



Tutorial 3: HTML and CSS

CS 104

Spring, 2024-25

TA: Kritin Gupta

Credits: Sabyasachi & Guramrit Singh



Topics

- Visual Studio Code
- HTML tags (review)
- CSS
- Exercises



Visual Studio Code

- Introduction
- Demo

VS Code



Some pros of **VSCode** :-

- ❖ It is free and open-source.
- ❖ It is cross-platform, so it can be used on Windows, macOS, and Linux.
- ❖ It is extensible, with a large number of extensions available to add new features and functionality.
- ❖ It is customizable, so you can change the look and feel to suit your preferences.
- ❖ It is lightweight and fast, so it can be used on even older computers.
- ❖ It has a large and active community, with a lot of resources available online.
- ❖ It is a popular choice for web development, data science etc
- ❖ It has built-in git support
- ❖ It has intellisense, an intelligent code completion feature.

Installation

- ❖ For [Ubuntu](#), [Windows](#), [MacOS](#)

```
aria@aria-IdeaPad-Slim-5-14IAH8:~/Desktop$ sudo snap install --classic code
[sudo] password for aria:
code 0ee08df0 from Visual Studio Code (vscode**) installed
aria@aria-IdeaPad-Slim-5-14IAH8:~/Desktop$ code --version
1.85.1
0ee08df0cf4527e40edc9aa28f4b5bd38bbff2b2
x64
aria@aria-IdeaPad-Slim-5-14IAH8:~/Desktop$ mkdir CS104
aria@aria-IdeaPad-Slim-5-14IAH8:~/Desktop$ cd CS104
aria@aria-IdeaPad-Slim-5-14IAH8:~/Desktop/CS104$ touch homepage.html
aria@aria-IdeaPad-Slim-5-14IAH8:~/Desktop/CS104$ code .
aria@aria-IdeaPad-Slim-5-14IAH8:~/Desktop/CS104$
```

Useful Features

- An integrated terminal
- Auto-save
- Syntax highlighting
- Many, many extensions like IntelliSense, Live Preview
- VS Code IntelliSense is provided for JavaScript, TypeScript, JSON, HTML, CSS, SCSS, and Less out of the box.
- Can have a richer IntelliSense by installing language extensions





HTML tags (review)

- VSCode preview extension
- Tags

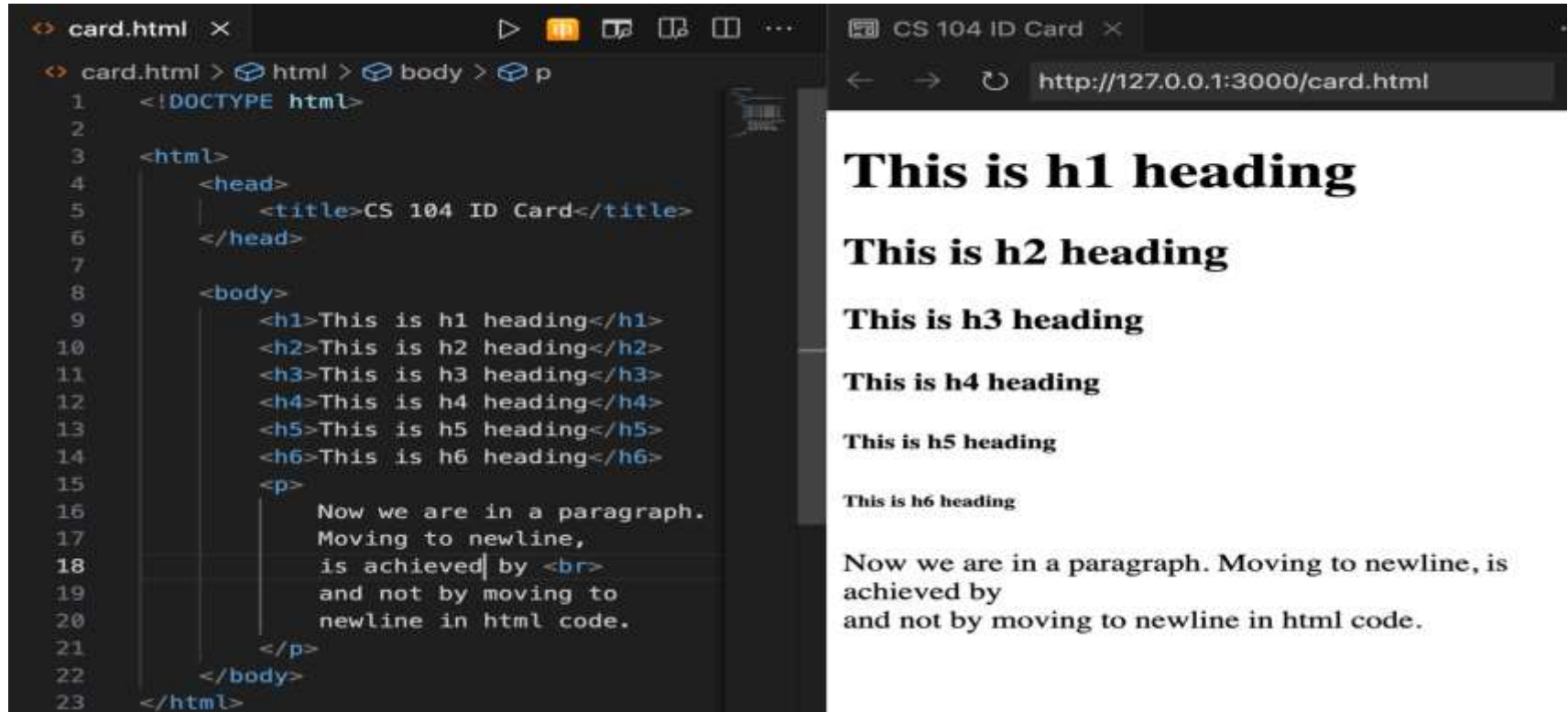
HTML Document

<> card.html >  html >  head

```
1  <!DOCTYPE html>
2
3  <html>
4      <head>
5      |
6      </head>
7
8      <body>
9
10     </body>
11 </html>
```

- ❖ `<!DOCTYPE html>` declares the document type to be **HTML**.
- ❖ The HTML document is enclosed between `<html>` and `</html>` tags.
- ❖ HTML document is divided in two parts, head and body.
- ❖ The head element enclosed within `<head>` and `</head>` represents a collection of metadata for the Document.
- ❖ The body element present in `<body>` and `</body>` represents the content of the document.

Title, headings and paragraphs.



The image shows a side-by-side comparison of an HTML file in a code editor and its rendered output in a web browser. The code editor on the left shows the HTML structure of 'card.html', including the DOCTYPE, head with a title 'CS 104 ID Card', and a body containing six headings (h1 to h6) and a paragraph with a line break. The browser on the right displays the rendered page, where the headings are formatted in bold and increasing sizes, and the paragraph text is wrapped across multiple lines.

```
card.html x
card.html > html > body > p
1 <!DOCTYPE html>
2
3 <html>
4   <head>
5     <title>CS 104 ID Card</title>
6   </head>
7
8   <body>
9     <h1>This is h1 heading</h1>
10    <h2>This is h2 heading</h2>
11    <h3>This is h3 heading</h3>
12    <h4>This is h4 heading</h4>
13    <h5>This is h5 heading</h5>
14    <h6>This is h6 heading</h6>
15    <p>
16      Now we are in a paragraph.
17      Moving to newline,
18      is achieved by <br>
19      and not by moving to
20      newline in html code.
21    </p>
22  </body>
23 </html>
```

CS 104 ID Card x
http://127.0.0.1:3000/card.html

This is h1 heading

This is h2 heading

This is h3 heading

This is h4 heading

This is h5 heading

This is h6 heading

Now we are in a paragraph. Moving to newline, is achieved by
and not by moving to newline in html code.

- ❖ The title of the webpage is written within `<title>` and `</title>` in the head element
- ❖ There are 6 heading levels, from `<h1>` to `<h6>`, largest to smallest.
- ❖ The paragraph is written inside `<p>` and `</p>`, whereas `
` is used to break line.

Tables, Lists and Formatting

The screenshot displays a code editor on the left and a web browser on the right. The code editor shows the HTML source code for a file named 'tables.html'. The browser displays the rendered output of this code.

Code Editor Content:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>CS100 Tut-3</title>
5 </head>
6 <body>
7 <h3>Major Attractions of India</h3>
8 <table>
9 <tr>
10 <th>City</th>
11 <th>Food</th>
12 <th>Place</th>
13 </tr>
14 <tr>
15 <td>Cutack</td>
16 <td>Dahi Vada</td>
17 <td>Barabati Fort</td>
18 </tr>
19 <tr>
20 <td>Hyderabad</td>
21 <td>Dum Biryani</td>
22 <td>Charminar</td>
23 </tr>
24 <tr>
25 <td>Amritsar</td>
26 <td>Kulcha</td>
27 <td>Golden Temple</td>
28 </tr>
29 </table>
30 <p>Based on the table above we conclude some points</p>
31 <ol>
32 <li>Amritsar is famous for Biryani</li>
33 <li>Cutack is famous for</li>
34 <li>Barabati Fort</li>
35 <li>Dahi Vada</li>
36 </ol>
37 <p>India is beautiful</p>
38 <p>25<sup>2</sup> = 625</p>
39 <dl>
40 <dt>Barabati Fort</dt>
41 <dd>Barabati Fort is a 987 CE fort built by Marakata Keshari of Somavanshi dynast</dd>
42 <dt>Charminar</dt>
43 <dd>The Charminar (four minarets) is a monument located in Hyderabad, Telangana,</dd>
44 <dt>Golden Temple</dt>
45 <dd>Golden Temple is a gurudwara located in Amritsar, Punjab.</dd>
46 </dl>
47 </body> </html>
```

Browser Content:

Major Attractions of India

City	Food	Place
Cutack	Dahi Vada	Barabati Fort
Hyderabad	Dum Biryani	Charminar
Amritsar	Kulcha	Golden Temple

Based on the table above we conclude some points

- Amritsar is famous for Biryani Kulcha
-
- Cutack is famous for
 - Barabati Fort
 - Dahi Vada
- India is beautiful
- $p_0 = 25^2 = 625$

Barabati Fort

Barabati Fort is a 987 CE fort built by Marakata Keshari of Somavanshi dynasty in Cutack, Odisha.

Charminar

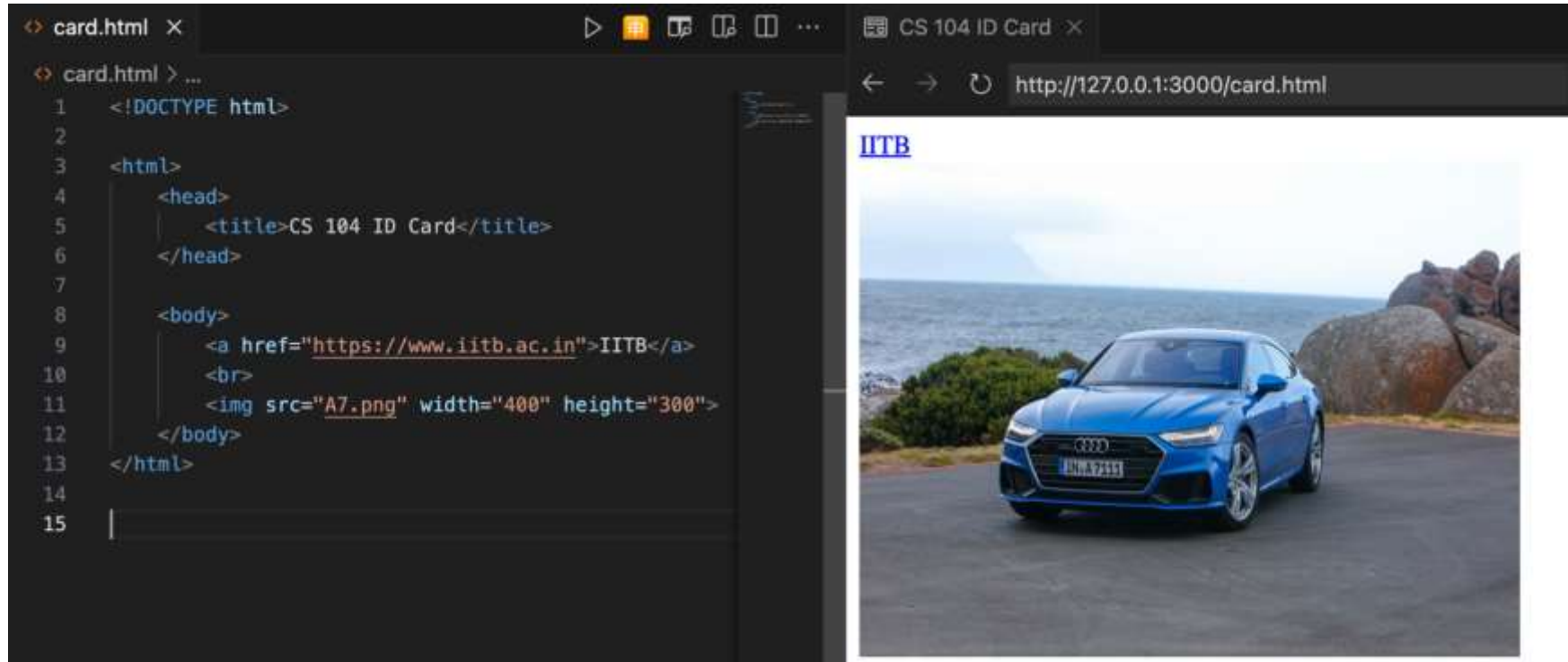
The Charminar (four minarets) is a monument located in Hyderabad, Telangana, India.

Golden Temple

Golden Temple is a gurudwara located in Amritsar, Punjab.

- ❖ Tables are created using `<table>` tag. `<tr>` represents table rows. `<th>` and `<td>` are table header and table data respectively.
- ❖ List items are defined inside `` tags. `` creates ordered lists, and `` creates unordered lists.
- ❖ Try out nesting `` inside `` and same for ``!
- ❖ Definition lists `<dl>` creates a list of terms with definition.
- ❖ `` (bold) and `<i>` (italic) are text styling tags.
- ❖ `<sub>` and `<sup>` produce subscripting and superscripting respectively.
- ❖ `<mark>`, `<ins>` and `` are used to highlight, underline and strikethrough

Links and images




- ❖ The url is written in `href` attribute of `<a>` tag, text enclosed in between `<a>` and `` is displayed on the webpage. When clicked, the page redirects to the url, to open on a new tab, set `target` attribute to `"_blank"`.
- ❖ The image location is given to the `src` attribute of `` tag, height and width can be adjusted.


Videos and iframe

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <title>Videos</title>
5 </head>
6 <body>
7   <p>The following shows a video from our local directory</p>
8   <br>
9   <video src="video.mp4" height="200" controls></video>
10  <br>
11  <p>The following is fetched from youtube and is embedded
12  using iframe</p>
13  <iframe width="400" height="300"
14  src="https://www.youtube.com/embed/shxiermazzq?si=atrjh_cnuozvll0a"
15  title="Youtube video player"
16  frameborder="0"
17  allow="accelerometer; autoplay; clipboard-write; encrypted-media;
18  gyroscope; picture-in-picture; web-share"
19  referrerpolicy="strict-origin-when-cross-origin" allowfullscreen></iframe>
20 </body>
21 </html>
```

The following shows a video from our local directory



The following is fetched from youtube and is embedded using iframe



- ❖ Like images, video location is given to the **src** attribute of the **<video>** tag. The **controls** attribute adds video controls like play, pause and volume.
- ❖ The **<iframe>** tag can be used to display a youtube video in the web page.

Favicon and Comments

<> favicon.html x

<> favicon.html > html

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <title>CS108 Webpage</title>
5   <!-- Favicons are small icons that appear in the browser tab of your webpage. -->
6   <link rel="icon" href="logo.png" type="image/x-icon">
7 </head>
8 <body>
9   <p>Favicons are small icons that appear in the browser tab of your webpage.</p>
10 </body>
11 </html>
```



- ❖ Everything between `<!--` and the `-->` will be hidden from the display on the web page.
- ❖ To setup a favicon, we `<link>` our current document to the icon in the `href` attribute, we also specify the type of linked resource to be an icon by setting `rel` and `type` attributes as shown

Forms

```
ms.html > html > body > form > label
<body>
  <form action="/tables.html" method="post">
    <!-- text input -->
    <label for="name">Name</label>
    <input type="text" id="name" name="name">
    <br><br>

    <!-- radio buttons -->
    <label>Grad Year</label>
    <input type="radio" id="2025" name="batch" val
    <label for="2025">2025</label>
    <input type="radio" id="2026" name="batch" val
    <label for="2026">2026</label>
    <input type="radio" id="2027" name="batch" val
    <label for="2027">2027</label>
    <br><br>

    <!-- checkboxes -->
    <label>Courses you've taken</label>
    <input type="checkbox" id="cs101" name="cs101"
    <label for="cs101">CS101</label>
    <input type="checkbox" id="cs108" name="cs108"
    <label for="cs108">CS108</label>
    <input type="checkbox" id="ma106" name="ma106"
    <label for="ma106">MA106</label>
    <br><br>

    <!-- button -->
    <label for="clickme">Clickable Button</label>
    <input type="button" value="Click me" name="cl
    <br><br>

    <!-- submit button -->
    <input type="submit" value="Submit">
  </form>
</body>
```

← → ↺ http://127.0.0.1:3000/forms.html

Name

Grad Year ☐ 2025 ☐ 2026 ☐ 2027

Courses you've taken ☐ CS101 ☐ CS108 ☐ MA106

Clickable Button

- ❖ The `for` attribute in `<label>` corresponds to the `id` attribute of `<input>`.
- ❖ The `name` attribute is used for server side script.
- ❖ In case of buttons and submit input, the `value` attribute determines the text on the button.
- ❖ `<select>` tag lists a dropdown. More on this in Exercise!

- ❖ We use `<form>` to create a **HTML** form, `action` attribute specified the target URL where the data will be sent, `method` attribute defines the **HTTP** method to be used.
 - In case of get, the data is displayed in the URL
- ❖ The `<input>` tags are used to take any input from user side. The attribute `type` specifies what kind of input. By default it is **"text"**
 - `type = "radio"`: Single Choice
 - `type = "checkbox"`: Multi Choice
 - `type = "button"`: Clickable button
 - `type = "submit"`: Submitting form data to a handler that is specified in the action attribute of `<form>`



CSS

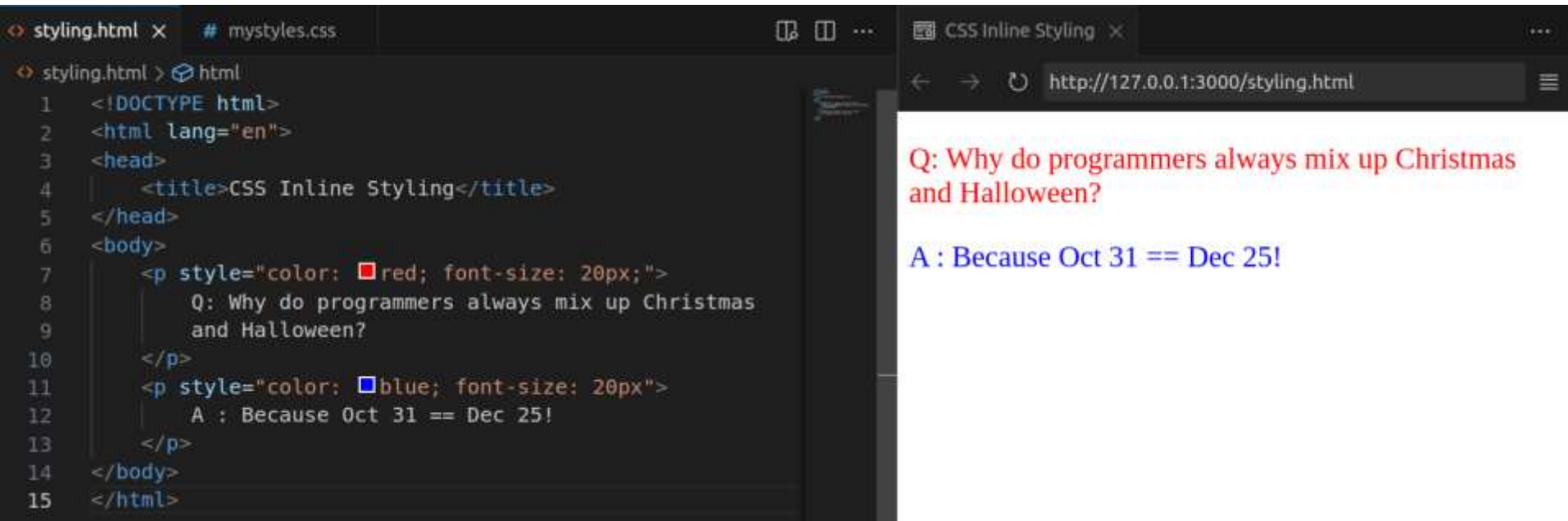
- Class and ID
- Inline Styling
- Internal Styling
- External Styling
- Div element

Class and ID



- ❖ Classes are used to group HTML elements together and apply styles or behavior to multiple elements simultaneously.
 - `<tag class="classname">Content</tag>`
- ❖ Elements can have multiple classes separated by spaces
 - `<tag class="class1 class2">Content</tag>`
- ❖ IDs are used to uniquely identify a single HTML element on a page. IDs must be unique within a document. No two elements should share the same ID.
 - `<tag id="elementID">Content</tag>`
- ❖ Classes and IDs are commonly used in CSS to apply styles to specific groups of elements:
 - `.classname { /* styles */ }`
 - `#elementID { /* styles */ }`
 - `tagname { /* styles */ }`
- ❖ Classes are often used in JavaScript for selecting and manipulating groups of elements, whereas IDs are to target and manipulate specific elements.

Inline Styling



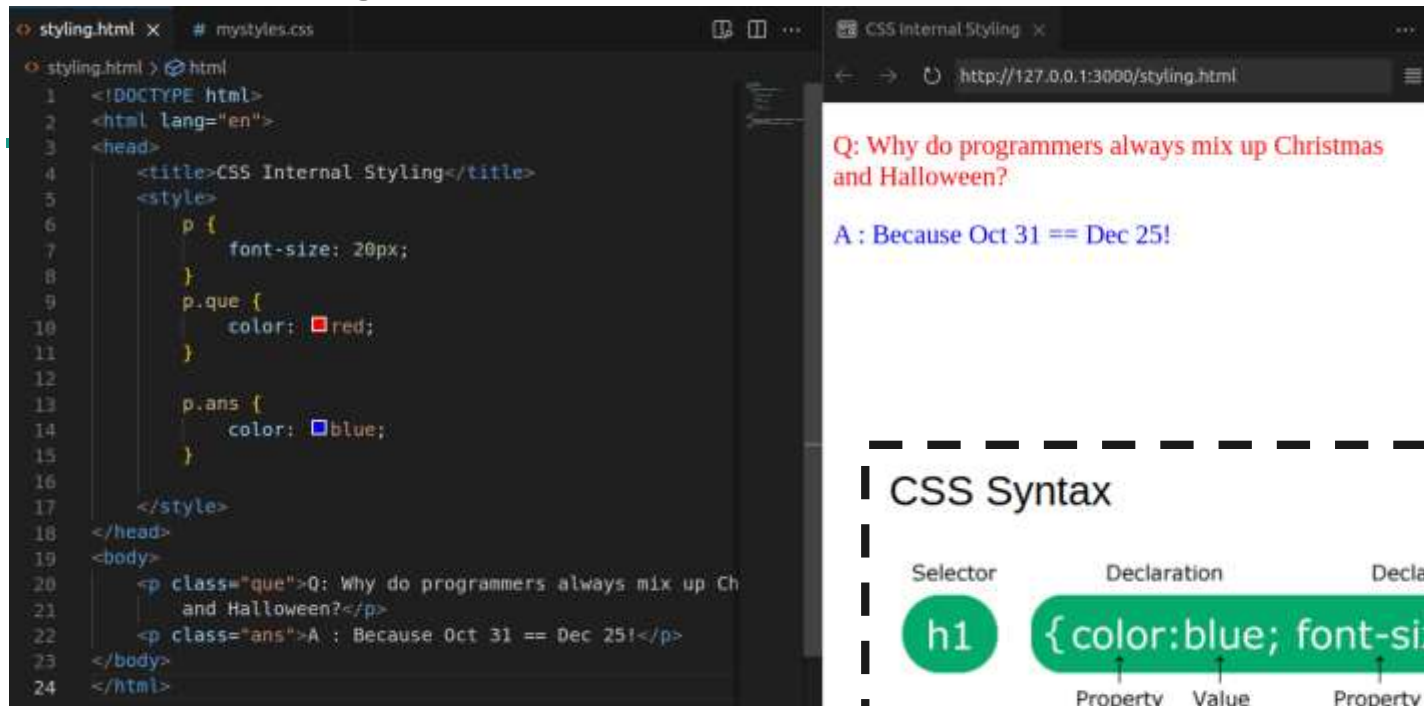
The image shows a web browser window on the right and a code editor on the left. The browser window displays the rendered HTML from the code editor. The code editor shows the following HTML code:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <title>CSS Inline Styling</title>
5 </head>
6 <body>
7   <p style="color: red; font-size: 20px;">
8     Q: Why do programmers always mix up Christmas
9     and Halloween?
10  </p>
11  <p style="color: blue; font-size: 20px">
12    A : Because Oct 31 == Dec 25!
13  </p>
14 </body>
15 </html>
```

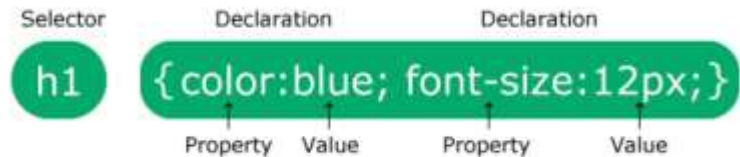
The browser window shows the rendered output of this code. The text "Q: Why do programmers always mix up Christmas and Halloween?" is displayed in red, and the text "A : Because Oct 31 == Dec 25!" is displayed in blue. Both lines of text are in a larger font size than the default.

- ❖ Inline styling involves applying styles directly to individual HTML elements using the `style` attribute.
- ❖ In this example, the `style` attribute is used to set the color to `red` for the Question and `blue` for the Answer and the font size is `20px` for both the `<p>` tags.

Internal Styling



CSS Syntax

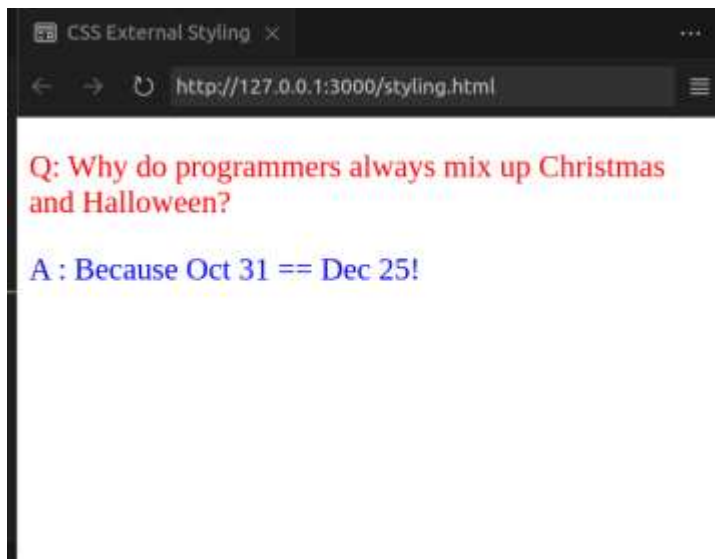


- ❖ Internal styling is achieved by placing CSS rules within a <style> tag in the HTML document, typically within the <head> section.
- ❖ Follow the CSS syntax to add styles to different elements, classes, ids. Selector `p` describes styling for all <p> tags, Selector `p.que` describes styling for all <p> tags with `class = "que"`

External Styling

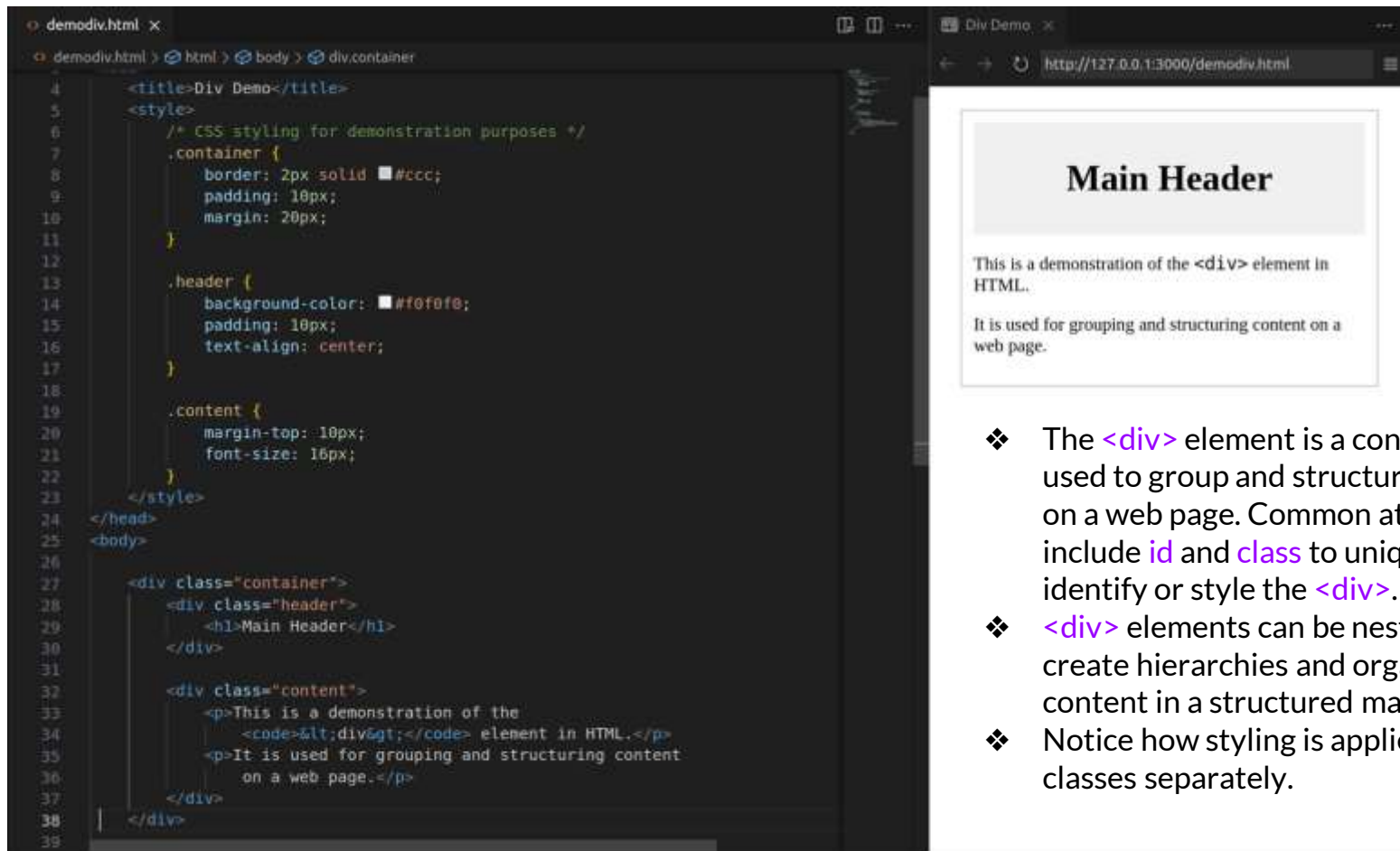
```
styling.html x
styling.html > html
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <title>CSS External Styling</title>
5   <link rel="stylesheet" href="mystyles.css">
6 </head>
7 <body>
8   <p class="que">Q: Why do programmers always mix up
9     Christmas
10    and Halloween?</p>
11   <p class="ans">A : Because Oct 31 == Dec 25!</p>
12 </body>
13 </html>

# mystyles.css x
# mystyles.css > p.ans
1 p {
2   font-size: 20px;
3 }
4 p.que {
5   color: red;
6 }
7
8 p.ans {
9   color: blue;
10 }
```



- ❖ External styling involves linking an external CSS file to the HTML document using the `<link>` element within the `<head>` section.
- ❖ To read more about CSS selectors: https://www.w3schools.com/css/css_selectors.asp

<div> element



The image shows a code editor on the left and a web browser on the right. The code editor displays the HTML and CSS for a demo page. The CSS defines three classes: .container (border, padding, margin), .header (background-color, padding, text-align), and .content (margin-top, font-size). The HTML structure uses these classes to create a container, a header, and a content area. The browser on the right shows the rendered page with a grey header and a white content area.

```
demodiv.html x
demodiv.html > html > body > div.container
4   <title>Div Demo</title>
5   <style>
6     /* CSS styling for demonstration purposes */
7     .container {
8       border: 2px solid #ccc;
9       padding: 10px;
10      margin: 20px;
11    }
12
13    .header {
14      background-color: #f0f0f0;
15      padding: 10px;
16      text-align: center;
17    }
18
19    .content {
20      margin-top: 10px;
21      font-size: 16px;
22    }
23  </style>
24 </head>
25 <body>
26
27   <div class="container">
28     <div class="header">
29       <h1>Main Header</h1>
30     </div>
31
32     <div class="content">
33       <p>This is a demonstration of the
34         <code><div></code> element in HTML.</p>
35       <p>It is used for grouping and structuring content
36         on a web page.</p>
37     </div>
38   </div>
39
```

Div Demo x
http://127.0.0.1:3000/demodiv.html

Main Header

This is a demonstration of the `<div>` element in HTML.

It is used for grouping and structuring content on a web page.

- ❖ The `<div>` element is a container used to group and structure content on a web page. Common attributes include `id` and `class` to uniquely identify or style the `<div>`.
- ❖ `<div>` elements can be nested to create hierarchies and organize content in a structured manner.
- ❖ Notice how styling is applied to div classes separately.

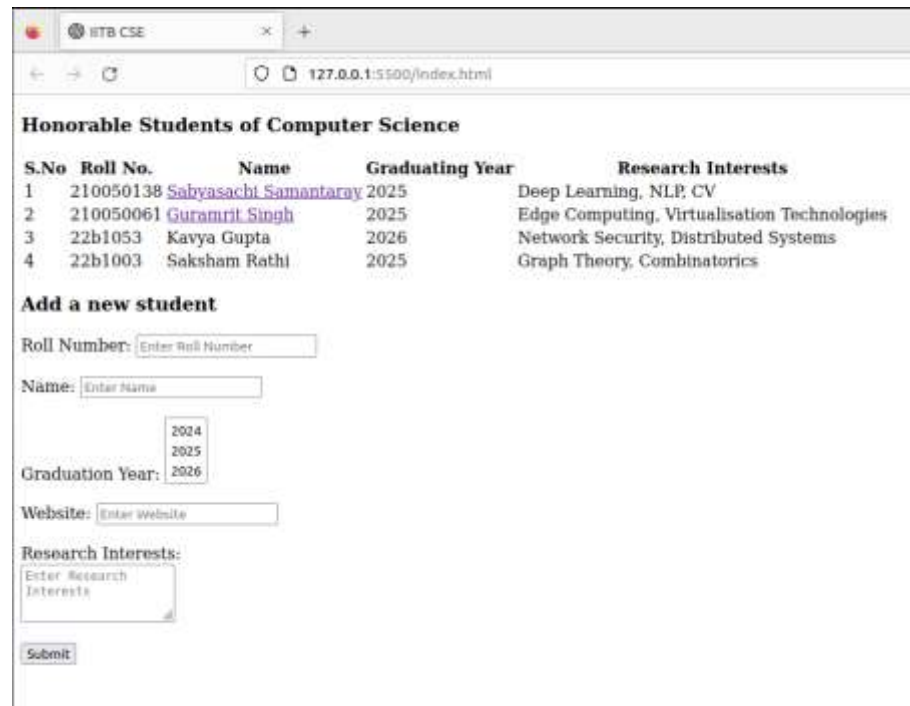


Exercises

- HTML
- CSS

Exercise 1 (Template with HTML)

- ❖ A simple template of a IITB CSE student information page.
- ❖ Deliverables:
 - Favicon and Title of the web page
 - Table of students with their information
 - Hyperlinked names with their webpages
 - A form to collect information of a new student to be added
 - All input sections except Website is mandatory
 - Dropdown for Graduation Year
 - Placeholder Text as shown in Figure
 - Submit Button(Non Functional as of Now)
- ❖ Solution: [index.html](#) (provided)



S.No	Roll No.	Name	Graduating Year	Research Interests
1	210050138	Sabyasachi Samantary	2025	Deep Learning, NLP, CV
2	210050061	Guramrit Singh	2025	Edge Computing, Virtualisation Technologies
3	22b1053	Kavya Gupta	2026	Network Security, Distributed Systems
4	22b1003	Saksham Rathi	2025	Graph Theory, Combinatorics

Add a new student

Roll Number:

Name:

Graduation Year:

Website:

Research Interests:

Exercise 2 (Styling with CSS)

- ❖ In extension to the previous exercise, now add styling to enhance the visual appeal of the HTML template.
- ❖ Additionally, remove the default features of the `<a>` tag, where in the links are blue and underlined. Instead now, it shows underline only when hovered.
- ❖ Do all the styling in file `styles.css` and link it with the html file.
- ❖ Solution: `styles.css` (provided)
- ❖ Coming Up (Next Week):
 - On submitting details of a new student, it updates the table above.

The screenshot shows a web browser displaying a web application. At the top, there's a title "Honorable Students of Computer Science" above a table. The table has five columns: S.No, Roll No., Name, Graduating Year, and Research Interests. It contains four rows of student data. Below the table is a form titled "Add a new student". The form includes input fields for Roll Number, Name, and Website, a dropdown menu for Graduation Year (with options 2024, 2025, 2026, 2027), and a text area for Research Interests. A green "Submit" button is at the bottom of the form.

S.No	Roll No.	Name	Graduating Year	Research Interests
1	210050138	Sabyasachi Samantaray	2025	Deep Learning, NLP, CV
2	210050061	Guramrit Singh	2025	Edge Computing, Virtualisation Technologies
3	22b1053	Kavya Gupta	2026	Network Security, Distributed Systems
4	22b1003	Saksham Rathi	2026	Graph Theory, Combinatorics

Add a new student

Roll Number:

Name:

Graduation Year:

Website:

Research Interests:



Thank You !!!