

CSC-336

Web Technologies

Lecture 2:

Topics:

- Lists, Links, Images & Multi-Page Websites, File Path, BoilerPlate
- Introduction to CSS, How to Add CSS, CSS Selectors,
- CSS Properties, Inspect Element

Muhammad Naveed Shaikh

Department of Computer Engineering
naveedshaikh@cuiatd.edu.pk

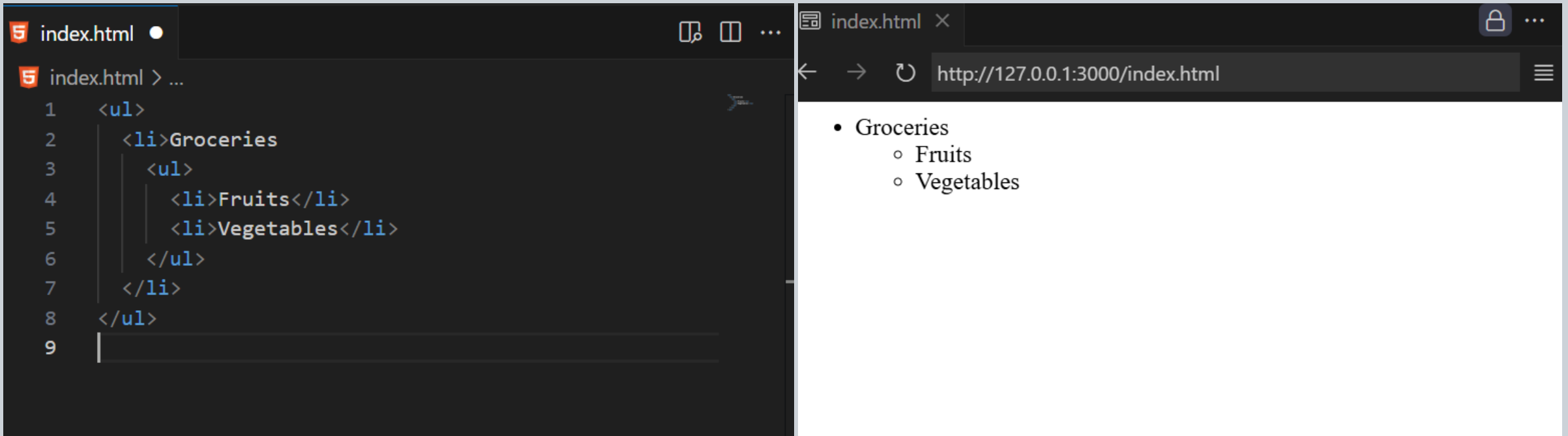


Topics from Previous Week

- HTML Structure
 - Headings,
 - Paragraphs
 - Formatting ,
 and <hr> tags
 - Lists, Anchor, Image element

Lists in HTML

- Types of Lists
 - Unordered Lists () → Bulleted points
 - Ordered Lists () → Numbered points
 - Nested Lists → A list inside another list



The image displays a side-by-side comparison of HTML code and its rendered output. On the left, a code editor window titled 'index.html' shows the following code:

```
1 <ul>
2   <li>Groceries
3     <ul>
4       <li>Fruits</li>
5       <li>Vegetables</li>
6     </ul>
7   </li>
8 </ul>
9
```

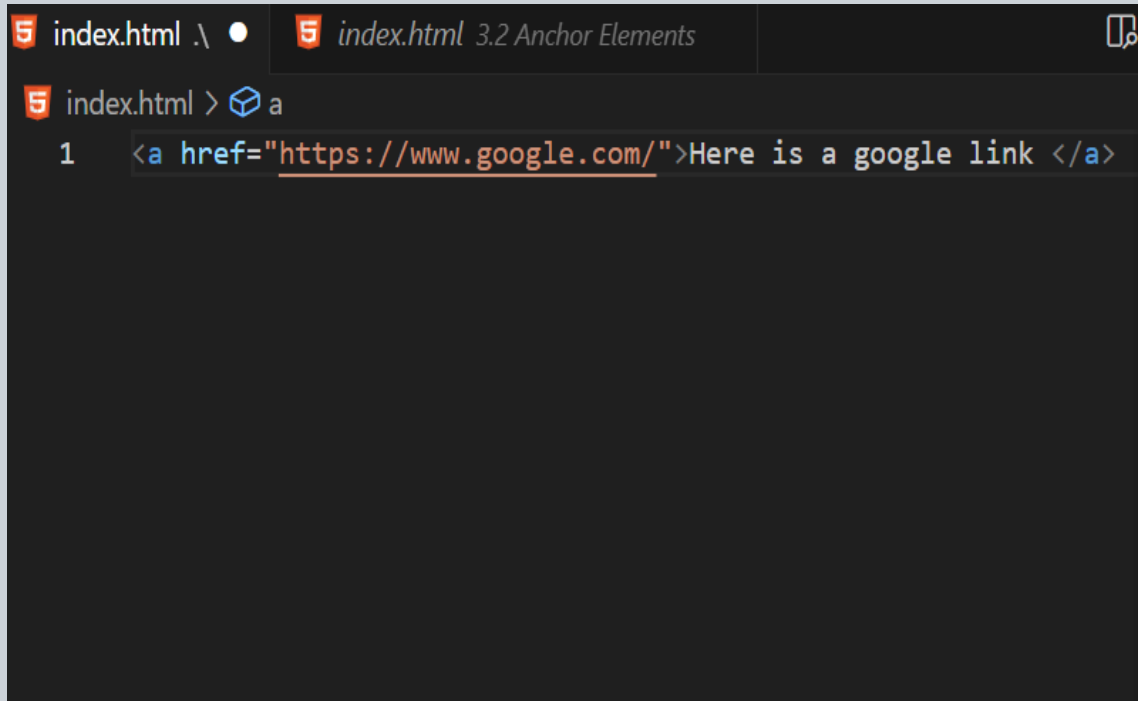
On the right, a web browser window shows the rendered result of this code at the URL 'http://127.0.0.1:3000/index.html'. The browser displays a bulleted list where 'Groceries' is the main item, and 'Fruits' and 'Vegetables' are indented sub-items, each preceded by a small circle.

Practice Problem for Nested List

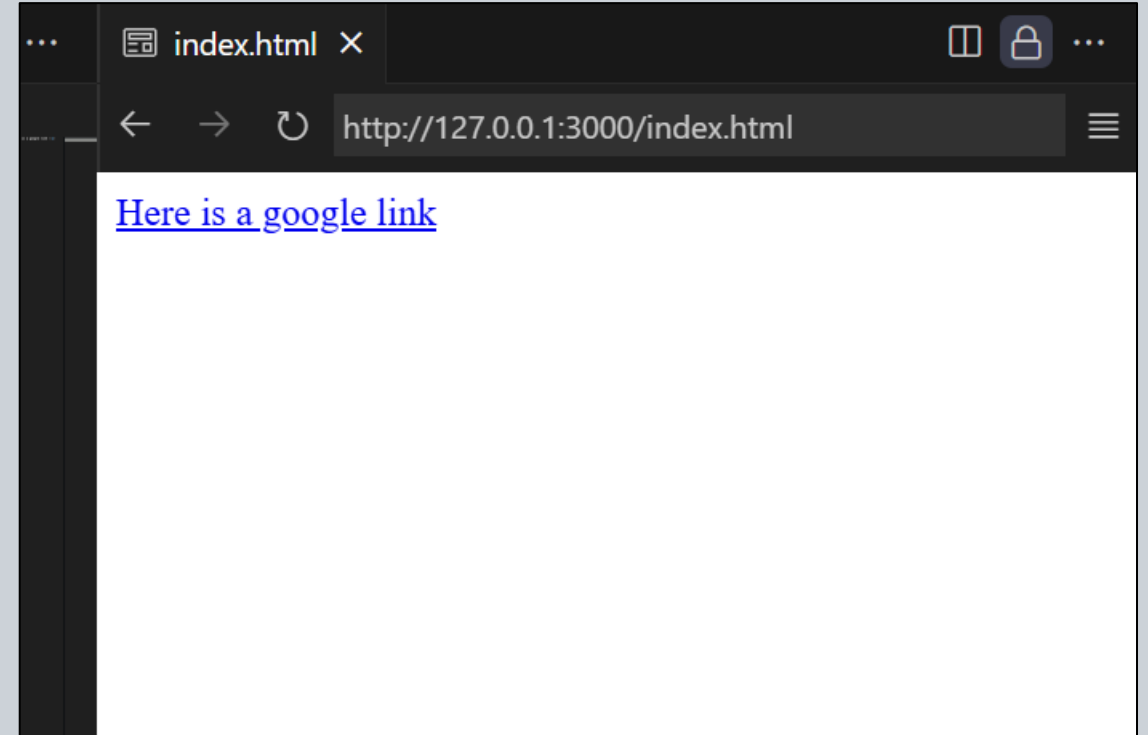
- A
- B
 - 1. B1
 - 2. B2
 - B2a
 - B2aa
 - B2ab
 - B2b
 - B2c
 - 3. B3
 - 1. B31
 - 2. B32
- C

Anchor Element

`<tag attribute="value"> Content</tag>`



A screenshot of a code editor with a dark theme. The top bar shows two tabs: 'index.html .' and 'index.html 3.2 Anchor Elements'. The editor content shows a file named 'index.html' with a cursor at line 1, column 1. The code being typed is `Here is a google link `. The URL is underlined in the code.



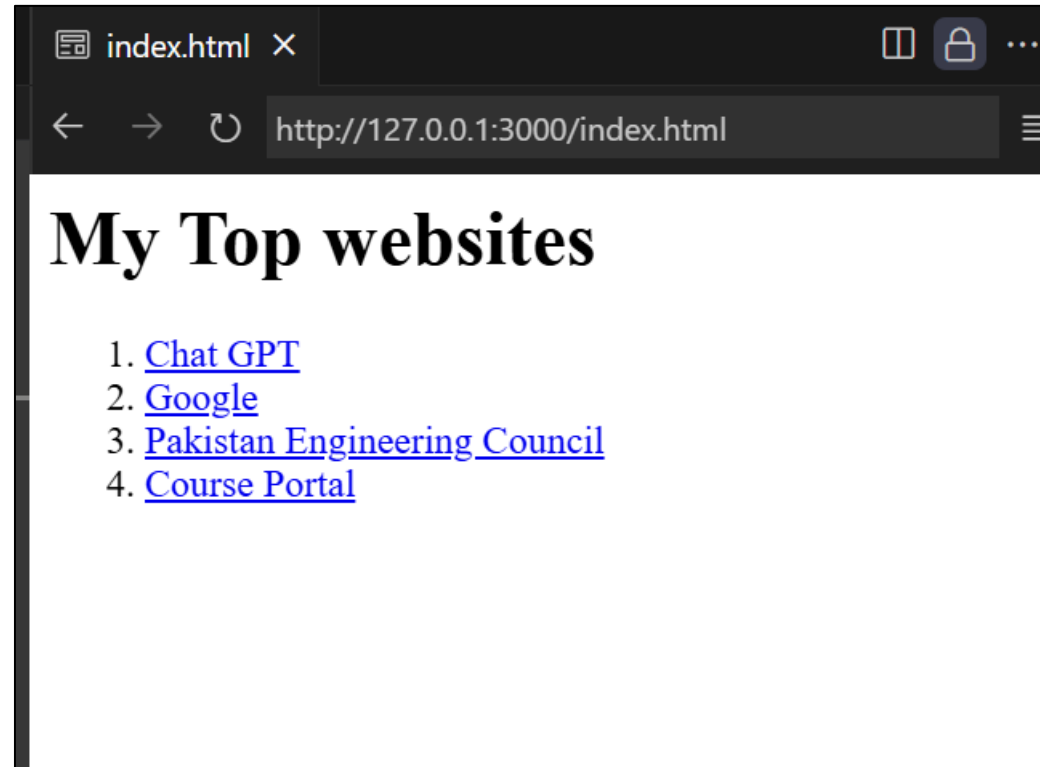
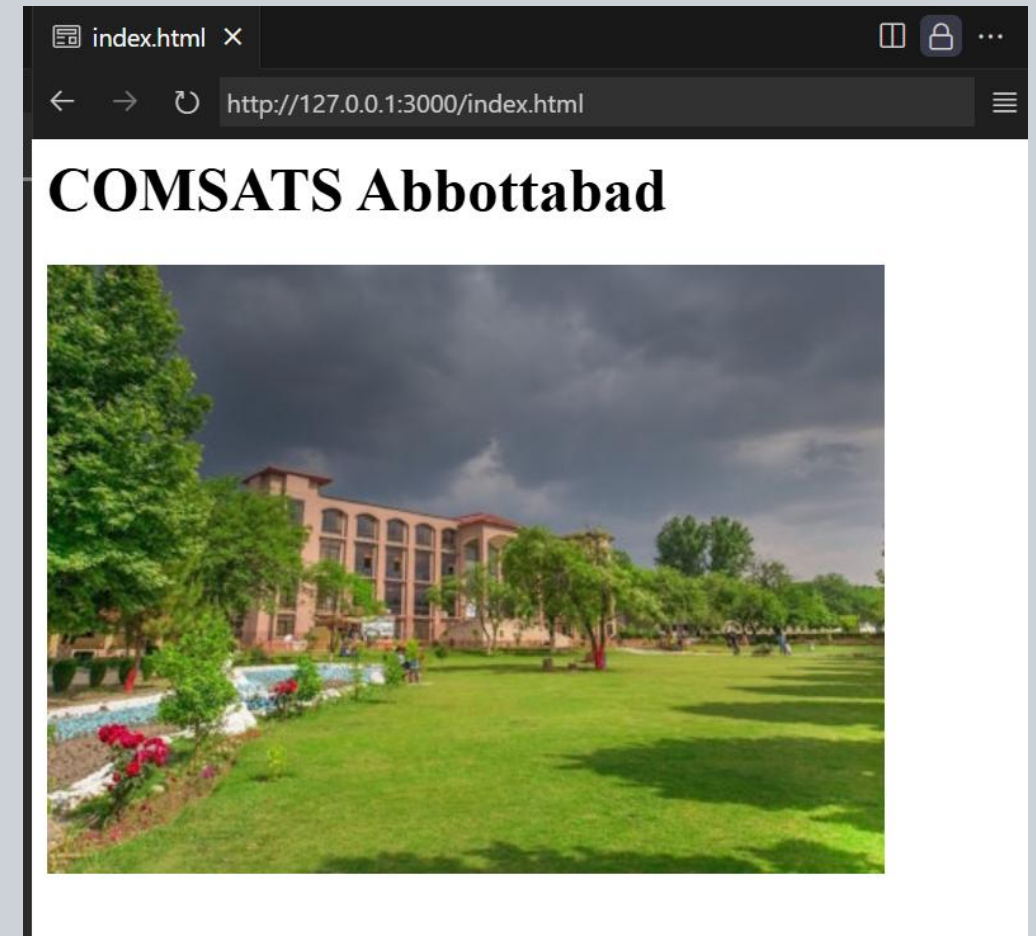


Image Element

- `img < src =“ ” alt=“ ”>`
 - Relative Address (Drive)
 - Absolute Address (Internet)

```
index.html •  
index.html > img  
1 <h1>COMSATS Abbottabad</h1>  
2 
```



Birthday Invite page (Practice Problem)

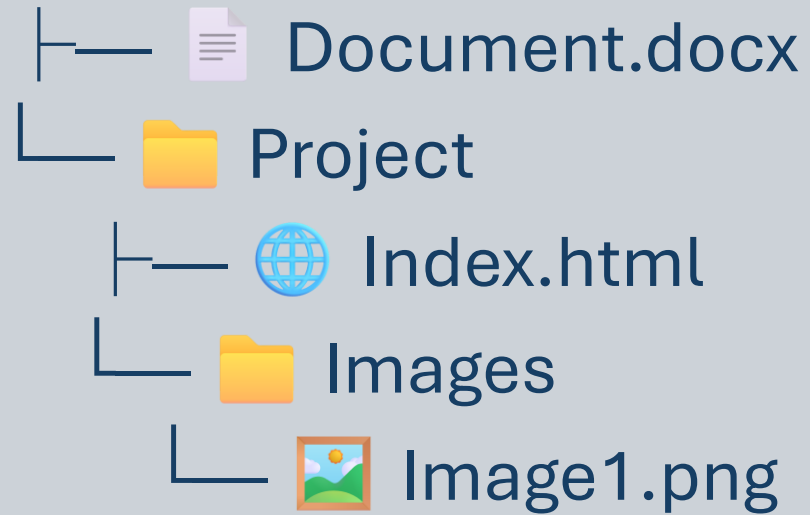
- Birthday Page



File Paths

- Absolute Paths → Relative to Root

 Root



C:/Project/Images/Image.png

/Project/Images/Image.png

File Paths

- Relative Path



Root



Document.docx



Project



Index.html



Image



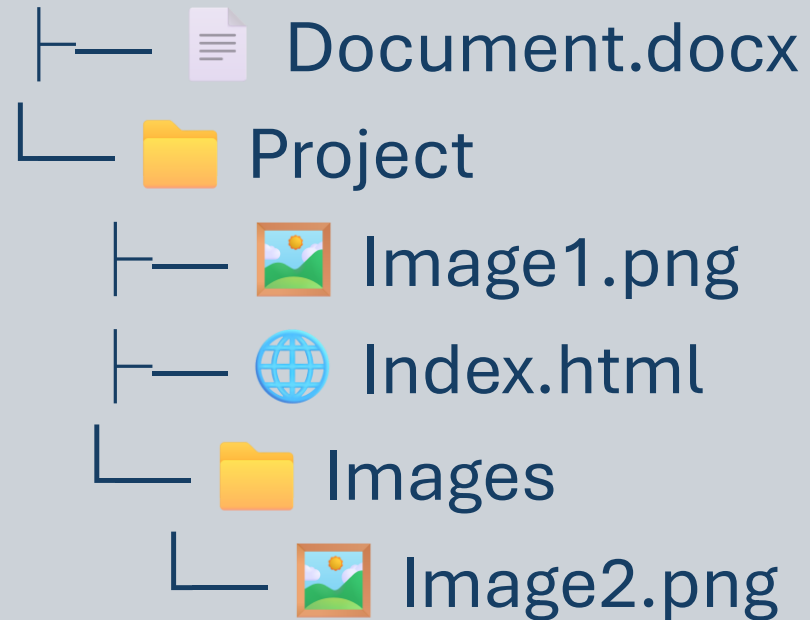
Image1.png

/Image/Image1.png

File Paths

- Special Characters

 Root



../Document.docx

./Image1.png

./Images/Image2.png

- Upper Level (../)
- Current Directory (./)

Multi-Page Website

- It is good practice to build multi page website



HTML Boilerplate

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>Website Name</title>
  </head>

  <body>
    <h1> Hello World!</h1>
  </body>
</html>
```

<!DOCTYPE html>

It defines the html version.

```
<!DOCTYPE html>
```

```
<html lang="en">
```

it defines the language of the web

```
</html>
```

HTML Boilerplate

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
  <head>
```

it defines the all important settings of the webpage

```
  </head>
```

```
</html>
```


HTML Boilerplate

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
  <head>
```

```
    <meta charset="UTF-8">
```

```
  </head>
```

```
</html>
```

Metadata = data about data,
information about the document that
browsers, search engines, and social media use.

UTF-8 :

Ensures text displays correctly, supports emojis &
multiple languages

HTML Boilerplate

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
  <head>
```

```
    <meta charset="UTF-8">
```

```
    <title>Website Name</title>  it is the name of webpage.
```

```
  </head>
```

```
</html>
```

HTML Boilerplate

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>Website Name</title>
  </head>

  <body>

  </body>
</html>
```

It contains actual web content to be displayed on webpage

HTML Boilerplate

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>Website Name</title>
  </head>

  <body>
    <h1> Hello World!</h1>
  </body>
</html>
```

- What is CSS?
 - CSS = Cascading Style Sheets
 - Cascading → Styles flow down like water
 - Browser chooses the winning style
 - when there are multiple rules for the same element.
- Decision order:
Importance → Specificity → Order

CSS Cascading Explained

How Styles Flow Like a Waterfall

- CSS = Cascading Style Sheets
- Cascading → Styles flow down like water
- Browser chooses the winning style when there are multiple rules for an element
- Decision order:
Importance → Specificity → Order



External CSS
(Lowest Priority)

```
style.css  
p { color: green; }
```

Internal CSS

```
#intro  
{ color: blue; }
```

Inline CSS
(Highest Priority)

```
<p style='color: red;'>
```

Final Applied Style
Color = **Red**

How to ADD CSS

- Three ways:

Inline

```
<tag style="css"/>
```

Internal

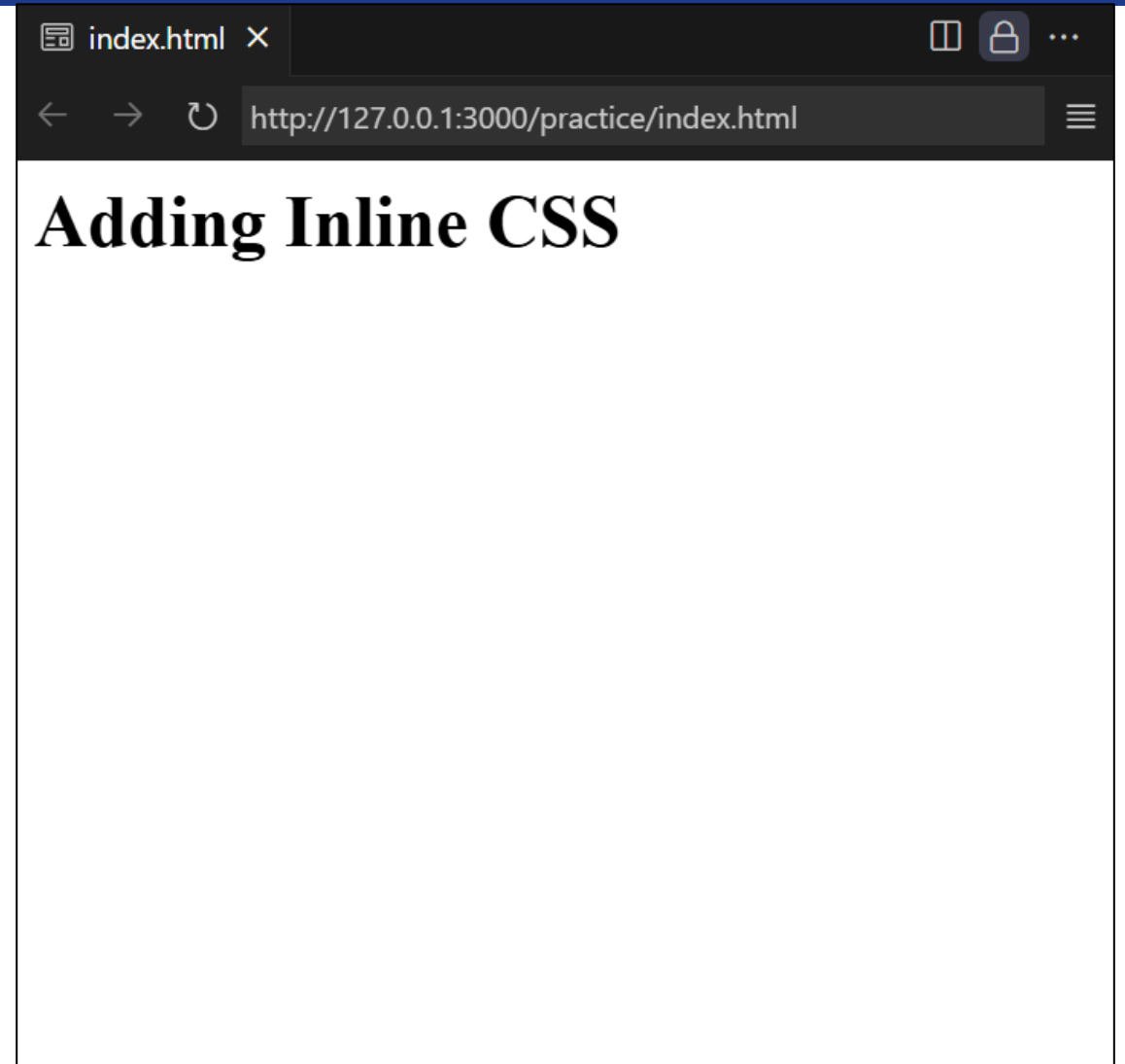
```
<style>css</style>
```

External

```
<link href="style.css"/>
```

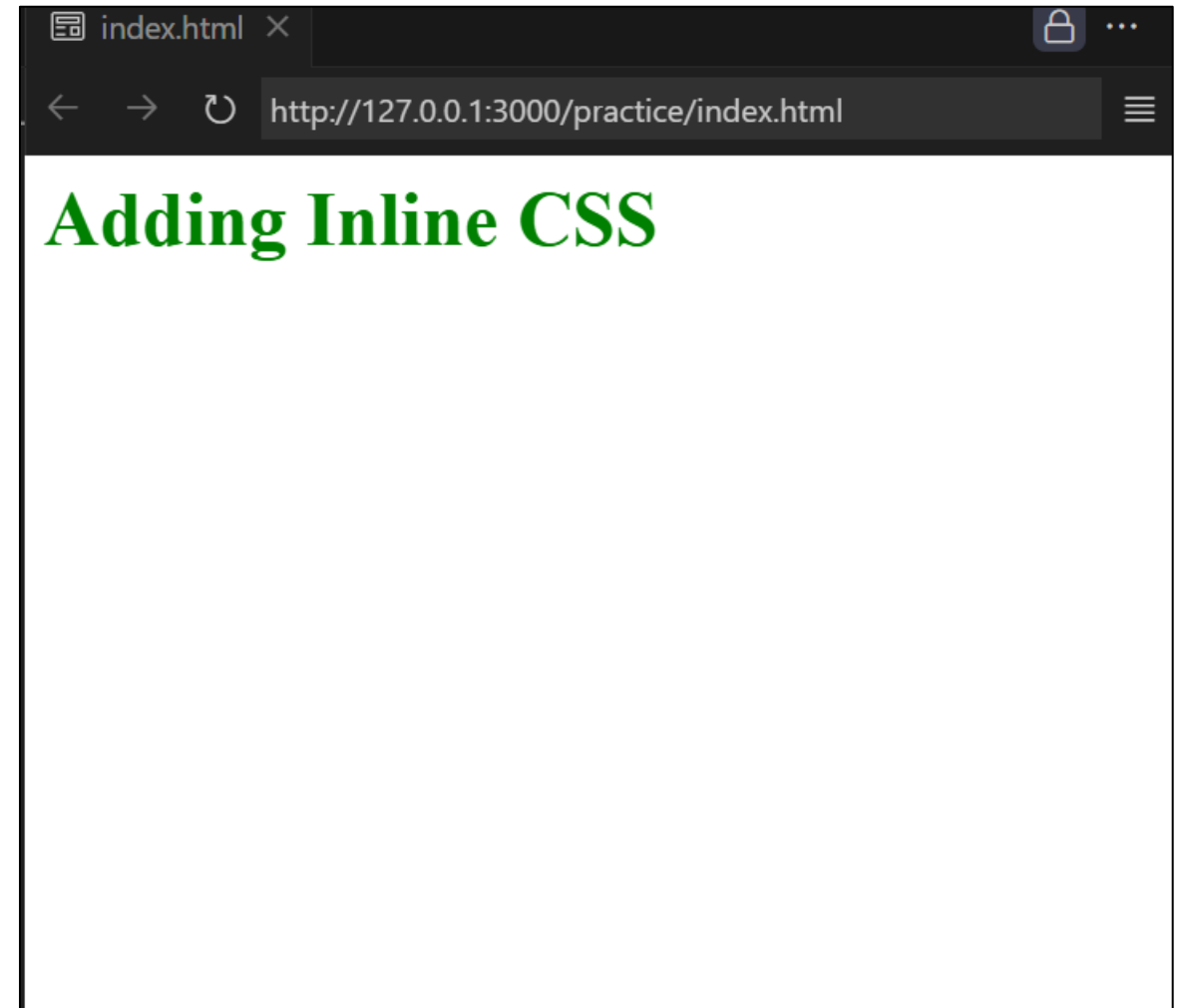
Inline CSS

`<h1>Adding Inline CSS</h1>`



Inline CSS

```
<h1 style="color: green;">Adding  
Inline CSS</h1>
```



Internal

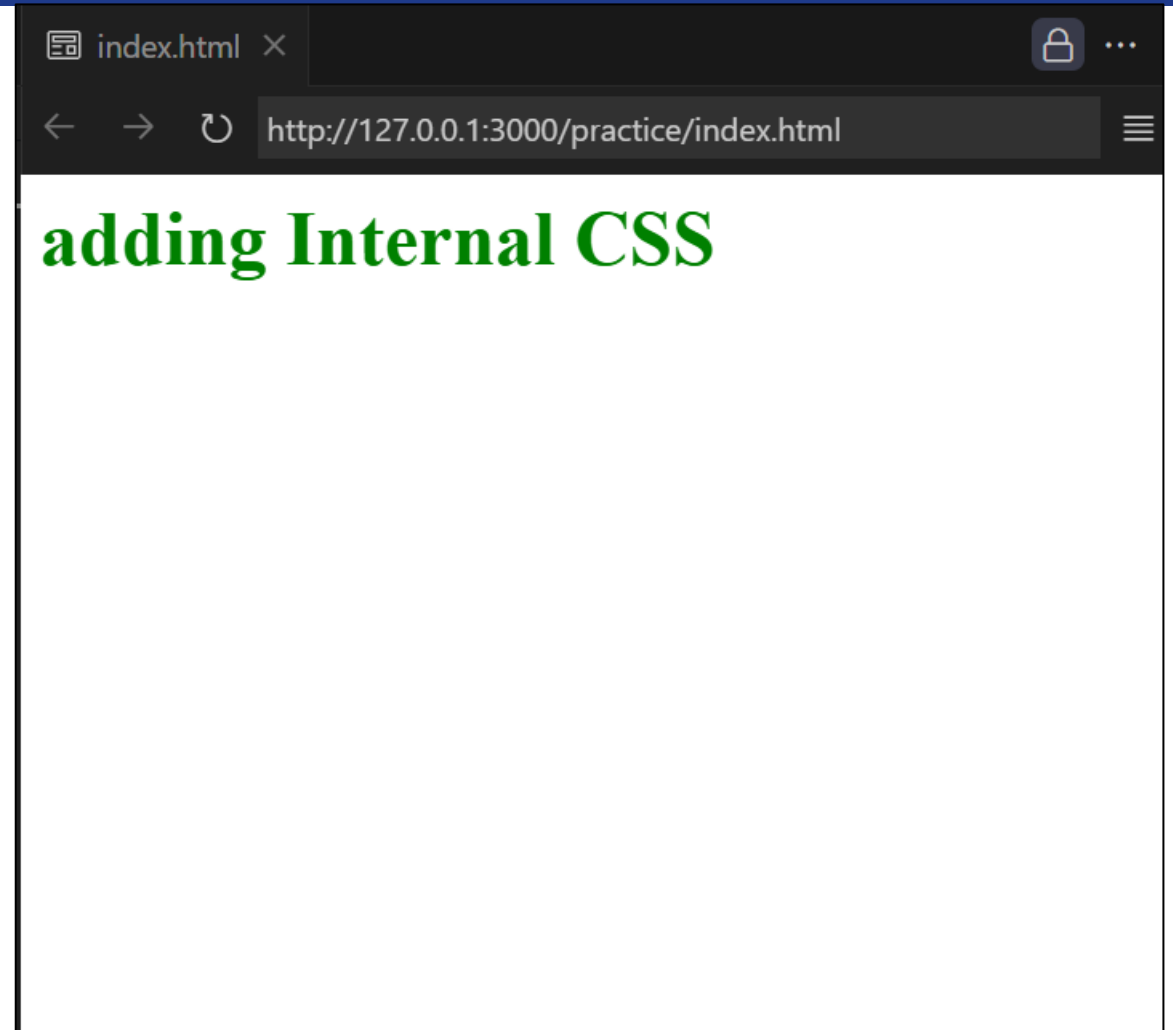
```
<html>  
  <head>  
  
  </head>  
  
  <body>  
    <h1>adding Internal CSS</h1>  
  </body>  
</html>
```



Internal

```
<html>
  <head>
    <style>
      h1{ color:green }
    </style>
  </head>

  <body>
    <h1>adding Internal CSS</h1>
  </body>
</html>
```



External

```
index.html • styles.css
practice > index.html > html
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Document</title>
7
8
9 </head>
10 <body>
11   <h1>Adding CSS Externally</h1>
12 </body>
13 </html>
```

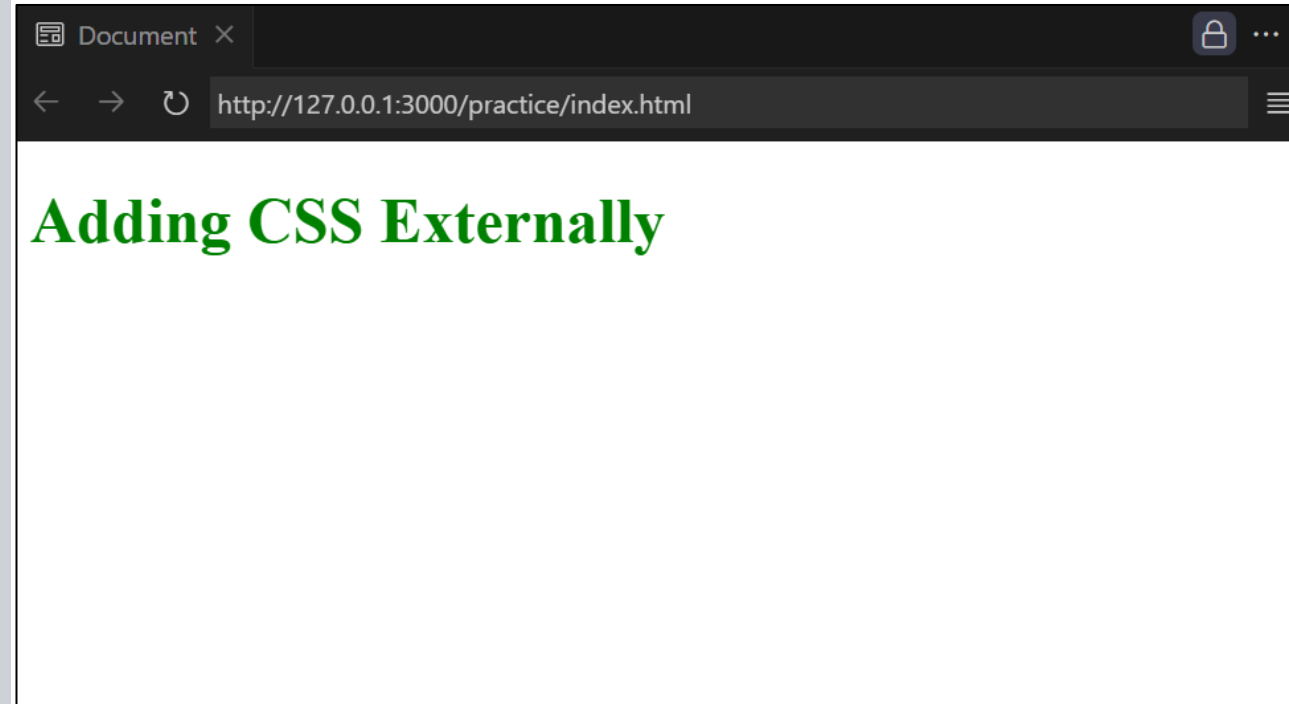


External

- Two files

```
index.html • styles.css
practice > index.html > html
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Document</title>
7   <link rel="stylesheet" href="./styles.css">
8
9 </head>
10 <body>
11   <h1>Adding CSS Externally</h1>
12 </body>
13 </html>
```

```
index.html • styles.css
practice > styles.css > ...
1
2   h1{ color: green;}
3
```



CSS Selectors

- Definition:
CSS selectors are patterns used to select HTML elements to apply styles.
- They connect HTML and CSS by telling the browser which elements to style.
- Without selectors, CSS wouldn't know where to apply rules.

ELEMENT SELECTORS

```
h1 { color: blue;}
```

- Syntax:

```
element {CSS Rule;}
```

Examples:

- `p{text-align:center;}`
- `h2{background-color: orange;}`

ELEMENT SELECTORS

```
h1 { color: blue;}
```

- Syntax:

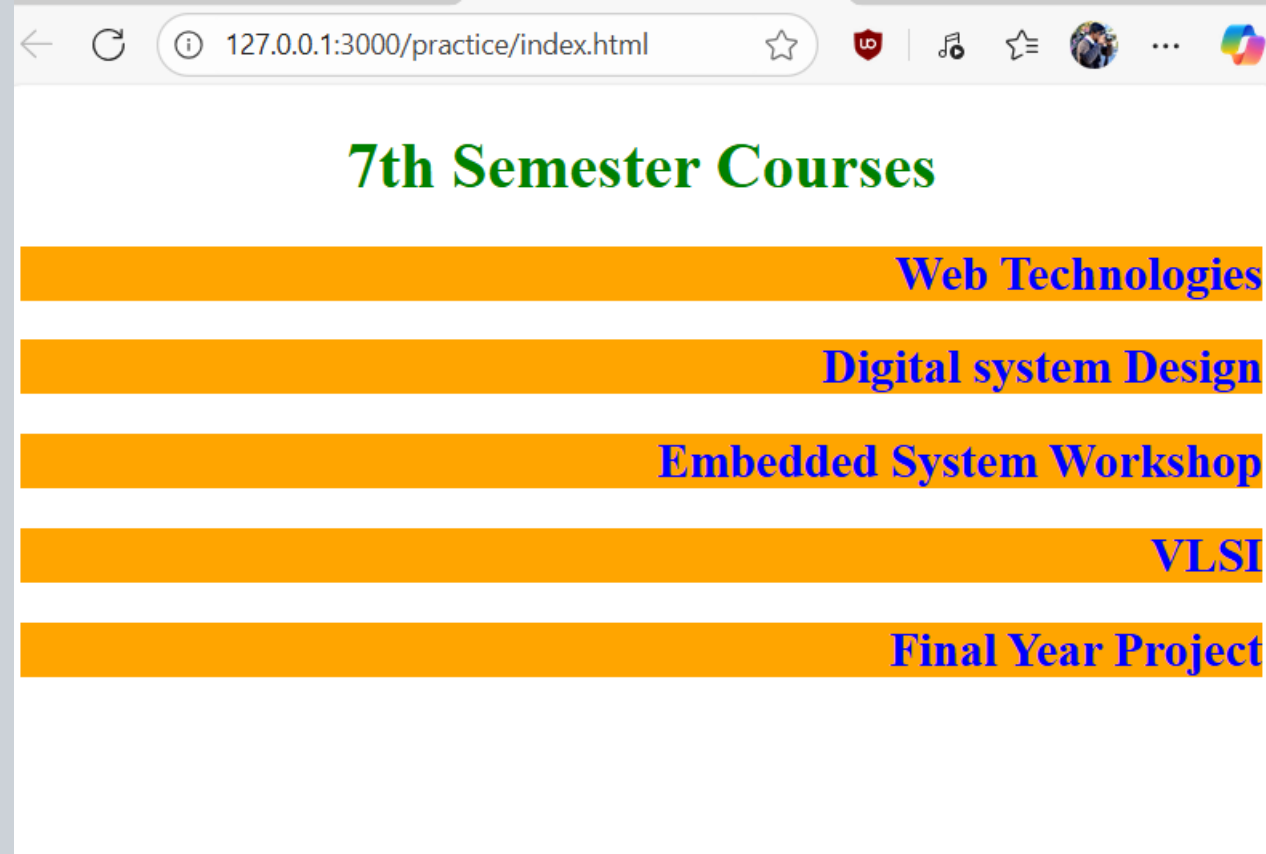
```
element {CSS Rule;}
```

Examples:

- `p{text-align:center;}`
- `h2{background-color: orange;}`

Practice Case Study#1

- Design this web page using only Element Selector



Class SELECTOR

```
.Green_Heading { color: green;}
```

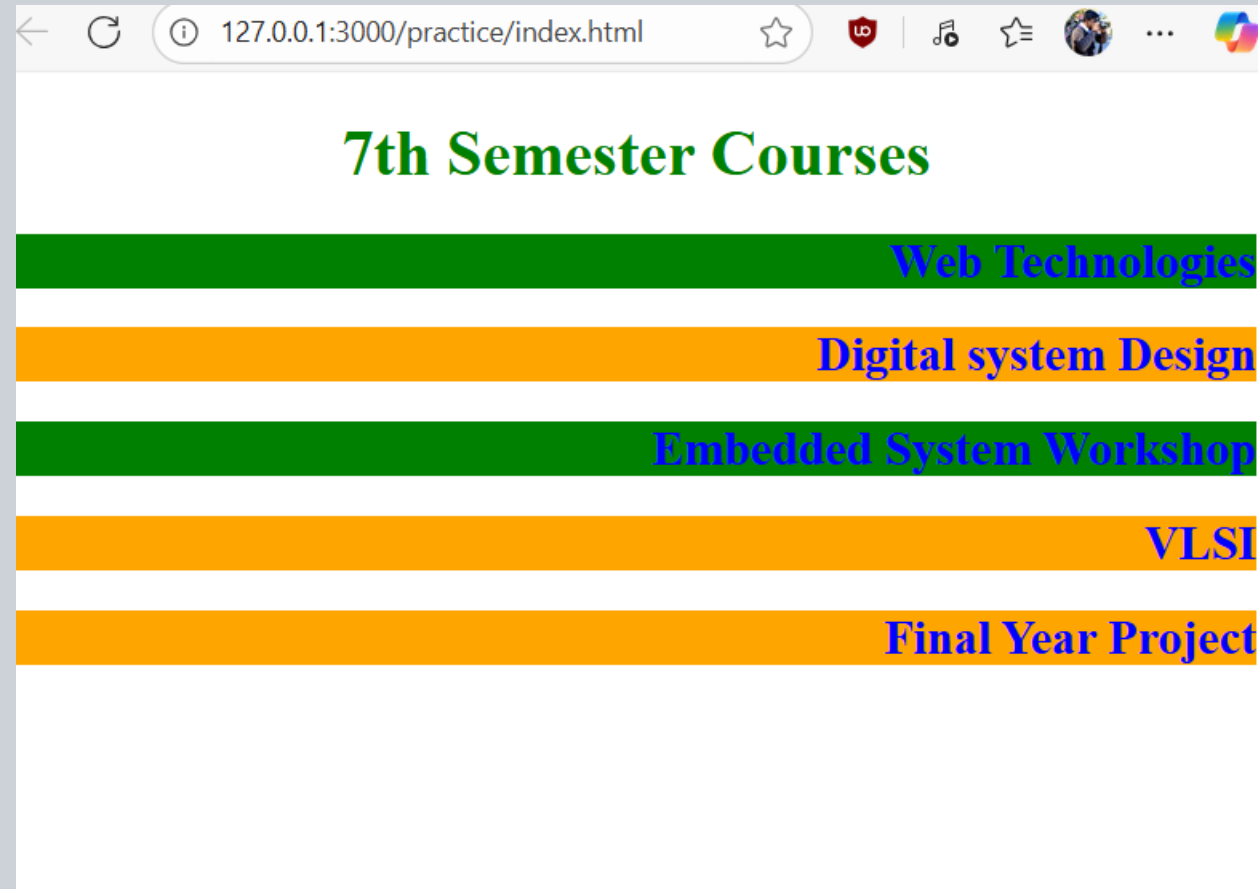
- Syntax:

```
.classname{CSS Rule;}
```

classname is attribute defined using the class keyword in any html element

Practice Case Study#2

- Modify the case study#1 using class selector



Class Selector

- Multiple element can have same class

```
<h1 class="red-color"> Heading1 </h1>  
<p class="red-color"> paragraph </p>
```

- One tag/element can be declared in multiple classes

```
<h1 class="blue"> heading1 </h1>  
<h2 class="blue col"> column1</h2>  
<h3 clas="col"> heading3</h3>
```

ID SELECTOR

#Green { color: green;}

- Syntax:

#id_name{CSS Rule;}

Id_name is attribute with keyword ID defined in any html element.

ID SELECTOR

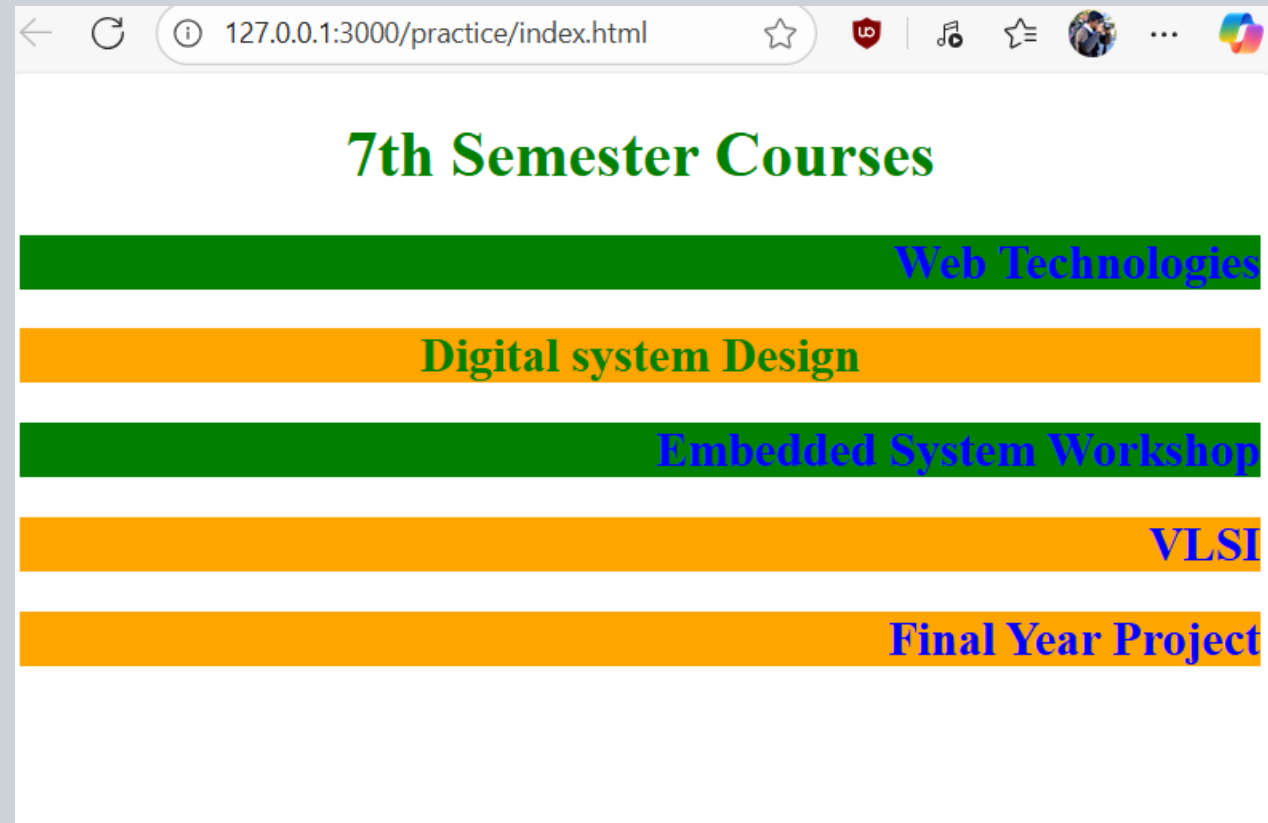
- Some of the examples of class definitions.

```
<h1 id="one" class="red-color" > Heading1 </h1>  
<p id="Two" class="red-color"> paragraph </p>  
<h1 id="card" class="blue"> heading1 </h1>  
<h2 id="item" class="blue col"> column1</h2>  
<h3 clas="col"> heading3</h3>
```

- Key difference between class and ID selector:
 - ID is unique to each element
 - While many element can be assigned to same class.

Practice Case Study#3

- Modify the case study#2 using ID selector



Attribute Selector

```
p[href="https://www.google.com"] { color: green; }
```

- Syntax:

```
element[attr]{CSS Rule}
```

attr → attribute

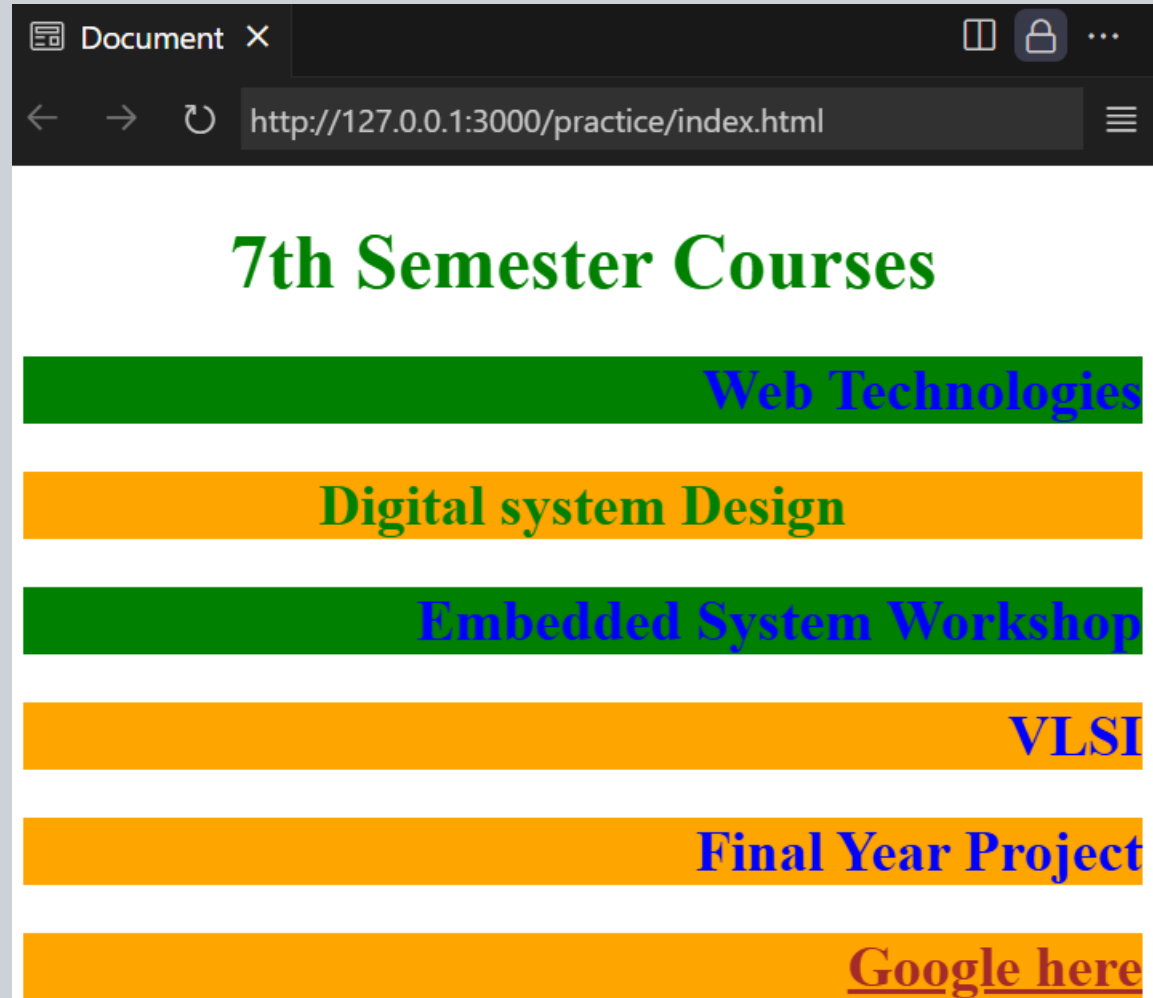
```
element[attr=value]{CSS Rule}
```

attr → attribute, value → attribute value

The CSS **attribute selector** matches elements based on the element having a given attribute explicitly set, with options for defining an attribute value or substring value match.

Practice Case Study#4

- Modify the case study#3 using attribute selector



Universal Selector

```
*{ color: green;}
```

- Syntax:

```
*{CSS Rule}
```

The CSS universal selector (*) matches elements of any type.

Practice Case Study#5

- Modify the case study#4 using universal selector
- Hint: italic and font size



CSS Properties-CSS Colors

- Text color
`h1{ color : green;}`
- Background color
`.h2{background-color: blue;}`
- Links: MDN
 - <https://developer.mozilla.org/en-US/docs/Web/CSS/named-color>
 - <https://colorhunt.co/> freely themes for web designs

CSS Properties- Font Properties

- Font-size = 20px
- 1px = 1/96 th inch = 0.26mm → square = (height * width) → height = width = 1px
- 1pt = 1/72th inch = 0.35mm → height = width = 0.35mm
 - Try to remember 12pts in Microsoft word
- 1em = 100% of parent
- 1rem = 100% of root

CSS Properties- Font Properties

- For this code
 - Font size of h1=??
 - Font size of h2=??

```
<html>
  <head>
    <style>
      html{font-size: 10px;}
      body{ font-size: 20px;}
      h1{ font-size: 2em;}
      h2{font-size:1rem;}
    </style>
  </head>
  <body>
    <h1> Hading1</h1>
    <h2>Heading2</h2>
  </body>
</html>
```

1. Keyword

- `h2{font-weight: normal}`
- Bold/Normal

2. Relative to Parent/Root

- Bolder/lighter (+100/-100)

3. Number

- 100-900

CSS-Properties (Font-family)

h1{font-family: Helvetica, Sans-serif}

```
h1{  
font-family: Helvetica, Sans-serif  
}
```

- In case of multiword name for family.
 - Use quotes like “Times New Roman”
- Custom font
 - <https://fonts.google.com/>
 - Use this method to link the any of your own choice font family in webpage.

- Open Google Chrome (or any modern browser).
- Right-click anywhere on the page → select Inspect.
- OR press the shortcut keys:
 - Ctrl + Shift + I (Windows/Linux)
 - CMD+ Option + I (Mac)
- Developer Tools panel will open (usually at bottom or right side).
- Explore tabs like Elements, Style, Computed and CSS overview
 - How to add new CSS Rule
- Important links by Angella Yu Development course
 - <https://appbrewery.github.io/just-add-css/>
 - <https://appbrewery.github.io/css-inspection/>

Credit: Angela Yu & App Brewery (Udemy Course)