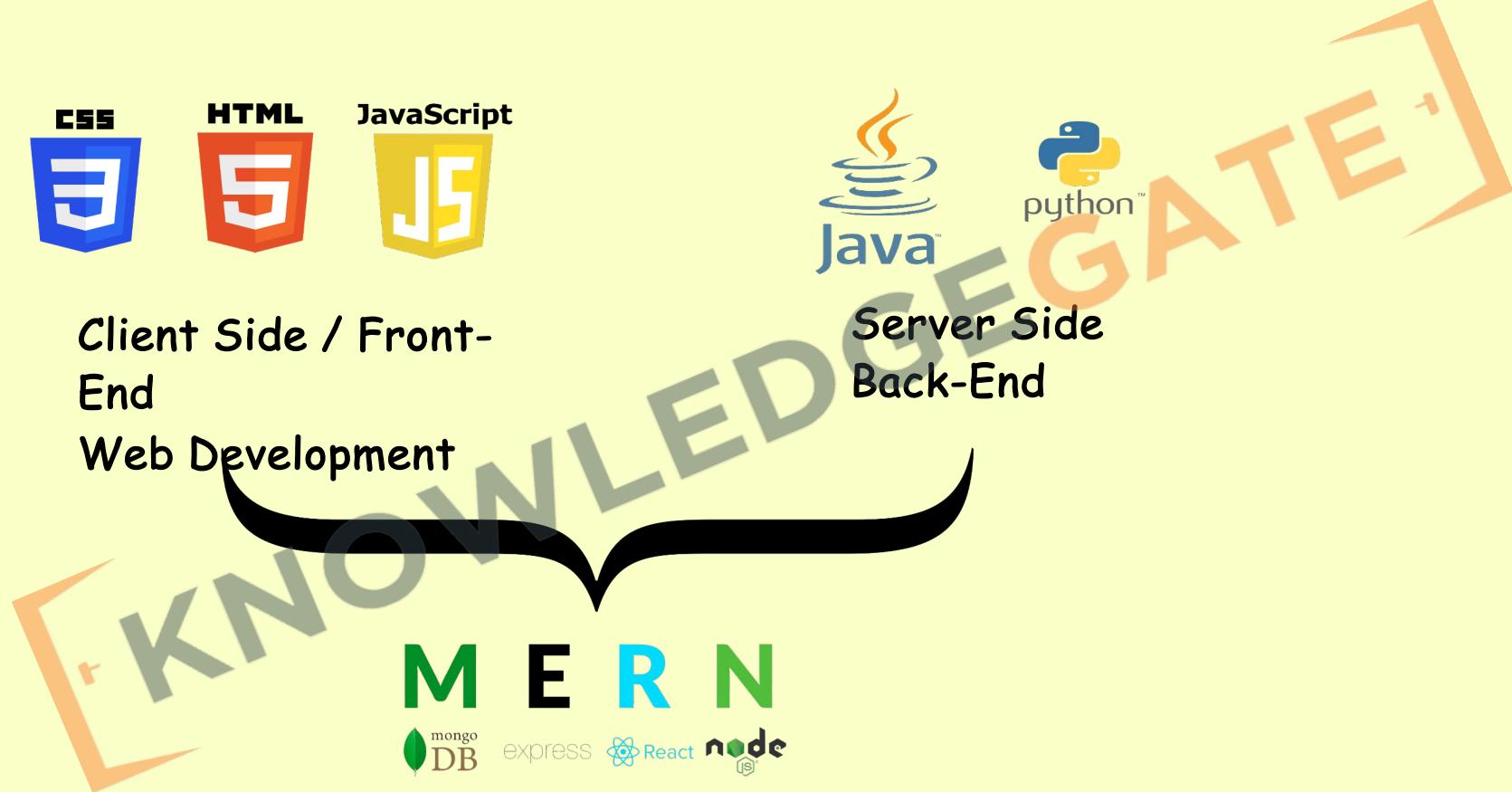


express

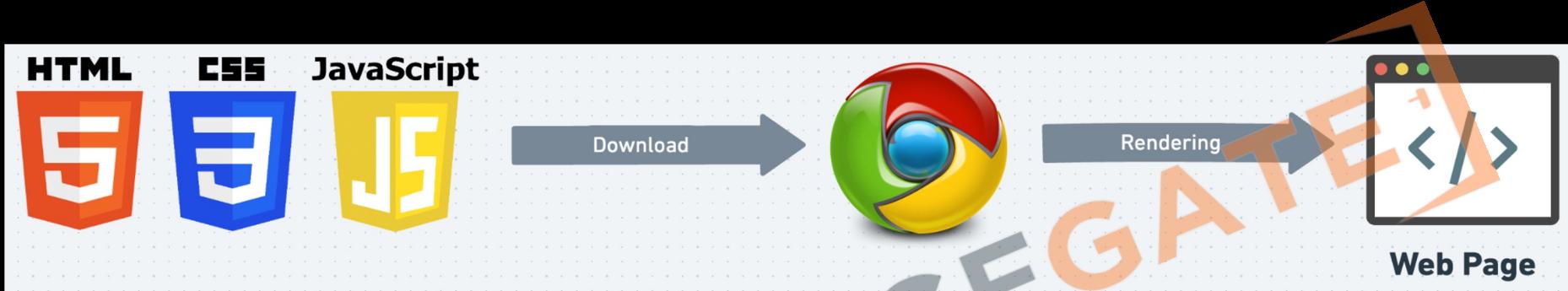


node

Full Stack



Role of Browser



1. **Displays Web Page:** Turns HTML code into what you see on screen.
2. **User Clicks:** Helps you interact with the web page.
3. **Updates Content:** Allows changes to the page using JavaScript.
4. **Loads Files:** Gets HTML, images, etc., from the server.



HTML

(Hypertext Markup Language)

1. **Structure:** Sets up the layout.
2. **Content:** Adds text, images, links.
3. **Tags:** Uses elements like `<p>`, `<a>`.
4. **Hierarchy:** Organizes elements in a tree.





CSS

(Cascading Style Sheets)

1. **Style:** Sets the look and feel.
2. **Colors & Fonts:** Customizes text and background.
3. **Layout:** Controls position and size.
4. **Selectors:** Targets specific **HTML elements.**





JS

(Java Script)

1. JavaScript has nothing to do with Java
2. **Actions:** Enables interactivity.
3. **Updates:** Alters page without reloading.
4. **Events:** **Responds** to user actions.
5. **Data:** Fetches and sends info to server.





Importance of CSS



Premium Website



Premium Brand



Premium Customer



High Salary Developer

Course Levels

Level 1 CSS Basics

Level 2 Color System and Background

Level 3 Text Properties

Level 4 Box Model CSS

Level 5 Display & Position

Level 6 Flexbox, Grid & Media Queries

Level 7 Animation, Transition & Transform

Project:  Myntra Clone

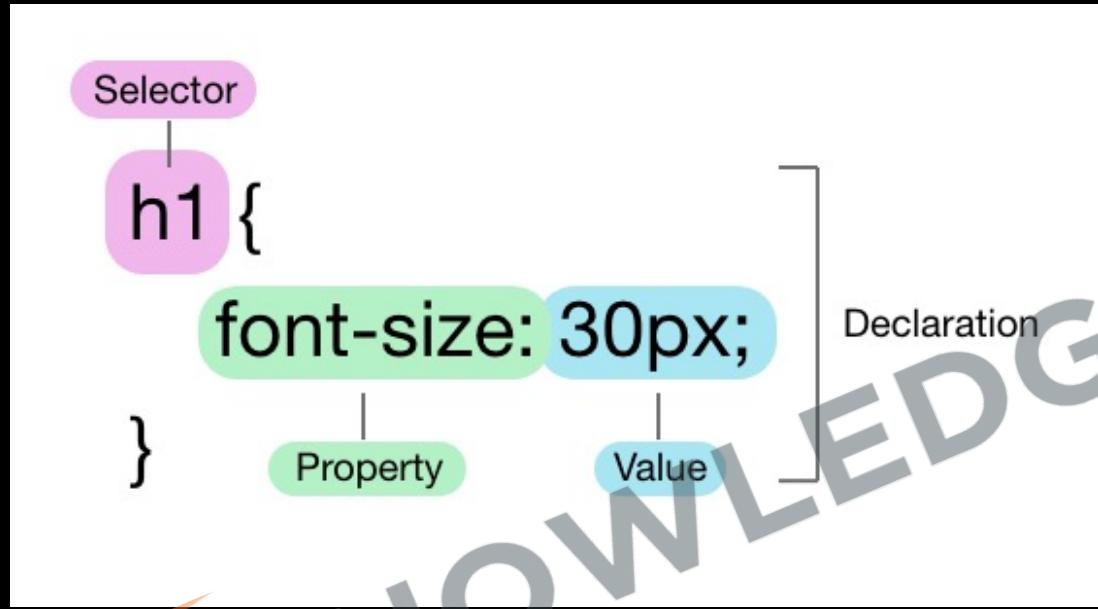


Level 1

CSS Basics

1. Basic Syntax
2. Color Property
3. Including Styles
4. HTML Refresher
5. Comments
6. MDN Documentation
7. Browser Tools
8. Selectors

1. Basic Syntax



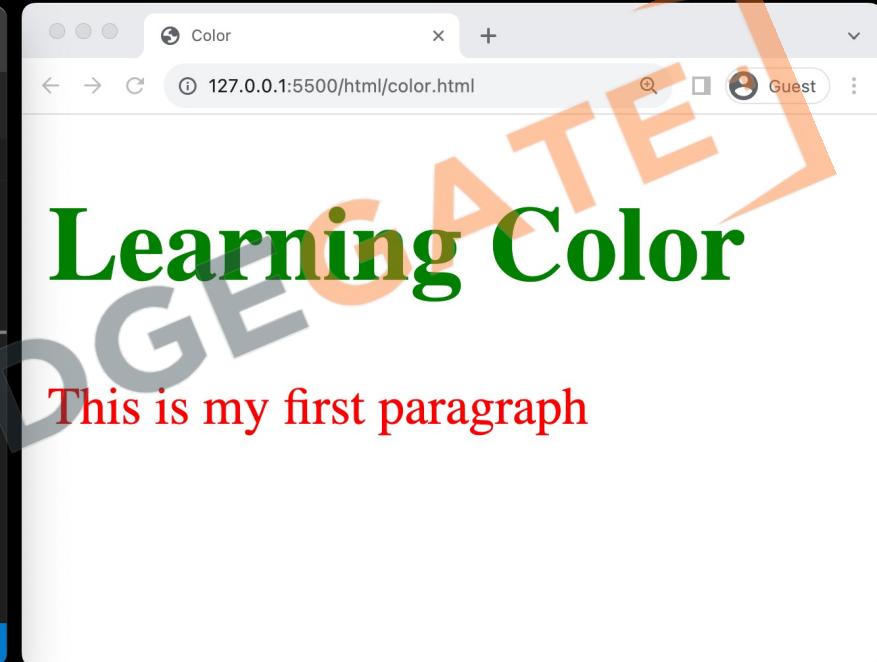
- **Selector:** The HTML element that you want to style.
- **Property:** The attribute you want to change (like font, color, etc.).
- **Value:** The specific style you want to apply to the property (like red, bold, etc.).

2. Color Property

A screenshot of a code editor window titled "color.html". The code is as follows:

```
html > color.html >
html > head > style > p
7   <style>
8     h1 {
9       color: green;
10    }
11   p {
12     color: red;
13   }
14 </style>
15 </head>
16 <body>
17   <h1>Learning Color</h1>
18   <p>This is my first paragraph</p>
19 </body>
```

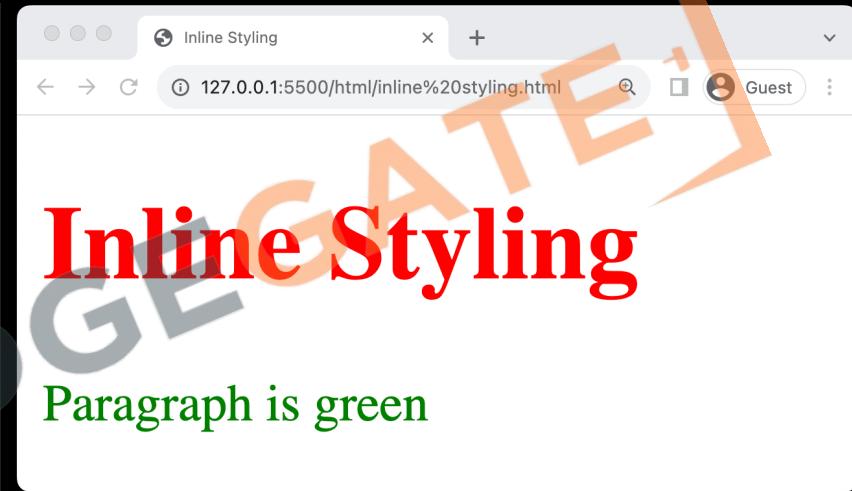
The editor has a dark theme with light-colored syntax highlighting. A sidebar on the left contains icons for file operations, search, and user profile.



- **Definition:** The CSS color property defines the text color or foreground color in an HTML element.
- **Enhancement:** Use it to emphasize sections and elevate webpage aesthetics.

3. Including Styles (Inline Styling)

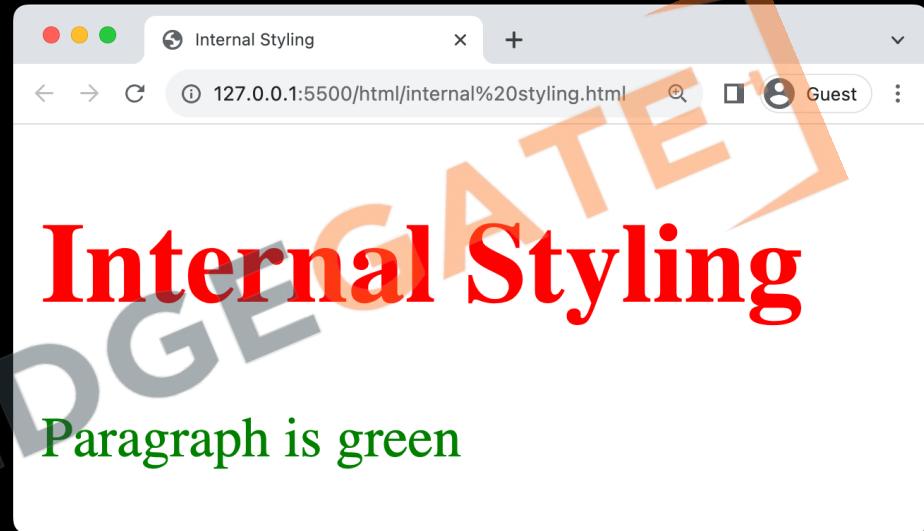
```
html > inline styling.html > html
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4  |   <title>Inline Styling</title>
5  </head>
6  <body>
7  |   <h1 style="color: red;">Inline Styling</h1>
8  |   <p style="color: green;">Paragraph is green</p>
9  </body>
10 </html>
```



- **Direct Application:** Apply styles directly to **HTML** elements using the **style** attribute.
- **One-off Changes:** Ideal for **single**, unique style alterations.
- **Can Be Cluttered:** May lead to **cluttered** HTML if used extensively.
- **Limited Reusability:** Reduces the **reusability** of CSS rules in larger projects.

3. Including Styles (Internal Styling)

```
3 <head>
4     <title>Internal Styling</title>
5     <style>
6         h1 {color: red;}
7         p {color: green;}
8     </style>
9 </head>
10 <body>
11     <h1>Internal Styling</h1>
12     <p>Paragraph is green</p>
13 </body>
14 </html>
```

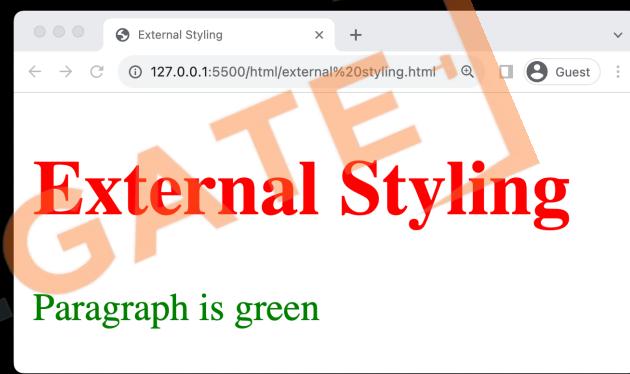


- **Embedded CSS:** Styles are placed within `<style>` tags in the HTML head section.
- **Cleaner than Inline:** More organized compared to inline styles.
- **Reusable Styles:** Allows for some reuse of styles across the page.

3. Including Styles (External Styling)

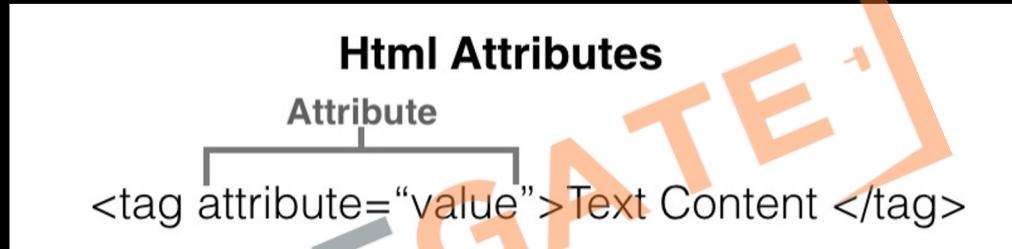
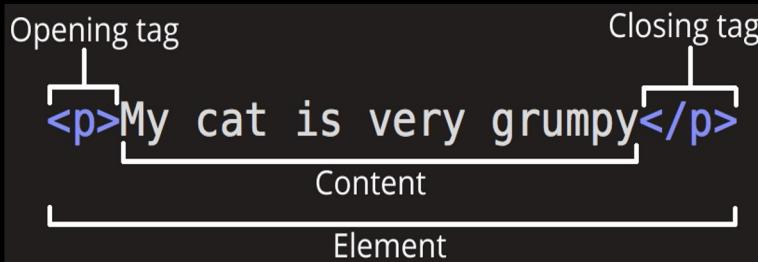
```
3 <head>
4     <title>External Styling</title>
5     <link rel="stylesheet" href="../css/
6         external styling.css">
7 </head>
8 <body>
9     <h1>External Styling</h1>
10    <p>Paragraph is green</p>
11 </body>
```

```
css > # external styling.css > ↗ p
1   ↘ h1 {
2       color: red;
3   }
4
5   ↘ p {
6       color: green;
7   }
```



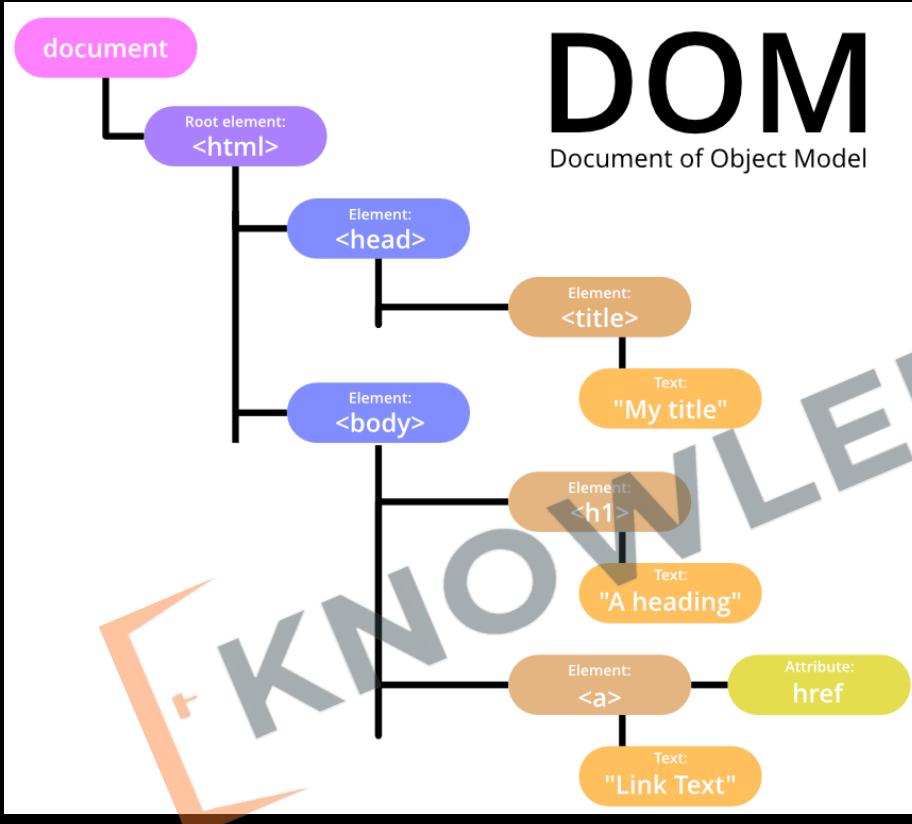
- **Separate CSS File:** Stores styles in a **separate .css file**, linked to HTML.
- **Reusable:** Enables style **reuse** across multiple webpages.
- **Link in HTML:** Use the **<link>** tag within the **<head>** section to link the CSS.
- **Relative or Absolute Path:** The href attribute can contain a relative or absolute path to the CSS file.

4. HTML Refresher (Tags & Attributes)



1. Elements that are used to create a website are called HTML Tags.
2. Tags can contain content or other HTML tags.
3. Define elements like text, images, links

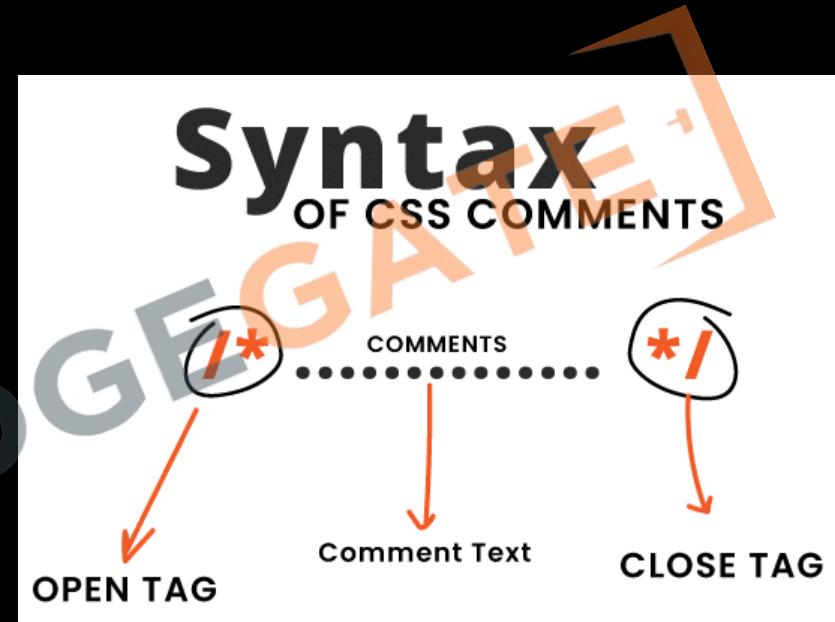
4. HTML Refresher (DOM)



- 1. Structure Understanding:** Helps in understanding the **hierarchical structure** of a webpage, crucial for applying targeted CSS styles.
- 2. Dynamic Styling:** Enables learning about dynamic styling, allowing for **real-time changes** and interactivity through CSS.

5 Comments

1. Used to add **notes** in HTML or CSS code
2. **Not displayed** on the web page
3. Syntax: `<!-- Comment here -->`
4. Helpful for **code organization**
5. Can be multi-line or single-line

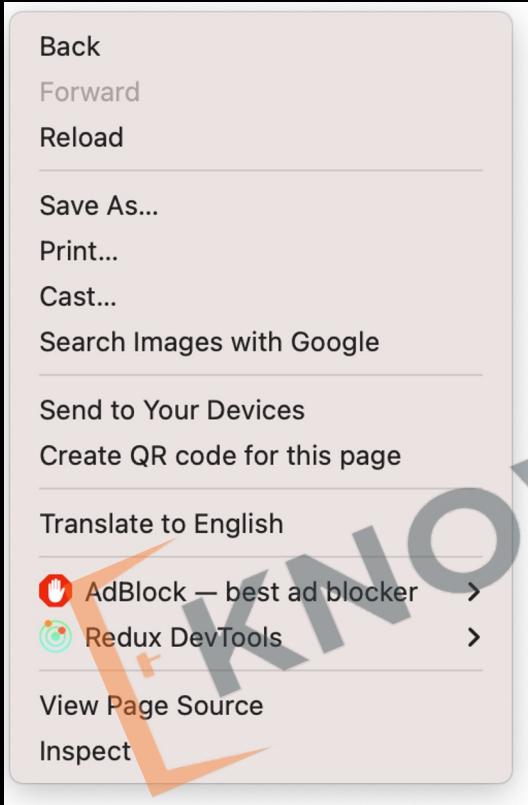


6.MDN Documentation

```
# external styling.css U X
css > # external styling.css
1 h1 { Sets the color of an element's text
2   | (Edge 12, Firefox 1, Safari 1, Chrome 1, IE 3, Opera 3)
3   }
4
5 p { Syntax: <color>
6   | MDN Reference
7   | color: green;
8 }
```

1. For Official resource for CSS, visit
<https://developer.mozilla.org/>
2. Includes examples for real-world use
3. Updated with latest CSS-3 features
4. Trusted by developers worldwide

7. Browser Tools (Inspect element)



1. Allows **real-time editing** of HTML/CSS
2. Useful for debugging and testing
3. Shows **element hierarchy** and layout
4. **Highlights** selected elements on page

7. Browser Tools (css Live Edit)

The screenshot shows a YouTube channel page for 'kg coding'. The channel has 10M subscribers and 1 video. It features a banner with the text 'CODING MADE' and logos for Angular, Vue.js, MERN, C, C++, and Java. Below the banner, there's a profile picture of a man with glasses, a bio section with 'Coming Soon..... >', and links to 'knowledgegate.in' and '1 more link'. At the bottom, there are tabs for 'HOME', 'VIDEOS', 'LIVE', and 'PLAYLISTS'. A section titled 'Our Other Channels' lists 'KNOWLEDGE GATE' with 594K subscribers and 'KnowledgeGate Placement Prep' with 105K subscribers.

```
<div>
  <div>
    <div id="channel-name" data-yt="c4-tabbed-header-renderer">
      <!--css-build:shady-->
      <!--css-build:shady-->
      <div id="container" class="style-scope ytd-channel-name">
        <t-formatted-string id="text" link-inherit-color title class="style-scope ytd-channel-name">KG Coding</t-formatted-string> &#9733;
      </div>
      <p><tp-yt-paper-tooltip fit-to-visible-bounds class="style-scope ytd-channel-name" role="tooltip" ip="" tabindex="-1"></p></tp-yt-paper-tooltip>
    </div>
    <ytd-badge-supported-renderer class="style-scope ytd-channel-name" disable-upgrade hidden></ytd-badge-supported-renderer>
  </div>
  <p class="style-scope ytd-c4-tabbed-header-renderer" hidden></p>
  <span class="meta-item style-scope ytd-c4-tabbed-header-renderer"></span>
  <span class="meta-item style-scope ytd-c4-tabbed-header-renderer" hidden></span>
  <span class="meta-item style-scope ytd-c4-tabbed-header-renderer">
    <t-formatted-string id="subscriber-count" class="style-scope ytd-c4-tabbed-header-renderer" aria-label="13.8K subscribers">10M subscribers</t-formatted-string>
    <span aria-hidden="true" class="delimiter style-scope ytd-c4-tabbed-header-renderer"></span>
  </span>
  <span class="meta-item style-scope ytd-c4-tabbed-header-renderer"></span>
  <div id="channel-tagline" class="style-scope ytd-c4-tabbed-header-renderer"></div>
  <div id="channel-header-links" class="style-scope ytd-c4-tabbed-header-renderer"></div>
  <div id="buttons" class="style-scope ytd-c4-tabbed-header-renderer"></div>
  </div>
  <div id="links-holder" class="style-scope ytd-c4-tabbed-header-renderer"></div>
  </div>
  </div>
  <tp-yt-app-toolbar sticky class="style-scope ytd-c4-tabbed-header-renderer" style="transform: translate3d(0px, 0px, 0px); "></tp-yt-app-toolbar>
</div>
</tp-yt-app-header>
<div id="contentContainer" class="style-scope tp-yt-app-header-layout" style="padding-top: 364px;"></div>
</tp-yt-app-header-layout>
</ytd-c4-tabbed-header-renderer>
</div>
<div id="alerts" class="style-scope ytd-browse"></div>
<ytd-channel-legal-info-renderer class="style-scope ytd-browse" disable-upgrade hidden></ytd-channel-legal-info-renderer>
<ytd-playlist-sidebar-renderer class="style-scope ytd-browse" disable-upgrade hidden></ytd-playlist-sidebar-renderer>
<div>
```

Changed Channel Name color

7. Browser Tools (JS Live Edit)

The screenshot shows a YouTube channel page for 'KONLEGATE'. The main video thumbnail features a man in a white t-shirt with 'KONLEGATE' printed on it, standing in front of a wall with various technology logos (Angular, Vue.js, MongoDB, Express, React, Node.js) and the word 'CODING MADE SIMPLE'. The channel has 1.4M subscribers. Below the thumbnail are navigation links for 'HOME', 'VIDEOS', 'LIVE', and 'PLAYLISTS'. On the right side, the browser's developer tools are open, specifically the Network tab. A large watermark reading 'ACKNOWLEDGED' is overlaid across the center of the screen.

Styles Computed Layout Event Listeners DOM Breakpoints Properties Accessibility

:hov .cls + ↗ featured:1

Filter

Search

Console What's New

top Filter Default levels 5 Issues: 1 4 1 hidden

com/s/desktop/462a8050/sjoin/custom-elements-ess-adapt.../k8-opt/bin/cnird_party/va...
vascript/custom_elements/fast-shim.js.sourcemap: HTTP error: status code 404,
net::ERR_HTTP_RESPONSE_CODE_FAILURE

⚠ DevTools failed to load source map: Could not load content for https://www.youtube...
com/s/desktop/462a8d5d/j...bin/web-animations-next-lite.min.vfset/web-animations-nex...
t-lite.min.js.map: HTTP error: status code 404, net::ERR_HTTP_RESPONSE_CODE_FAILURE

⚠ DevTools failed to load source map: Could not load content for https://www.youtube...
com/s/desktop/462a8d5d/j...bin/webcomponents-sd.vfset/bl..._party/javascript/polymer/v...
/webcomponentsjs/webcomponents-sd.js.sourcemap: HTTP error: status code 404,
net::ERR_HTTP_RESPONSE_CODE_FAILURE

LegacyDataMixin will be applied to all desktop_polymer_enable_wil icons.js:4482
legacy elements.
Set '_legacyUndefinedCheck: true' on element class to enable.

GET https://googleads.g.doubleclick.net/pagead/id
net::ERR_BLOCKED_BY_CLIENT VM15538:201

GET https://www.google.com/pagead/lv www.google.com/pagea...5CoWhrV 7RpGsmwVg:1
z?evtid=ACd6KtzZ40h0LZgbnYYZ8gxW6cxP
BX2-4...inAppBootstrap%3AUnclassified&az=1&sigh=AB9vU40PfuZ9Fiv1V5CoWhrV 7RpGsmwVg
net::ERR_BLOCKED_BY_CLIENT

GET https://www.google.co.in/pagead/ www.google.co.in/pag...5CoWhrV 7RpGsmwVg:1
lvz?evtid=ACd6KtzZ40h0LZgbnYYZ8gxW6c
xPBX2...inAppBootstrap%3AUnclassified&az=1&sigh=AB9vU40PfuZ9Fiv1V5CoWhrV 7RpGsmwVg
net::ERR_BLOCKED_BY_CLIENT

chrome-extension://invalid/ net::ERR_FAILED cast sender.js:10

chrome-extension://invalid/ net::ERR_FAILED cast sender.js:10

GET https://static.doubleclick.net/instream/ad_status.js
net::ERR_BLOCKED_BY_CLIENT spf.js:34

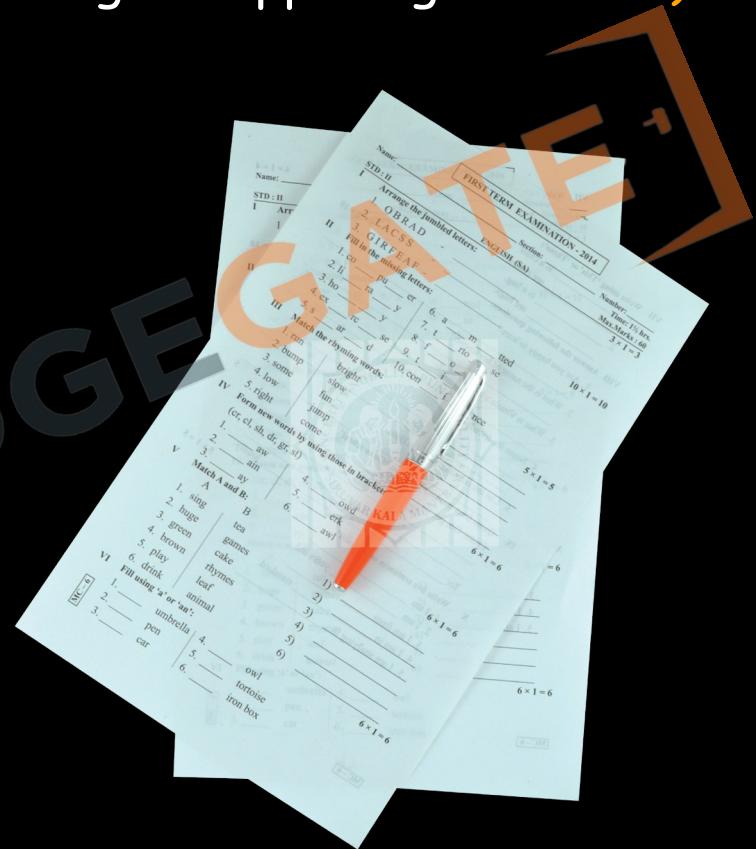
> document.getElementById("inner-header-container").style.visibility = 'hidden'
< 'hidden'

⚠ The resource https://i.ytimg.com/generate_204 was preloaded using link featured:1
preload but not used within a few seconds from the window's load event. Please make
sure it has an appropriate 'as' value and it is preloaded intentionally.

7. Browser Tools (*Changes happening at Client*)

1. Changes made are **temporary**
2. Affect **only** the current session
3. **Not saved** to the server
4. Reset upon page **reload**
5. Useful for **testing**, not permanent fixes

Like: If you change the question in your question paper that has no effect on actual exam.



7. Browser Tools (css Specific element)

The screenshot shows the Chrome DevTools interface with the 'Elements' tab selected. The DOM tree on the left shows the structure of the page, including the root element and various components like the head and body. The styles panel on the right displays the CSS rules applied to the selected element, with the 'Styles' tab circled in red. The styles listed include:

```
<!DOCTYPE html>
<html lang="en">
  <head> ...
  </head>
  <body>
    ...
      <h1>External Styling</h1> == $0
    </body>
</html>
```

external styling.css:1

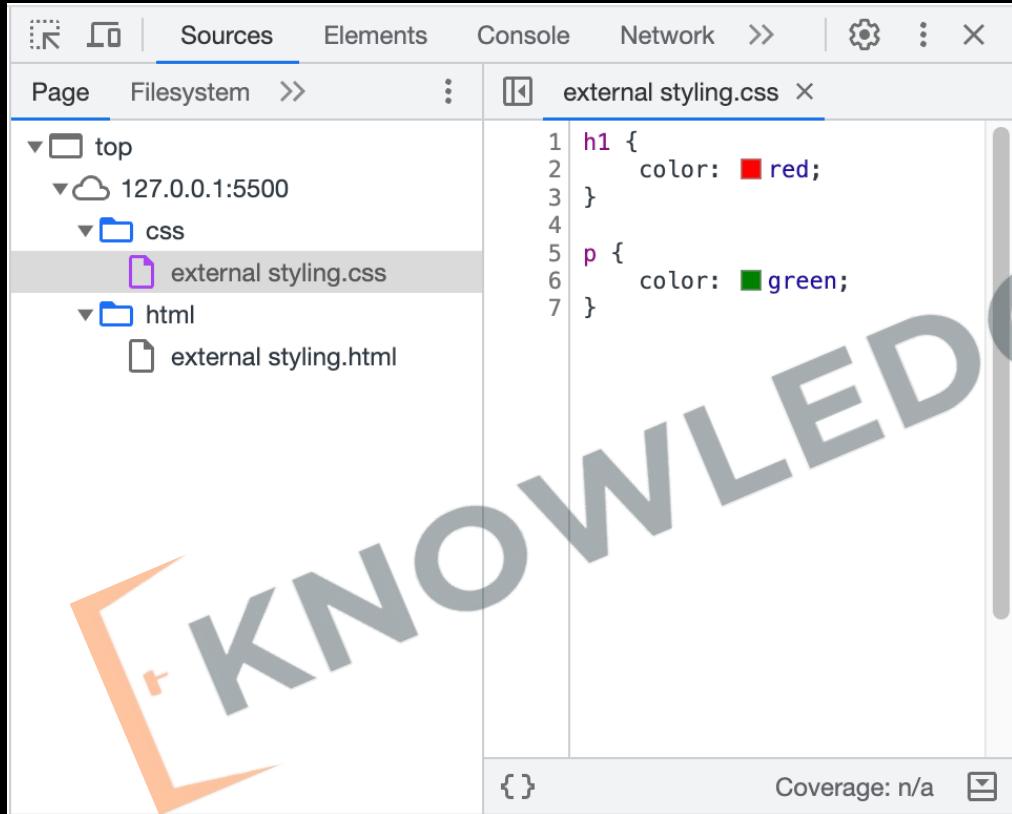
user agent stylesheet

element.style { }
h1 { color: red; }
h1 { display: block; font-size: 2em; margin-block-start: 0.67em; margin-block-end: 0.67em; margin-inline-start: 0px; margin-inline-end: 0px; font-weight: bold; }

Box Model diagram (margin): margin: 21.440; border: -; padding: -; size: 249.200x36.800; margin: 21.440.

- Styles Panel
- Box Model
- Changes happening only at client

7. Browser Tools (Source tab)



The screenshot shows the 'Sources' tab in the Chrome DevTools interface. On the left, the file tree displays 'top', '127.0.0.1:5500', 'css' (with 'external styling.css' selected), and 'html' (with 'external styling.html'). The main pane shows the contents of 'external styling.css':

```
1 h1 {  
2   color: red;  
3 }  
4  
5 p {  
6   color: green;  
7 }
```

At the bottom of the DevTools window, there is a status bar with the text 'Coverage: n/a'.

KNOWLEDGE GATE

7. Browser Tools (Network tab)

Sources Elements Console Network Performance Memory Application >

Preserve log Disable cache No throttling

Filter Invert Hide data URLs

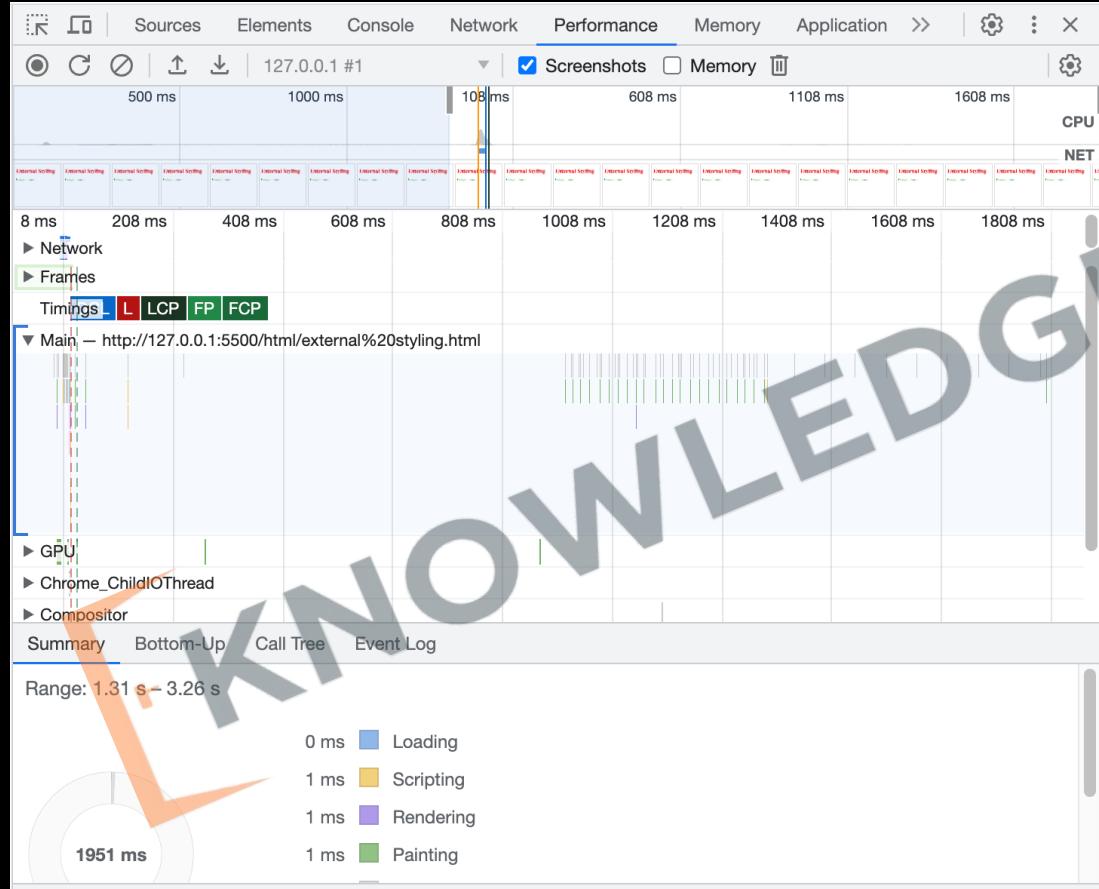
All | Fetch/XHR JS CSS Img Media Font Doc WS Wasm Manifest Other Has blocked cookies

Blocked Requests 3rd-party requests

10 ms 20 ms 30 ms 40 ms 50 ms 60 ms 70 ms 80 ms 90 ms

Name	Status	Type	Initiator	Size	Time	Waterfall
external%20styling.html	304	docu...	Other	295 B	4 ms	
ws	101	webso...	<u>external stylin...</u>	0 B	Pending	

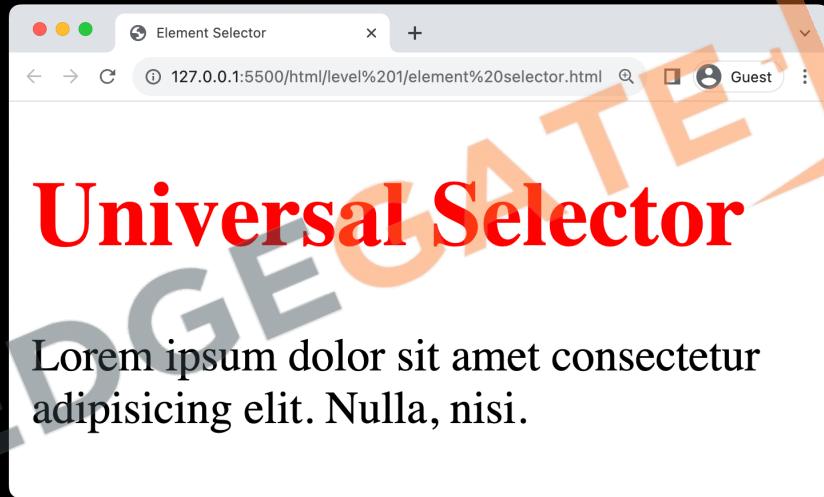
7. Browser Tools (Performance tab)



KNOWLEDGE GATE

8. Selectors (Element selector)

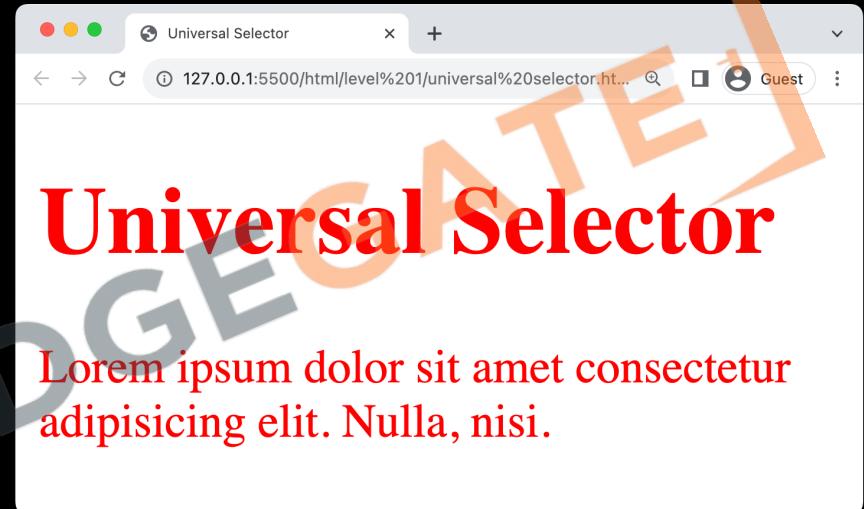
```
3 <head>
4     <title>Element Selector</title>
5     <style>
6         h1 {
7             color: red
8         }
9     </style>
10    </head>
11    <body>
12        <h1>Universal Selector</h1>
13        <p>Lorem ipsum dolor sit amet consectetur
14           adipisicing elit. Nulla, nisi.</p>
```



- **Targets Elements:** Selects HTML elements based on their **tag name**.
- **Syntax:** Simply use the **element's name**
- **Uniform Styling:** Helps in applying **consistent styles** to all instances.
- **Ease of Use:** Straightforward and **easy** to implement for basic styling.

8. Selectors (Universal selector)

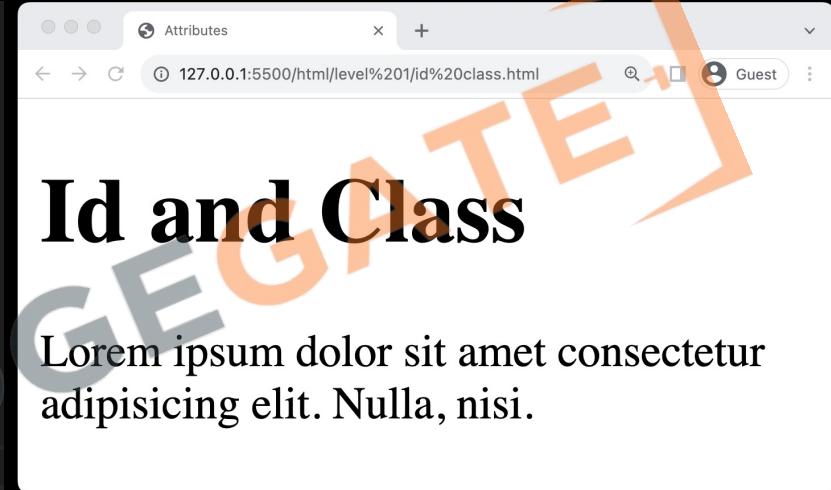
```
3 <head>
4     <title>Universal Selector</title>
5     <style>
6         * {
7             color: red
8         }
9     </style>
10    </head>
11    <body>
12        <h1>Universal Selector</h1>
13        <p>Lorem ipsum dolor sit amet consectetur
14            adipisicing elit. Nulla, nisi.</p>
```



- **Matches All:** Targets and styles **all elements** on a webpage.
- **Syntax:** Utilized as an **asterisk (*)**.
- **Resets Styles:** Commonly used to **reset margins** and paddings globally.
- **Broad Styling:** Useful for setting universal attributes like **font or color**.
- **Usage Caution:** Can cause style **conflicts** due to its wide-reaching effects.

8. Selectors (id & class property)

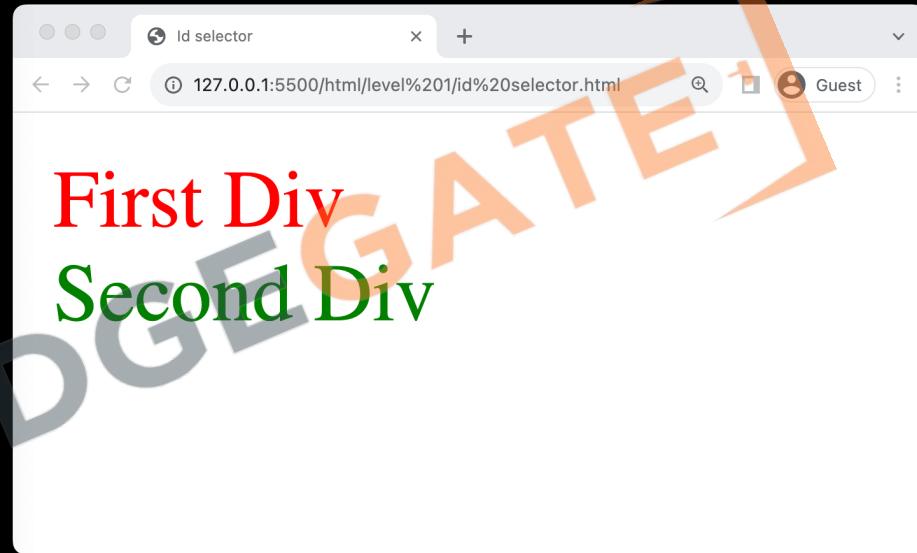
```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4  |   <title>Attributes</title>
5  </head>
6  <body>
7  |   <h1 id="top_heading">Id and Class</h1>
8  |   <p class="article">Lorem ipsum dolor sit amet
9  |       consectetur adipisicing elit. Nulla, nisi.</p>
9  </body>
10 </html>
```



- **ID Property:** Assigns a unique identifier to a single HTML element.
- **Class Property:** Allows grouping of multiple HTML elements to style them collectively.
- **Reusable Classes:** Class properties can be reused across different elements for consistent styling.
- **Specificity and Targeting:** Both properties assist in targeting specific elements or groups of elements for precise styling.

8. Selectors (Id selector)

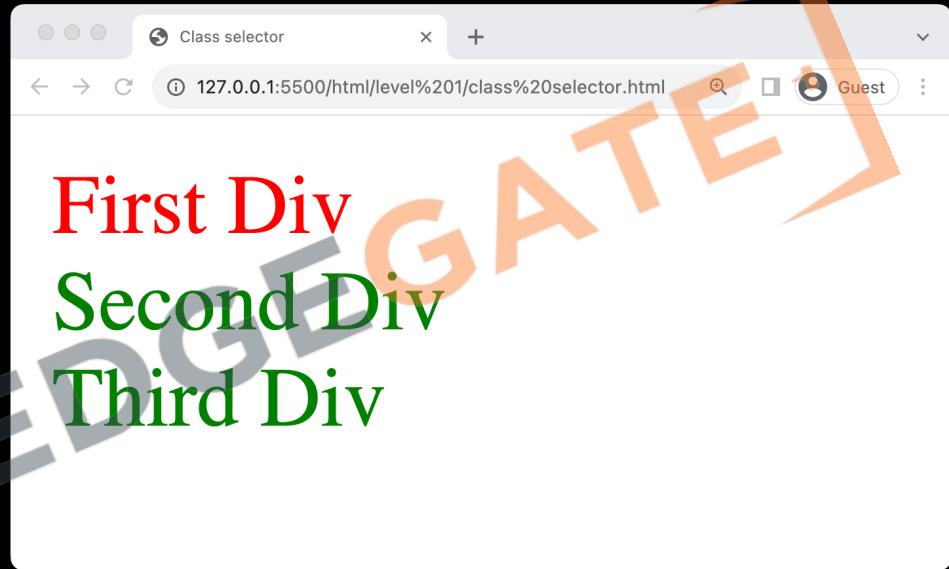
```
3 <head>
4     <title>Id selector</title>
5     <style>
6         #first { color: red; }
7         #second { color: green; }
8     </style>
9 </head>
10 <body>
11     <div id="first">First Div</div>
12     <div id="second">Second Div</div>
13 </body>
```



- **Unique Identifier:** Targets a specific element with a **unique ID** attribute.
- **Syntax:** Uses the **hash (#)** symbol
- **Single Use:** Each ID should be used **once per page** for uniqueness.
- **Specific Targeting:** Ideal for styling **individual, distinct** elements.

8. Selectors (Class selector)

```
<head>
    <title>Class selector</title>
    <style>
        #first { color: red; }
        .second { color: green; }
    </style>
</head>
<body>
    <div id="first">First Div</div>
    <div class="second">Second Div</div>
    <div class="second">Third Div</div>
</body>
```



- **Group Styling:** Allows styling of **multiple elements** grouped under a class.
- **Syntax:** Utilizes the **dot (.)** symbol.
- **Reusable:** Can be used on multiple elements for **consistent styling**.
- **Versatility:** Ideal for applying styles to a category of elements.

8. Selectors (Group selector)

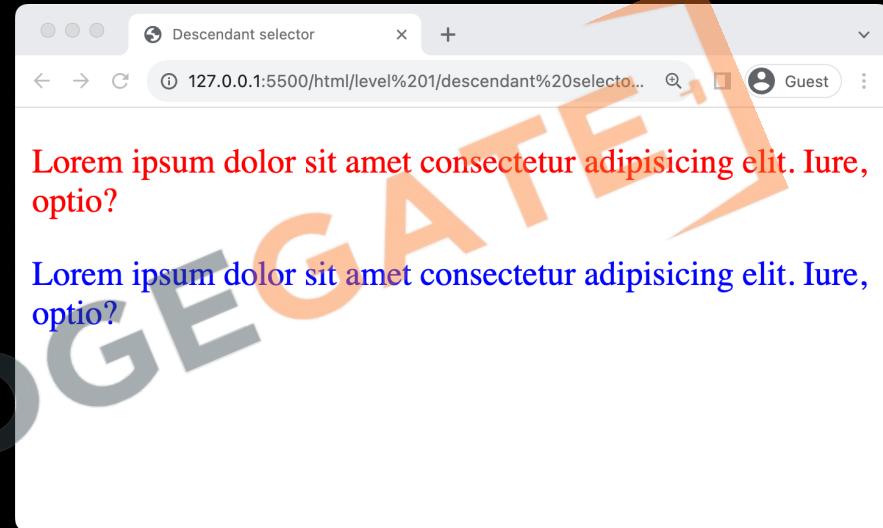
```
<head>
  <title>Group selector</title>
  <style>
    h1, h2, h3 {
      color: red
    }
  </style>
</head>
<body>
  <h1>Heading 1</h1>
  <h2>Heading 2</h2>
  <h3>Heading 3</h3>
  <h4>Heading 4</h4>
</body>
```



- **Multiple Elements:** Styles multiple elements simultaneously.
- **Syntax:** Separates selectors with commas.
- **Efficiency:** Reduces code redundancy and saves time.

8. Selectors (Descendant selector)

```
<head>
  <title>Descendant selector</title>
  <style>
    div p { color: red }
    p { color: blue }
  </style>
</head>
<body>
  <div>
    <p>
      Lorem ipsum dolor sit amet consectetur adipisicing elit. Iure, optio?
    </p>
  </div>
  <p>
    Lorem ipsum dolor sit amet consectetur adipisicing elit. Iure, optio?
  </p>
</body>
```



- **Nested Targeting:** Styles elements **nested** within a specified element.
- **Syntax:** Separate selectors with **spaces**.
- **Hierarchy-Based:** Allows styling based on the **hierarchical** structure of HTML.
- **Specific Styling:** Facilitates more **targeted** and specific styling of elements.

Level 1 Revision

CSS Basics

1. Basic Syntax
2. Color Property
3. Including Styles
4. HTML Refresher
5. Comments
6. MDN Documentation
7. Browser Tools
8. Selectors

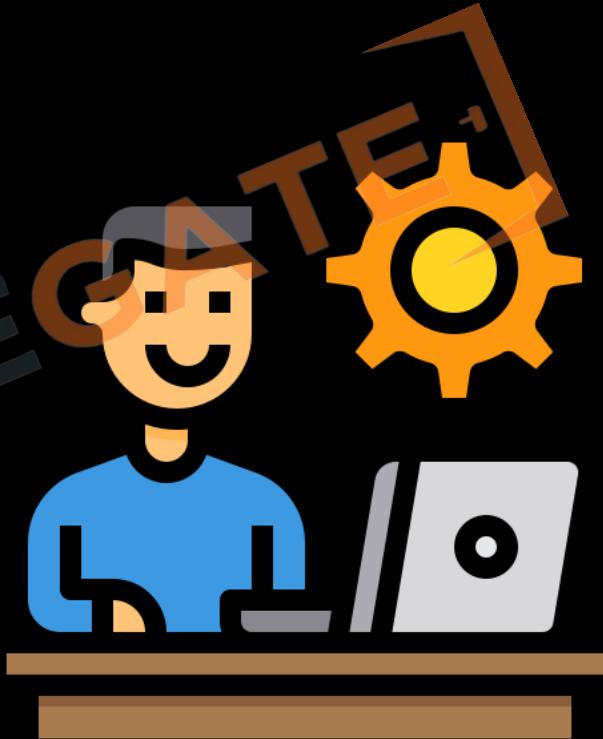


DYNAMIC KNOWLEDGE

Practice Set Level 1

CSS Basics

- Create a heading and set the text color red.
- Create a div with id `#heading`, include CSS using all 3 ways `line`, `style tag` and `external`, and observe priority.
- Add `comments` to your `CSS` class
- Create a `div`, `paragraph` and `heading` and use id Selector, element selector and class selector for them.
- Create two divs with id `first` and `second` and define color for both using group selector.



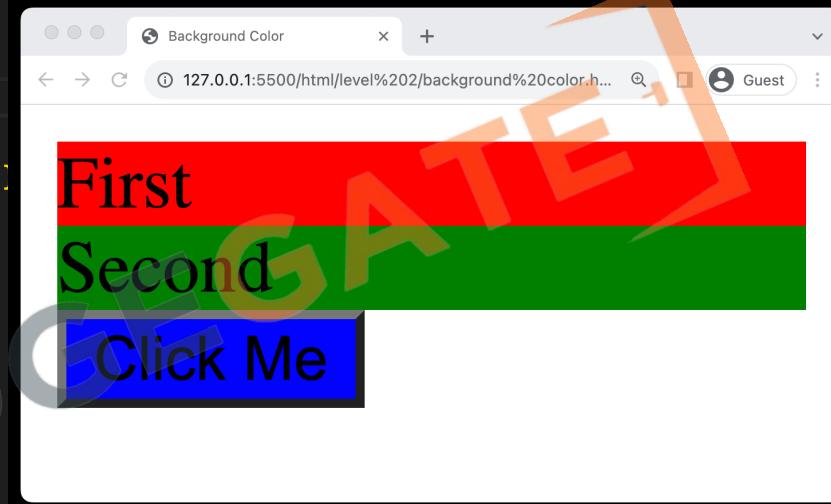
Level 2

Color System & Background

- 9. **Background-Color Property**
- 10. **Color System**
- 11. **Absolute Units**
- 12. **Height & Width Property**
- 13. **Background-Image Property**
- 14. **Visibility Property**

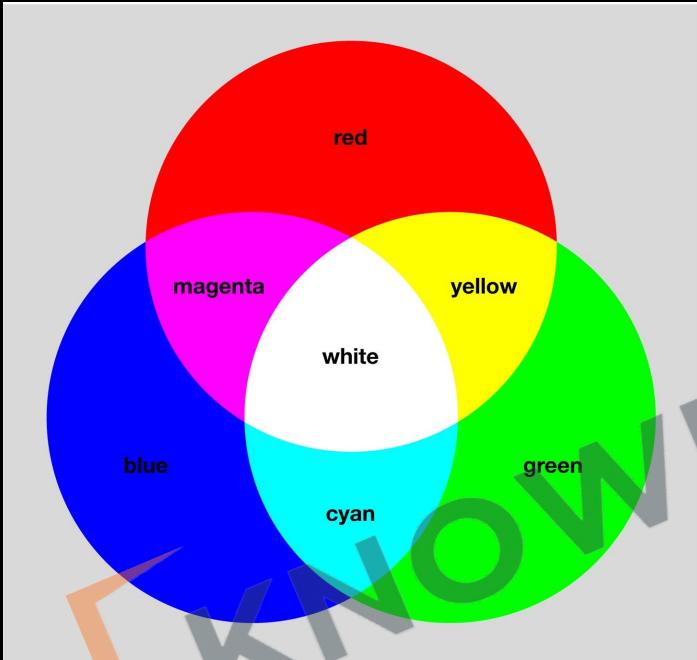
9. Background Color

```
<head>
    <title>Background Color</title>
    <style>
        #first { color: black; background-color: red; }
        #second { color: black; background-color: green; }
        button { color: black; background-color: blue; }
    </style>
</head>
<body>
    <div id="first">First</div>
    <div id="second">Second</div>
    <button>Click Me</button>
</body>
```



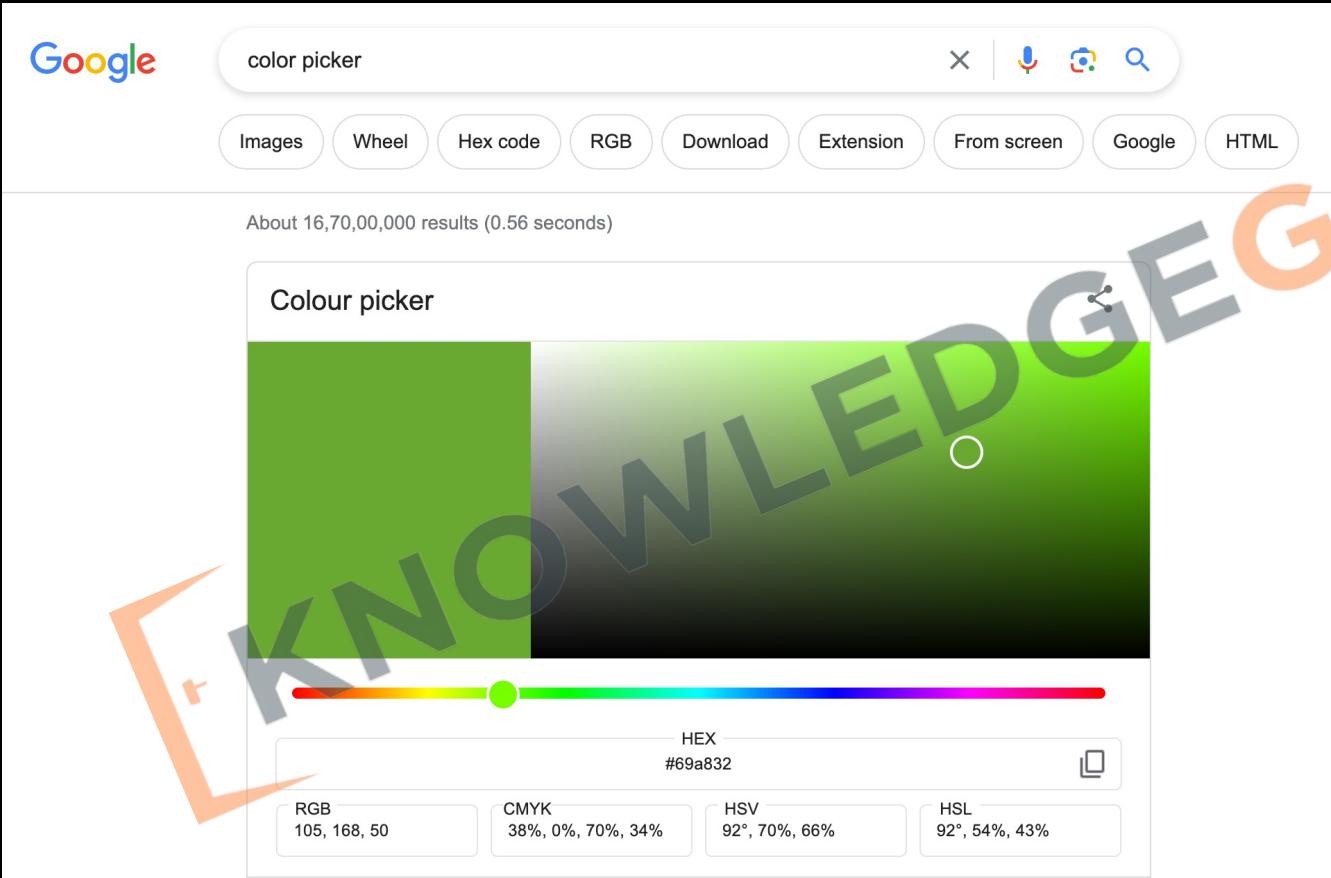
- **Definition:** Sets the background color of an element.
- **Syntax:** Utilized as `background-color: color;`
- **Visual Appeal:** Enhances the visual appeal and contrast of webpage elements.

10. Color System (Color Theory)



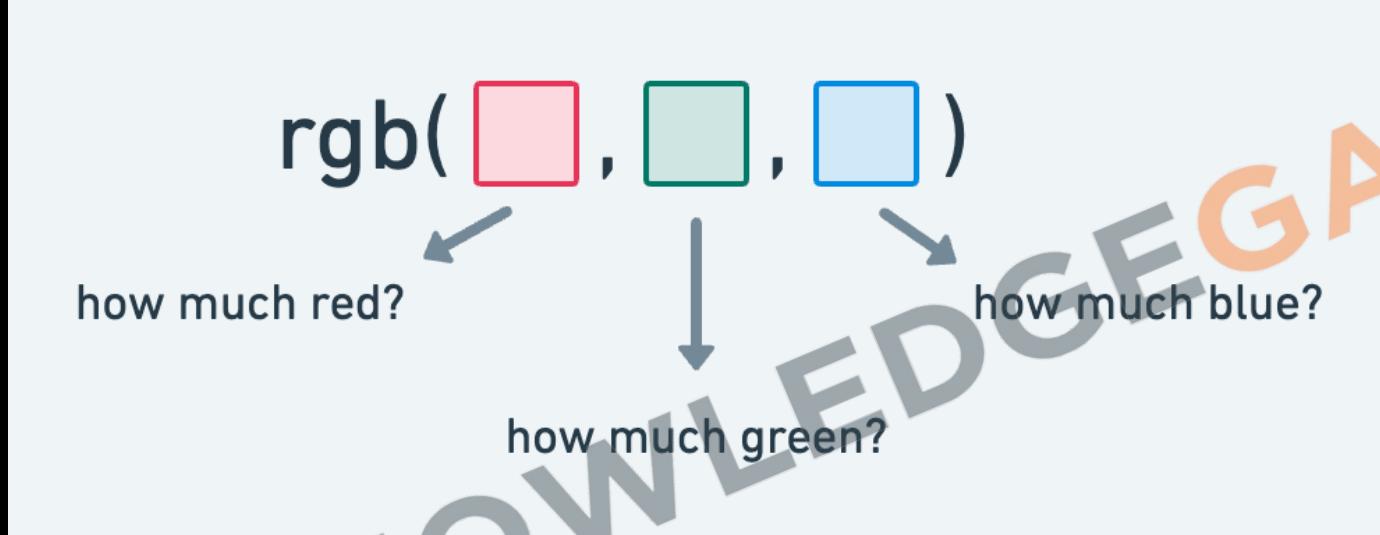
- **RGB Model:** Creates colors by mixing Red (R), Green (G), and Blue (B) light sources.
- **Additive Model:** More light means increased brightness.
- **Primary Colors:** R, G, and B are the foundational colors.
- **White & Black:** All combined yield white; absence equals black.
- **Color Depth:** Allows for millions of color variations.

10. Color System (color picker)



KNOWLEDGE

10. Color System (RGB Color Model)



- **Three Channels:** Consists of Red (R), Green (G), and Blue (B) channels to create a variety of colors.
- **Syntax:** Utilized as `rgb(r, g, b)` where r, g, and b are values between 0 and 255.

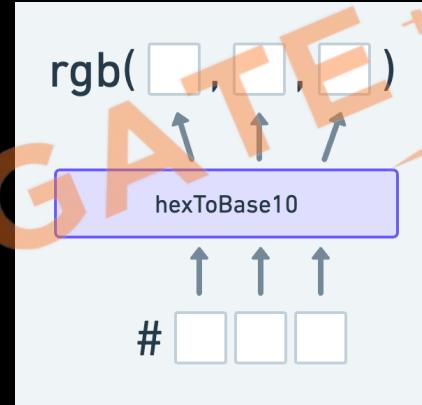
10. Color System (RGB Color Model)

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>RGB Color</title>
</head>
<body>
    <div style="background-color: red; color: black; padding: 10px; margin-bottom: 5px;">First</div>
    <div style="background-color: green; color: black; padding: 10px; margin-bottom: 5px;">Second</div>
    <div style="background-color: blue; color: black; padding: 10px; margin-bottom: 5px;">Third</div>
    <div style="background-color: #1e8449; color: white; padding: 10px; margin-bottom: 5px;">Fourth</div>
</body>
</html>
```



10. Color System (HEX Color Model)

- **Hexadecimal Codes:** Represents colors using hexadecimal values, consisting of **6 digits** combined from numbers and letters (A-F).
- **Syntax:** Written as **#RRGGBB**
- **Easy Color Matching:** Facilitates easy color matching with graphic design tools and branding colors.
- **Web Standards:** Widely supported and a common standard for defining colors in web design



10. Color System (HEX Color Model)

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Hex Color</title>
</head>
<body>
    <div style="background-color: #ff0000">First</div>
    <div style="background-color: #00ff00">Second</div>
    <div style="background-color: #0000ff">Third</div>
    <div style="background-color: #402ae9">Fourth</div>
</body>
</html>
```



10. Color System (Alpha Channel)

- **RGBA:** RGB's extension, includes alpha for opacity control (0-1 range).
- **Transparency Control:** Facilitates the adjustment of transparency levels in colors.
- **Visual Effects:** Enables the creation of visual effects like shadows and overlays.
- **Layering:** Assists in layering elements with varying degrees of visibility.



10. Color System (Alpha Channel)

```
<h1 style="color: □rgb(255,0,0,0.1);">First</div>
<h1 style="color: □rgb(255,0,0,0.25);">First</div>
<h1 style="color: □rgb(255,0,0,0.5);">First</div>
<h1 style="color: □rgb(255,0,0,0.75);">First</div>
<h1 style="color: □rgb(255,0,0,1.0);">First</div>
```



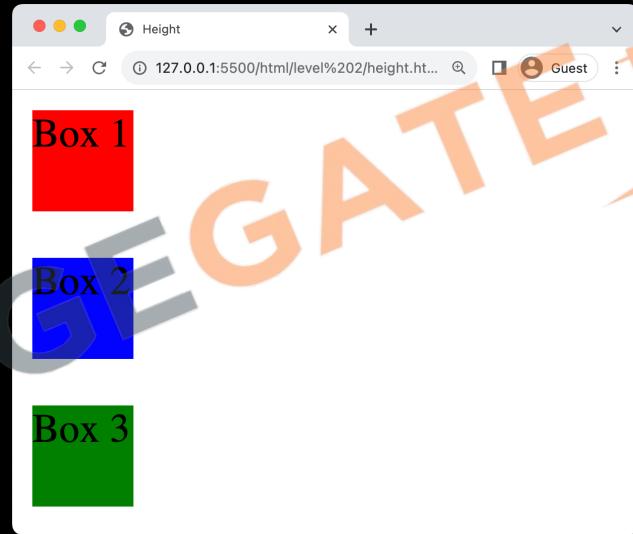
11. Absolute Units

- **Definition:** Pixels (px) are fixed-size units, representing a dot on a computer screen.
- **Precision:** Allows for precise control over element dimensions.
- **Graphics & Web Design:** Commonly used in graphics and web design for setting font sizes, margins, and more.
- **Cross-Browser Consistency:** Provides consistency across different browsers.
- **High-DPI Displays:** Can vary in appearance on high-DPI (dots per inch) displays.



12. Height & Width Property

```
<head>
  <title>Height</title>
  <style>
    .box { height: 40px; width: 40px; }
    #box1 {background-color: red;}
    #box2 {background-color: blue;}
    #box3 {background-color: green;}
  </style>
</head>
<body>
  <div id="box1" class="box">Box 1</div> <br>
  <div id="box2" class="box">Box 2</div> <br>
  <div id="box3" class="box">Box 3</div>
</body>
```



- **Dimensions Control:** Used to specify the **height** and **width** of elements.
- **Unit Variability:** Can use units like pixels (px)
- **Box Model Component:** Influences padding, border, and margin.
- **Min and Max Values:** Can utilize min-height, max-height, min-width, and max-width to set restrictions on dimensions.

13. Background image Property

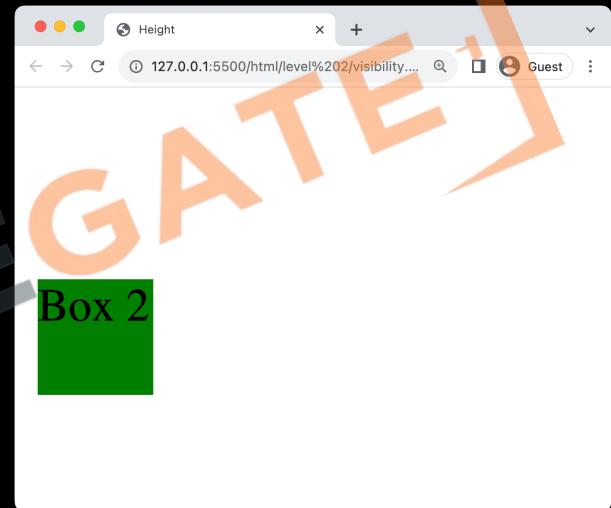
- **Usage:** Adds an **image** as a background to elements.
- **Syntax:** Defined using **background-image:**
`url('path/to/image');`
- **Repetition:** Control image repetition using **background-repeat**.
- **Positioning:** Adjust image position using **background-position**.
- **Size Control:** Manipulate image size using **background-size**.
- **Background-Attachment:** Sets whether the background image **scrolls** with the element or remains fixed.
- Shorthand (**color, image, repeat, attachment, position**)

```
<head>
  <title>Background Image</title>
  <style>
    #box1 {
      background-image: url(../../images/css.png);
      height: 500px;
      width: 500px;
    }
  </style>
</head>
<body>
  <div id="box1"></div>
</body>
```



14. Visibility Property

```
<head>
  <title>Visibility</title>
  <style>
    .box { height: 40px; width: 40px; }
    #box1 {background-color: red; visibility: hidden;}
    #box2 {background-color: green; visibility: visible;}
  </style>
</head>
<body>
  <div id="box1" class="box">Box 1</div> <br>
  <div id="box2" class="box">Box 2</div> <br>
</body>
```



- **Usage:** Controls the **visibility** of elements without changing the layout.
- **Values:** Can take visible, hidden, or collapse as values.
- **Space Occupancy:** Even when hidden, the element **occupies space**.
- **Interactivity:** Hidden elements are not accessible to **user interactions**.

Level 2 Revision

Color System & Background

9. Background-Color Property
10. Color System
11. Absolute Units
12. Height & Width Property
13. Background-Image Property
14. Visibility Property



Practice Set Level 2

Color System & Background

- Create a div bar with text and background color with opacity
- Change the color of the main content
- Add background image to one div
- Use background shorthand property



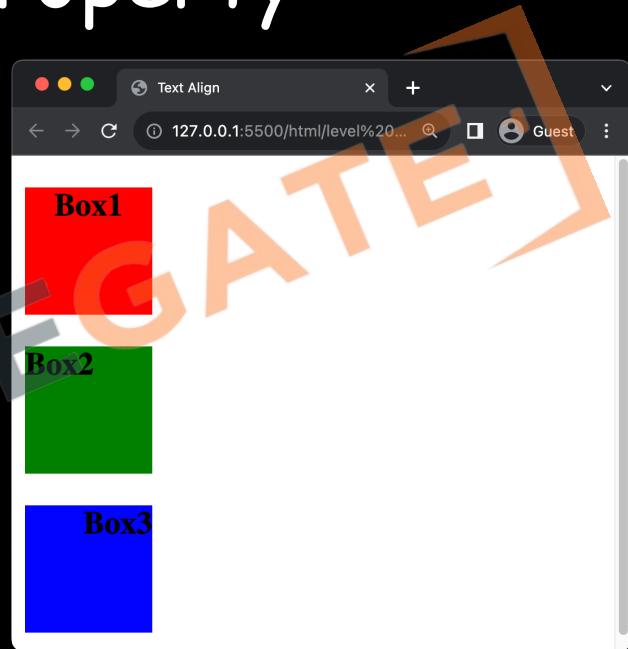
Level 3

Text Properties

- 15. **Text-Align** Property
- 16. **Text-Decoration** Property
- 17. **Text-Transform** Property
- 18. **Line-Height** Property
- 19. **Font** Properties
- 20. **Font** Family
- 21. **Icon** using fonts

15. Text-Align Property

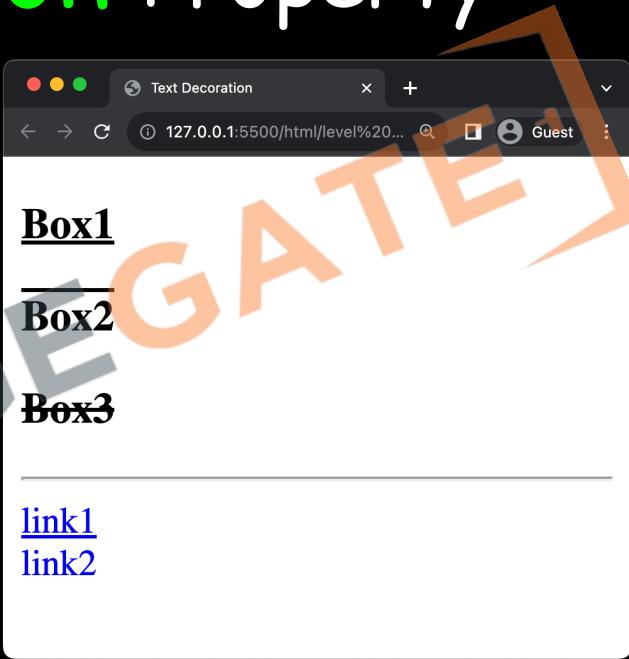
```
<head>
  <title>Text Align</title>
  <style>
    .box {height: 100px; width: 100px;}
    #box1 {background-color: red; text-align: center;}
    #box2 {background-color: green; text-align: left;}
    #box3 {background-color: blue; text-align: right;}
  </style>
</head>
<body>
  <h3 id="box1" class="box">Box1</h3>
  <h3 id="box2" class="box">Box2</h3>
  <h3 id="box3" class="box">Box3</h3>
</body>
```



- **Usage:** Controls the horizontal alignment of text within an element.
- **Values:** Can take values like left, right, center, and justify.
- **Visual Appeal:** Enhances readability and visual appeal by organizing text neatly.

16. Text-Decoration Property

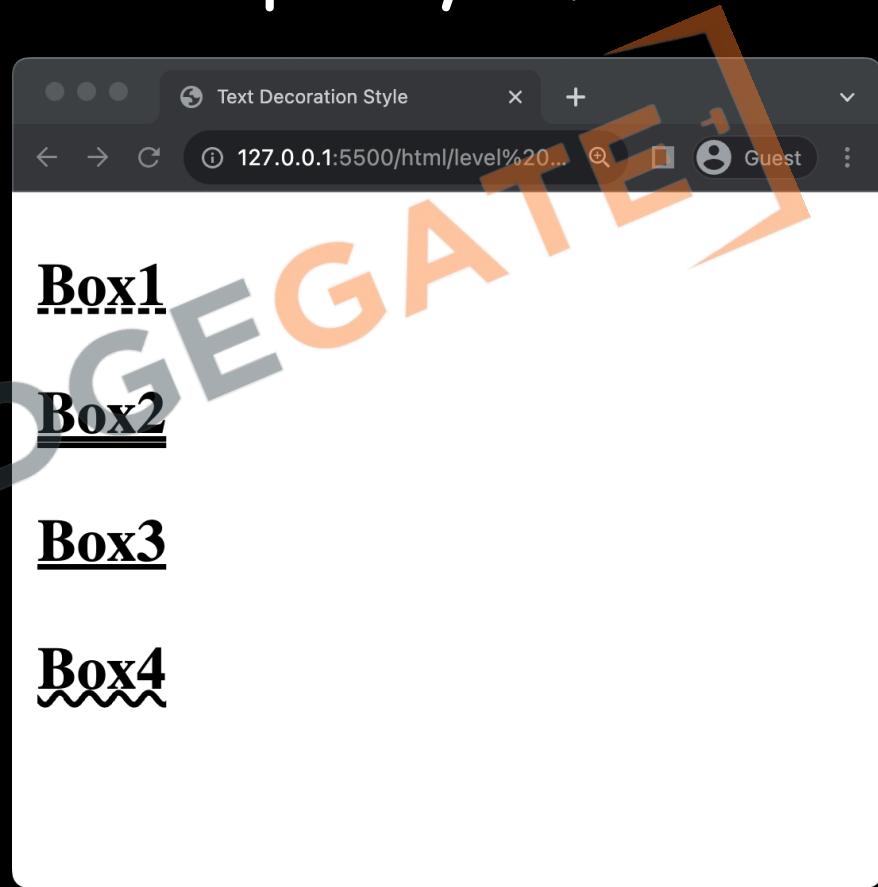
```
<head>
    <title>Text Decoration</title>
    <style>
        #box1 {text-decoration: underline;}
        #box2 {text-decoration: overline;}
        #box3 {text-decoration: line-through;}
    </style>
</head>
<body>
    <h3 id="box1" class="box">Box1</h3>
    <h3 id="box2" class="box">Box2</h3>
    <h3 id="box3" class="box">Box3</h3> <hr>
    <a href="#">link1</a> <br>
    <a href="#" style="text-decoration: none;">link2</a>
</body>
```



- **Usage:** Modifies the appearance of inline text.
- **Values:** Options include **none**, **underline**, **overline**, and **line-through**.
- **Hyperlinks:** Commonly used to remove underlines from hyperlinks for aesthetic purposes.

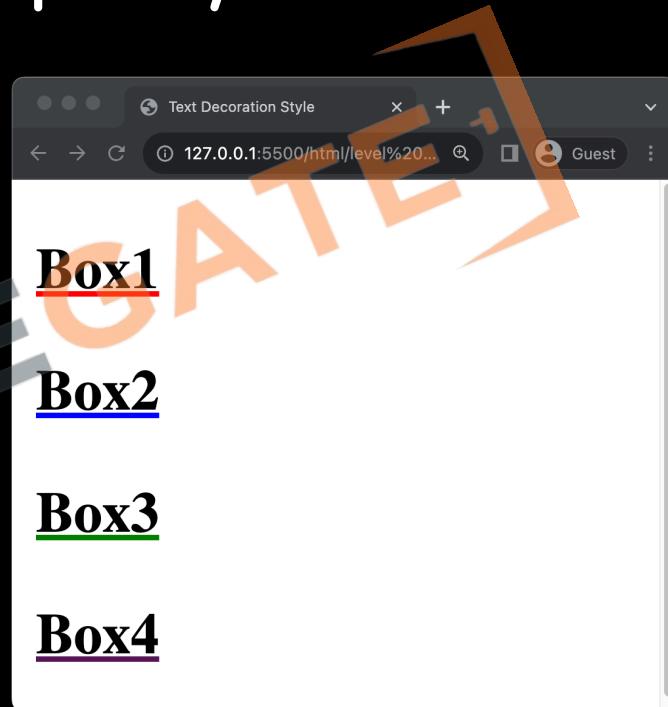
16. Text-Decoration Property (style)

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Text Decoration Style</title>
  <style>
    .box {text-decoration: underline;}
    #box1 {text-decoration-style: dashed;}
    #box2 {text-decoration-style: double;}
    #box3 {text-decoration-style: solid;}
    #box4 {text-decoration-style: wavy;}
  </style>
</head>
<body>
  <h3 id="box1" class="box">Box1</h3>
  <h3 id="box2" class="box">Box2</h3>
  <h3 id="box3" class="box">Box3</h3>
  <h3 id="box4" class="box">Box4</h3>
</body>
</html>
```



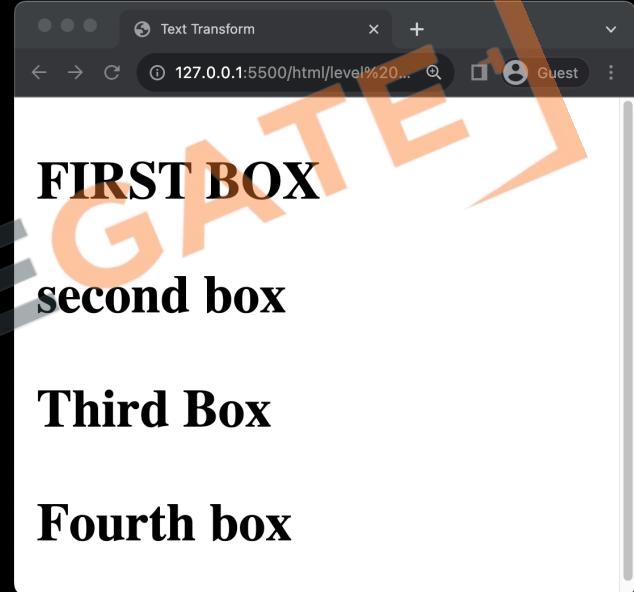
16. Text-Decoration Property (color)

```
<head>
  <title>Text Decoration Style</title>
  <style>
    .box {text-decoration: underline;}
    #box1 {text-decoration-color: red;}
    #box2 {text-decoration-color: blue;}
    #box3 {text-decoration-color: green;}
    #box4 {text-decoration-color: purple; color: purple; font-weight: bold;}
  </style>
</head>
<body>
  <h3 id="box1" class="box">Box1</h3>
  <h3 id="box2" class="box">Box2</h3>
  <h3 id="box3" class="box">Box3</h3>
  <h3 id="box4" class="box">Box4</h3>
</body>
```



17. Text-Transform Property

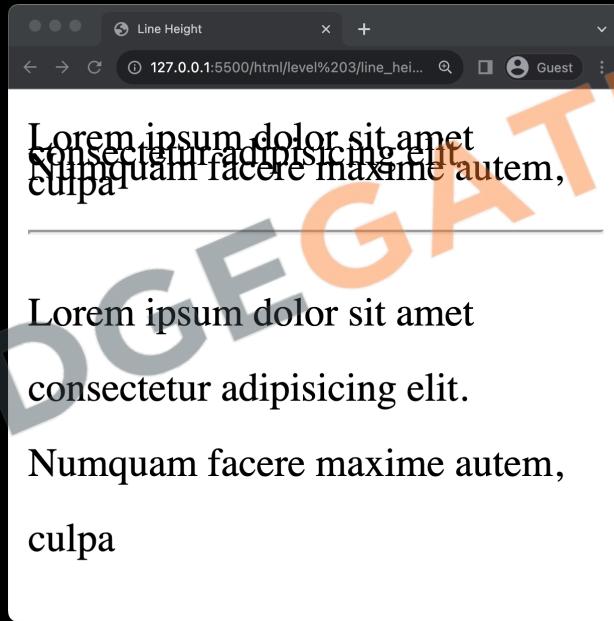
```
<head>
  <title>Text Transform</title>
  <style>
    #box1 {text-transform: uppercase;}
    #box2 {text-transform: lowercase;}
    #box3 {text-transform: capitalize;}
    #box4 {text-transform: none;}
  </style>
</head>
<body>
  <h3 id="box1" class="box">First box</h3>
  <h3 id="box2" class="box">Second box</h3>
  <h3 id="box3" class="box">Third box</h3>
  <h3 id="box4" class="box">Fourth box</h3>
</body>
```



- **Usage:** Controls the **capitalization** of text.
- **Common Values:** Can be **uppercase**, **lowercase**, or **capitalize**.
- **None Value:** **none** value **disables** text transformations.
- **Typography:** Useful for setting text style and improving **typography** in web design.

18. Line Height

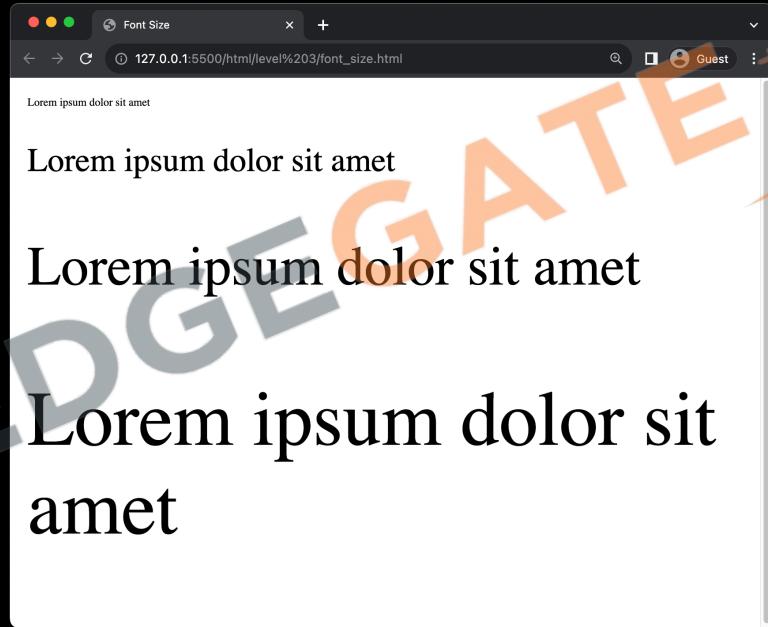
```
<head>
    <title>Line Height</title>
    <style>
        #first { line-height: 6px; }
        #second { line-height: 30px; }
    </style>
</head>
<body>
    <p id="first">Lorem ipsum dolor sit amet
    consectetur adipisicing elit. Numquam facere
    maxime autem, culpa</p> <hr>
    <p id="second">Lorem ipsum dolor sit amet
    consectetur adipisicing elit. Numquam facere
    maxime autem, culpa</p>
</body>
```



- **Usage:** Adjusts the amount of **space above and below** inline elements.
- **Readability:** Enhances text readability by preventing overcrowding.
- **Vertical Spacing:** Useful for **controlling vertical spacing** between lines of text.

19. Font Property (font-size)

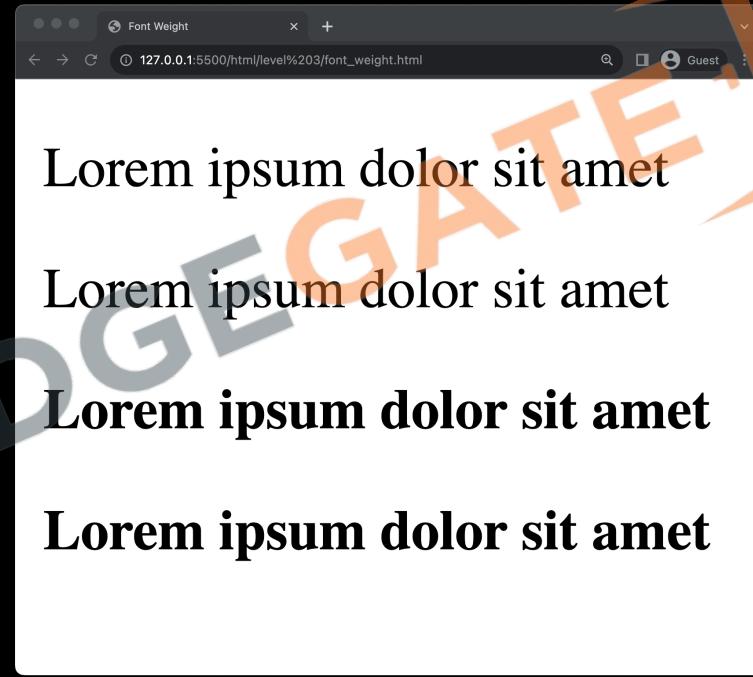
```
<head>
  <title>Font Size</title>
  <style>
    #first {font-size: 5px;}
    #second {font-size: 15px;}
    #third {font-size: 25px;}
    #fourth {font-size: 35px;}
  </style>
</head>
<body>
  <p id="first">Lorem ipsum dolor sit amet</p>
  <p id="second">Lorem ipsum dolor sit amet</p>
  <p id="third">Lorem ipsum dolor sit amet</p>
  <p id="fourth">Lorem ipsum dolor sit amet</p>
</body>
```



- **Usage:** Sets the **size of the font** in web content.
- **Responsiveness:** Helps in creating **responsive designs** adaptable to various screen sizes.
- **Readability:** Crucial for ensuring the **readability** of text on websites.

19. Font Property (font-weight)

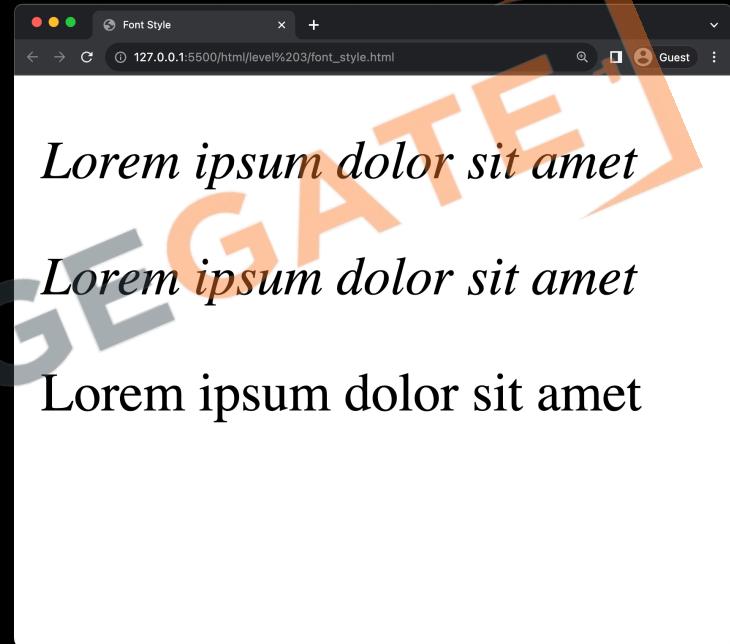
```
<head>
  <title>Font Weight</title>
  <style>
    #first {font-weight: 100;}
    #second {font-weight: 400;}
    #third {font-weight: 600;}
    #fourth {font-weight: 900;}
  </style>
</head>
<body>
  <p id="first">Lorem ipsum dolor sit amet</p>
  <p id="second">Lorem ipsum dolor sit amet</p>
  <p id="third">Lorem ipsum dolor sit amet</p>
  <p id="fourth">Lorem ipsum dolor sit amet</p>
</body>
```



- **Usage:** Defines the **thickness** of characters in a font.
- **Values:** Can take values like **normal**, **bold**, **bolder**, or numeric values (**100 to 900**).
- **Text Emphasis:** Utilized to emphasize text or **create contrast** between text elements.

19. Font Property (font-style)

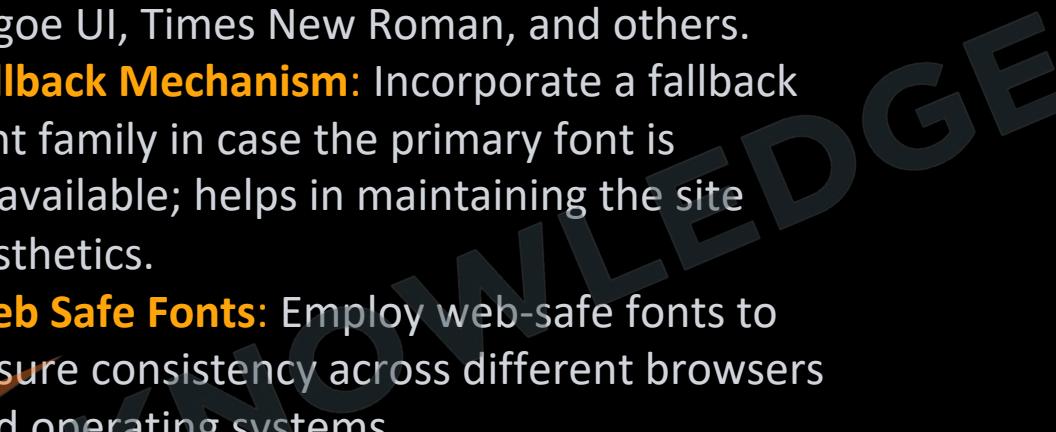
```
<head>
    <title>Font Style</title>
    <style>
        #first {font-style: italic;}
        #second {font-style: oblique;}
        #third {font-style: normal;}
    </style>
</head>
<body>
    <p id="first">Lorem ipsum dolor sit amet</p>
    <p id="second">Lorem ipsum dolor sit amet</p>
    <p id="third">Lorem ipsum dolor sit amet</p>
</body>
```



- **Usage:** Controls the **style** of the font, mainly affecting its inclination.
- **Values:** Common values are **normal**, **italic**, and **oblique**.
- **Text Formatting:** Useful for **highlighting** or distinguishing certain text segments.

20. Font Family

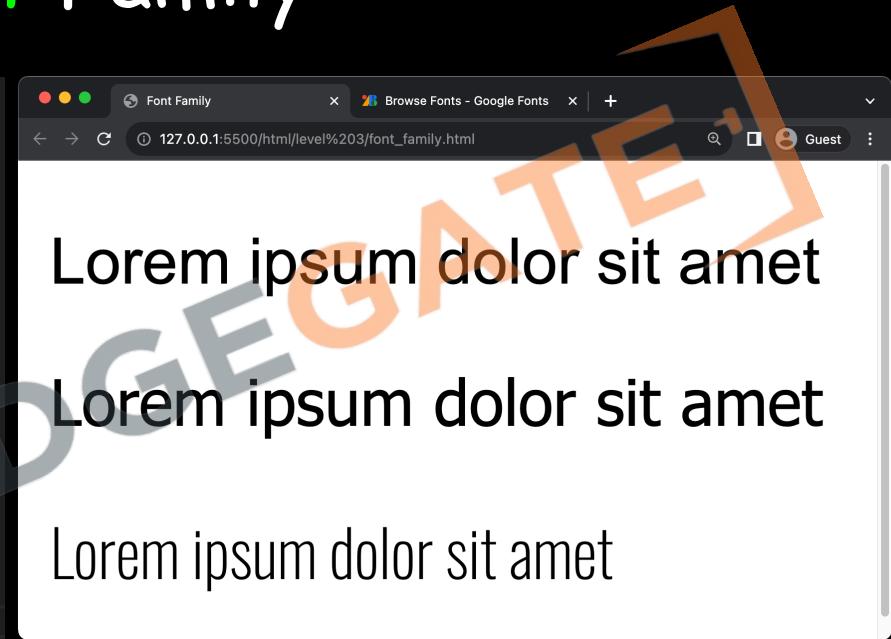
- **Usage:** Defines which font should be used for text within an element.
- **Specific Fonts:** Common choices include Arial, Segoe UI, Times New Roman, and others.
- **Fallback Mechanism:** Incorporate a fallback font family in case the primary font is unavailable; helps in maintaining the site aesthetics.
- **Web Safe Fonts:** Employ web-safe fonts to ensure consistency across different browsers and operating systems.
- **Generic Family:** Always end the font family list with a generic family like serif or sans-serif as a last resort option.



Arial Narrow
Book Antiqua
Cambria
Century Gothic
Consolas
COPPERPLATE
Georgia
Impact
Lucida Sans Unicode
Papyrus
Script MT Bold
Tahoma
Times New Roman
Verdana

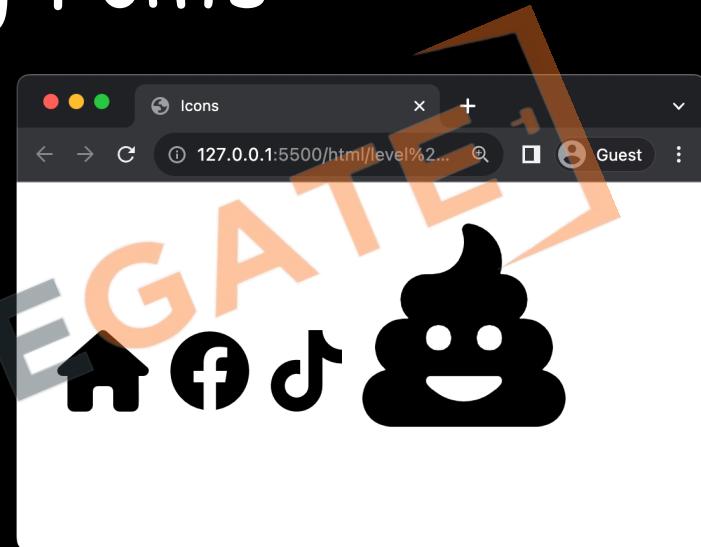
20. Font Family

```
<head>
  <title>Font Family</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=Oswald:wght@200&
display=swap" rel="stylesheet">
  <style>
    #first {font-family: Arial, Helvetica, sans-serif;}
    #second {font-family: 'Segoe UI', Tahoma, Geneva, Verdana,
    sans-serif;}
    #third {font-family: 'Oswald', sans-serif;}
  </style>
</head>
<body>
  <p id="first">Lorem ipsum dolor sit amet</p>
  <p id="second">Lorem ipsum dolor sit amet</p>
  <p id="third">Lorem ipsum dolor sit amet</p>
</body>
```



21. Icons using Fonts

```
<head>
  <title>Icons</title>
  <script src="https://kit.fontawesome.com/43290fa92d.js" crossorigin="anonymous"></script>
</head>
<body>
  <i class="fa-solid fa-house"></i>
  <i class="fa-brands fa-facebook"></i>
  <i class="fa-brands fa-tiktok"></i>
  <i class="fa-solid fa-poo" style="font-size: 40px;"></i>
</body>
```



Using <https://fontawesome.com>

Level 3 Revision

Text Properties

15. **Text-Align** Property
16. **Text-Decoration** Property
17. **Text-Transform** Property
18. **Line-Height** Property
19. **Font** Properties
20. **Font Family**
21. **Icon** using fonts



Practice Set Level 3

Text Properties

- Create an **Heading** at the centre and make capitalized
- Use Font family for the whole page to **TimeNewRoman**
- Create one div inside another div. Set id and text **outer** to outer div, set id and text **inner** to inner div. Set outer div text size to 25px. Set inner to 10px.
- Use icons from **fontawesome.com** and use icons of **LinkedIn** and **Github**



Level 4

Box Model

- 22. What is Box Model
- 23. Padding Property
- 24. Margin Property
- 25. Border Property

22. What is Box Model

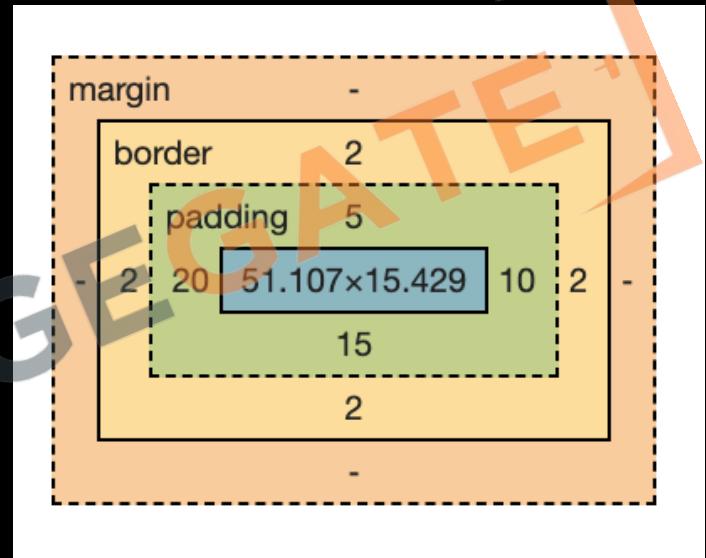
मिठाई भण्डार पे चलो

- **Core Concept:** Central concept in CSS that outlines the design and layout of elements on the web page.
- **Components:** Consists of four main components - **margin**, **border**, **padding**, and **content**.
- **Margin:** The space outside the border, separating the element from others.
- **Border:** The outline that encapsulates the padding and content.
- **Padding:** The space between the border and the actual content, providing a buffer.
- **Content:** The innermost layer where text, images, or other media are housed.



23. Padding Property

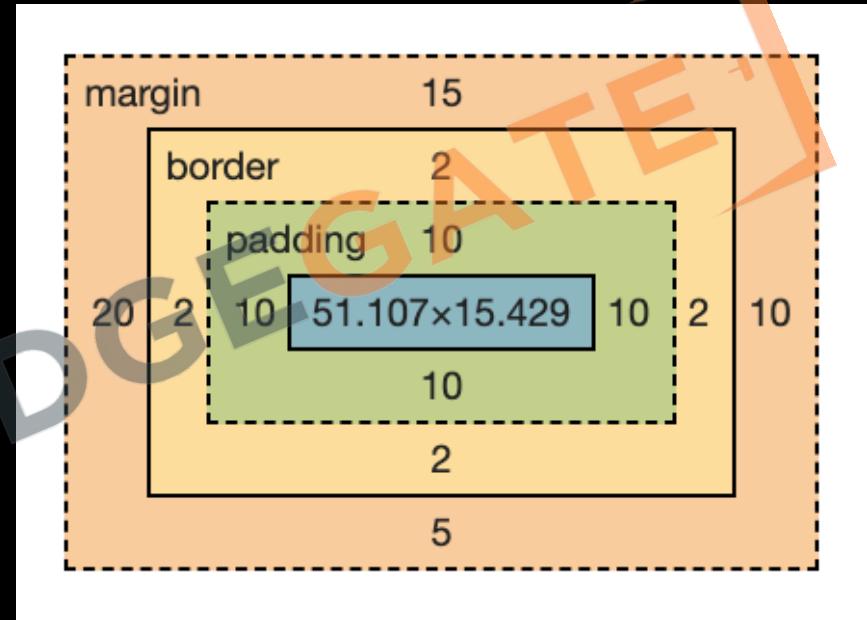
```
<head>
  <title>Padding</title>
  <style>
    * { margin: 0; padding: 0; }
    #button1 {
      padding: 5px 10px 15px 20px;
      background-color: aquamarine;
    }
  </style>
</head>
<body>
  <button id="button1">Click Me</button>
</body>
```



- Usage:** Defines the space between the content of an element and its border.
- Individual Sides:** Allows setting padding for individual sides using `padding-top`, `padding-right`, `padding-bottom`, and `padding-left`.
- Shorthand:** Can use shorthand property `padding` to set all sides at once, e.g., `padding: 10px 20px 10px 20px`.

24. Margin Property

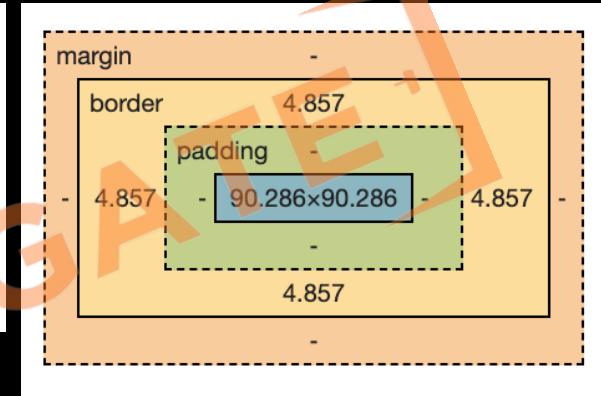
```
<head>
    <title>margin</title>
    <style>
        * { margin: 0; padding: 0; }
        #button1 {
            margin: 15px 10px 5px 20px;
            padding: 10px;
            background-color: aquamarine;
        }
    </style>
</head>
<body>
    <button id="button1">Click Me</button>
</body>
```



- **Functionality:** Sets the space around elements, separating them from others.
- **Individual Sides:** Customizable for **top**, **right**, **bottom**, and **left** sides.
- **Shorthand:** Allows quick setup, e.g., `margin: 10px 20px.` (clockwise)
- **Auto Value:** Can be used for central alignment with `auto` value.

24. Border Property

```
<head>
    <title>Border</title>
    <style>
        * { margin: 0; padding: 0; }
        #button1 {
            height: 100px; width: 100px;
            border: 5px dashed black;
            background-color: aquamarine;
        }
    </style>
</head>
<body>
    <button id="button1">Click Me</button>
</body>
```



- **Usage:** Creates an outline around **HTML** elements.
- **Components:** Defined by **width**, **style**, and **color** attributes.
- **Styles:** Includes options like **solid**, **dashed**, and **dotted**.
- **Shorthand:** Can set attributes at once, e.g., **border: 2px solid black**.

24. Border Property (border radius)

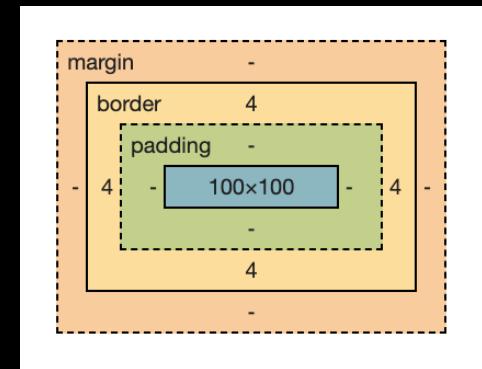
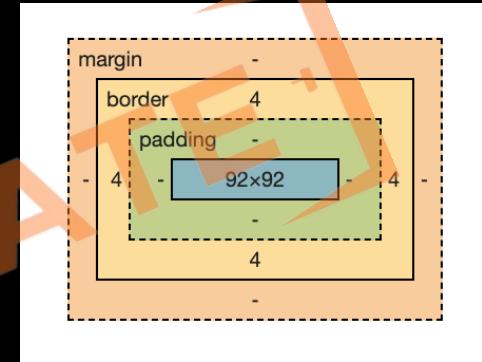
```
<head>
  <title>Border Radius</title>
  <style>
    * { margin: 0; padding: 0; }
    #button1 {
      height: 100px; width: 100px;
      border: 5px solid black;
      border-radius: 30px;
      background-color: aquamarine;
    }
  </style>
</head>
<body>
  <button id="button1">Click Me</button>
</body>
```



- **Usage:** Used to **create rounded corners** for elements.
- **Individual Corners:** Allows setting different radii for each corner.
- **Shorthand:** e.g., `border-radius: 10px 20px`.

24. Border Property (box sizing)

```
<head>
  <title>Box Sizing</title>
  <style>
    * { margin: 0; padding: 0; }
    button {
      height: 100px; width: 100px;
      border: 4px solid black;
      background-color: aquamarine;
    }
    #button1 { box-sizing: border-box; }
    #button2 { box-sizing: content-box; }
  </style>
</head>
<body>
  <button id="button1">Click Me</button>
  <button id="button2">Click Me</button>
</body>
```



Level 4 Revision

Box Model

- 22. What is Box Model
- 23. Padding Property
- 24. Margin Property
- 25. Border Property



Practice Set Level 4

Box Model

- Create a div with height and width **200px** and observe different values in the box model by inspecting.
- Create a **button** and give **10px padding** to top and bottom and **15px** on the sides.
- Add **100px margin** to the button on all sides.
- Add dotted border with color **red** and so much that the button becomes a **circle**.
- Create **two boxes** with different box-sizing values and observe changes in box model.

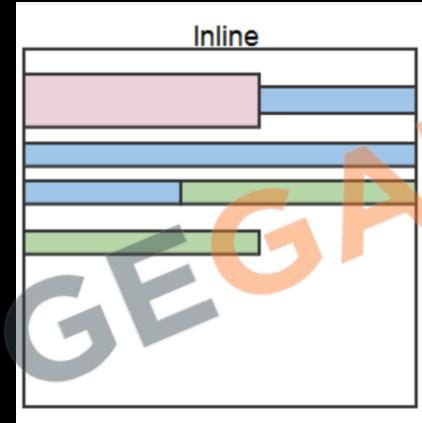
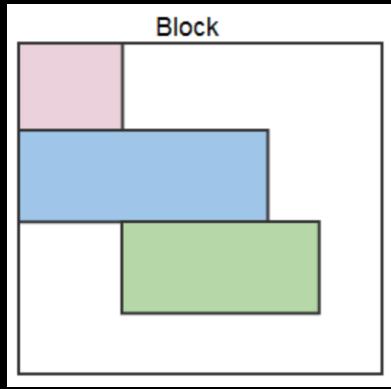


Level 5

Display and Position

- 26. **Display** Property
- 27. **Responsive** Websites
- 28. **Relative** Units
- 29. **Position** Property
- 30. **Semantic** Tags

26. Display Property (Block / Inline Elements)



Block Elements

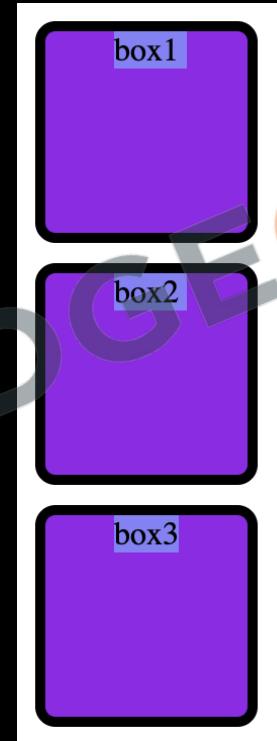
- **New Line:** Start on a new line.
- **Full Width:** Take up all horizontal space.
- **Styling:** Can have margins and padding.
- **Size:** Width and height can be set.
- **Examples:** <div>, <p>, <h1>, , .

Inline Elements

- **Flow:** Stay in line with text.
- **Width:** Just as wide as the content.
- **No Break:** No new line between elements.
- **Limited Styling:** Can't set size easily.
- **Examples:** , <a>, , .

26. Display Property (Block)

```
.box {  
    height: 100px;  
    width: 100px;  
    background-color: blueviolet;  
    margin: 10px;  
    text-align: center;  
    border: 5px solid black;  
    border-radius: 10px;  
    display: block;  
}
```

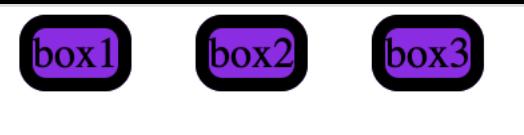


```
<head>  
    <title>Display Block</title>  
    <link rel="stylesheet" href="../../css/  
    level 5/display.css">  
</head>  
<body>  
    <div id="parent">  
        <div id="div1" class="box">box1</div>  
        <div id="div2" class="box">box2</div>  
        <div id="div3" class="box">box3</div>  
    </div>  
</body>
```

26. Display Property (Inline)

```
.box {  
    height: 100px;  
    width: 100px;  
    background-color: blueviolet;  
    margin: 10px;  
    text-align: center;  
    border: 5px solid black;  
    border-radius: 10px;  
  
    display: inline;  
}
```

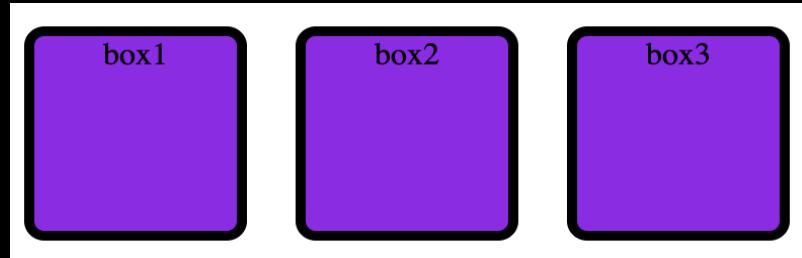
```
<!DOCTYPE html>  
<html lang="en">  
  <head>  
    <title>Display Inline</title>  
    <link rel="stylesheet" href="../../css/  
      level 5/display_inline.css">  
  </head>  
  <body>  
    <div id="parent">  
      <div id="div1" class="box">box1</div>  
      <div id="div2" class="box">box2</div>  
      <div id="div3" class="box">box3</div>  
    </div>  
  </body>  
</html>
```



26. Display Property (Inline-Block)

```
.box {  
    height: 100px;  
    width: 100px;  
    background-color: blueviolet;  
    margin: 10px;  
    text-align: center;  
    border: 5px solid black;  
    border-radius: 10px;  
    display: inline-block;  
}
```

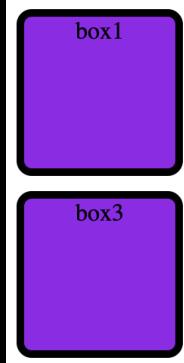
```
<head>  
    <title>Display Inline Block</title>  
    <link rel="stylesheet" href="../../css/  
        level 5/display_inline_block.css">  
</head>  
<body>  
    <div id="parent">  
        <div id="div1" class="box">box1</div>  
        <div id="div2" class="box">box2</div>  
        <div id="div3" class="box">box3</div>  
    </div>  
</body>
```



26. Display Property (None)

```
.box {  
    height: 100px;  
    width: 100px;  
    background-color: blueviolet;  
    margin: 10px;  
    text-align: center;  
    border: 5px solid black;  
    border-radius: 10px;  
}  
  
#div2 {  
    display: none;  
}
```

```
<head>  
    <title>Display None</title>  
    <link rel="stylesheet" href="../../css/  
        level 5/display_none.css">  
</head>  
<body>  
    <div id="parent">  
        <div id="div1" class="box">box1</div>  
        <div id="div2" class="box">box2</div>  
        <div id="div3" class="box">box3</div>  
    </div>  
</body>
```



27. Responsive Website



KNOWLEDGE GATE¹

27. Responsive Website



1. Adapts layout for different screen sizes
2. Flexible layouts
3. Optimizes images and assets
4. Enhances user experience on mobile and desktop

28. Relative Units

CSS Units Cheat Sheet

px

Absolute pixel value

%

A percentage of the parent element.
100% is the width of the parent element

em

Relative to the font size of the element

rem

Relative to the font size of
the root element

vh

Relative to 1% of the viewport's height

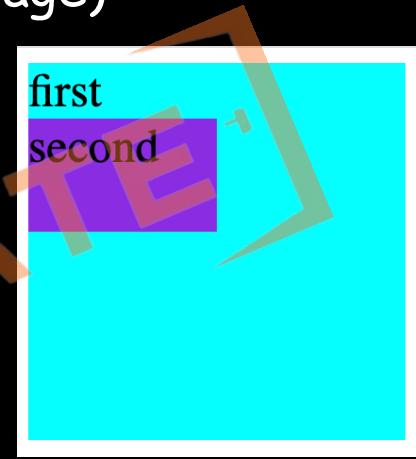
vw

Relative to 1% of the viewport's width

28. Relative Units (Percentage)

```
#first {  
    height: 200px;  
    width: 200px;  
    background-color: #aqua;  
    font-size: 25px;  
}  
  
#second {  
    background-color: #blueviolet;  
    width: 50%;  
    height: 30%;  
}
```

```
<body>  
    <div id="first">  
        first  
        <div id="second">  
            second  
        </div>  
    </div>  
</body>
```

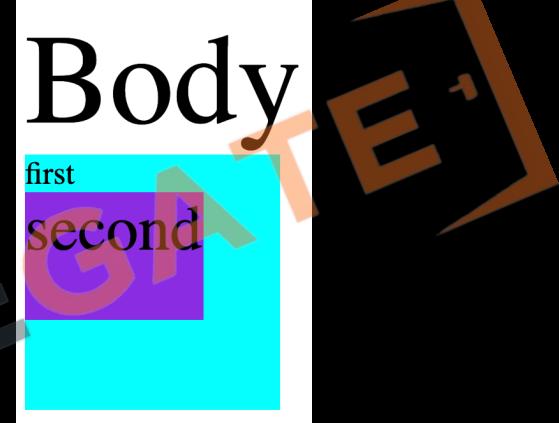


- **Relative Sizing:** Facilitates dynamic sizing **relative to parents**.
- **Adaptability:** Ensures **responsiveness** across various screens.
- **Dimensions:** Quickly set width and height as a percentage.

28. Relative Units (EM)

```
body {  
    font-size: 100px;  
}  
  
#first {  
    height: 200px;  
    width: 200px;  
    background-color: #aqua;  
    font-size: 25px;  
}  
  
#second {  
    background-color: #blueviolet;  
    width: 70%;  
    height: 50%;  
    font-size: 2em;  
}
```

```
<body>  
    Body  
    <div id="first">  
        first  
        <div id="second">  
            | second  
            </div>  
        </div>  
    </body>
```

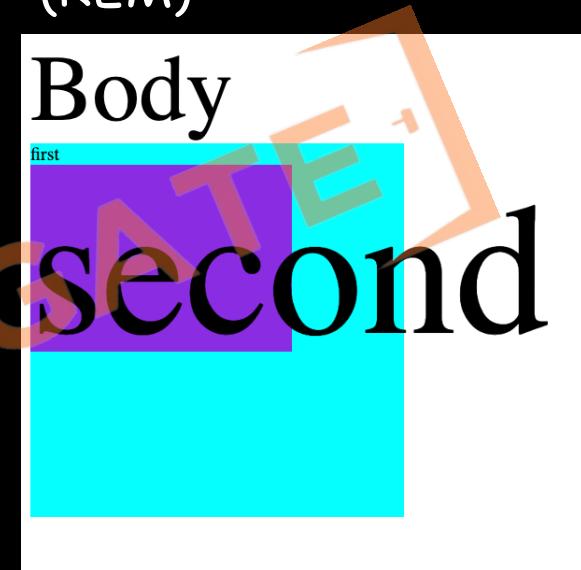


- **Relative Unit:** Sized relative to the parent element's font size.
- **Scalability:** Facilitates easy scaling of elements for responsive design.
- **Font Sizing:** Commonly used for setting font sizes adaptively.

28. Relative Units (REM)

```
* {  
    font-size: 50px;  
}  
  
#first {  
    height: 200px;  
    width: 200px;  
    background-color: #aqua;  
    font-size: 10px;  
}  
  
#second {  
    background-color: #blueviolet;  
    width: 70%;  
    height: 50%;  
    font-size: 2rem;  
}
```

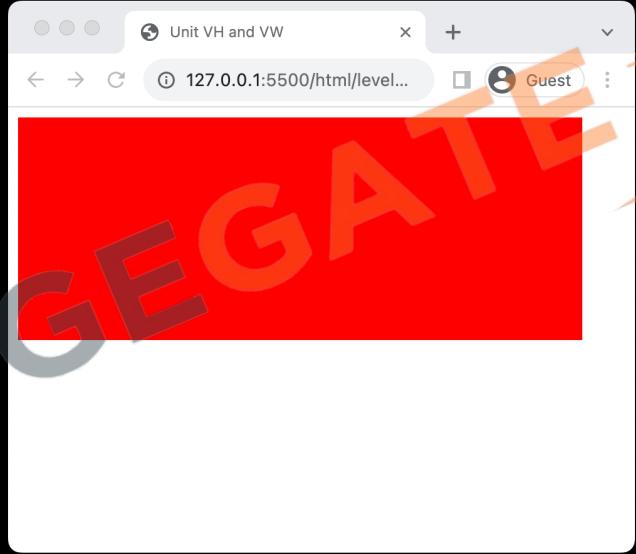
```
<body>  
    Body  
    <div id="first">  
        first  
        <div id="second">  
            | second  
            </div>  
        </div>  
    </body>
```



- **Relative Sizing:** Facilitates dynamic sizing **relative to root element**.
- **Adaptability:** Ensures **responsiveness** across various screens.
- **Dimensions:** Quickly set width and height as a percentage.

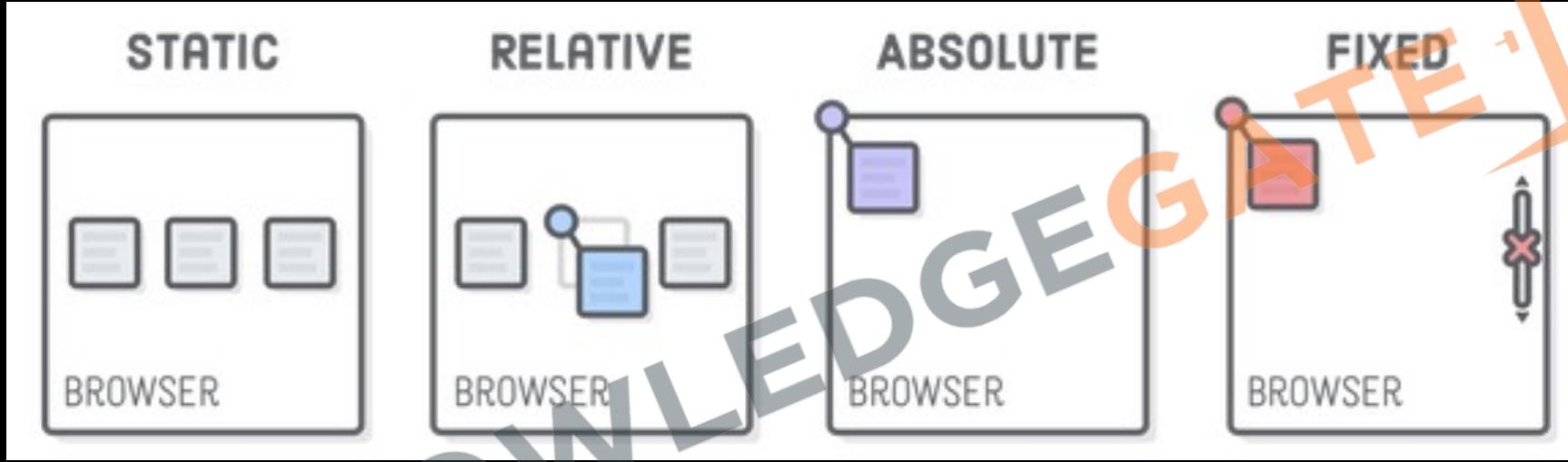
28. Relative Units (vw/vH)

```
<head>
  <title>Unit VH and VW</title>
  <style>
    #first {
      height: 50vh;
      width: 90vw;
      background-color: red;
    }
  </style>
</head>
<body>
  <div id="first"></div>
</body>
```



- **Viewport Relative Units:** Units based on viewport's width (vw) or height (vh) for responsive design.
- **Responsive Layouts:** Essential for creating adaptive layouts; e.g., height: 100vh for full-screen sections.
- **Element Sizing:** Useful for defining heights and widths that scale with the viewport.

29. Position Property

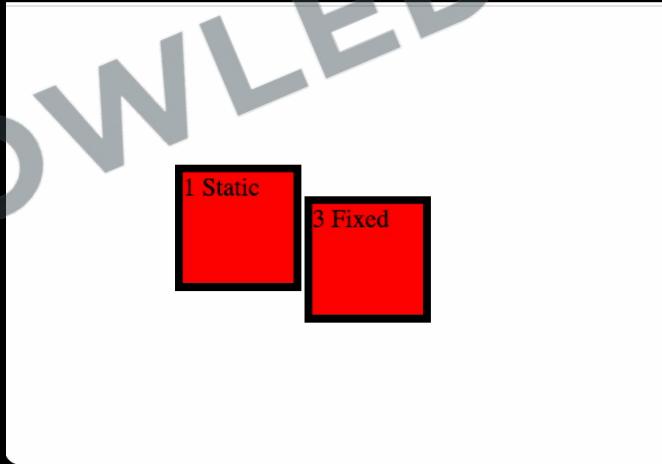


- **Static (default)** : Elements follow the normal document flow. (`top`, `right`, `bottom`, `left`, `z-index` would not work)
- **Relative**: Element's position adjusted from its normal position.
- **Absolute**: Positions element relative to the nearest positioned ancestor.
- **Fixed**: Element positioned relative to the `viewport`, does not move on scroll.

29. Position Property

```
div {  
    height: 70px;  
    width: 70px;  
    background-color: red;  
    border: 5px solid black;  
    margin: 20px;  
}  
  
#div1 {  
    position: static;  
}  
  
#div2 {  
    position: relative;  
    top: 20px;  
    left: 90px;  
}  
  
#div3 {  
    position: fixed;  
    top: 20px;  
    left: 90px;  
}  
  
#div4 {  
    position: absolute;  
    top: 200px;  
    left: 200px;  
}
```

```
<head>  
    <title>Position</title>  
    <link rel="stylesheet" href="../../css/level 5/positions.css">  
</head>  
<body>  
    <div id="div1">1 Static</div>  
    <div id="div2">2 Relative</div>  
    <div id="div3">3 Fixed</div>  
    <div id="div4">4 Absolute</div>  
</body>
```

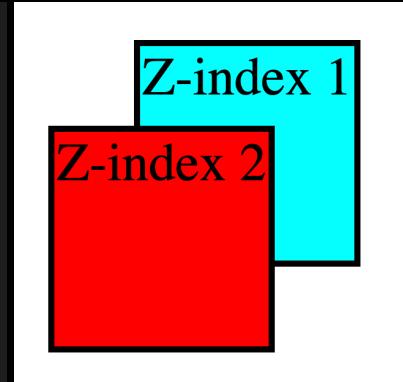


29. Position Property (z index)

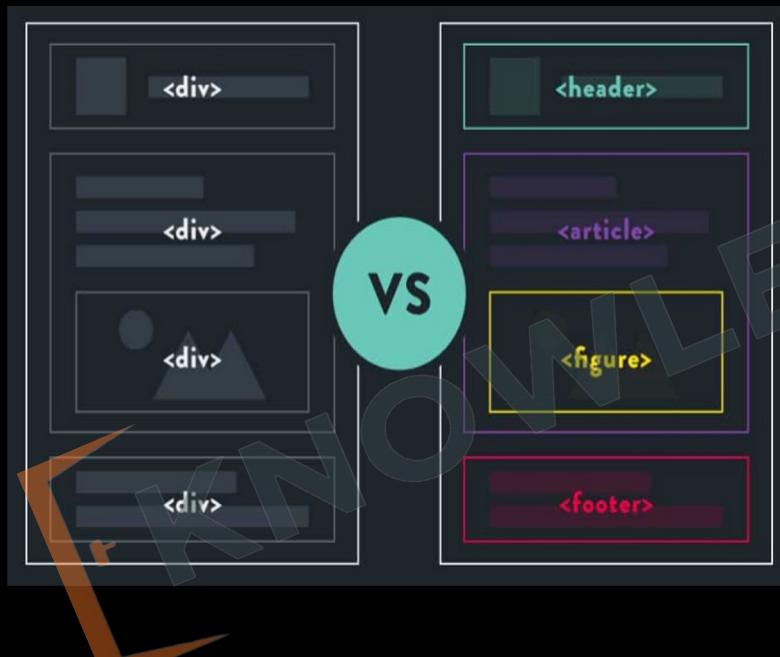
```
.container {  
    position: relative;  
}  
  
.box1, .box2 {  
    position: absolute;  
    border: 3px solid black;  
    width: 100px;  
    height: 100px;  
    text-align: center;  
    font-size: 25px;  
}  
  
.box1 {  
    background-color: red;  
    left: 20px;  
    top: 60px;  
    z-index: 2;  
}  
  
.box2 {  
    background-color: aqua;  
    left: 60px;  
    top: 20px;  
    z-index: 1;  
}
```

- **Stacking Order:** Determines the **stacking order** of elements along the Z-axis.
- **Position Context:** Only applies to elements with position set to relative, absolute, fixed, or sticky.
- **Integer Values:** Accepts **integer values**, including negative numbers.
- **Higher Values:** An element with a **higher z-index value appears above others.**

```
<head>  
    <title>Z-Index</title>  
    <link rel="stylesheet" href="../../css/  
        level 5/z-index.css">  
</head>  
<body>  
    <div class="container">  
        <div class="box1">Z-index 2</div>  
        <div class="box2">Z-index 1</div>  
    </div>  
</body>
```



30. Semantic Tags



30. Semantic Tags

Semantic Tags

- **Meaningful:** Describe content.
- **SEO:** Good for search engines.
- **Accessibility:** Useful for screen readers.
- Examples: `<header>`, `<footer>`, `<article>`, `<section>`, `<nav>`.

Non-Semantic Tags

- **Generic:** No specific meaning.
- **For Styling:** Used for layout.
- **No SEO:** Not SEO-friendly.
- Examples: `<div>`, ``, `<i>`, ``.

Level 5 Revision

Display and Position

- 26. Display Property
- 27. Responsive Websites
- 28. Relative Units
- 29. Position Property
- 30. Semantic Tags



Practice Set Level 5

Display and Position

- Create a webpage with header, footer, and a content area.
 - Header
 - Create a nav bar with links.
 - Main
 - Create a div with width and height, background green and border radius 50%
 - Create Three divs with container height and width 100px. Display inline block.
 - Set the correct position property for the single div element to ensure it remains at the right side of the page and does not shift when scrolling.
 - Use z-index to place the div on top of another div.
 - Footer
 - Add text in footer.



Level 6

Flex Box, Grid and Media Queries

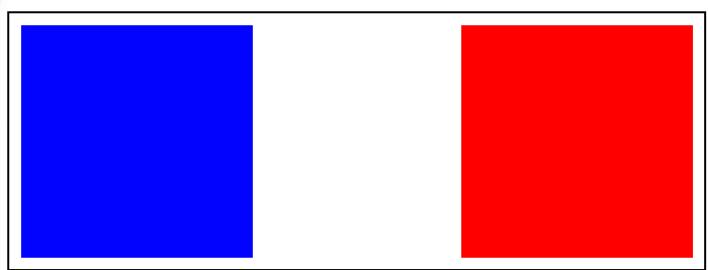
- 31. **Float** Property
- 32. What is **Flexbox**?
- 33. **Flex Model**
- 34. **Flexbox Direction**
- 35. **Properties:** Flexbox container
- 36. **Properties:** Flex Items
- 37. **Grid Layout**
- 38. **Media Queries**

31. Float Property

```
.container {  
    height: 110px;  
    width: 300px;  
    border: 1px solid #000;  
}  
  
.box {  
    width: 100px;  
    height: 100px;  
    margin: 5px;  
}  
  
.box1 {  
    background-color: red;  
    float: right;  
}  
  
.box2 {  
    background-color: blue;  
    float: left;  
}
```

- **Element Alignment:** Allows elements to be aligned to the left or right within their containing element.
- **Values:** Can take values like "left", "right", or "none" to determine the floating direction.
- **Old Layout Technique:** Less commonly used with the advent of Flexbox.

```
<head>  
    <title>Float Property</title>  
    <link rel="stylesheet" href="../../css/level 6/float.css">  
</head>  
<body>  
    <div class="container">  
        <div class="box box1"></div>  
        <div class="box box2"></div>  
    </div>  
</body>
```

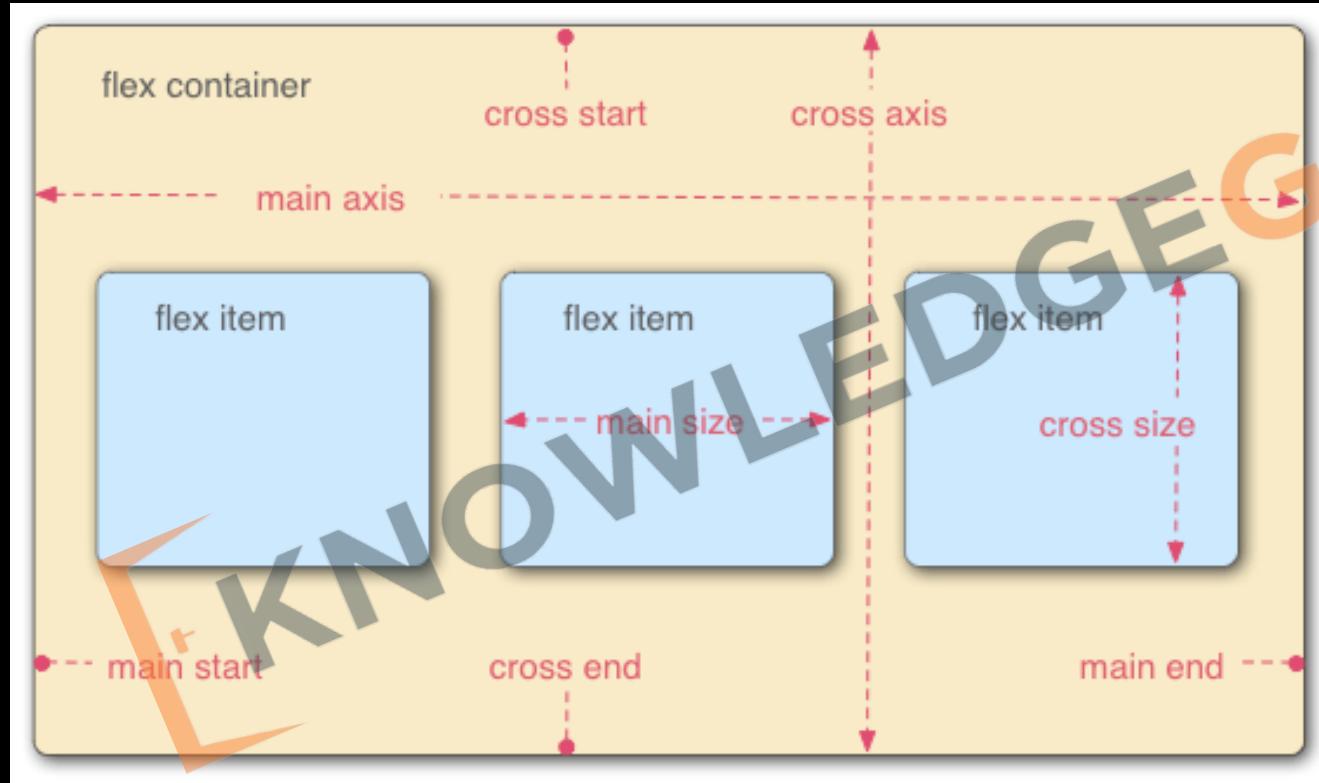


32. What is Flexbox?

Flexbox is a one-dimensional layout method for arranging items in rows or columns.
Items *flex* (expand) to fill additional space or shrink to fit into smaller spaces.



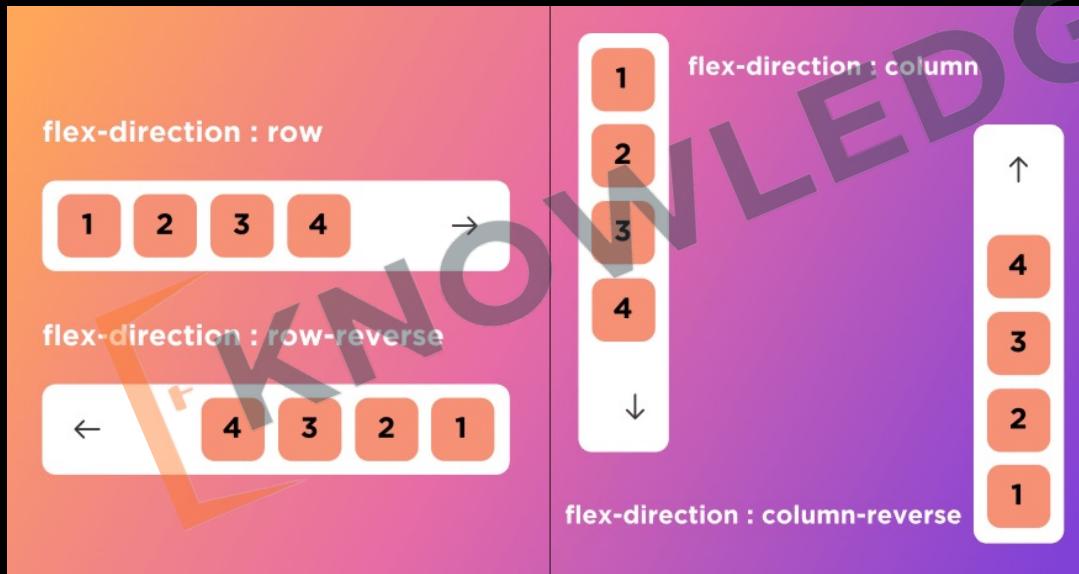
33. Flex Model



display: flex

34. Flexbox Direction

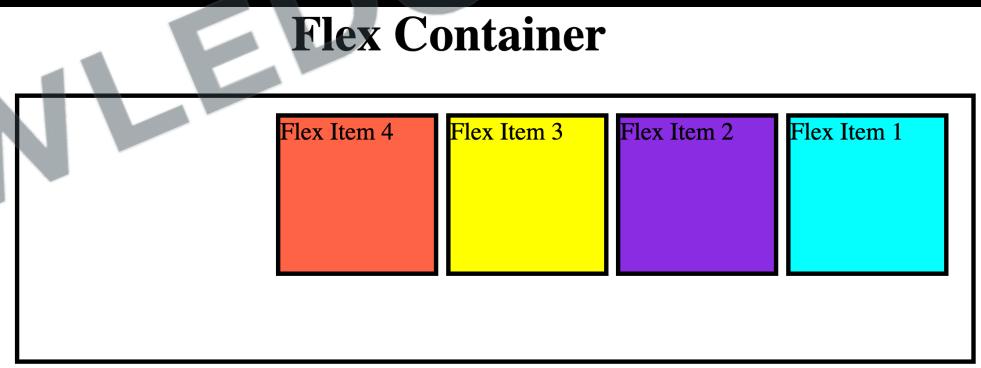
- **Property Name:** flex-direction is the property used to define the direction in a flex container.
- **Row Layout:** row value aligns the flex items horizontally, in a left-to-right fashion.
- **Column Layout:** column value stacks the flex items vertically, from top to bottom.
- **Reverse Direction:** Adding -reverse to row or column (as in row-reverse or column-reverse) reverses the order of the items.



34. Flexbox Direction

```
* {  
    margin: 0;  
    padding: 0;  
}  
  
.box {  
    height: 100px;  
    width: 100px;  
    border: 3px solid black;  
    margin-right: 5px;  
}  
  
#heading {margin-left: 200px;}  
  
#container {  
    height: 150px;  
    width: 600px;  
    padding: 10px;  
    margin: 20px;  
    border: 3px solid black;  
  
    display: flex;  
    flex-direction: row-reverse;  
}  
  
#box1 { background-color: aqua; }  
#box2 { background-color: blueviolet; }  
#box3 { background-color: yellow; }  
#box4 { background-color: tomato; }
```

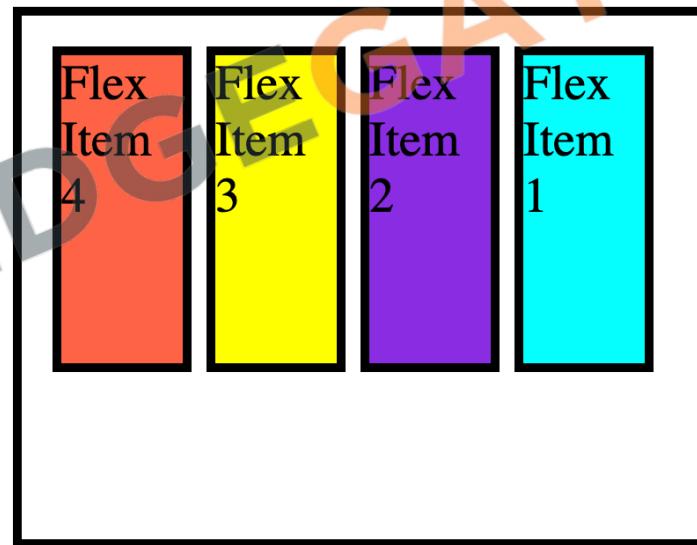
```
<body>  
    <h1 id="heading">Flex Container</h1>  
    <div id="container">  
        <div id="box1" class="box">Flex Item 1</div>  
        <div id="box2" class="box">Flex Item 2</div>  
        <div id="box3" class="box">Flex Item 3</div>  
        <div id="box4" class="box">Flex Item 4</div>  
    </div>  
</body>
```



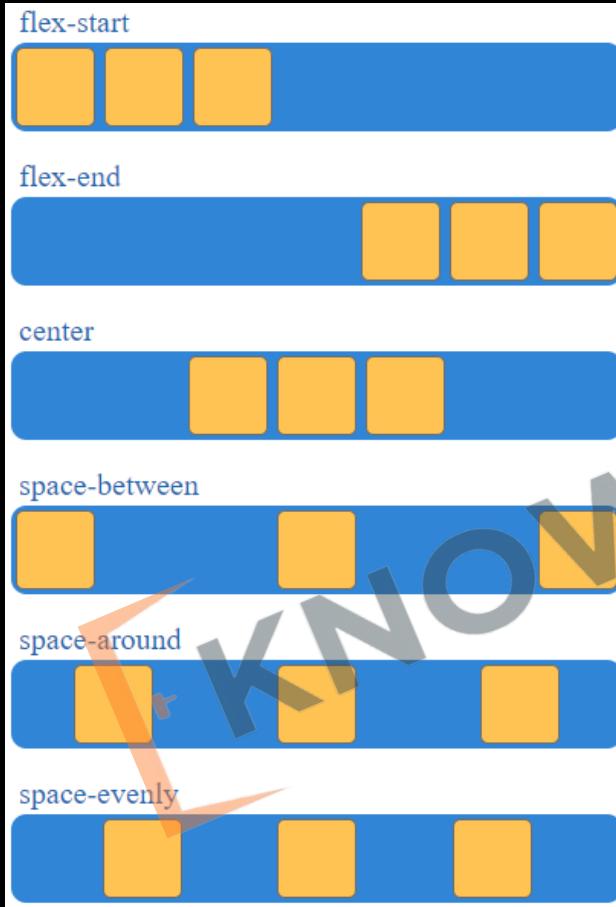
34. Flexbox Direction

```
.box {  
    height: 100px;  
    width: 100px;  
    border: 3px solid black;  
    margin-right: 5px;  
}  
  
#heading {margin-left: 50px;}  
  
#container {  
    height: 150px;  
    width: 200px;  
    padding: 10px;  
    margin: 20px;  
    border: 3px solid black;  
  
    display: flex;  
    flex-direction: row-reverse;  
}
```

Flex Container



35. Properties: Flexbox container (Justify Content)



- **Alignment:** Aligns flex items along the main axis.
- **flex-start:** Items align to the start of the flex container.
- **flex-end:** Items align to the end of the flex container.
- **Center:** Items are centered within the flex container.
- **space-between/space-around/space-evenly:** Distributes space between items evenly.

35. Properties: Flexbox container (Justify Content)

Flexbox Container

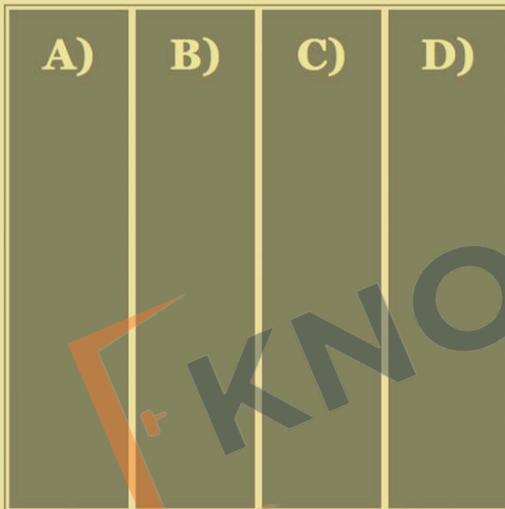


```
display: flex;  
justify-content: space-between;
```

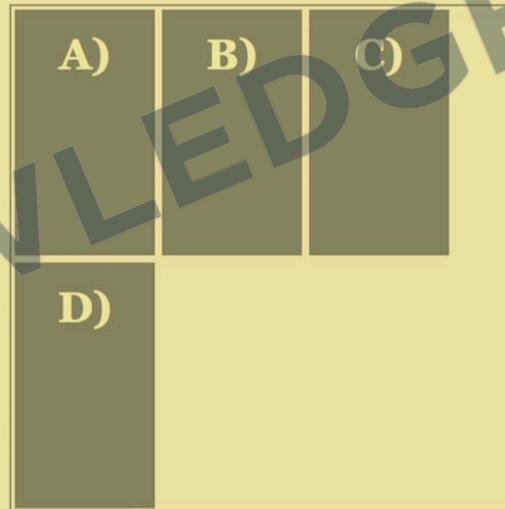
35. Properties: Flexbox container (Flex Wrap)

`flex-wrap:`

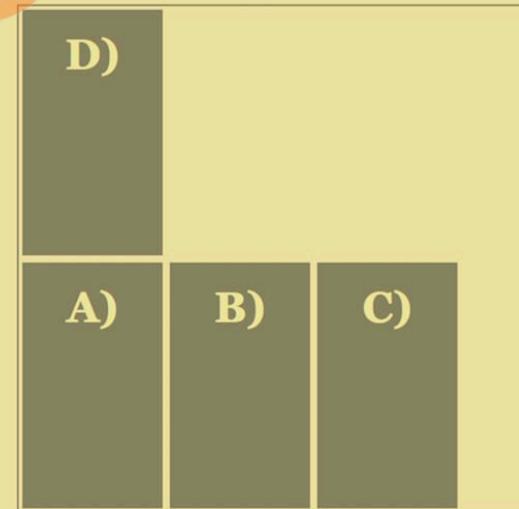
`nowrap (default)`



`wrap`



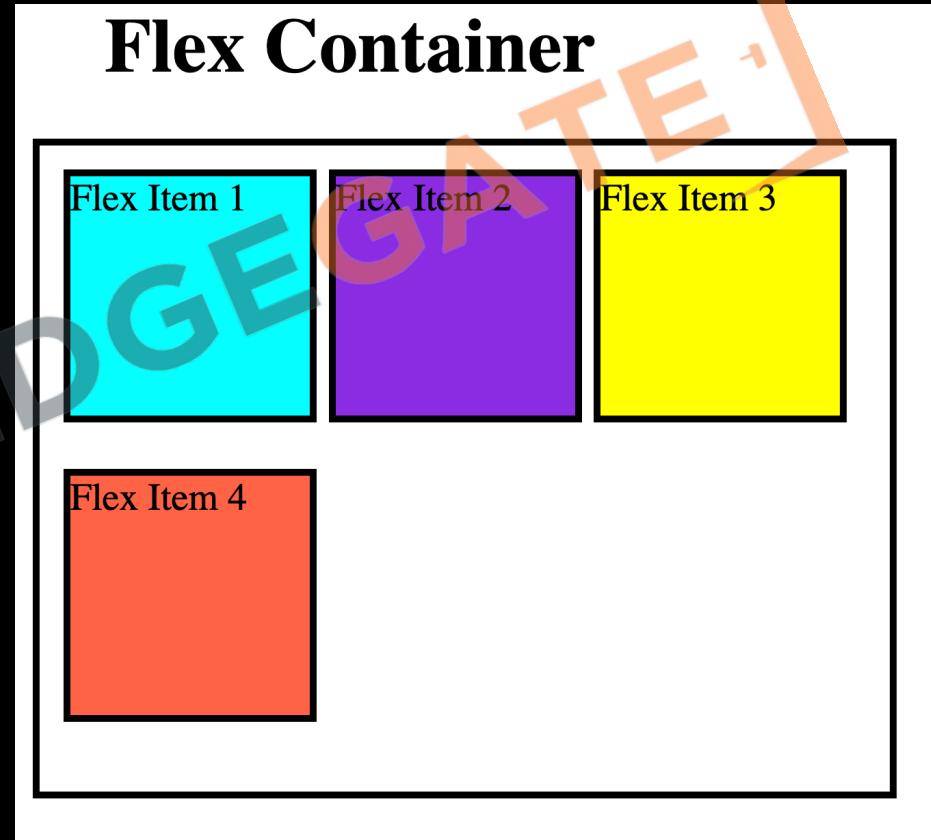
`wrap-reverse`



KNOWLEDGE GATE

35. Properties: Flexbox container (Flex Wrap)

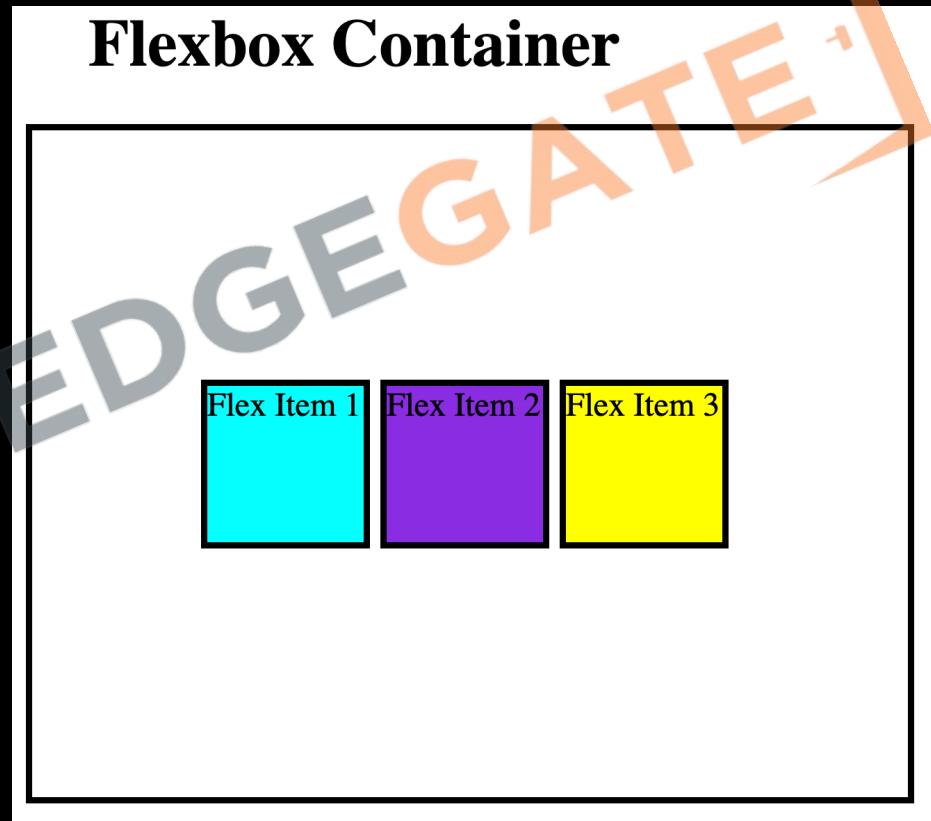
```
#container {  
    height: 250px;  
    width: 335px;  
    padding: 10px;  
    margin: 20px;  
    border: 3px solid black;  
  
    display: flex;  
    flex-wrap: wrap;  
}
```



35. Properties: Flexbox container (Align Items)

This property is used to align the flex container's items along the cross-axis, which is perpendicular to the main axis.

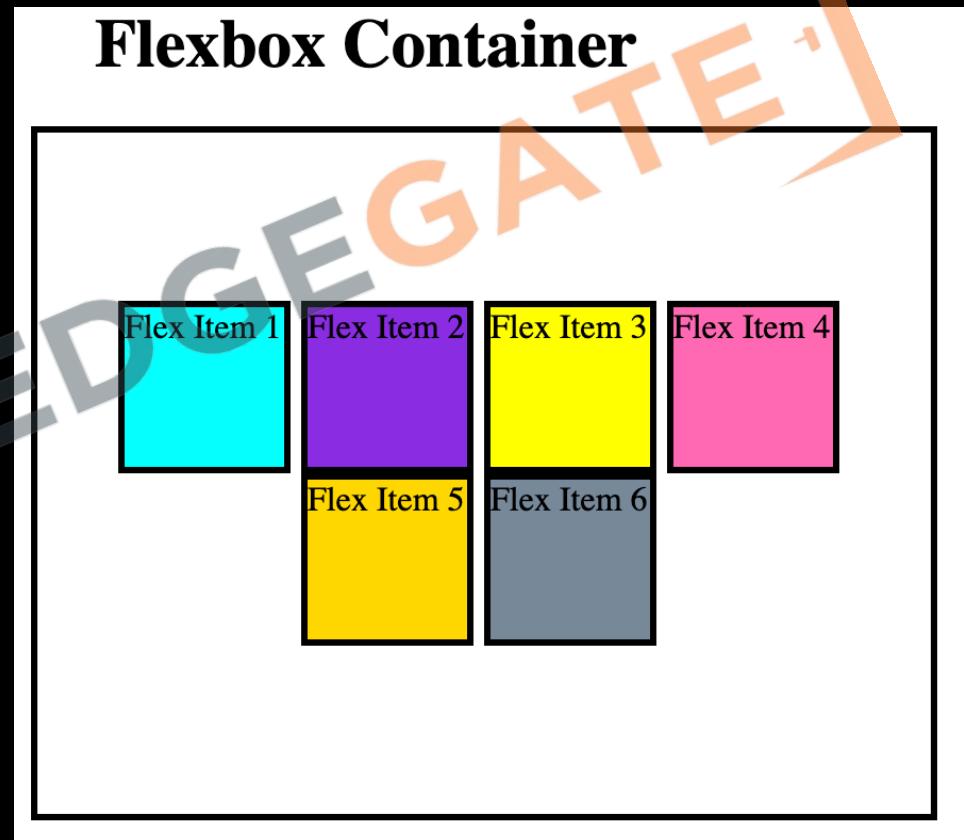
```
display: flex;  
flex-direction: row;  
justify-content: center;  
align-items: center;
```



35. Properties: Flexbox container (Align Content)

It is utilized to adjust the spacing between flex lines within a flex container, particularly when there is extra space along the cross-axis.

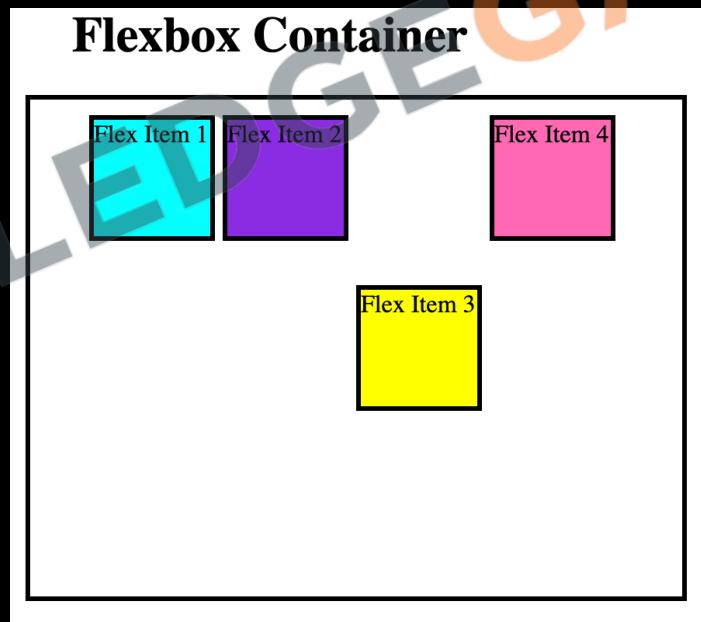
```
display: flex;  
flex-direction: row;  
justify-content: center;  
flex-wrap: wrap;  
align-content: center;
```



36. Properties: Flex Items (Align Self)

```
#container {  
    height: 300px;  
    width: 400px;  
    padding: 10px;  
    margin: 20px;  
    border: 3px solid black;  
  
    display: flex;  
    flex-direction: row;  
    justify-content: center;  
    flex-wrap: wrap;  
    align-items: start;  
}  
  
#box1 { background-color: aqua; }  
#box2 { background-color: blueviolet; }  
#box4 { background-color: hotpink; }  
#box3 {  
    background-color: yellow;  
    align-self: center;  
}
```

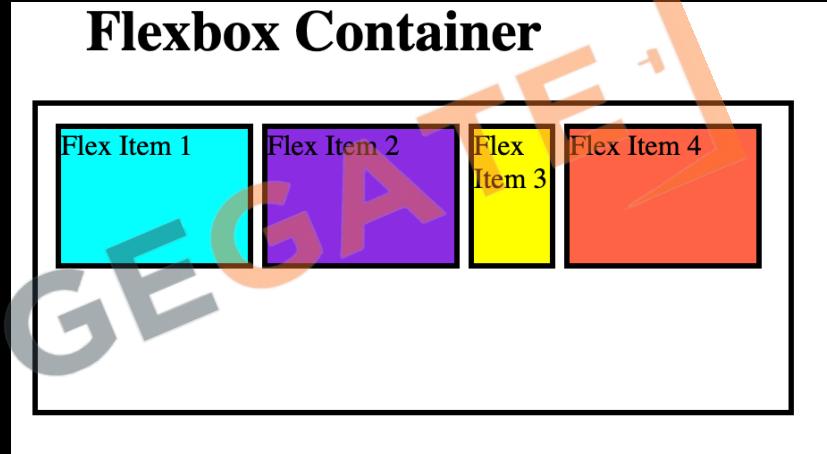
Allows individual flex items to **override** the container's **align-items** property, aligning them differently along the cross-axis.



36. Properties: Flex Items (Flex Shrink)

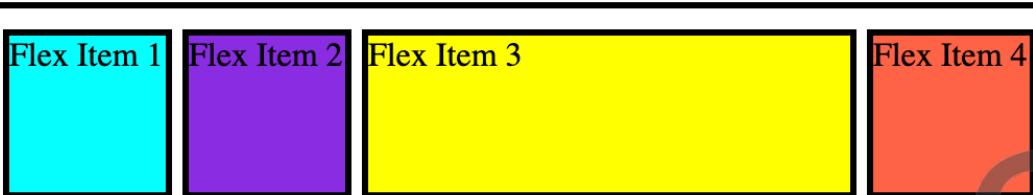
```
#box3 {  
    background-color: yellow;  
    flex-shrink: 4;  
}
```

The "flex-shrink" property in CSS determines how much a **flex item** **will shrink relative to other items** in the flex container if there is insufficient space.



36. Properties: Flex Items (Flex Grow)

Flexbox Container



```
#box3 {  
    background-color: yellow;  
    flex-grow: 1;  
}
```

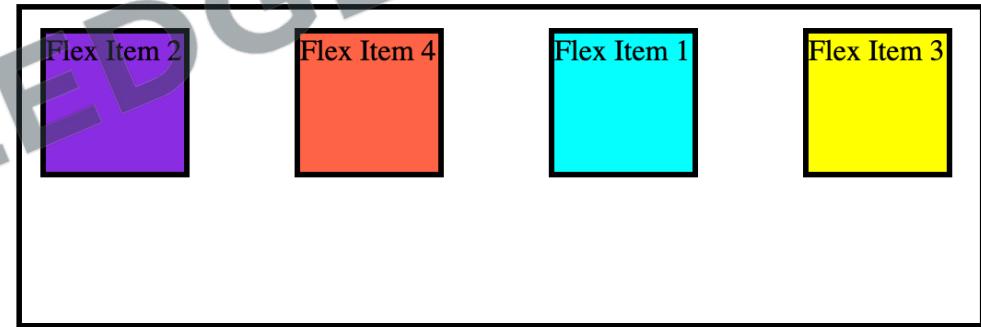
The "flex-grow" property in CSS specifies how much a **flex item** will grow relative to other items in the flex container when additional space is available.

36. Properties: Flex Items (Order)

```
#box1 {  
    background-color: #aqua;  
    order: 3  
}  
  
#box2 {  
    background-color: #blueviolet;  
    order: 1;  
}  
  
#box3 {  
    background-color: #yellow;  
    order: 4;  
}  
  
#box4 {  
    background-color: #tomato;  
    order: 2;  
}
```

The "order" property in CSS allows you to define the sequence in which flex items appear within the flex container, **overriding their original order** in the HTML.

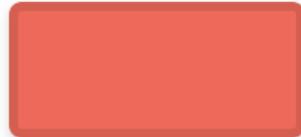
Flexbox Container



37. Grid

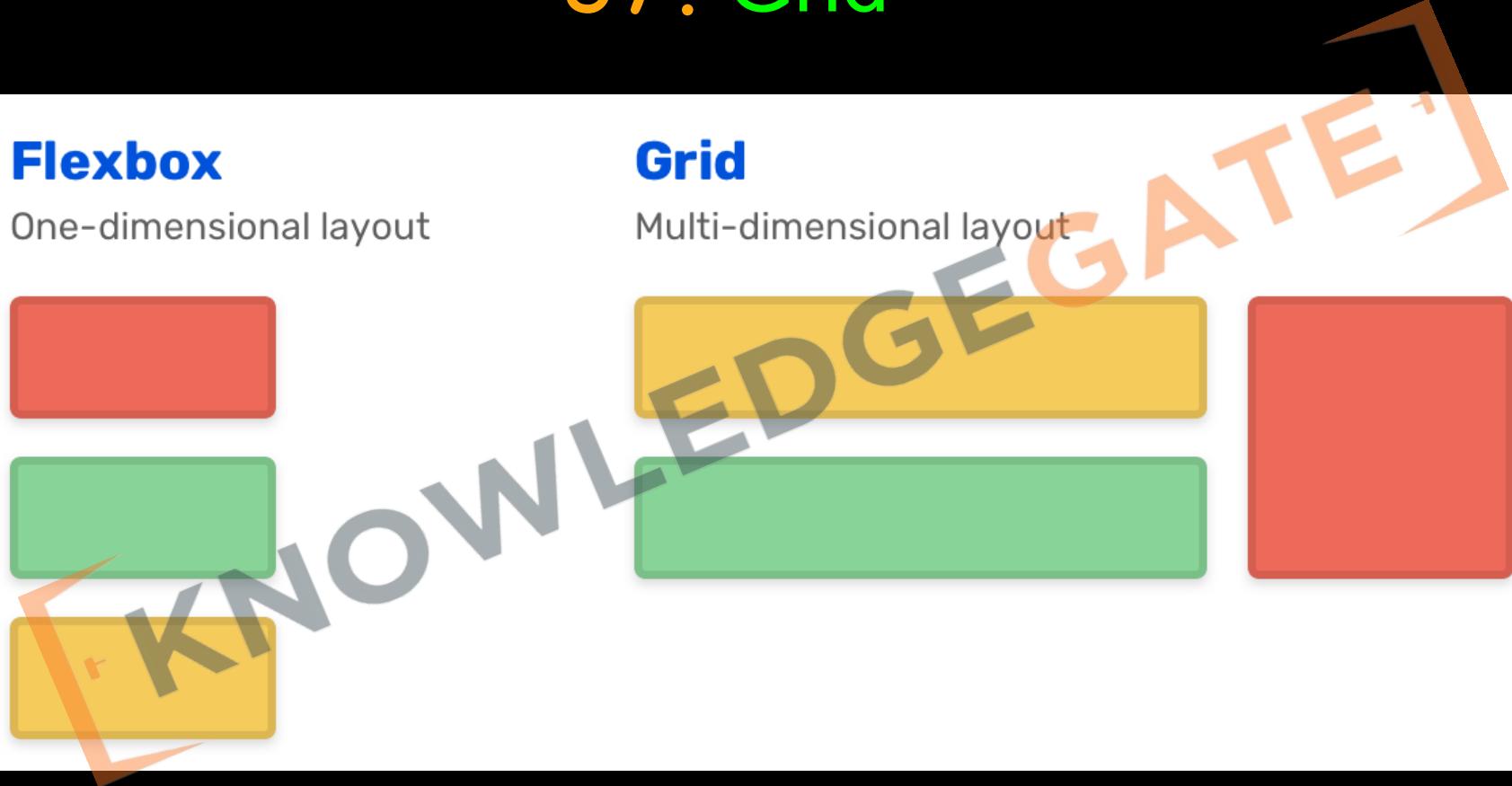
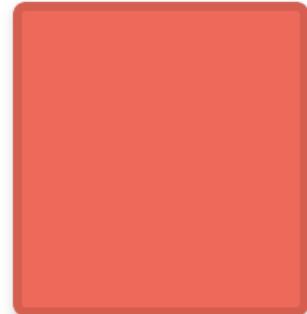
Flexbox

One-dimensional layout



Grid

Multi-dimensional layout



37. Grid

```
.container {  
    display: grid;  
    grid-template-columns: 50px 50px;  
    grid-template-rows: 50px 50px;  
}  
  
.item1 {  
    grid-column: 1 / 2;  
    grid-row: 1 / 2;  
    background-color: #lightblue;  
}  
  
.item2 {  
    grid-column: 2 / 2;  
    grid-row: 1 / 2;  
    background-color: #lightgreen;  
}  
  
.item3 {  
    grid-column: 1 / 2;  
    grid-row: 2 / 2;  
    background-color: #lightpink;  
}  
  
.item4 {  
    grid-column: 2 / 2;  
    grid-row: 2 / 2;  
    background-color: #lightyellow;  
}
```

```
<div class="container">  
    <div class="item1">Item 1</div>  
    <div class="item2">Item 2</div>  
    <div class="item3">Item 3</div>  
    <div class="item4">Item 4</div>  
</div>
```



- 2D layout system for rows & columns.
- Activate with `display: grid;`.
- Children become grid items.
- Define structure with `grid-template` properties.
- Individual units called grid cells.

38. Media Queries

declaration

Media Type

```
@media screen and (max-width: 768px){  
    .container{  
        // Write styles here  
    }  
}
```

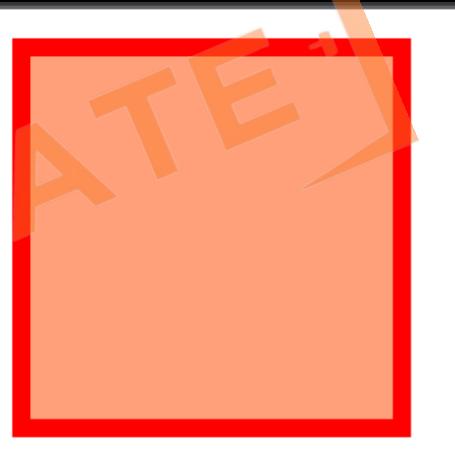
styles to apply
when all conditions
are met

Specifying amount
of screen to cover

- Tailor styles for specific device characteristics.
- Use to create responsive web designs.
- Apply styles based on conditions like screen size.
- Syntax: `@media (condition) { CSS rules }`.
- Can combine multiple conditions using and, or.

38. Media Queries (width)

```
.box {  
    height: 100px;  
    width: 100px;  
    background-color: lightsalmon;  
}  
  
@media screen and (width: 250px) {  
    .box {  
        border: 5px solid red;  
    }  
}
```



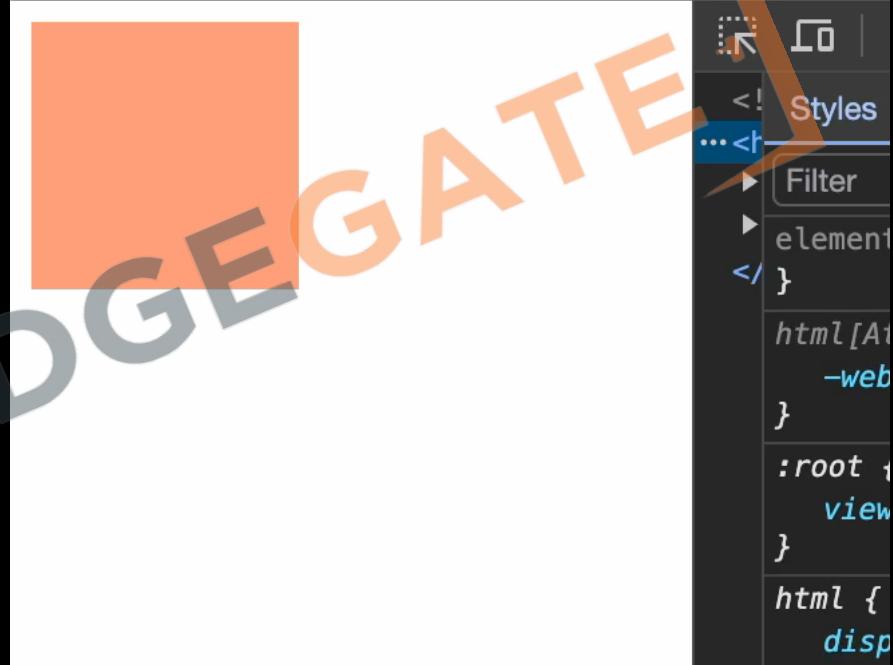
38. Media Queries (min-width)

```
.box {  
    height: 100px;  
    width: 100px;  
    background-color: lightcoral;  
}  
  
@media screen and (min-width: 300px) {  
    .box {  
        height: 150px;  
        width: 150px;  
    }  
}
```



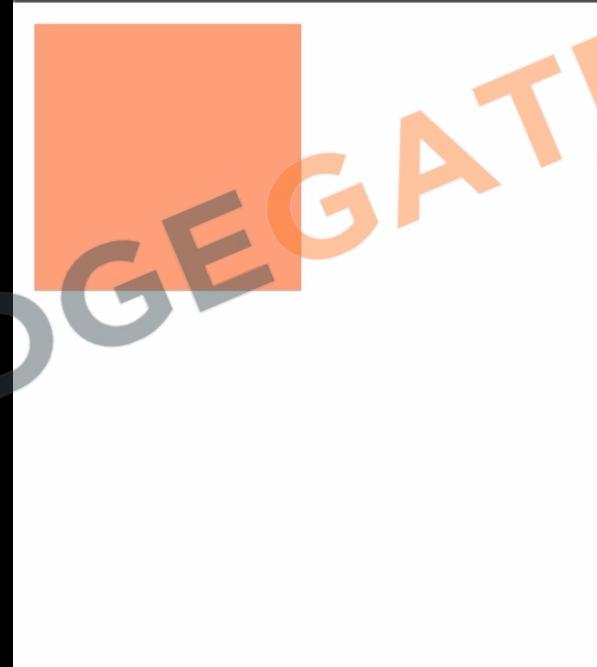
38. Media Queries (max-width)

```
.box {  
    height: 100px;  
    width: 100px;  
    background-color: lightsalmon;  
}  
  
@media screen and (max-width: 250px) {  
    .box {  
        height: 50px;  
        width: 50px;  
    }  
}
```



38. Media Queries (combination)

```
.box {  
    height: 100px;  
    width: 100px;  
    background-color: lightsalmon;  
}  
  
@media screen and (min-width: 250px)  
and (max-width: 300px) {  
    .box {  
        border-radius: 50%;  
    }  
}
```

A screenshot of a browser's developer tools, specifically the 'Styles' panel. It shows a tree structure of styles being applied to an element. At the top is 'element.style'. Below it are rules for 'Filter', 'element.style', and 'html [Attribute] -webkit-l'. Further down are rules for ':root' and 'html' with properties like 'view-transition' and 'display'. The 'Styles' tab is highlighted in blue.

Level 6 Revision

Flex Box, Grid and Media Queries

- 31. **Float** Property
- 32. What is **Flexbox**?
- 33. **Flex Model**
- 34. **Flexbox Direction**
- 35. **Properties:** Flexbox container
- 36. **Properties:** Flex Items
- 37. **Grid Layout**
- 38. **Media Queries**



Practice Set Level 6

Flex Box, Grid and Media Queries

- Create the same nav bar created in the last Practice Set but do the spacing using flexbox.
- Use **flexbox** to centre an image inside a div.
- Create a div with three boxes where two boxes have fixed size, but the **third box grows and shrinks** with container size.
- Create a div of 50px by 50px with color green. It should respond to following conditions:
 - Keep color **green** under 300px of view port size
 - Change color to **red** from 300px to 400px
 - Change color to **blue** after 400px



Level 7

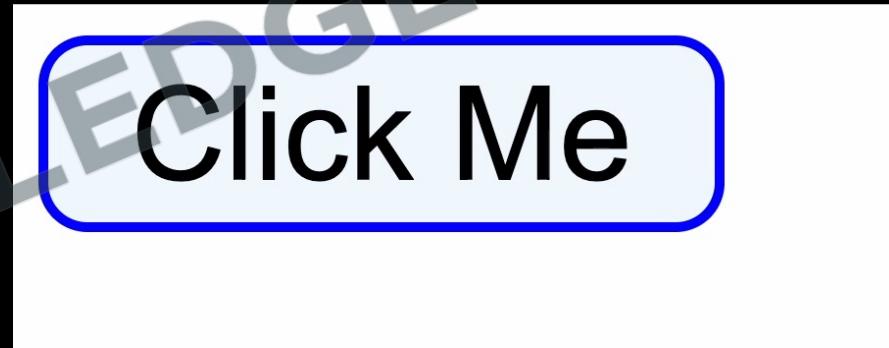
Animation, Transition & Transform

- 39. Pseudo Classes
- 40. Transitions
- 41. CSS Transform
- 42. Animation

39. Pseudo Classes

```
.btn {  
    height: 20px;  
    width: 70px;  
    border: 1px solid blue;  
    border-radius: 5px;  
    background-color: aliceblue;  
}  
  
.btn:hover {  
    height: 25px;  
    width: 80px;  
    border: 1px solid red;  
}  
  
.btn:active {  
    height: 25px;  
    width: 80px;  
    border: 1px solid red;  
    background-color: indianred;  
}
```

- Used to define **special states** of HTML elements.
- Syntax: **selector:pseudo-class { styles }**.
- Common examples: `:hover`, `:active`, `:first-child`.
- Target elements based on their position or user action.



40. Transitions



CSS transition is a property that enables smooth animation between changes in CSS property values

- **transition-property:** Defines which CSS properties will transition.
- **transition-duration:** Sets how long the transition lasts.
- **transition-timing-function:** Controls the speed curve of the transition.
- **transition-delay:** Specifies a delay before the transition starts.

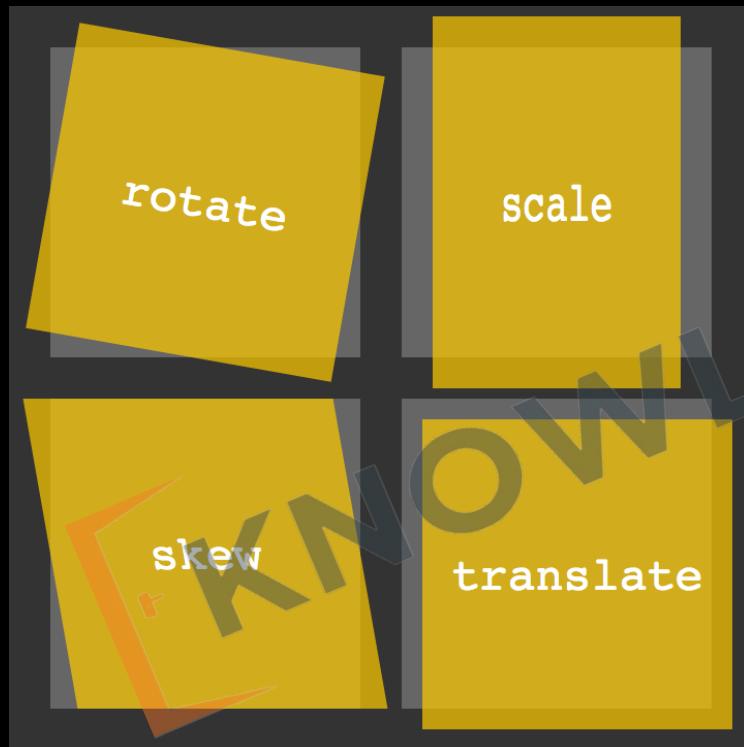
40. Transitions

```
.btn {  
    height: 20px;  
    width: 70px;  
    border: 1px solid blue;  
    border-radius: 5px;  
    background-color: aliceblue;  
    transition-property: all;  
    transition-duration: 1s;  
    transition-timing-function: ease-in-out;  
    transition-delay: 1s;  
    /*transition: all 1s ease-in-out 1s;*/  
}  
  
.btn:hover {  
    height: 25px;  
    width: 80px;  
    border: 1px solid red;  
}  
  
.btn:active {  
    height: 25px;  
    width: 80px;  
    border: 1px solid red;  
    background-color: indianred;  
}
```



Click Me

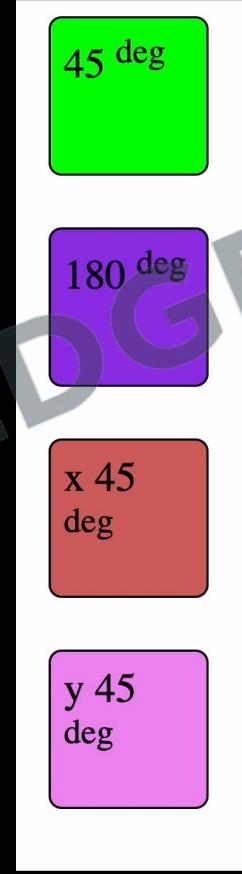
41. CSS Transform



- Allows modification of an element's shape and position.
- Can perform operations like `rotate`, `scale`, and `translate`.
- Does not affect the layout of surrounding elements.
- Used to create visual effects like 3D space transformations.
- Implemented with functions like `rotate()`, `scale()`, and `translate()`.

41. CSS Transform (Rotate)

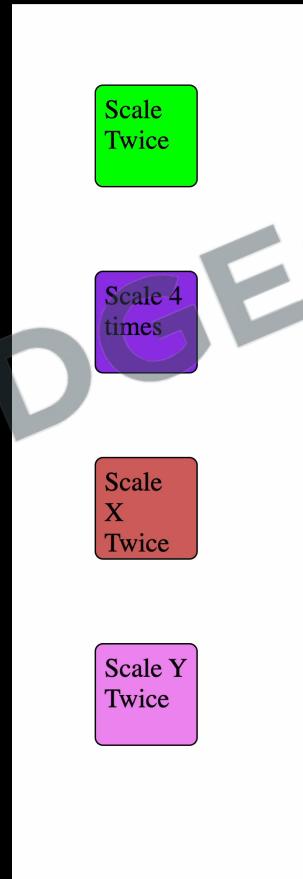
```
.box {  
    height: 50px;  
    width: 50px;  
    padding: 5px;  
    margin: 20px;  
    border: 1px solid black;  
    border-radius: 5px;  
    transition-property: all;  
    transition-duration: 1s;  
    transition-timing-function: ease-in-out;  
}  
  
#box1 { background-color: lime; }  
#box1:hover { transform: rotate(45deg); }  
  
#box2 { background-color: blueviolet; }  
#box2:hover { transform: rotate(180deg); }  
  
#box3 { background-color: indianred; }  
#box3:hover { transform: rotatex(45deg); }  
  
#box4 { background-color: violet; }  
#box4:hover { rotatey(45deg); }
```



- Rotates an element around a fixed point.
- Defined using the `rotate()` function within the `transform` property.
- Default rotation point is the element's center.

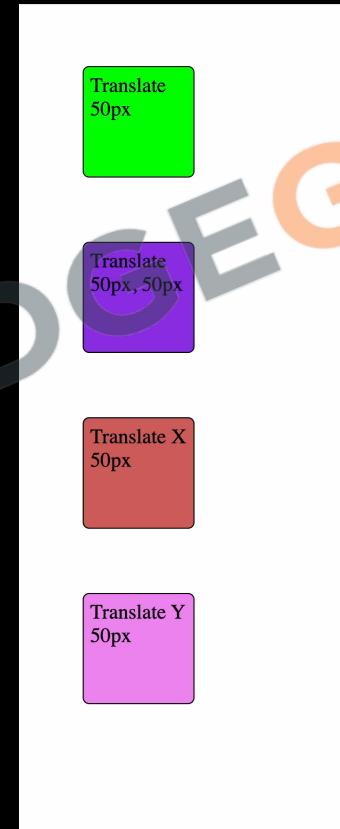
41. CSS Transform (Scale)

```
.box {  
    height: 50px;  
    width: 50px;  
    padding: 5px;  
    margin: 50px;  
    border: 1px solid black;  
    border-radius: 5px;  
    transition-property: all;  
    transition-duration: 1s;  
    transition-timing-function: ease-in-out;  
}  
  
#box1 { background-color: lime; }  
#box1:hover { transform: scale(2); }  
  
#box2 { background-color: blueviolet; }  
#box2:hover { transform: scale(4); }  
  
#box3 { background-color: indianred; }  
#box3:hover { transform: scalex(2); }  
  
#box4 { background-color: violet; }  
#box4:hover { transform: scaley(3); }
```



41. CSS Transform (Translate)

```
.box {  
    height: 75px;  
    width: 75px;  
    padding: 5px;  
    margin: 50px;  
    border: 1px solid black;  
    border-radius: 5px;  
    transition-property: all;  
    transition-duration: 1s;  
    transition-timing-function: ease-in-out;  
}  
  
#box1 { background-color: lime; }  
#box1:hover { transform: translate(50px); }  
  
#box2 { background-color: blueviolet; }  
#box2:hover { transform: translate(50px, 50px); }  
  
#box3 { background-color: indianred; }  
#box3:hover { transform: translateX(50px); }  
  
#box4 { background-color: violet; }  
#box4:hover { transform: translateY(50px); }
```



41. CSS Transform (Skew)

```
.box {  
    display: inline-block;  
    height: 75px;  
    width: 75px;  
    padding: 5px;  
    margin: 50px;  
    border: 1px solid black;  
    border-radius: 5px;  
    transition-property: all;  
    transition-duration: 1s;  
    transition-timing-function: ease-in-out;  
}  
  
#box1 { background-color: lime; }  
#box1:hover { transform: skew(45deg); }  
  
#box2 { background-color: blueviolet; }  
#box2:hover { transform: skew(90deg); }
```



42. Animation

The keyframes at rule rule start with "@keyframes" keyword

animation name which is specified in animation-name property



specify where animation should end.
You can write 100% as well instead of "to"

Specify when the style change will happen.
You can write 0% as well, which is same as "from"

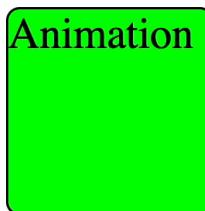
CSS properties.

42. Animation Properties

- **animation-name:** Specifies the name of the @keyframes defined animation.
- **animation-duration:** Defines the total time the animation takes to complete one cycle.
- **animation-timing-function:** Controls the pacing of the animation (e.g., linear, ease-in).
- **animation-delay:** Sets a delay before the animation starts, allowing for a pause before initiation.
- **animation-iteration-count:** Indicates the number of times the animation should repeat.
- **animation-direction:** Specifies the direction of the animation, allowing for reverse or alternate cycles.

42. Animation

```
.box {  
    height: 75px;  
    width: 75px;  
    border: 1px solid black;  
    border-radius: 5px;  
    position: absolute;  
    left: 10;  
    background-color: lime;  
  
    animation-name: ghumakkad;  
    animation-duration: 4s;  
    animation-timing-function: ease-in-out;  
    animation-delay: 0s;  
    animation-iteration-count: 4;  
  
    animation-direction: alternate;  
  
    /* animation: ghumakkad 4s ease-in-out 0s 4  
    alternate; */  
}  
  
@keyframes ghumakkad {  
    from {left: 10px}  
    to {left: 300px}  
}
```



42. Animation

Animation

```
@keyframes ghumakkad {  
    0% {left: 10px; top: 0px}  
    50% { left: 150px; top: 100px }  
    100% {left: 300px; top: 0px}  
}
```

Level 7 Revision

Animation, Transition & Transform

- 39. Pseudo Classes
- 40. Transitions
- 41. CSS Transform
- 42. Animation



Practice Set Level 7 Revision

Animation, Transition & Transform

- Create a webpage with a progress bar that showcases a smooth loading animation. The progress bar should fill up from 0 to 100% with a smooth transition effect and a slight bounce when it reaches 100%.



Project

Myntra GATE Clone



e-KNOWLEDGE GATE