

PREFACE

ABOUT THE BOOK

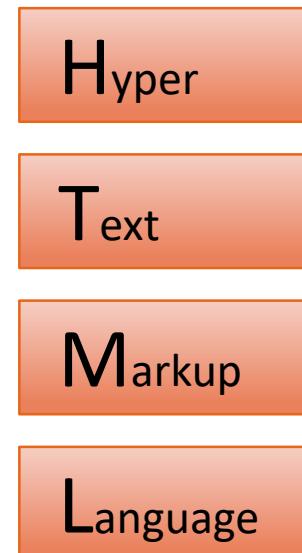
This book contains 100 very important HTML & CSS interview questions.

ABOUT THE AUTHOR

Jagmohan Rai has around 3.5 years of experience in software development.
He helps candidates in clearing technical interview in tech companies.

Q. What is **HTML**? Differences between **HTML & HTML5**? Advantages of **HTML5**? **V. IMP.**

- ❖ HTML(HyperText Markup Language) is the standard **markup language** used to create web pages.
- ❖ Markup language meaning a language which define the **structure of a document using elements** like headings, paragraphs, links, lists, and more.
- ❖ **HTML is not a programming language**, it's a markup language like XML.



```
<!DOCTYPE html>
<html>
|  <head>
|    <title>HTML</title>
|  </head>
|  <body>
|    <a href="https://abc.com">Link</a>
|    <h1>Heading Text</h1>
|    <p>Para Text</p>
|  </body>
|</html>
```

Q. What is **HTML**? Differences between **HTML & HTML5**? Advantages of **HTML5**? **V. IMP.**

- ❖ HTML5 is the fifth and **latest version** of HTML.

Old HTML/ HTML4

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"  
"http://www.w3.org/TR/html4/strict.dtd">  
<html>  
  <head>  
    <title>Old HTML Example</title>  
    <meta http-equiv="Content-Type"  
          content="text/html; charset=UTF-8" />  
  </head>  
  <body>  
    <h1>Tag</h1>  
    <p>Paragraph</p>  
    <a href="https://abc.com">Link</a>  
  </body>  
</html>
```

HTML/ HTML5

```
<!DOCTYPE html>  
<html>  
  <head>  
    <meta charset="UTF-8" />  
    <title>HTML5 Example</title>  
  </head>  
  <body>  
    <header>  
      <h1>Tag</h1>  
    </header>  
    <section>  
      <p>Paragraph</p>  
      <a href="https://abc.com">Link</a>  
    </section>  
  </body>  
</html>
```

5 Advantages of HTML5:

1. New Semantic Elements: `<header>`, `<nav>`, `<article>`, `<section>`, `<aside>`, `<footer>`

2. Form Input Types: `<input type="date">`, `<input type="email">`

3. Audio and Video Support: `<audio>`, `<video>`

4. Mobile Compatibility

5. Simpler Code

Q. What is the difference between **HTML** and **XHTML**?

- ❖ XHTML(eXtensible HyperText Markup Language) is a markup language that follows the **rules of XML** to define the structure of web pages.

XHTML

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html
PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>XHTML Example</title>
  </head>
  <body>
    <h1>XHTML</h1>
    
  </body>
</html>
```

HTML/ HTML5

```
<!DOCTYPE html>
<html>
  <head>
    <title>HTML</title>
  </head>
  <body>
    <h1>HTML</h1>
    
  </body>
</html>
```

Q. What is the difference between **HTML** and **XHTML**?

HTML/ HTML5	XHTML
<p>1. HTML has more lenient syntax rules. Closing tag is not mandatory for some elements. <code>.</code></p>	<p>XHTML follows stricter syntax rules. All tags must be properly nested and closed. <code></code></p>
<p>2. HTML is not case-sensitive. <code><DIV></code>, <code><P></code> will work.</p>	<p>XHTML is case-sensitive. <code><div></code>, <code><p></code> will work.</p>
<p>3. HTML is widely supported by all browsers and web platforms.</p>	<p>XHTML has limited support by browsers.</p>

Q. What is the role of DOCTYPE in HTML? **V. IMP.**

- ❖ DOCTYPE(Document Type) declaration specifies the **version** of HTML.
- ❖ DOCTYPE tells the browser which version of HTML it is and how to interpret the code.

```
<!DOCTYPE html>
<html>
  <head>
    <title>DocType</title>
  </head>
  <body>
    <h1>DocType</h1>
  </body>
</html>
```

```
<!-- Different kind of Doctypes -->
<!-- HTML 4.01 Strict -->
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">

<!-- XHTML 1.0 Strict: -->
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<!-- HTML5 (Preferred for modern web development): -->
<!DOCTYPE html>
```

Q. What if you remove <!DOCTYPE html> from your HTML?

- ❖ Then browsers can still render the page, but they will not be able to validate the version of HTML, therefore it may lead to some **compatibility issues** with SEO or debugging.

```
<!DOCTYPE html>
<html>
  <head>
    <title>DocType</title>
  </head>
  <body>
    <h1>DocType</h1>
  </body>
</html>
```

Q. Difference between Head & Body in HTML? Where to place JS link reference? **V. IMP.**

- ❖ The `<head>` element is where you place **meta-information** (information about the document). For example, `<title>`, `<meta>`, `<link>`, `<script>`, `<style>` are normally kept under head element.

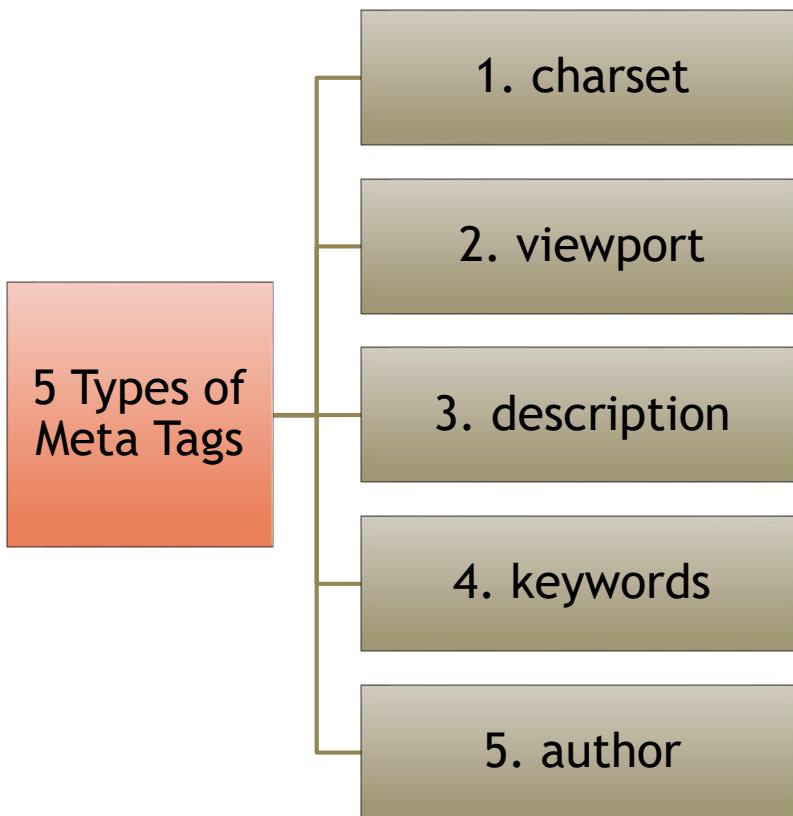
- ❖ The `<body>` element is where you place the **actual content** of your HTML web page. For example, `<div>`, `<h1>`, `<p>`, ``, `<a>` are normally kept under body element.

- ❖ Head will load before the body, therefore, if you are manipulating HTML elements in your JS functions, then place the JS link at the end of the body section because until then all the elements will be loaded. Else, place it inside the head tag.

```
<!DOCTYPE html>
<html>
<head>
  <title>Sample Page</title>
  <meta charset="UTF-8">
  <link rel="stylesheet" href="styles.css">
  <script src="script.js"></script>
</head>
<body>
  <h1>Hello, World!</h1>
  <p>This is a sample webpage.</p>
  
</body>
</html>
```

Q. What are Meta Tags? What are the 5 types of meta tags?

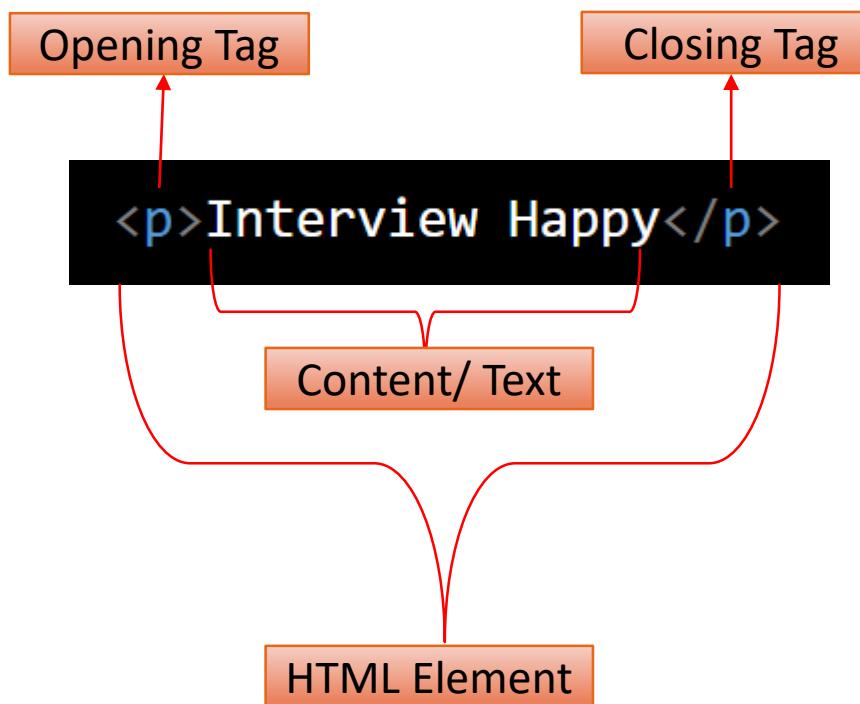
- ❖ Meta tags in HTML are elements used to provide metadata or additional information about a web page.



```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8" />
    <title>Example Page</title>
  </head>
  <body>
    <h1>Title Example</h1>
  </body>
</html>
```

Q. What are HTML elements? What is the difference between Element & Tag? **V. IMP.**

- ❖ HTML elements are the **building blocks** of web pages.
- ❖ Element consists of a start tag, content, and an end tag.
- ❖ Tag is a specific part of an element that denotes the beginning or end of that element.



```
<!DOCTYPE html>
<html>
  <head>
    <title>Title Element</title>
  </head>
  <body>
    <h1>Heading Element</h1>
    <ul><!-- List Element -->
      | <li>List Item 1</li>
      | <li>List Item 2</li>
    </ul>
    <p>Paragraph Element</p>
    <div>Division Element</div>
    
    <a href="~/AnchorElement.html"></a>
  </body>
</html>
```

Q. What are the roles and uses of the **<div>** element in HTML? **V. IMP.**

❖ The **<div>**(division) element in HTML is a **container** that is used to **group and structure** the content on a webpage.

❖ **Top 3 uses of <div> element:**

1.1. Grouping & structuring Content:

- It allows you to group together related elements.

1.2. Styling and Layout:

- **<div>** elements are used to apply common styles or css to grouped elements.

1.3. Scripting:

- JavaScript and other scripting languages can target **<div>** elements to manipulate their content or behavior.

```
<body>
  <div class="container">
    <h1>Heading 1</h1>
    <p>Paragraph 1</p>
  </div>
</body>
```

Heading 1

Paragraph 1

Q. What is the role of **header**, **main**, **section**, **footer** & **address** elements in HTML?

- ❖ **<header>**, **<main>**, **<section>** and **<footer>** are semantic elements, used to define the layout and **structure of a webpage** in a meaningful and organized way.

1.<header
 >

Contains the header content of website.

1.<main>

Contains the main content of the document.

1.<footer>

contains the group related content.

1.<address
 >

provides contact information.

Website Header

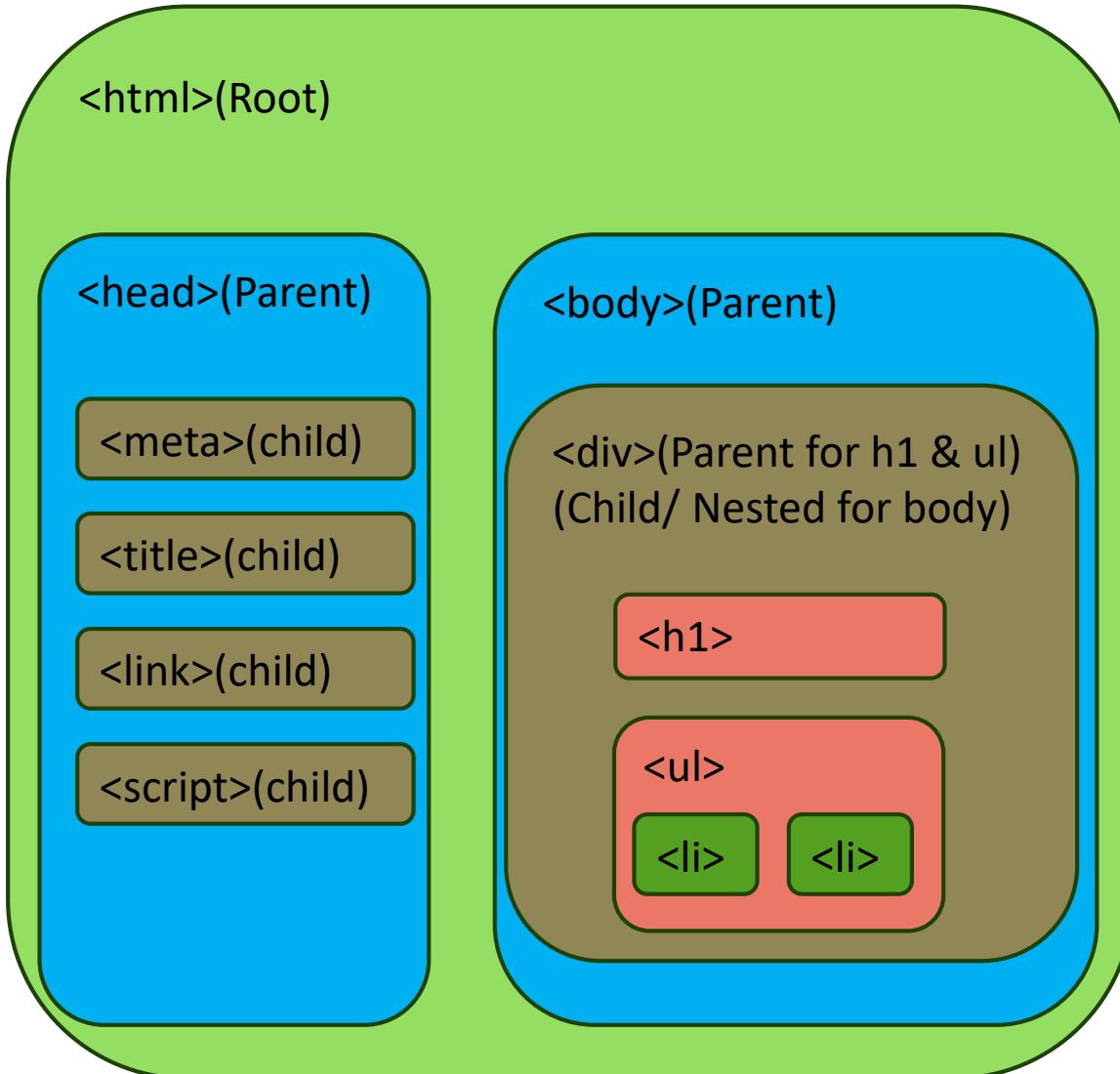
Section 1

Section 2

India

```
<body>
  <header>
    <h1>Website Header</h1>
  </header>
  <main>
    <section id="section1">
      <h2>Section 1</h2>
    </section>
    <section id="section2">
      <h2>Section 2</h2>
    </section>
  </main>
  <footer>
    <address>India</address>
  </footer>
</body>
```

Q. What are Root, Parent, Child & Nested elements? V. IMP.



```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <title>Nested Elements</title>
    <link rel="stylesheet" href="styles.css" />
    <script src="script.js"></script>
  </head>
  <body>
    <div>
      <h1>Inside container</h1>
      <ul>
        <li>Item 1</li>
        <li>Item 2</li>
      </ul>
    </div>
  </body>
</html>
```

Q. What are Root, Parent, Child & Nested elements? **V. IMP.**

1. The **root element** is the highest-level element in the hierarchy of an HTML document. In HTML5, the root element is `<html>`.
2. A **parent element** is an element that contains other child elements.
3. A **child or nested element** is an element that is contained within a parent element. Child and nested elements are same.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <title>Nested Elements</title>
    <link rel="stylesheet" href="styles.css" />
    <script src="script.js"></script>
  </head>
  <body>
    <div>
      <h1>Inside container</h1>
      <ul>
        <li>Item 1</li>
        <li>Item 2</li>
      </ul>
    </div>
  </body>
</html>
```

Q. What are Empty Elements?

- ❖ An empty element in HTML is an element that doesn't need content between opening and closing tags.
- ❖ Empty elements are also called a **self-closing** or void elements.
- ❖ Empty elements in HTML: , <input>,
, <hr>, <meta>, <link>, <area>, <base>, <col>, <embed>.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Empty Elements Example</title>
    <link rel="stylesheet" href="style.css" />
    <meta charset="UTF-8" />
  </head>
  <body>
    
    <input type="text" />
    <br />
  </body>
</html>
```

Q. What are **Semantic Elements** in HTML? Is div a semantic element? **V. IMP.**

- ❖ Semantic elements in HTML are elements that provide **meaning to the content** they contain.
- ❖ <div> is not a semantic element, because div is a general-purpose structural element. It doesn't give any meaning to the content.

Top 5 Semantic elements

1. <header>

2. <main>

3. <section>

4. <footer>

5. <address>

```
<body>
  <header>
    |   <h1>Website Header</h1>
  </header>
  <main>
    |   <section id="section1">
    |     <h2>Section 1</h2>
    |   </section>
  </main>
  <aside>
    |   <h2>Aside Content</h2>
  </aside>
  <footer>
    |   <address>India</address>
  </footer>
</body>
```

Q. What are the 3 differences between Block-Level and Inline Elements? **V. IMP.**

1. Block-level elements create "blocks" of content.
`<div>, <p>, <h1>, , , <table>, <form> etc.`
2. By default, Block-level elements start on a new line.
3. You can set both width and height for block-level elements.

```
<!DOCTYPE html>
<html>
  <head>
    <title>BlockLevel Elements</title>
  </head>
  <body>
    <div>block-level element</div>
    <p>block-level element</p>
  </body>
</html>
```

1. Inline elements length depends on their content length.
`, <a>, , , , <input>,
 etc.`
2. Inline elements do not start on a new line.
3. You can't set width and height for inline elements.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Inline Elements</title>
  </head>
  <body>
    <p>My name is<strong> Happy</strong></p>
    <p>Click <a href="https://abc.com">here</a></p>
  </body>
</html>
```

Q. What are **HTML Attributes**? What are the **Types** of HTML attributes? **V. IMP.**

❖ **Types of attributes:**

1.1. Common Attributes (Global Attributes)

- Example: class, id, style, data-*
- They are common and applicable for all the elements.

1.2. Specific Attributes

- Example: href, src, alt, width, height, target, rel, type, value, name, placeholder, disabled, readonly, checked, selected
- These are specific to the elements.

Q. What are the **Id**, **Style** & **Class** attributes of an element? When to use what? **V. IMP.**

- ❖ **id** attribute is used to **uniquely identify** an element on a page.
 - The primary purpose of the id attribute is to allow JavaScript and CSS to target and manipulate specific elements.
- ❖ **style** attribute allows you to apply **inline styles** directly to an individual element.
- ❖ **class** attribute is used to group together multiple elements that **share common styles**.
 - classes are recommended for large website not inline styles.

```
<html>
  <head>
    <title>Id, Style and Class</title>
    <style>
      .highlighted {
        color: red;
      }
    </style>
  </head>
  <body>
    <h1 id="uniqueId">Main heading</h1>

    <p style="color: blue">Para 1</p>

    <p class="highlighted">Para 2</p>
    <p class="highlighted">Para 3</p>
  </body>
</html>
```

Q. What will happen if two elements have **same ids**?



- ❖ May be in browser there is no direct impact, but it will be considered as **invalid HTML**.
- ❖ Invalid HTML means it can lead to **unexpected behavior** in your webpage. For example, problems can occur in css styling and JavaScript interactions.

```
<html lang="en">
  <head>
    <title>uplicate IDs</title>
  </head>
  <body>
    <div id="uniqueId1">First Element</div>
    <p id="uniqueId1">Second Element</p>
  </body>
</html>
```

Q. How to specify **Multiple Classes** for single element? What is the **style precedence**?



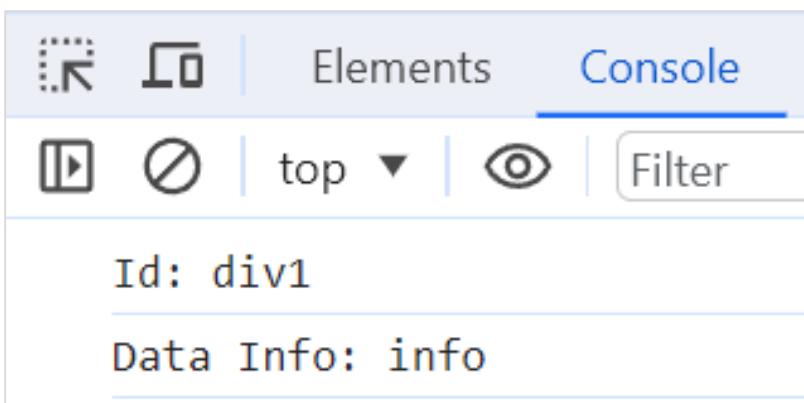
- ❖ To specify multiple classes for an element in HTML, you can simply add a **space-separated list of class names** within the class attribute.
- ❖ In case of multiple classes with same style attribute, style defined in the **last class** will take precedence.

single class
multiple classes

```
<html>
  <head>
    <title>Multiple Classes Example</title>
    <style>
      .class1 {
        color: red;
      }
      .class2 {
        font-size: 20px;
        color: green;
      }
    </style>
  </head>
  <body>
    <div class="class1">single class</div>
    <div class="class1 class2">multiple classes</div>
  </body>
</html>
```

Q. What are Data Attributes in HTML? **V. IMP.**

- ❖ Data attributes in HTML provide a way to add **custom data attributes** to add additional informational in elements.
- ❖ Data attributes are specified using the **data-** prefix.
- ❖ Data attributes can be accessed by dataset property in JS.



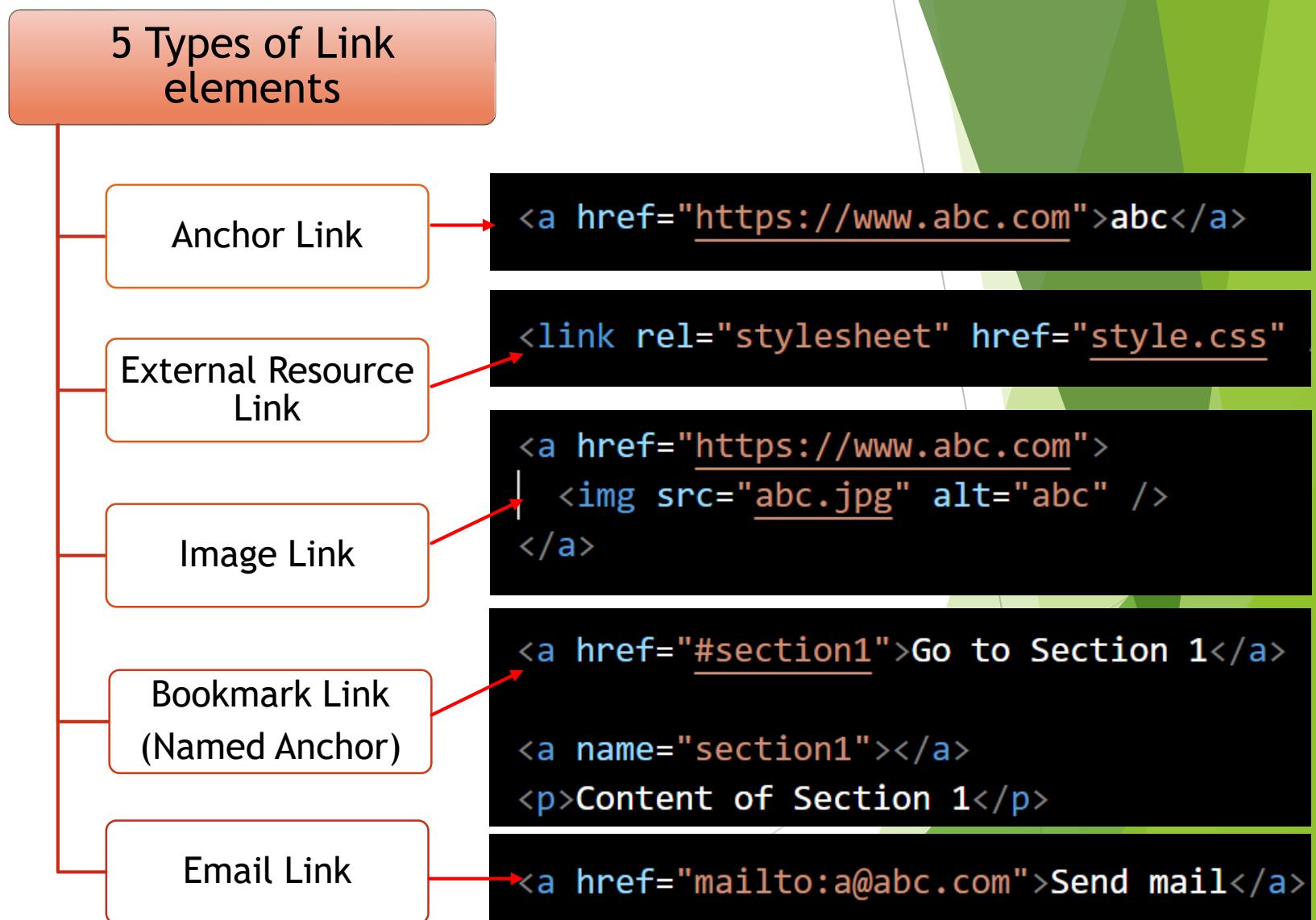
```
<body>
  <div id="div1" data-info="info">Data Attributes</div>

  <script>
    const element = document.querySelector("div");
    const id = element.id;
    const info = element.dataset.info;
    const dataid = element.dataset.id;

    console.log(`Id: ${id}`);
    console.log(`Data Info: ${info}`);
  </script>
</body>
```

Q. What are the 5 Types of Links in HTML? **V. IMP.**

1. Anchor Link (`<a>`) used for navigating from one webpage to another.
2. External Resource Link (`<link>`) used in the `<head>` section to connect to an external resource like a stylesheet or an icon.
3. Image Link (``) used to create a clickable image that leads to another webpage.
4. Bookmark Link (Named Anchor) points to a specific location within a webpage using a named anchor.
5. Email Link creates a clickable link that opens the user's default email client with a pre-filled email address.



Q. What is the difference between **Absolute** and **Relative URLs?** **V. IMP.**

❖ Absolute URLs:

1. Absolute URLs provide the **complete web address** of a resource.
2. Absolute URLs are typically used to link to resources on **different websites**.

```
<h2>Absolute URLs</h2>
<ul>
    <li><a href="http://www.abc.com">HTTP URL</a></li>
    <li><a href="ftp://ftp.abc.com/doc.pdf">FTP URL</a></li>
    <li><a href="mailto:info@abc.com">Mailto URL</a></li>
</ul>
```

❖ Relative URLs:

1. Relative URLs specify the location of a resource in **relation** to the current document. Full url is not required.
2. They are used when linking to resources within the **same website**.

```
<!-- example/index.html -->
<h2>Relative URLs</h2>
<ul>
    <li><a href="page.html">Same Directory</a></li>
    <li><a href="sub/page.html">Subdirectory</a></li>
    <li><a href="../page.html">Parent Directory</a></li>
    <li></li>
    <li><link rel="stylesheet" href="styles/styles.css"/></li>
    <li><script src="scripts/script.js"></script></li>
</ul>
```

Q. What is a **Fragment Identifier** in a URL? **V. IMP.**

- ❖ A fragment identifier is used to navigate to a **specific section** of the same webpage.
- ❖ Fragment identifier is preceded by a **# (hash)** symbol.

[Go to Section 1](#)

[Go to Section 2](#)

Section 1

Content for Section 1.

Section 2

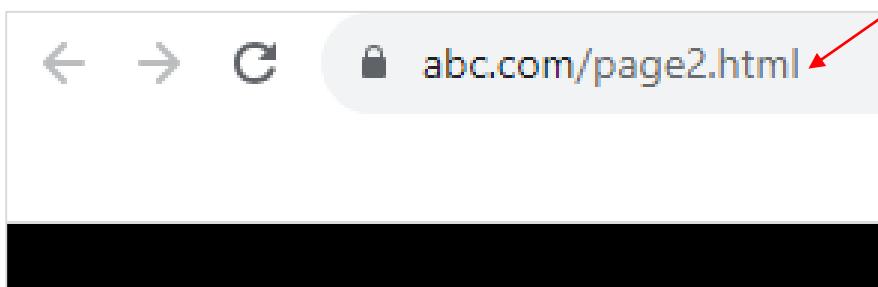
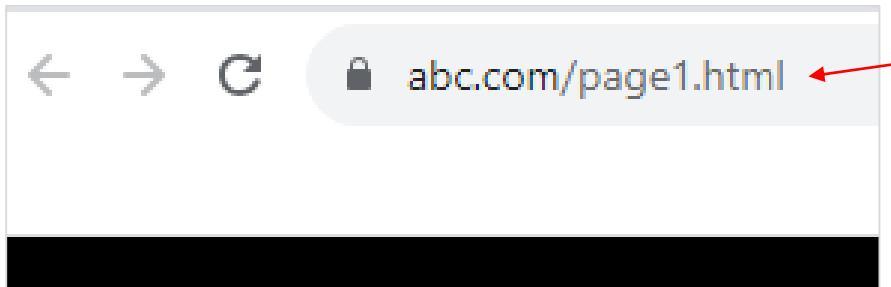
```
<body>
  <a href="#section1">Go to Section 1</a>
  <a href="#section2">Go to Section 2</a>

  <section id="section1">
    <h2>Section 1</h2>
    <p>Content for Section 1.</p>
    <div style="height: 500px">Test</div>
  </section>

  <section id="section2">
    <h2>Section 2</h2>
    <p>Content for Section 2.</p>
  </section>
</body>
```

Q. What is the purpose of the <base> element in HTML.?

- ❖ The <base> element in HTML is used to specify a **base URL** for relative URLs within a document.
- ❖ The <base> element is typically placed within the <head> section.



```
<html lang="en">
  <head>
    <title>Base Element Example</title>
    <base href="https://www.abc.com/" />
  </head>
  <body>
    <a href="page1.html">Link to Page 1</a>
    <a href="page2.html">Link to Page 2</a>
  </body>
</html>
```

Q. How do you add an **external stylesheet** in your HTML?



- ❖ **<link> element** is an empty element used to link **external resources** like stylesheets to the HTML document.
- ❖ **rel** attribute specifies the **relationship** between the current document and the linked resource.
- ❖ **href** attribute specifies the **location (URL)** of the external stylesheet.

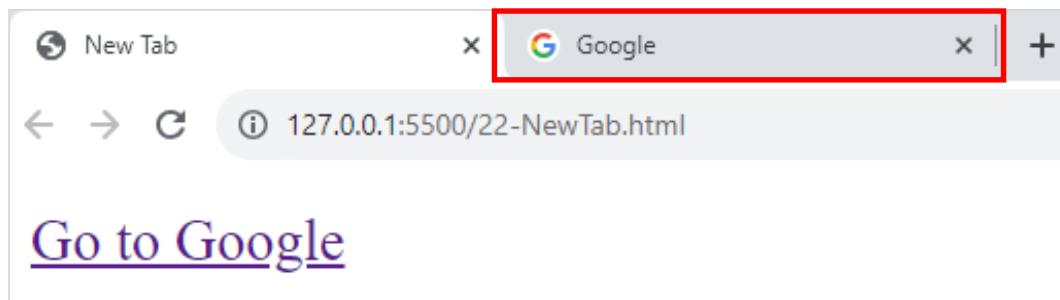
```
<html>
|   <head>
|   |   <link rel="stylesheet" href="styles.css" />
|   </head>
|   <body>
|   |   <!-- Body content -->
|   </body>
</html>
```

Q. How do you open a link in a new tab?



- ❖ Target attribute specifies how the linked content should be displayed.
- ❖ Target values: _blank(new tab), _self(default), _parent, _top

```
<html>
  <head>
    <title>New Tab</title>
  </head>
  <body>
    <a href="https://www.google.com" target="_blank">Go to Google</a>
  </body>
</html>
```



Q. What is a **Nested List** in HTML?

- ❖ A nested list in HTML is a list that is placed within another list item.

Technologies

- Backend
 - Java
 - .NET
 - Node
- Frontend
 - HTML
 - JS
 - React

```
<h1>Technologies</h1>
<ul>
  <li>
    Backend
    <ul>
      <li>Java</li>
      <li>.NET</li>
      <li>Node</li>
    </ul>
  </li>
  <li>
    Frontend
    <ul>
      <li>HTML</li>
      <li>JS</li>
      <li>React</li>
    </ul>
  </li>
</ul>
```

Q. What are table, tr, th, td elements? What are table advantages & disadvantages? **V. IMP.**

❖ Table elements:

1. <table> is the container for the entire table.
2. <tr>(table row) is used to define a row in the table.
3. <th>(table header) is used to represent the column headers.
4. <td>(table data) is used to represent the regular cells in a table.

Header 1	Header 2
Row 1, Cell 1	Row 1, Cell 2
Row 2, Cell 1	Row 2, Cell 2

Tables are a powerful tool for styling and displaying structured data.

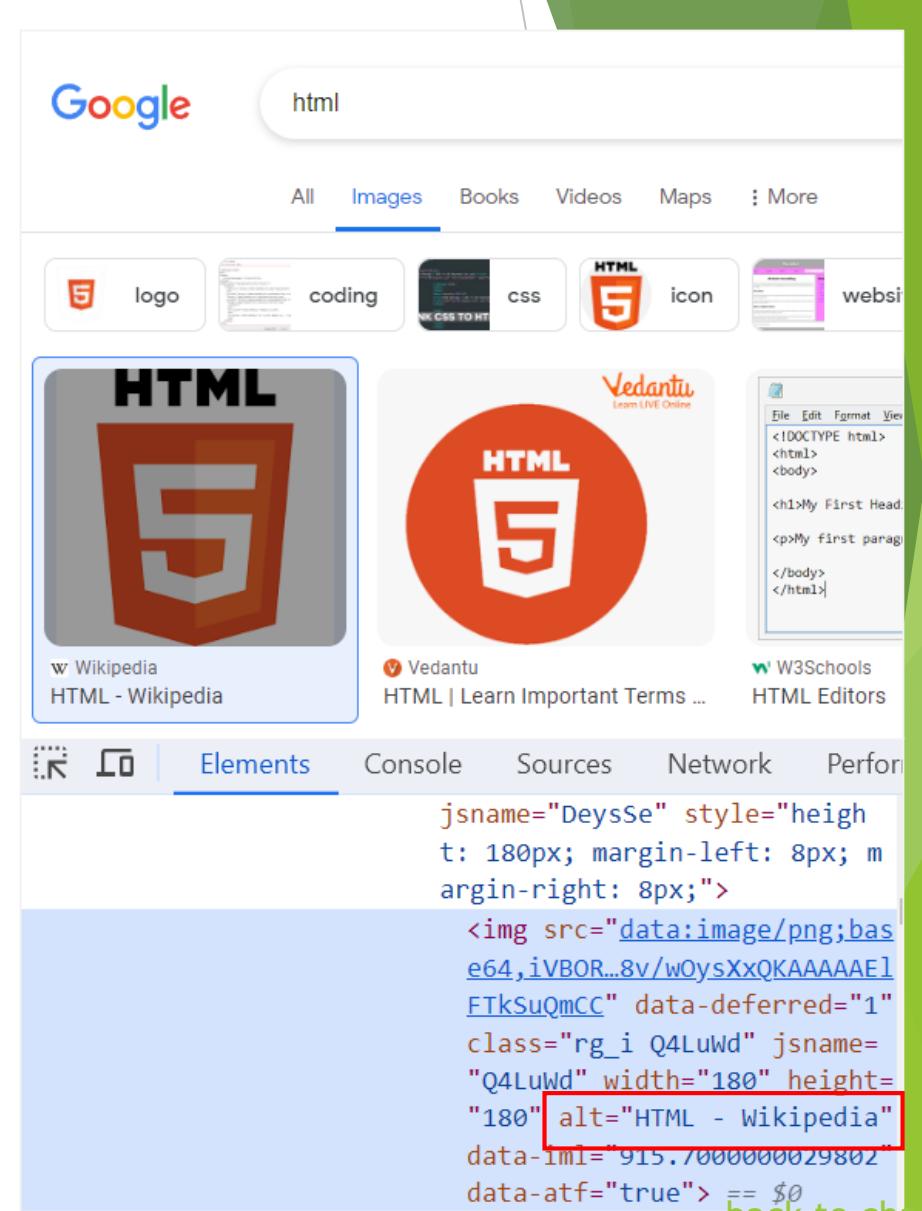
Tables multiple column's structure is not good for mobile devices(not responsive).

```
<body>
  <table border="1">
    <tr>
      <th>Header 1</th>
      <th>Header 2</th>
    </tr>
    <tr>
      <td>Row 1, Cell 1</td>
      <td>Row 1, Cell 2</td>
    </tr>
    <tr>
      <td>Row 2, Cell 1</td>
      <td>Row 2, Cell 2</td>
    </tr>
  </table>
</body>
```

Q. How to embed an **Image** in HTML? Why **alt** attribute is so important? **V. IMP.**

- ❖ element is used to embed an image in HTML.
- ❖ The src(soruce) attribute specifies the source file of the image.
- ❖ The alt(alternative text) attribute provides alternative text that describes the content of the image which is important for SEO(for searching).

```
<html lang="en">
  <head>
    <title>Image Example</title>
  </head>
  <body>
    
  </body>
</html>
```



Q. How to embed Video in HTML? Why we need multiple source elements?

1

- <video> is the element used to embed videos.

2

- **controls** attribute adds video controls like play, pause & volume.

3

- <source> element specifies multiple sources for the video. If one video format is not compatible with browser, then other will execute.

4

- **src** attribute specifies the path to the video file.

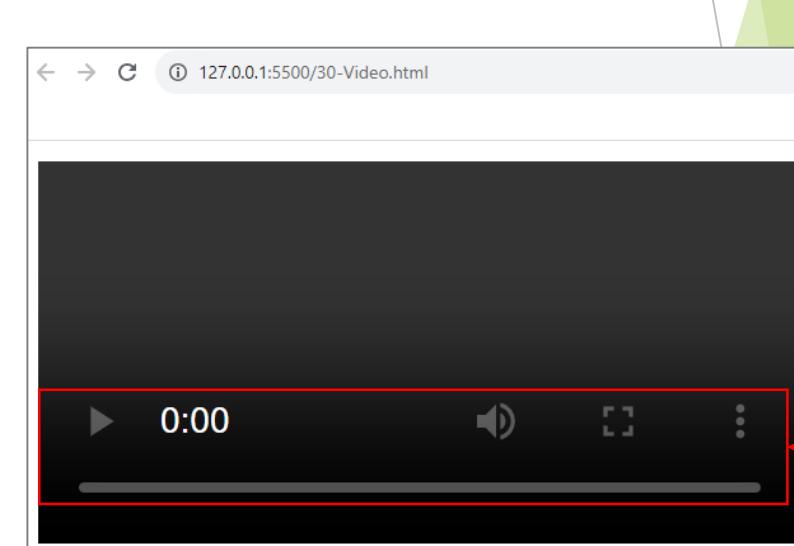
5

- **type** attribute indicates the MIME type.

6

- The **text "Your browser..."** is a fallback message for browsers that do not support the <video> element.

```
<body>
  <video controls>
    <source src="filepath.mp4" type="video/mp4" />
    <source src="filepath.webm" type="video/webm" />
    Your browser does not support the video tag.
  </video>
</body>
```



Q. What are **HTML Forms** and what are its **advantages?** **V. IMP.**

- ❖ Forms provide a way to collect and **submit** user data on a webpage.

Advantages of Form element

1. Data collection & organization is simple

2. Data validation is simple

3. Data submission on server is easy

4. Styling and Layout

5. Compatible with all browsers

Login Form

Username:

Password:

Q. What are HTML Form Elements and their Main Attributes? **V. IMP.**

Form elements & Attributes

1. form element

2. action attribute

3. method attribute

4. label element

5. for attribute

6. input element

7. type attribute

8. name attribute

```
<form action="/submit" method="post">
  <label for="username">Username:</label>
  <input type="text" id="username" name="username" />
  <br /><br />
  <label for="password">Password:</label>
  <input type="password" id="pwd" name="password" />
  <br /><br />
  <input type="submit" value="Submit" />
</form>
```

Login Form

Username: happy

Password:

Submit

Q. How to Validate for empty input field in form?



- ❖ **required** attribute is used to make the input filed mandatory.

```
<body>
  <h3>Login Form</h3>
  <form action="/submit" method="post">
    <label for="username">Username:</label>
    <input type="text" id="username" name="username" required />
    <br /><br />
    <label for="password">Password:</label>
    <input type="password" id="password" name="password" required />
    <br /><br />
    <input type="submit" value="Submit" />
  </form>
</body>
```

Login Form

Username: Happy

Password:

! Please fill out this field.

A screenshot of a web browser showing a login form. The form has two fields: 'Username' and 'Password'. The 'Username' field contains the text 'Happy'. The 'Password' field is empty. Below the 'Password' field is a red rectangular box highlighting the 'required' attribute in the HTML code. A tooltip-like message 'Please fill out this field.' with an exclamation mark icon appears over the empty 'Password' field.

Q. What are the **Types of Input** elements in a form?

Types of Input elements

Text Input

Password Input

File Input

Submit Button

Reset Button

Checkbox

Radio Button

Select

```
<input type="text" id="uid" name="uid" />  
<br /><br /><br />
```

```
<input type="password" id="pwd" name="pwd" />  
<br /><br /><br />
```

```
<input type="file" id="file" name="file" />  
<br /><br /><br />
```

```
<input type="submit" value="Submit" />  
<br /><br /><br />
```

```
<input type="reset" value="Reset" />  
<br /><br /><br />
```

Happy

.....

Choose File No file chosen

Submit

Reset

Q. How to create **radio buttons** and **checkboxes**? How to **group** them?

- ❖ Radio buttons are used when you want the user to **select exactly one option** from a group of options.

```
<label> <input type="radio" name="gender" value="male" /> Male </label><br />
<label> <input type="radio" name="gender" value="female" /> Female </label><br />
<label> <input type="radio" name="gender" value="other" /> Other </label><br />
```

Male
 Female
 Other

- ❖ Checkboxes are used when you want the user to be able to **select multiple options** from a list.

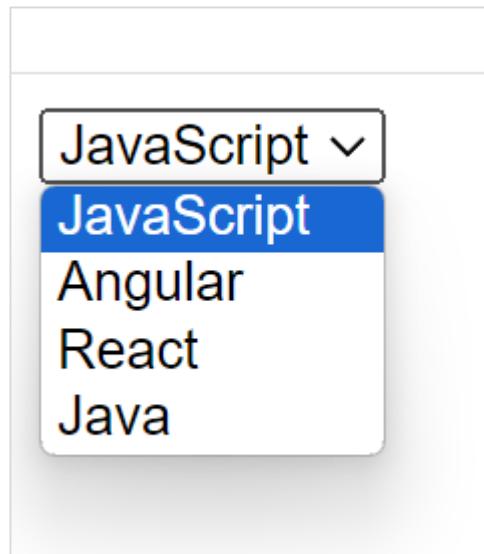
```
<label> <input type="checkbox" name="hobby" value="music" /> Music </label><br />
<label> <input type="checkbox" name="hobby" value="sports" /> Sports </label><br />
<label> <input type="checkbox" name="hobby" value="travel" /> Travel </label>
```

Music
 Sports
 Travel

- ❖ All radio buttons or checkboxes within a group should share the **same name** attribute.

Q. What is the purpose of the <select> element in a form?

- ❖ The <select> element in HTML is used to create a **dropdown menu** within a form.
- ❖ Dropdown menus are space-efficient compared to radio-buttons and checkboxes for displaying a **long list of options**.



```
<body>
  <form>
    <select name="Technology">
      <option value="1">JavaScript</option>
      <option value="2">Angular</option>
      <option value="3">React</option>
      <option value="4">Java</option>
    </select>
  </form>
</body>
```

Q. What are top 10 Best Practices for writing HTML code? **V. IMP.**

❖ HTML Bad Practices

```
<!-- 1. Missing DOCTYPE declaration -->
<html>
  <head>
    <!-- 2. Missing charset meta tag -->
    <title>Interview Happy</title>
  </head>
  <body>
    <!-- 3. Missing semantic header tag -->
    <div id="Header">
      <h1>Interview Happy</h1>
    </div>
    <!-- 4. Meaningless ID's and classes -->
    <div id="xyz" class="abc">
      <!-- 5. Incorrect capitalization of tags -->
      <P>Help in cracking interviews</P>
      <!-- 6. Missing Alt Text for Images -->
      
    </div>
  </body>
</html>
```

❖ HTML Best Practices

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport"
    content="width=device-width, initial-scale=1.0" />
    <title>Interview Happy</title>
    <link rel="stylesheet" href="styles.css" />
  </head>
  <body>
    <header>
      <h1>Interview Happy</h1>
    </header>
    <main>
      <section id="content">
        <p>Help in cracking interviews</p>
        
      </section>
    </main>
  </body>
</html>
```

Q. What is an HTML5 Boilerplate?

- ❖ An HTML5 boilerplate is a **basic template** for building modern websites using HTML5.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>HTML 5 Boilerplate</title>
    <link rel="stylesheet" href="style.css" />
  </head>
  <body>
    <!-- Website Content -->
    <script src="index.js"></script>
  </body>
</html>
```

Q. How do you optimize an HTML website for Performance?

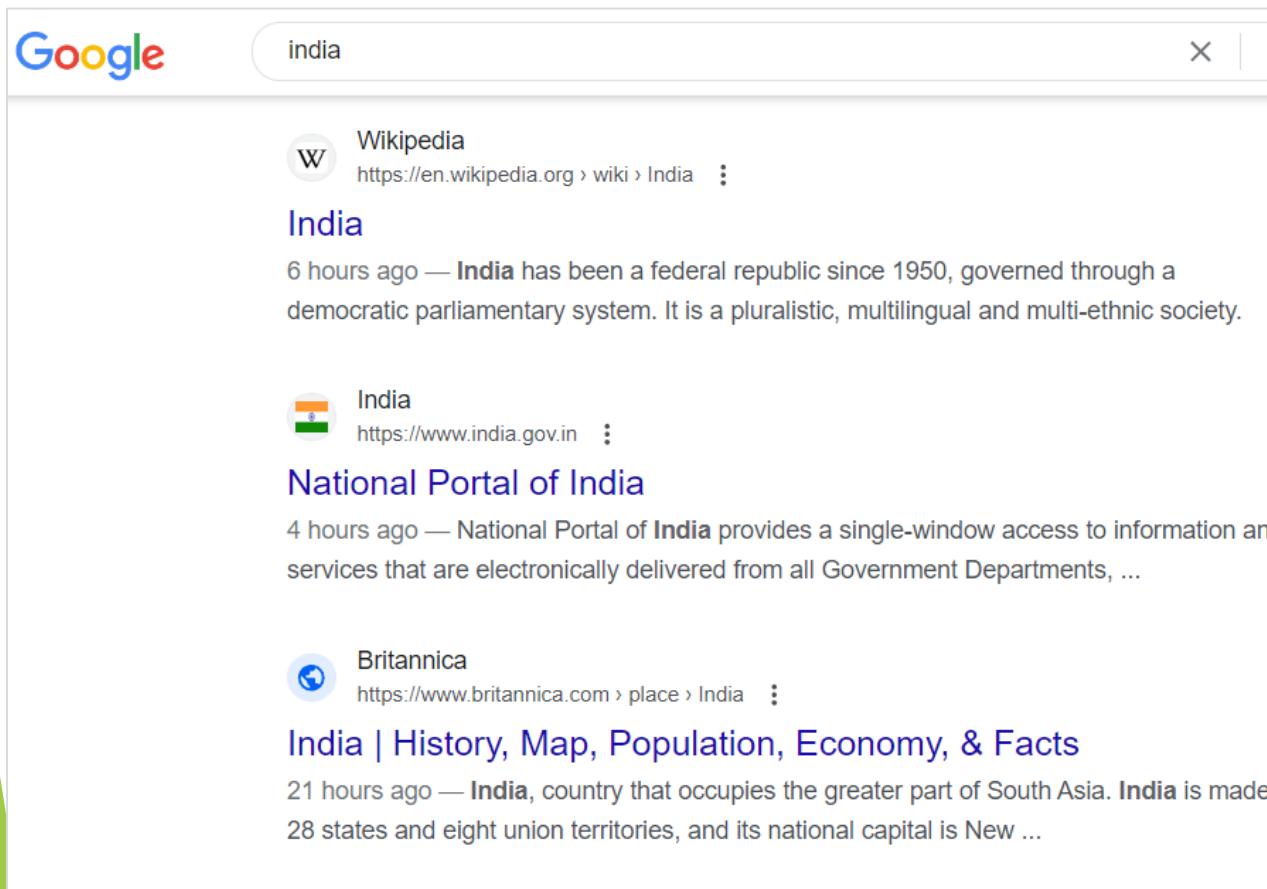
❖ 8 Best Practices for performance improvement in HTML:

1. Minimize HTTP Requests: Reduce the number of HTTP requests by combining CSS and JavaScript files.
2. Use External CSS and JavaScript Files
3. Minify CSS, JavaScript, and HTML: Remove unnecessary whitespace, comments & formatting to reduce file sizes.
4. Compress Images: Use tools to compress images and choose appropriate formats.
5. Leverage Browser Caching: Set cache headers to allow resources to be stored locally, reducing the need for repeated requests.
6. Load Critical Resources First.
7. Use Async and Await in JS.
8. Implement Lazy Loading to improve initial load times.



Q. What is SEO? What are 5 HTML Best Practices for SEO?

- ❖ SEO(Search Engine Optimization) is the practice of optimizing a website to improve its **visibility** and **ranking** in search engine results.



A screenshot of a Google search results page for the query "india". The results are as follows:

- Wikipedia**
https://en.wikipedia.org › wiki › India
India
6 hours ago — India has been a federal republic since 1950, governed through a democratic parliamentary system. It is a pluralistic, multilingual and multi-ethnic society.
- India**
https://www.india.gov.in
National Portal of India
4 hours ago — National Portal of India provides a single-window access to information and services that are electronically delivered from all Government Departments, ...
- Britannica**
https://www.britannica.com › place › India
India | History, Map, Population, Economy, & Facts
21 hours ago — India, country that occupies the greater part of South Asia. India is made up of 28 states and eight union territories, and its national capital is New ...

5 HTML Best Practices for SEO

1. Use Semantic HTML Elements

2. Optimize Page Titles

3. Use Meta Description Tags

4. Use Proper Heading Tags

5. Optimize Image Alt Attributes

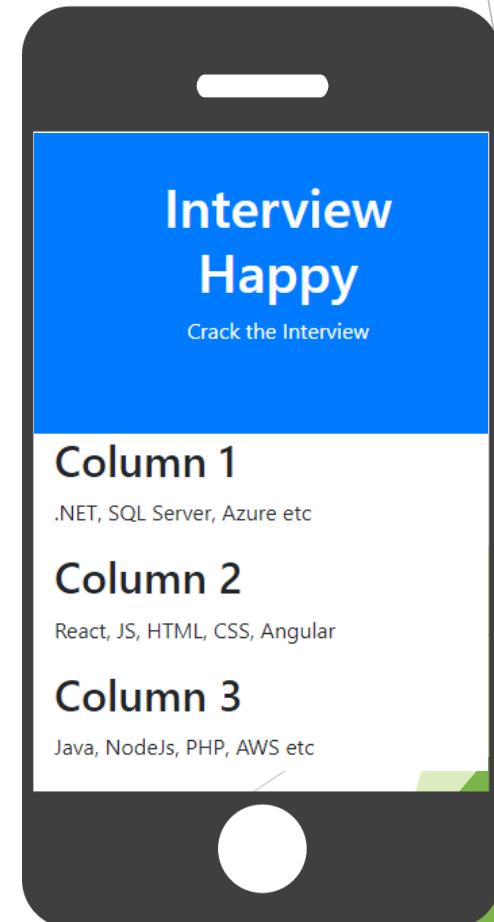
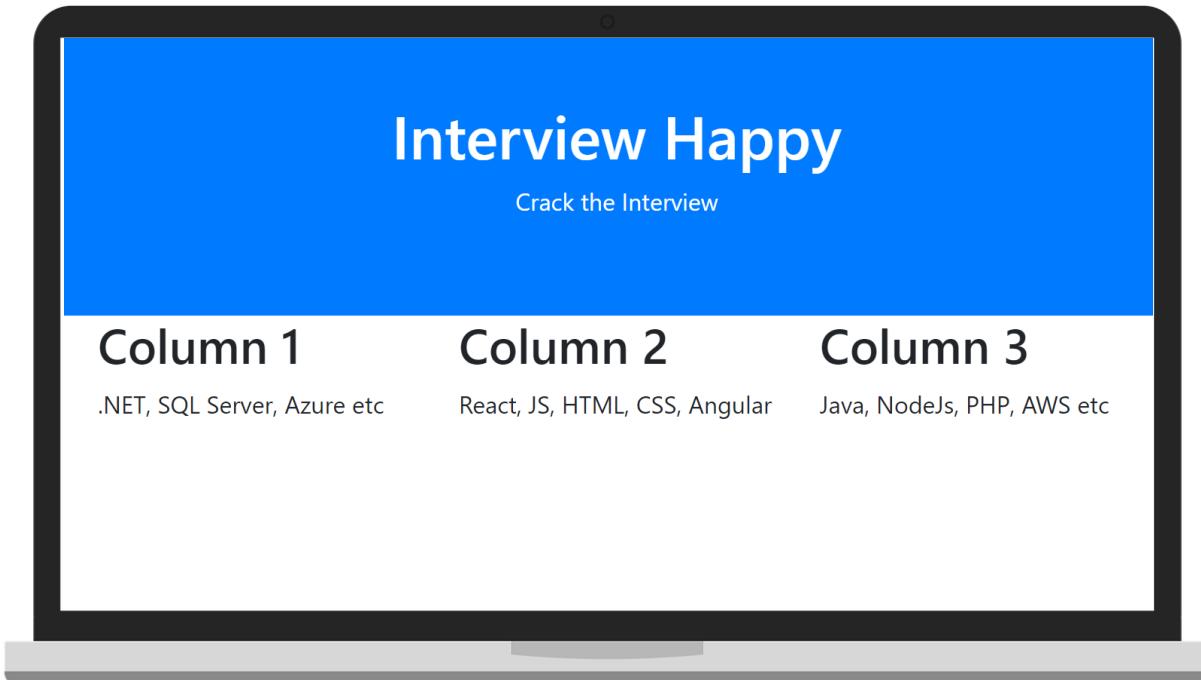
Q. What are the **5 steps** required for HTML website deployment?

❖ 5 Basic Steps for HTML website deployment:

- 1 • Choose a Hosting Provider (ex: aws/ azure)
- 2 • Purchase a Domain (ex: www.interviewhappy.com)
- 3 • Prepare Your Website Files(ex: HTML/ CSS/ JS/ Image)
- 4 • Upload Files to Hosting Server(using FTP, hosting panel, GIT)
- 5 • Configure SSL (Optional - Enable SSL for using HTTPS and enhance security)

Q. What is Responsive Design? **V. IMP.**

- ❖ Responsive design is a practice of creating web pages that **adapt** and display well on various devices and **screen sizes**.



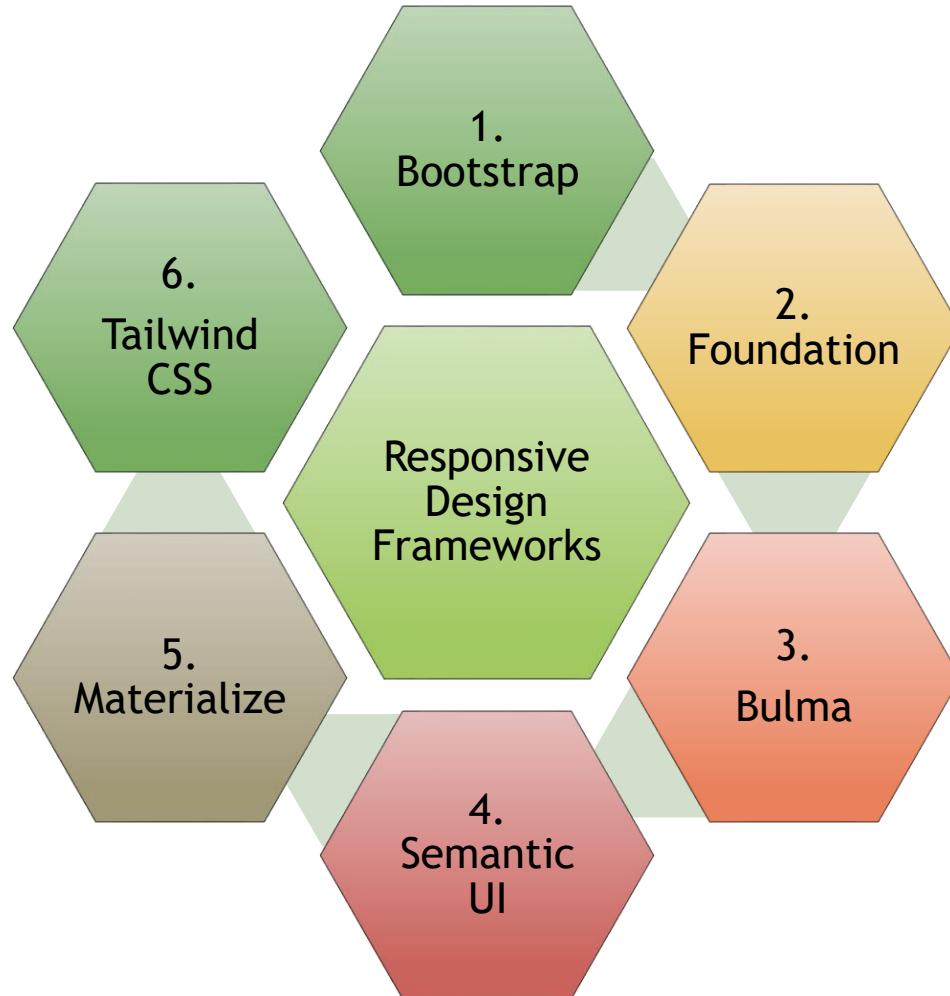
Q. What are the top 3 things required to make a website **Responsive** in HTML?

- 1. Set the Viewport Meta Tag :** This tag tells the browser to adjust the width of the page to match the screen's width.
- 2. Use Relative Units:** Instead of using fixed units like pixels, use relative units like percentages
- 3. Media Queries:** They set different sizes for different screen types.

```
.box {  
    width: 30%;  
}  
  
@media (max-width: 768px) {  
    .box {  
        width: 45%;  
    }  
}
```

```
<!DOCTYPE html>  
<html lang="en">  
  <head>  
    <meta charset="UTF-8" />  
    <meta name="viewport"  
          content="width=device-width, initial-scale=1.0" />  
    <title>Responsive HTML Example</title>  
    <link rel="stylesheet" href="39-MakeResponsive.css" />  
  </head>  
  <body>  
    <div class="container">  
      <div class="box" width="33%">Box 1</div>  
      <div class="box" width="33%">Box 2</div>  
      <div class="box" width="33%">Box 3</div>  
    </div>  
  </body>  
</html>
```

Q. What are some Responsive Design Frameworks for HTML & CSS?



Q. What are HTML Entities? How do you display special characters in HTML?

- ❖ HTML entities are special codes or sequences of characters used to represent reserved or special characters in HTML.

Top 10 Special Characters in HTML

Ampersand: &

Lesser than: <

Greater than: >

Euro symbol: €

Copyright symbol: ©

Trademark symbol: ™

Non-breaking space:

Double quotation mark: "

Single quotation mark: '

Registered trademark symbol: ®

```
<body>
  <h5>Top 10 Special Characters in HTML</h5>
  <p>Ampersand: &&lt;>&lt;&gt;</p>
  <p>Lesser than: &lt;</p>
  <p>Greater than: &gt;</p>
  <p>Euro symbol: &euro;</p>
  <p>Copyright symbol: &copy;</p>
  <p>Trademark symbol: &trade;</p>
  <p>Non-breaking space: &nbsp;</p>
  <p>Double quotation mark: &quot;</p>
  <p>Single quotation mark: &apos;</p>
  <p>Registered trademark symbol: &reg;</p>
</body>
```

Q. What are the **Top 5 Emerging Trends** in HTML development?

1. Web Components:

- Web Components are a set of web platform APIs that allow you to create custom, reusable HTML elements.

2. Progressive Web Apps (PWAs)

- PWAs are web applications that provide a native app-like experience to users.

3. Single Page Applications (SPAs):

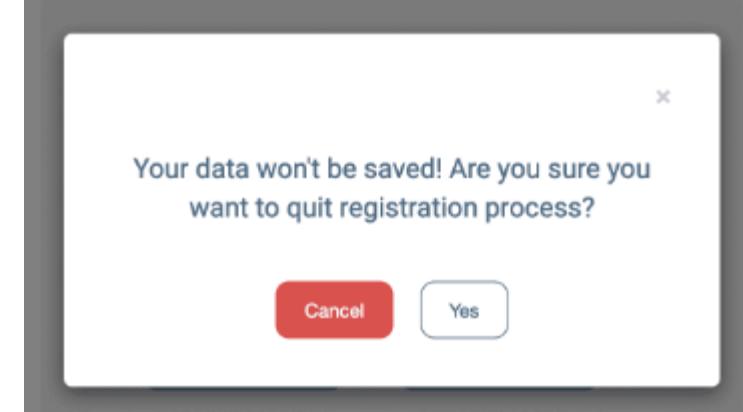
- SPAs load a single HTML page and dynamically update content as the user interacts with the app.

4. 3D Graphics and Virtual Reality (VR)

- VR refers to the integration of virtual reality experiences within a web browser using HTML, CSS, and JS. Example: WebGL & WebVR.

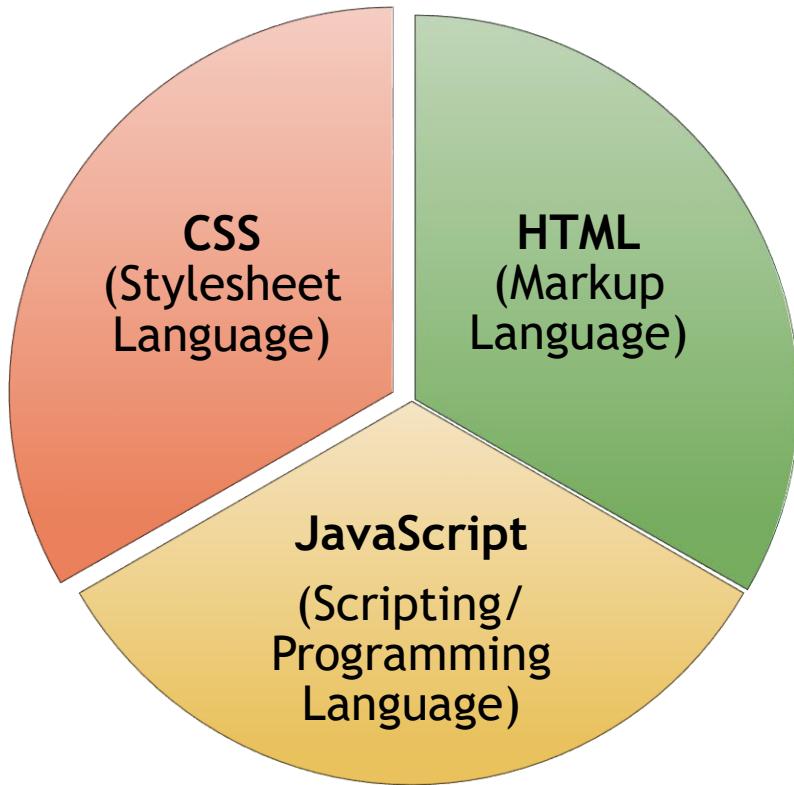
5. Augmented Reality (AR):

- AR combines the physical world with digital information or content. Example: ARCore and ARKit for web development.



Q. What is CSS? What are the 5 ways to Implement CSS in HTML? **V. IMP.**

- ❖ CSS(Cascading Style Sheets) is a **stylesheet language** used to control the **presentation** of web pages.

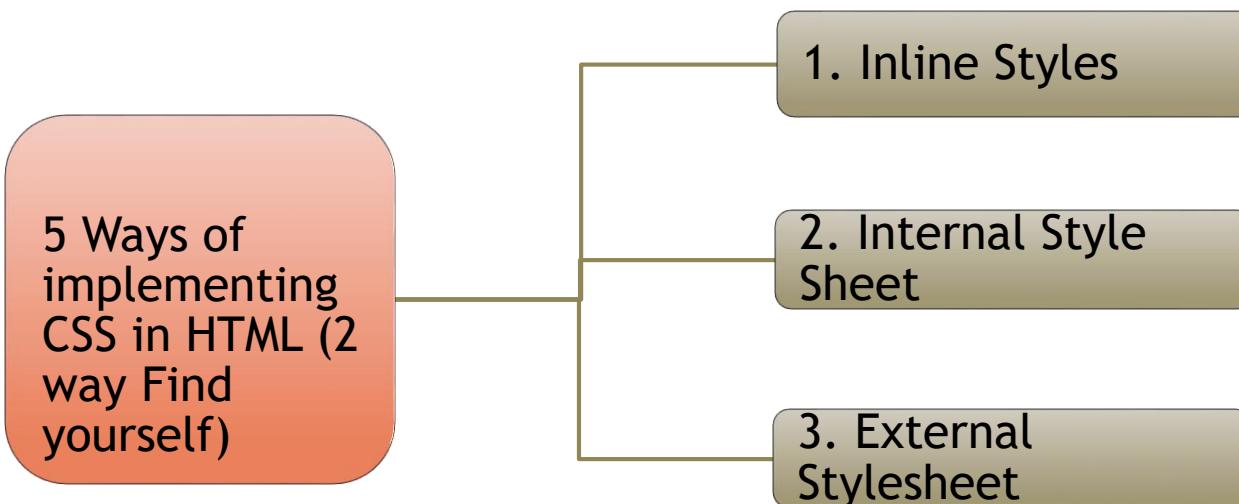


```
<html>
  <head>
    <meta charset="UTF-8" />
    <title>Inline CSS Example</title>
  </head>
  <body>
    <div style="background-color: #f0f0f0;">
      <h1 style="color: blue">Interview</h1>
      <p style="font-size: 30px">Happy</p>
    </div>
  </body>
</html>
```

Interview
Happy

CSS

Q. What is **CSS**? What are the 5 ways to **Implement CSS in HTML?** **V. IMP.**



Q. What is **Inline Style** in CSS? When to **use** it in real applications?

- ❖ Inline Styles apply styles directly to individual HTML elements using the **style attribute**.
- ❖ This method is suitable for applying styles to a **single element**.

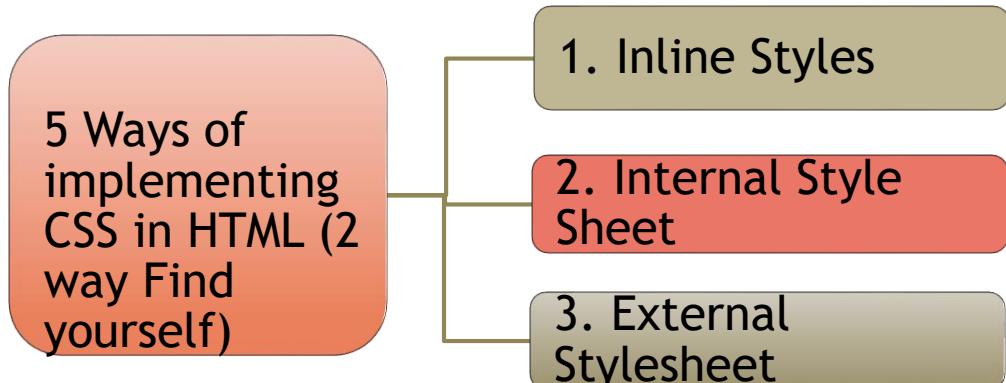
5 Ways of implementing CSS in HTML (2 way Find yourself)

- 1. Inline Styles
- 2. Internal Style Sheet
- 3. External Stylesheet

```
<html>
  <head> </head>
  <body>
    <h1 style="text-align: center">Inline Styles</h1>
    <p style="color: green">Paragraph</p>
  </body>
</html>
```

Q. What is Internal Stylesheet in CSS? When to use it in real applications?

- ❖ Internal Style Sheets can be implemented by adding the **<style>** element in the **<head>** section of HTML.
- ❖ This method is suitable for **smaller projects**.



```
<html>
  <head>
    <style>
      div {
        text-align: left;
      }
      .para {
        color: green;
      }
    </style>
  </head>
  <body>
    <div>Internal Style Sheet</div>
    <p class="para">Paragraph</p>
  </body>
</html>
```

Q. What is **External Stylesheet** in CSS? When to **use** it in real applications? **V. IMP.**

- ❖ In External Stylesheet, a **separate CSS file** is created(.css extension) and link it in the HTML file using the **<link> element**.
- ❖ This is most common and **recommended**.

5 Ways of implementing CSS in HTML (2 way Find yourself)

1. Inline Styles

2. Internal Style Sheet

3. External Stylesheet

```
# 2-CssTypes3.css > ...
1 h1 {
2   text-align: center;
3 }
4 p {
5   color: green;
6 }
```

```
<html>
| <head>
| | <link rel="stylesheet" href="2-CssTypes3.css">
| </head>
| <body>
| | <h1>Internal Style Sheet</h1>
| | <p>Paragraph</p>
| </body>
| </html>
```

Q. What are the **5 advantages** of External Stylesheet? **V. IMP.**

5 Advantages of CSS:

1. Separation of content(HTML) and presentation(Style).
2. Reusability of CSS classes in multiple elements.
3. Keeps things organized and **structured**.
4. Adapts to Different Devices.
5. Improves Website Speed.

1-Css2.css > ...

```
1 .container {  
2 | background-color: #f0f0f0;  
3 | padding: 20px;  
4 }  
5 .heading {  
6 | color: #333;  
7 }  
8 .paragraph {  
9 | font-family: Arial, sans-serif;  
10 }
```

Q. What are **Selectors** in CSS? How many **Types** of selectors are there? **V. IMP.**

- ❖ In CSS, selectors are patterns that are used to select and style HTML elements.

Types of Selectors

1. ID Selector(#)

2. Element Selector(no symbol)

3. Class Selector(.)

4. Universal Selector(*)

5. Descendant selector

6. Child Selector(>)

```
<html>
  <head>
    <link rel="stylesheet" href="3-Main.css" />
  </head>
  <body>
    <h1>Imported Styles</h1>
    <p>Paragraph</p>
  </body>
</html>
```

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

Q. Difference between **ID**, **Element** & **Class** selector? When to **use** which selector? **V. IMP.**

1. ID Selector (#)

- ID selectors are used for unique elements on a page, and they should only be assigned to one element.

```
/* # ID Selector */  
#my-id {  
    background-color: yellow;  
}
```

This background will be yellow.

2. Element Selector

- Element selectors are used when you want to apply a style to all elements of a specific type.

```
/* Element Selector */  
h2 {  
    font-size: 20px;  
}
```

This text font size will be 20px.

3. Class Selector (.)

- Class selectors are used when you want to apply a style to multiple elements with the same class.

```
/* . Class Selector */  
.my-class {  
    color: blue;  
}
```

This text will be blue.

Q. What are Universal selectors?

- ❖ Universal selector applied the specified styles to all elements, regardless of their type or attributes.

```
<body>
  <div id="my-id">
    | <p>Interview</p>
    | </div>

    <h2>Happy</h2>
  </body>
```

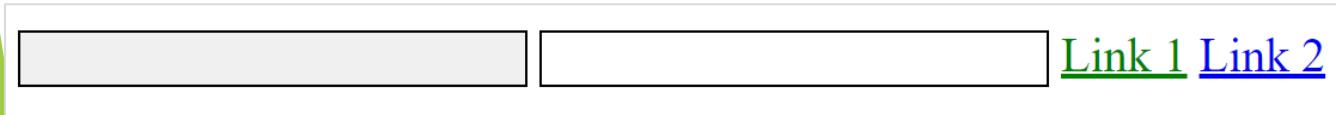
Interview
Happy

```
/* * Universal Selector */
* {
  color: blue;
  font-size: 24px;
}
```

Q. What are Attribute Selectors in CSS? What are its 5 Types?

- Attribute selectors in CSS target HTML elements based on their attributes and their values.

```
<input type="text"/>
<input type="password"/>
<a href="https://www.example.com">Link 1</a>
<a href="http://www.example.pdf">Link 2</a>
```



```
/* Attribute Exists Selector */
input[type] {
| border: 1px solid □black;
}

/* Attribute Equals Selector */
a[href="https://www.example.com"]
{
| color: □blue;
}

/* Attribute Contains Selector */
input[type*="text"] {
| background-color: ■#f0f0f0;
}

/* Attribute Starts With Selector */
a[href^="https://"]
{
| color: □green;
}

/* Attribute Ends With Selector */
a[href$=".pdf"] {
| text-decoration: underline;
}
```

back to chapter

Q. What are Pseudo-class Selector and Pseudo-element Selector?

- ❖ A pseudo-class selector is used to define the **special state** of an HTML element, like when a user hovers over an element.
- ❖ A pseudo-element selector targets a **specific part** of an element, such as ::before to insert content before an element.

```
<body>
  <div class="box">
    <p>Hover over me</p>
    <span>Content added before me</span>
  </div>
</body>
```

```
/* Pseudo-class Selector */
p:hover {
  color: blue;
}
```

Hover over me

```
/* Pseudo-element Selector */
span::before {
  content: "Yes: ";
  font-weight: bold;
}
```

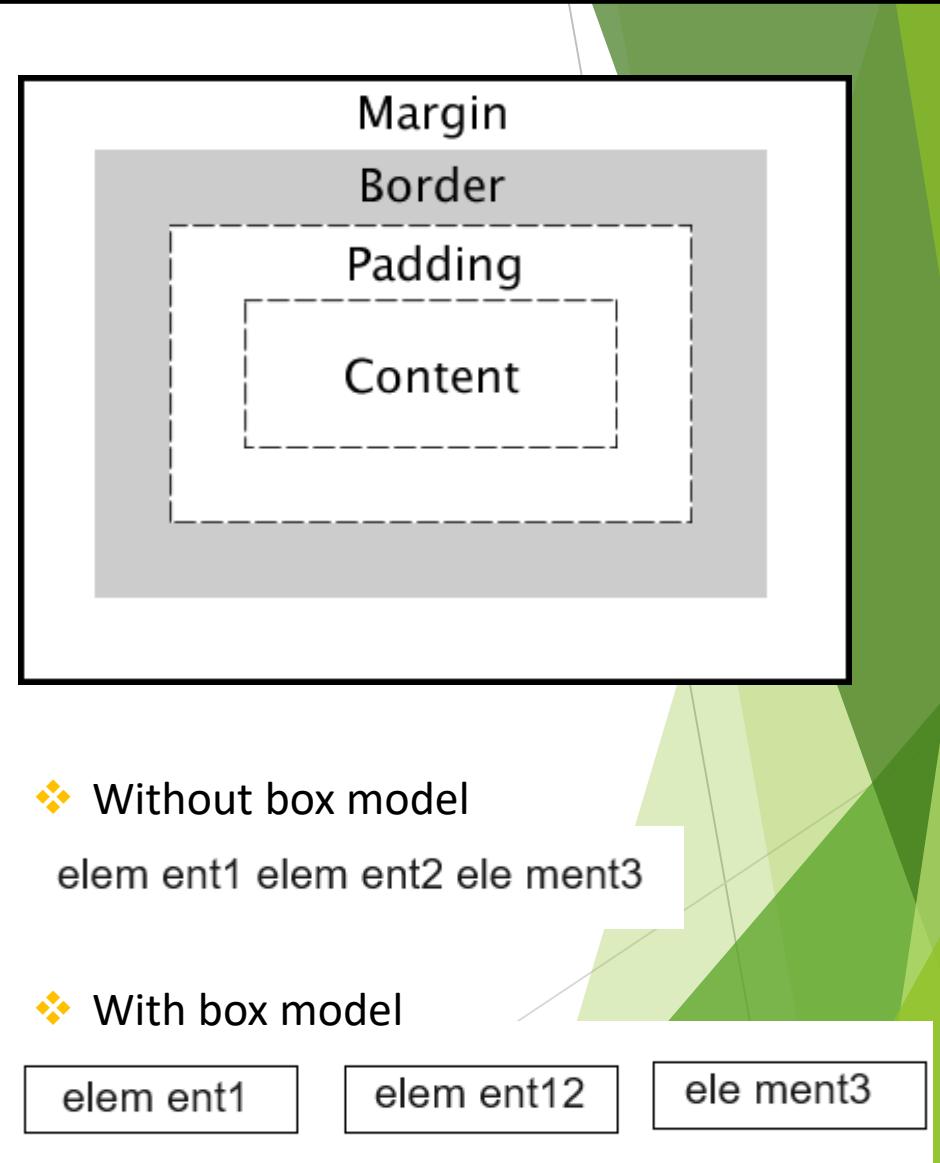
Yes: Content added before me

Q. Explain the concept of the Box Model in CSS. **V. IMP.**

- ❖ The "box model" describes the **layout of elements** on a webpage in terms of their dimensions and spacing.

- 1. **Content:** This is the actual content of the element, such as text, images, videos, etc.
- 2. **Padding:** This is the space between the content and the border.
- 3. **Border:** This is the line that separate the content from the surrounding elements.
- 4. **Margin:** This is the space between the borders of adjacent elements. Margins help control the space between elements.

- ❖ In CSS, you can control each of these components using various properties like width, height, padding border, margin.



Q. What is the difference between `display: inline` & `display: block`? V. IMP.

- ❖ Block elements take up the **full width** of their parent container by default
- ❖ Inline elements take up as much width as per the **width of content only**.

```
<body>
  <h3>Display Properties</h3>
  <div class="block">Block Element</div>
  Text1
  <div class="inline">Inline Element</div>
  Text2
</body>
```

```
.block {
  display: block;
  border: 1px solid black;
  padding: 20px;
  margin: 20px;
}
```

Block Element

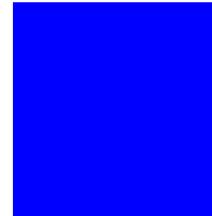
```
.inline {
  display: inline;
  border: 1px solid black;
  padding: 20px;
  margin: 20px;
}
```

Text1 Inline Element Text2

Q. What is z-index property in CSS? How same z-index elements will be displayed?

- ❖ z-index property in CSS is used to control the **stacking order** of positioned elements.
- ❖ Elements with a higher z-index value will be displayed on top of elements with a lower z-index value.
- ❖ If two elements have the same z-index, the one that comes later in the HTML source will appear on top.

```
<body>
  <div class="container">
    |  <div class="box1"></div>
    |  <div class="box2"></div>
    |
  </div>
</body>
```



```
.box1 {
  position: absolute;
  width: 100px;
  height: 100px;
  background-color: ■ red;
  z-index: 1;
}

.box2 {
  position: absolute;
  width: 100px;
  height: 100px;
  background-color: ■ blue;
  z-index: 2;
}
```

Q. What is the role of **float** property in CSS?

- ❖ The float property in CSS is used to specify how an element should align or "float" within its parent container.

```
<body>
  <div class="sidebar">
    <h2>Sidebar</h2>
  </div>

  <div class="main-content">
    <h1>Main Content</h1>
  </div>
</body>
```

```
.sidebar {
  width: 30%;
  float: right;
  background-color: lightblue;
  padding: 20px;
  box-sizing: border-box;
}

.main-content {
  width: 70%;
  padding: 20px;
  box-sizing: border-box;
  background-color: lightgreen;
}
```

Main Content

Sidebar

Q. What is the difference between position: relative/ absolute/ fixed? **V. IMP.**

- ❖ position: relative positions an element relative to its normal position of the page.
- ❖ position: absolute is positioned as per the parent element.
- ❖ position: fixed will fix the position of element even if the page is scrolled.

```
<body>
  <div class="relative-box">
    <h2>Relative Box</h2>
  </div>

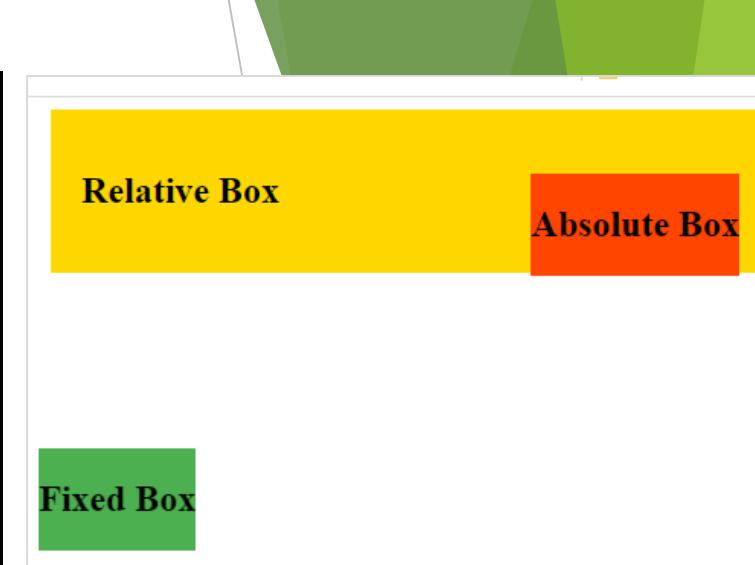
  <div class="absolute-box">
    <h2>Absolute Box</h2>
  </div>

  <div class="fixed-box">
    <h2>Fixed Box</h2>
  </div>
</body>
```

```
.relative-box {
  position: relative;
  left: 20px;
  padding: 20px;
  background-color: #ffd700;
}

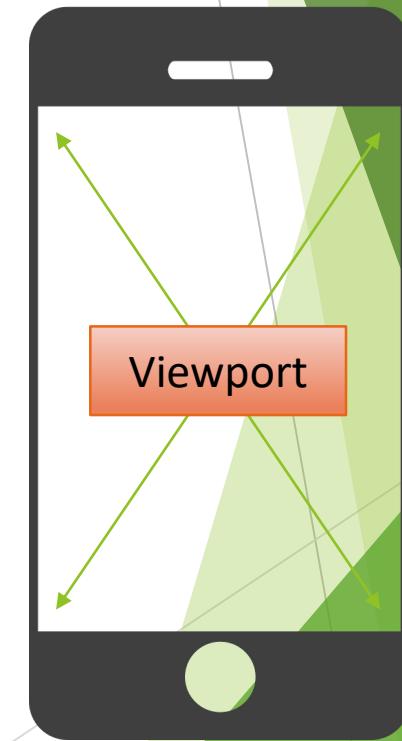
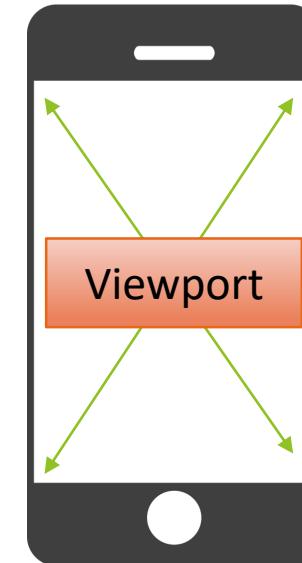
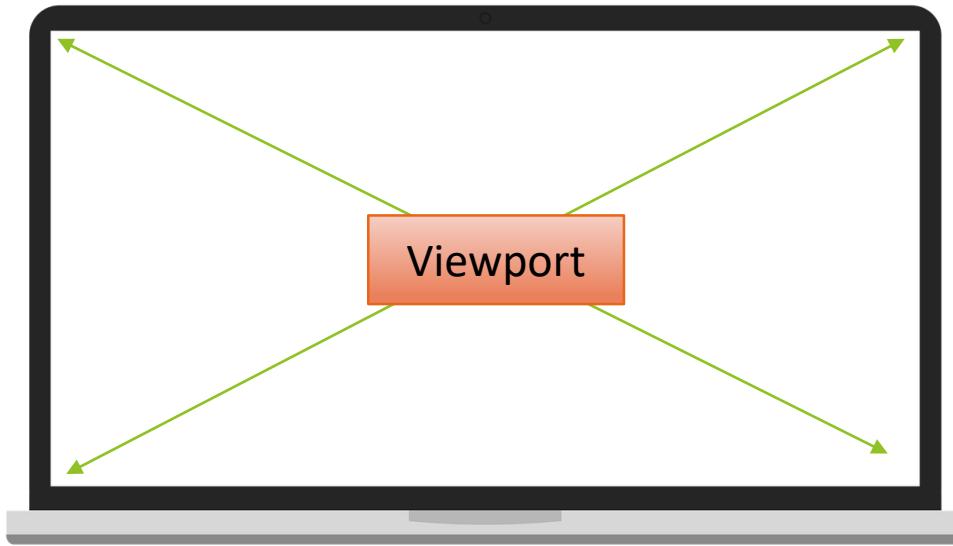
.absolute-box {
  position: absolute;
  top: 50px;
  right: 20px;
  background-color: #ff4500;
}

.fixed-box {
  position: fixed;
  bottom: 50px;
  left: 20px;
  background-color: #4caf50;
}
```



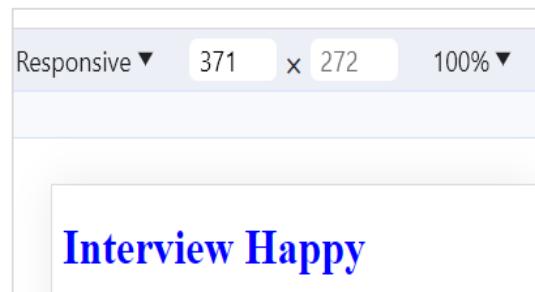
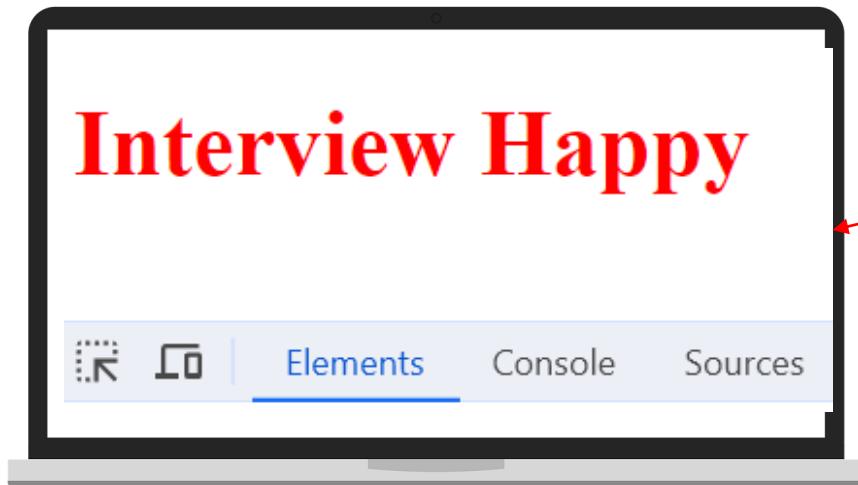
Q. What is the role of **meta viewport** tag? **V. IMP.**

- ❖ In web design, the "viewport" is the portion of the webpage that the user can see on their screen at any given time, without scrolling.



Q. What are Media Queries in CSS? **V. IMP.**

- Media queries in CSS allows you to apply different styles to elements based on the size of device screen.



```
<body>
  <h1>Interview Happy</h1>
</body>
```

```
h1 {
  color: red;
}
@media screen and (max-width: 768px) {
  h1 {
    color: blue;
  }
}
```

Q. What is **display: flex** in CSS?

- ❖ **display:flex** turns an element into a container that structures and aligns its child elements in a better way. It enables a flexbox layout for the container.
- ❖ **justify-content** property aligns flex items along the main axis of the flex container. It controls the spacing between and around items.

Home About Services Contact



Home

About

Services

Contact



```
<body>
  <nav class="nav-container">
    <a href="#">Home</a>
    <a href="#">About</a>
    <a href="#">Services</a>
    <a href="#">Contact</a>
  </nav>
</body>
```

```
.nav-container {
  display: flex;
  justify-content: space-around;
  background-color: #333;
  padding: 10px;
}
```

```
.nav-container a {
  color: white;
}
```

Q. What is CSS Grid Layout in CSS?

- ❖ The CSS Grid Layout Module offers a grid-based layout system, with rows and columns.

```
<body>
  <div class="grid-container">
    <div>1</div>
    <div>2</div>
    <div>3</div>
    <div>4</div>
    <div>5</div>
    <div>6</div>
  </div>
</body>
```

Without grid css

1
2
3
4
5
6

With grid css

```
.grid-container {
  display: grid;
  grid-template-columns: auto auto auto;
}
```

1	2	3
4	5	6

Q. What are the 5 best practices for creating a **Responsive Design** using CSS? **V. IMP.**

1. Use Responsive Units:

- Use relative units like percentages (%) or em units for sizes (width, height, padding, margin, etc.) instead of fixed units like pixels (px).
-

•2. Use Media Queries:

- Media queries allow you to apply different styles based on the characteristics of the device, such as screen width, height, and orientation. They are essential for creating a responsive design.
-

•3. Flexible Grid Systems:

- Use CSS Grid or Flexbox to create flexible layouts that can adapt to different screen sizes and orientations.
-

•4. Images and Media:

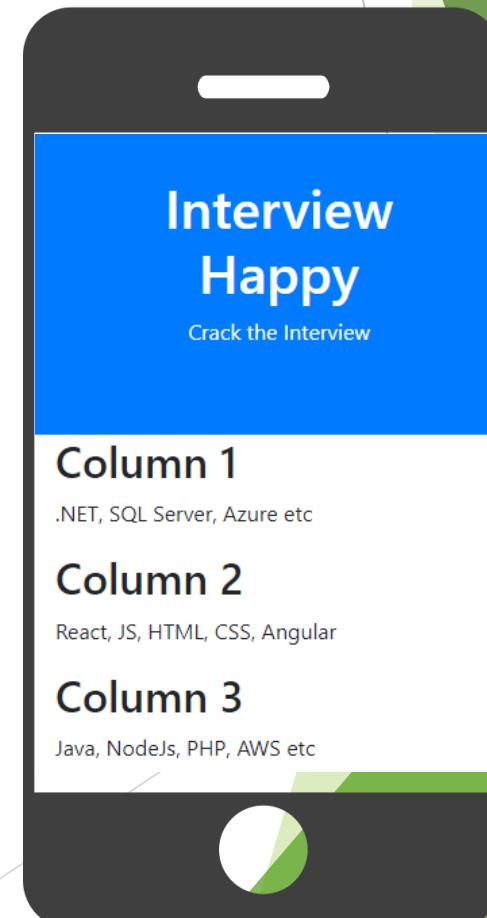
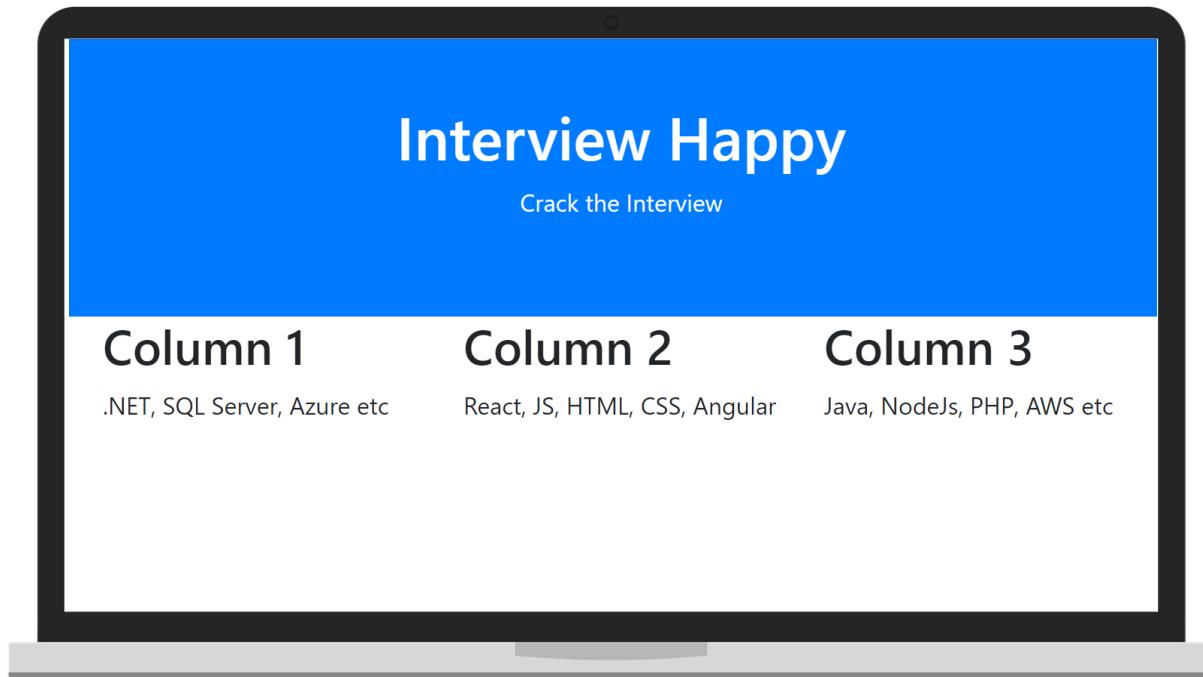
- Use max-width: 100%; on images to ensure they don't overflow their containers. This allows images to scale down proportionally on smaller screens.
-

•5. Viewport Meta Tag:

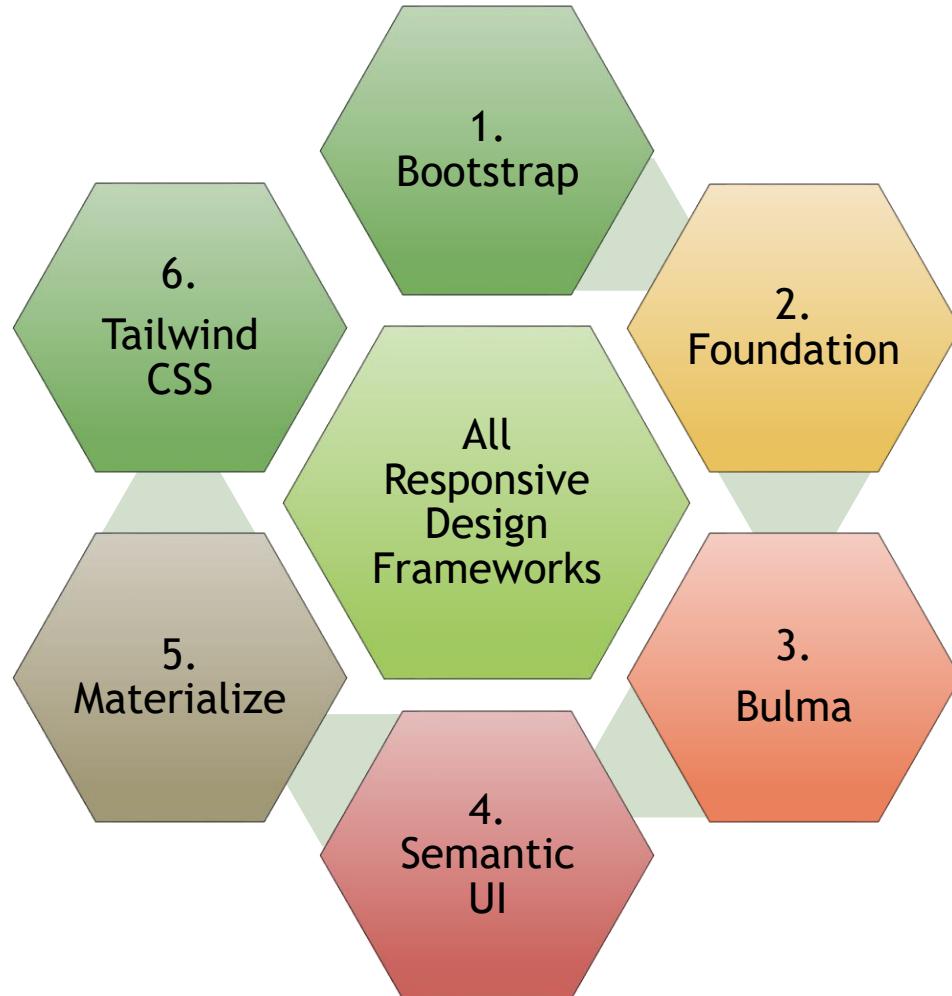
- Include the viewport meta tag in the <head> of your HTML to control how the page is displayed on mobile devices. It sets the initial scale, width, and zoom behavior.
-

Q. What is Bootstrap? What are the other 5 responsive design frameworks? **V. IMP.**

- ❖ Bootstrap is a popular open-source front-end framework which provides responsive and mobile-first CSS.



Q. What is Bootstrap? What are the other 5 responsive design frameworks? **V. IMP.**



Q. What are the 5 Advantages of using Bootstrap?

1. Rapid Development

- Bootstrap provides a wide range of ready-to-use components like navigation bars, buttons, forms, modals, and more. This saves time and effort in writing code from scratch.

2. Responsive Design

- Bootstrap is built with a mobile-first approach, meaning it prioritizes designing for smaller screens first and then scales up for larger screens. This ensures websites with Bootstrap are responsive for every device.

3. Cross-Browser Compatibility

- Bootstrap provides a consistent set of styles and components across different browsers, which helps in reducing cross-browser compatibility issues.

4. Large Community and Support

- This means there are plenty of resources, documentation, tutorials, and forums available for assistance.

5. Accessibility

- Bootstrap incorporates best practices for web accessibility, making it easier to create websites that are usable by people with disabilities.

Q. What are the **2 ways** to **include Bootstrap framework** for your website.

2 ways to include Bootstrap:

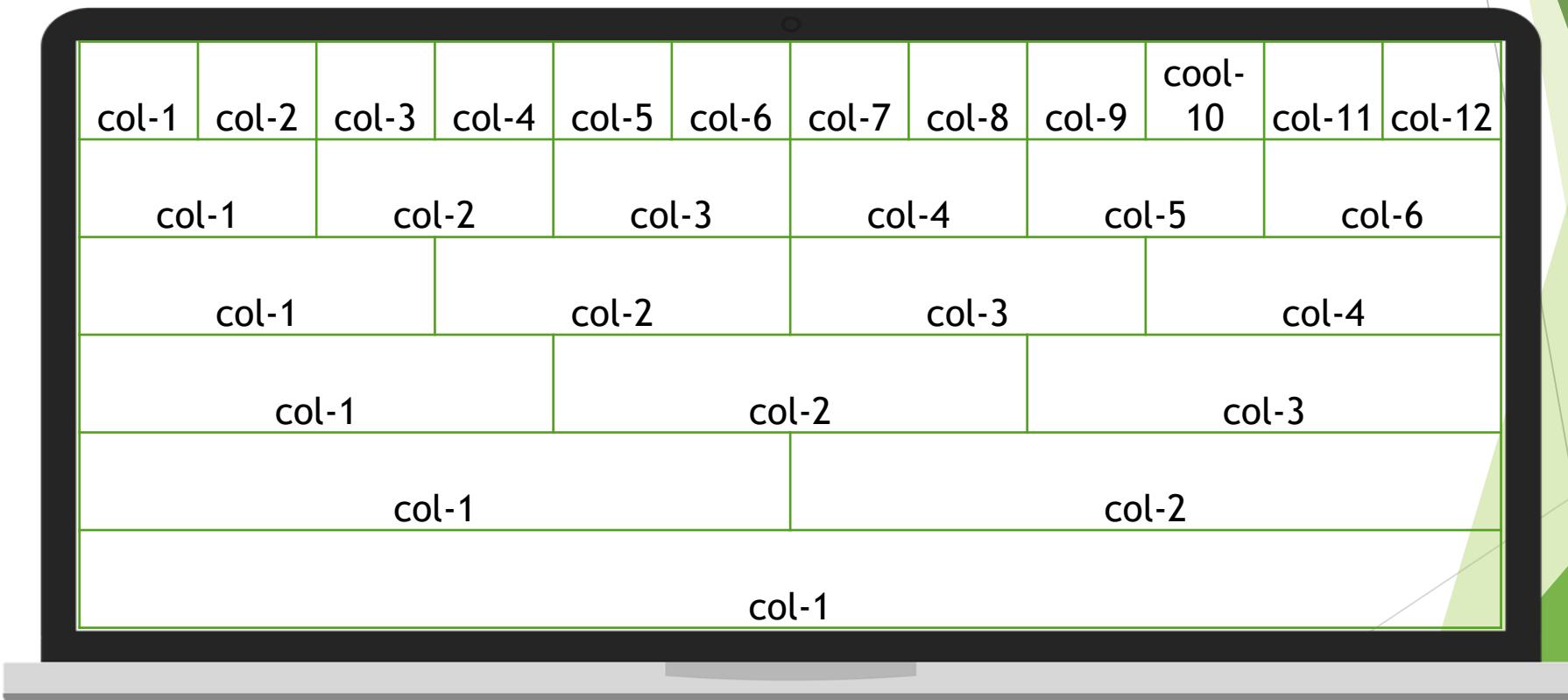
1. Put the link of Bootstrap CDN (Content Delivery Network) in head section.

2. Download the Bootstrap files and add them locally in your HTML document.

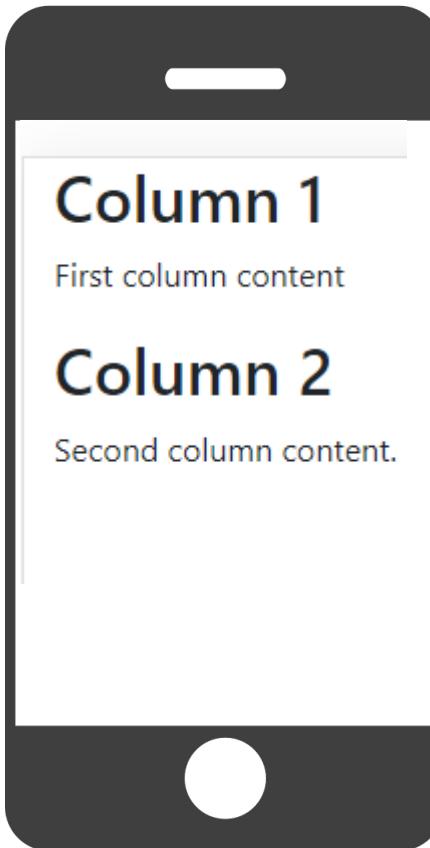
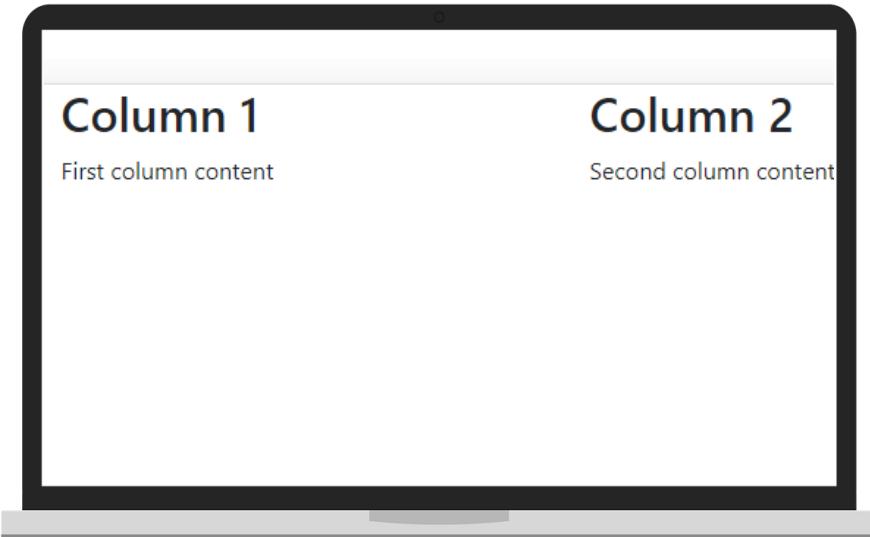
```
<head>
  <title>Bootstrap</title>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1" />
  <!-- Add Bootstrap CSS via CDN -->
  <link rel="stylesheet"
    | href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"
    |
  />
</head>
```

Q. Explain the Grid System in Bootstrap? **V. IMP.**

- ❖ Grid system is 12-column layout and is designed to adapt to various screen sizes.



Q. Explain the Grid System in Bootstrap? **V. IMP.**

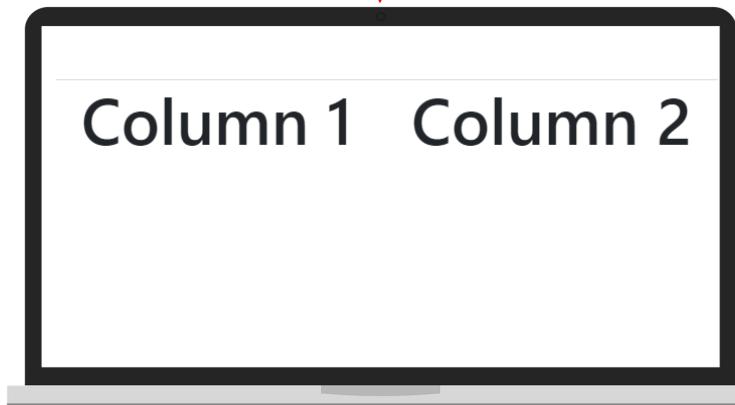


```
<body>
  <div class="container">
    <div class="row">
      <div class="col-md-6">
        <h2>Column 1</h2>
        <p>First column content</p>
      </div>
      <div class="col-md-6">
        <h2>Column 2</h2>
        <p>Second column content.</p>
      </div>
    </div>
  </div>
</body>
```

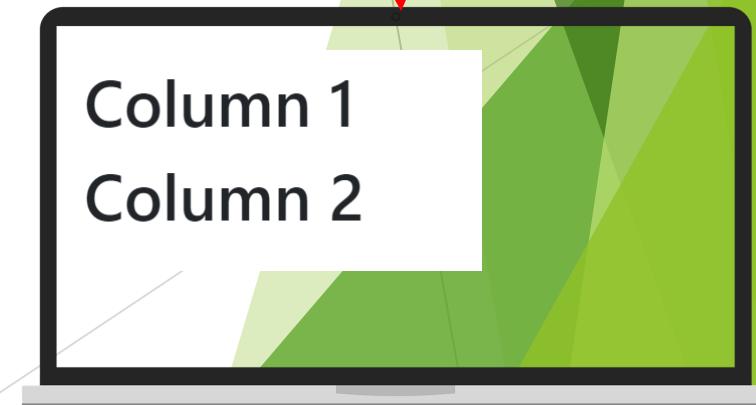
Q. What is the difference between col-xs, col-sm, col-md, col-lg & col-xl?

1. col-xs (Extra Small): Applies to extra small screens (phones).
2. col-sm (Small): Applies to small screens (tablets).
3. col-md (Medium): Applies to medium screens (laptops).
4. col-lg (Large): Applies to large screens (larger desktops).
5. col-xl (Extra Large): Applies to extra large screens (large TVs).

```
<div class="container">
  <div class="row">
    <div class="col-md-6">
      | <h2>Column 1</h2>
    </div>
    <div class="col-md-6">
      | <h2>Column 2</h2>
    </div>
  </div>
</div>
```



```
<div class="container">
  <div class="row">
    <div class="col-xl-6">
      | <h2>Column 1</h2>
    </div>
    <div class="col-xl-6">
      | <h2>Column 2</h2>
    </div>
  </div>
</div>
```

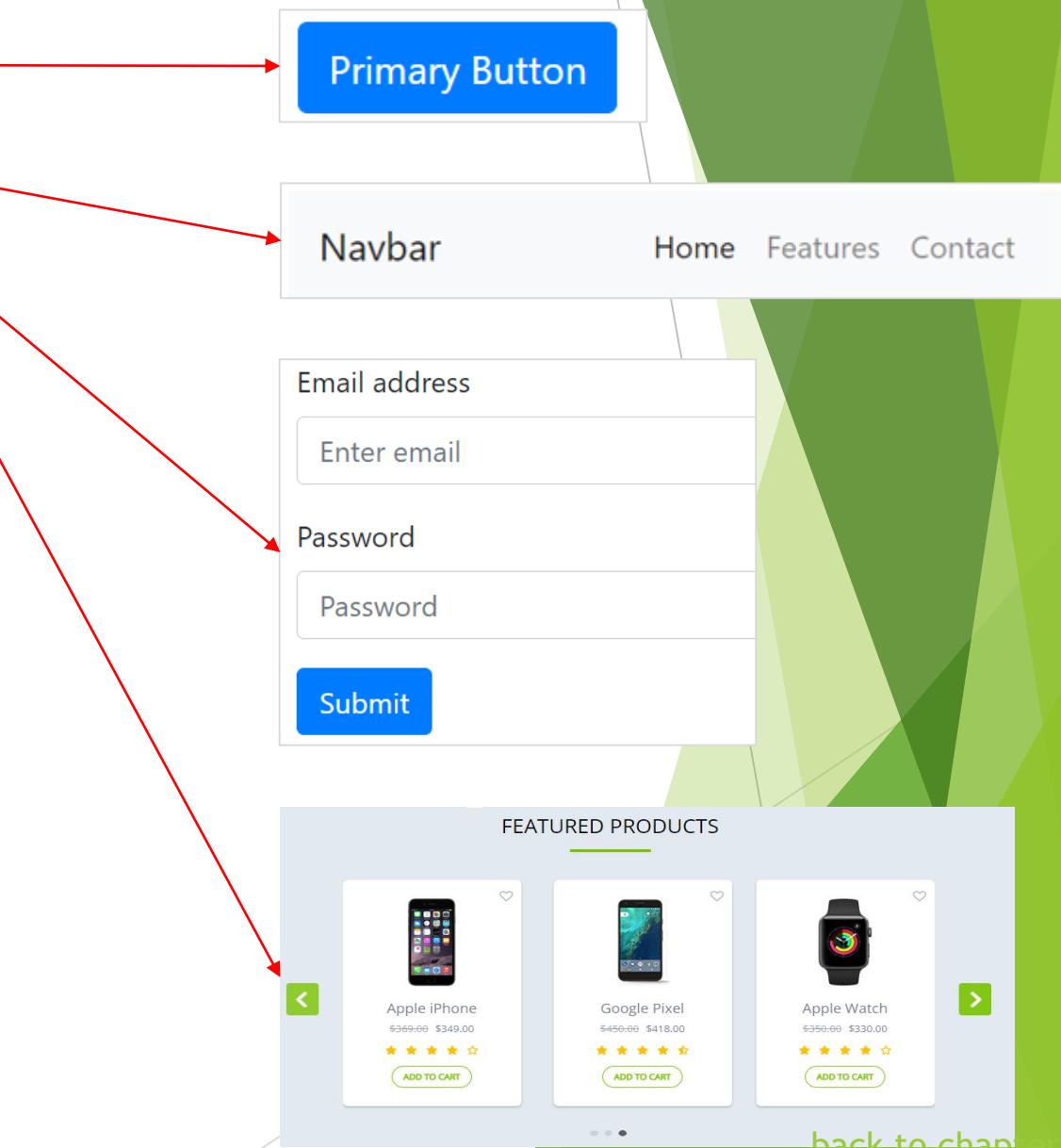


Q. What are Bootstrap Components? What are Top 10 bootstrap components? **V. IMP.**

- ❖ Bootstrap components are **pre-designed elements** that can be easily integrated into a web project.

Top 10 Bootstrap components

1. Buttons
2. Navbar
3. Forms
4. Carousel
5. Cards
6. Progress Bars
7. Pagination
8. Spinners
9. Badges
10. Modals

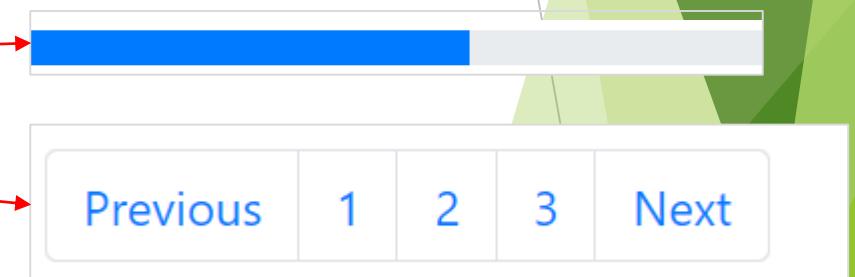
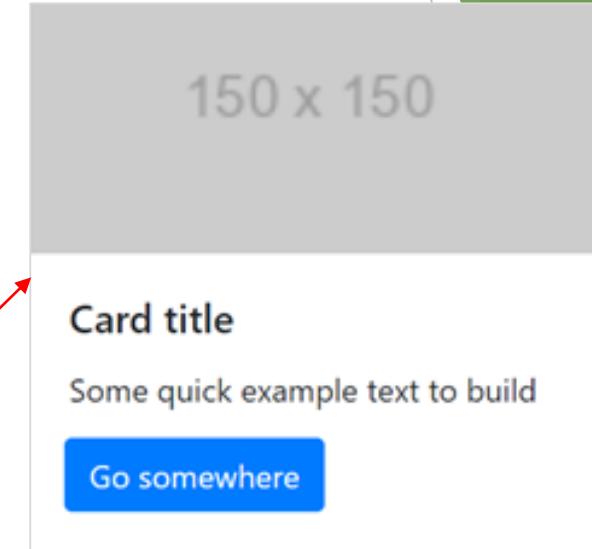


Q. What are Bootstrap Components? What are Top 10 bootstrap components? **V. IMP.**

- ❖ Bootstrap components are **pre-designed elements** that can be easily integrated into a web project.

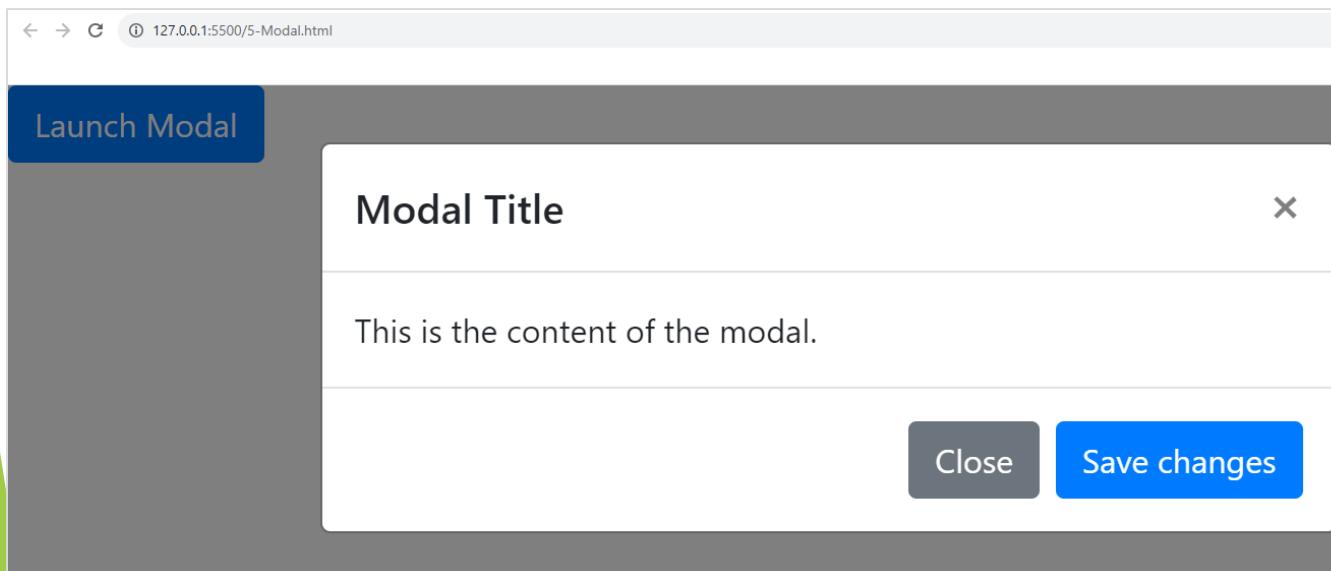
Top 10 Bootstrap components

1. Buttons
2. Navbar
3. Forms
4. Carousel
5. Cards
6. Progress Bars
7. Pagination
8. Spinners
9. Badges
10. Modals



Q. What is a Bootstrap Modal component?

- ❖ A Bootstrap modal is a lightweight, customizable, and interactive dialog box or popup window that appears on top of the main content of a web page.

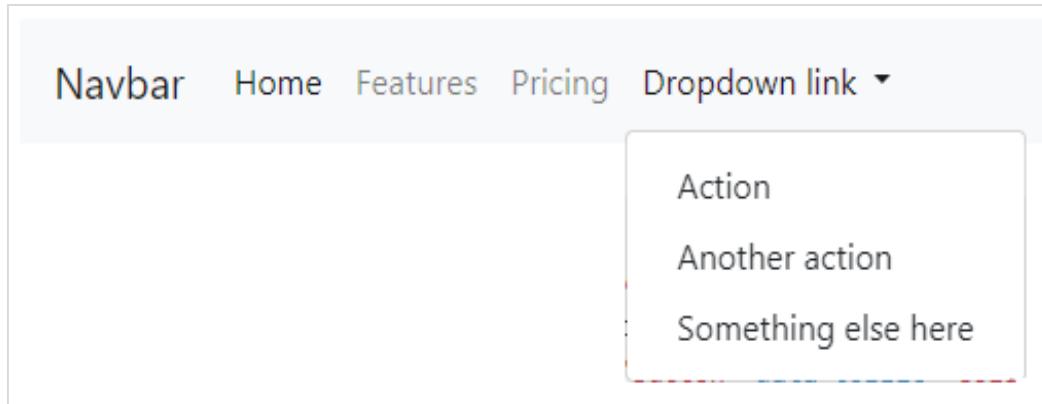


```
<body>
  <!-- Button to trigger the modal -->
  <button type="button" class="btn btn-primary" data-toggle="modal" data-target="#exampleModal">
    Launch Modal
  </button>

  <!-- Modal content -->
  <div class="modal fade" id="exampleModal" tabindex="-1" role="dialog" aria-labelledby="exampleModalLabel" aria-hidden="true">
    <div class="modal-dialog">
      <div class="modal-content">
        <div class="modal-header">
          <h5 class="modal-title" id="exampleModalLabel">Modal Title</h5>
          <button type="button" class="close" data-dismiss="modal" aria-label="Close">
            <span aria-hidden="true">&times;</span>
          </button>
        </div>
        <div class="modal-body">
          This is the content of the modal.
        </div>
        <div class="modal-footer">
          <button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>
          <button type="button" class="btn btn-primary" data-dismiss="modal">Save changes</button>
        </div>
      </div>
    </div>
  </div>
</body>
```

Q. What is a Bootstrap **Navigation** component?

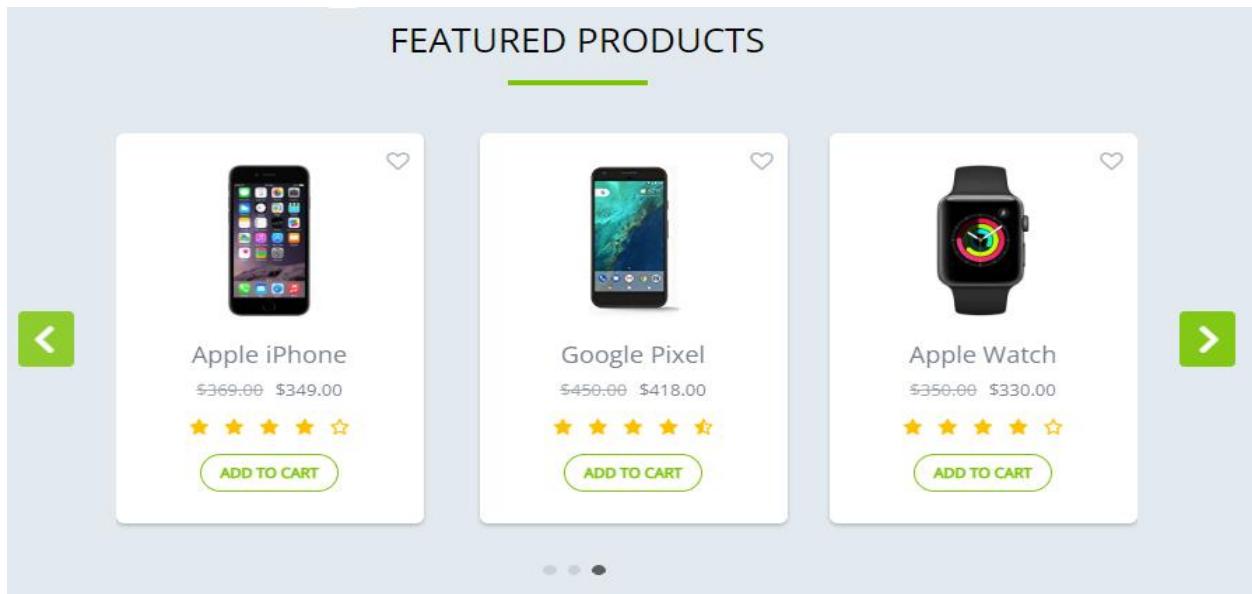
- ❖ Bootstrap's Navigation component provides a set of styles and components to create various types of navigation menus on a webpage.



```
<body>
  <nav class="navbar navbar-expand-lg navbar-dark">
    <a class="navbar-brand" href="#">Navbar</a>
    <button
      class="navbar-toggler"
      type="button"
      data-toggle="collapse"
      data-target="#navbarNav"
      aria-controls="navbarNav"
      aria-expanded="false"
      aria-label="Toggle navigation"
    >
      <span class="navbar-toggler-icon"></span>
    </button>
```

Q. What is Bootstrap **Carousel** component?

- ❖ Bootstrap's carousel component is a slideshow or image slider that allows you to display a series of images or content in a rotating manner.

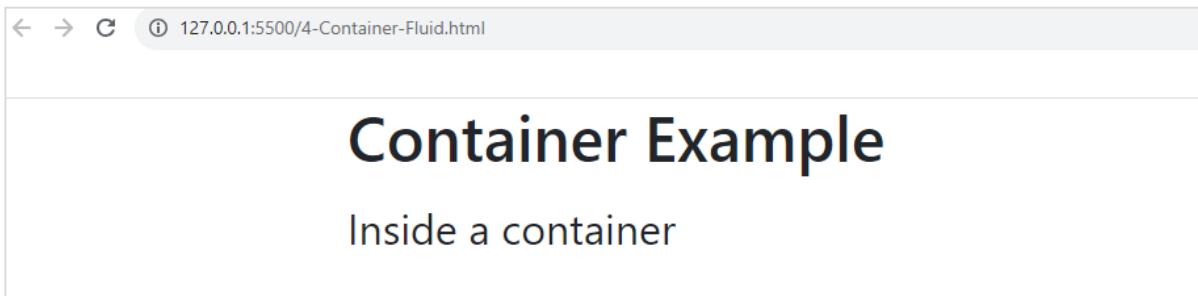


```
<div id="myCarousel" class="carousel slide">
  <!-- Indicators -->
  <ul class="carousel-indicators">
    <li data-target="#myCarousel" data-slide-to="0" class="active"></li>
    <li data-target="#myCarousel" data-slide-to="1"></li>
    <li data-target="#myCarousel" data-slide-to="2"></li>
  </ul>

  <!-- Slides -->
  <div class="carousel-inner">
    <div class="carousel-item active">
      
    </div>
    <div class="carousel-item">
      
    </div>
    <div class="carousel-item">
      
    </div>
  </div>
```

Q. Explain the difference between Bootstrap's **container** and **container-fluid**?

- ❖ .container class **centered the content** and add padding to the sides, ensuring that the content remains within a specified width.
- ❖ .container-fluid class provides a **full-width layout**, with the content extending to the edges of the screen.



```
<body>
  <div class="container">
    <h4>Container Example</h4>
    <p>Inside a container</p>
  </div>
</body>
```



```
<body>
  <div class="container-fluid">
    <h4>Container Example</h4>
    <p>Inside a container</p>
  </div>
</body>
```



All the best for your interviews

Never ever give up

Jagmohan Rai