# Module Name: HTML5

# HTML5

## OBJECTIVES

**1**   To equip students with the basic understanding of HTML5, its applications and Why it is preferred over Flash!

**2**   Students will learn about different Semantic/Form Elements introduced with HTML5

**3**   To make students able to implement different multimedia (Audio/Video) elements inside a HTML page

# HTML5

## OUTCOMES

- At the end of this module students are expected to learn

✓ Features added to HTML5 & its applications

✓ Create an HTML5 document & add different semantic elements & attributes to it

✓ Understand & apply form validations to HTML5 document

✓ Describe & implement the common audio/video formats available on the internet

# HTML5

- Document/Video Links

| Topics | URL |
|--------|-----|
| Basic HTML5 Tutorial | https://www.tutorialrepublic.com/html-tutorial/ |
| Online Web Tutorial (for quick reference) | https://www.w3schools.com/html/default.asp <br><br> https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/HTML5 <br><br> https://dev.w3.org/html5/html-author/ |
| HTML5 Tutorials | https://www.youtube.com/watch?v=kDyJN7qQETA <br><br> https://www.youtube.com/watch?v=g3L2jUmnuNA <br><br> https://www.youtube.com/watch?v=DPnqb74Smug |

# Content

- Introducing HTML5

- Semantic Elements

- Form Validations

- Multimedia

# Introducing HTML5

- HTML5 (particularly HTML5.2) is the latest version of the W3's HTML

- It redefines the rules and semantics of the existing HTML elements

- It adds new elements to support multimedia and web applications

- It also defines a parsing algorithm for generating DOM (Document Object Model) tree, even from ambiguous or poor quality markups

- To know more, head to W3's specifications for HTML5

    https://www.w3.org/TR/html52/index.html#contents

# Introducing HTML5

- Applications of HTML5

  - Various Tags
  - Multimedia Support
  - Form Validations
  - Local Storage
  - Offline pages
  - WebSQL
  - Cache Storage
  - Cookies
  - Drag n Drop
  - Canvas & SVG
  - Geolocation

To know the difference

→

HTML5 & Flash

Flashvhtml.com

# Semantic Elements

- Every HTML document is made from elements, represented by Tags

  - Tags (<... >)are a sequence of characters that marks where an elements start & stops

- The element name may be followed by an attribute (or series of attributes)

- An attribute defines how the instance of an element should behave

- E.g.



element          Attributes
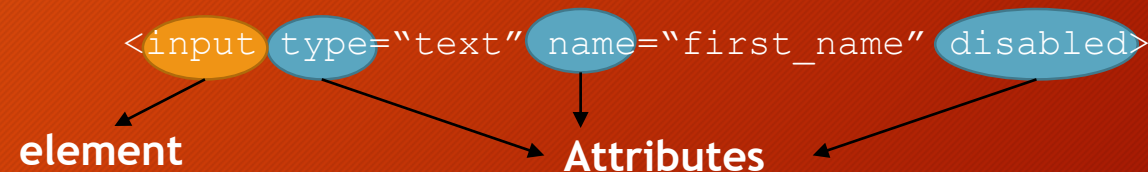
# Semantic Elements

- First HTML5 Document

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta http-equiv="X-UA-Compatible" content="ie=edge">
<title>Document</title>
</head>
<body>
<h1>Hello World!</h1>
</body>
</html>
```

This is how the browser knows that we're sending HTML5

Defines character encoding

Area of the window in which web content can be seen

Allows web developers to choose what version of IE the page should be rendered as

**Hello World!**

# Semantic Elements

- Defining Page Structure

  - Semantic elements refers to the way a given HTML element describes the meaning of its content

  - Main parts of a HTML5 page is divided up as follows:

    - **Header section** with a logo & title
    - **Navigation bar**
    - Body content divided up into three columns ⟶ Using &lt;section&gt;
    - **Articles** and ad blocks within the columns
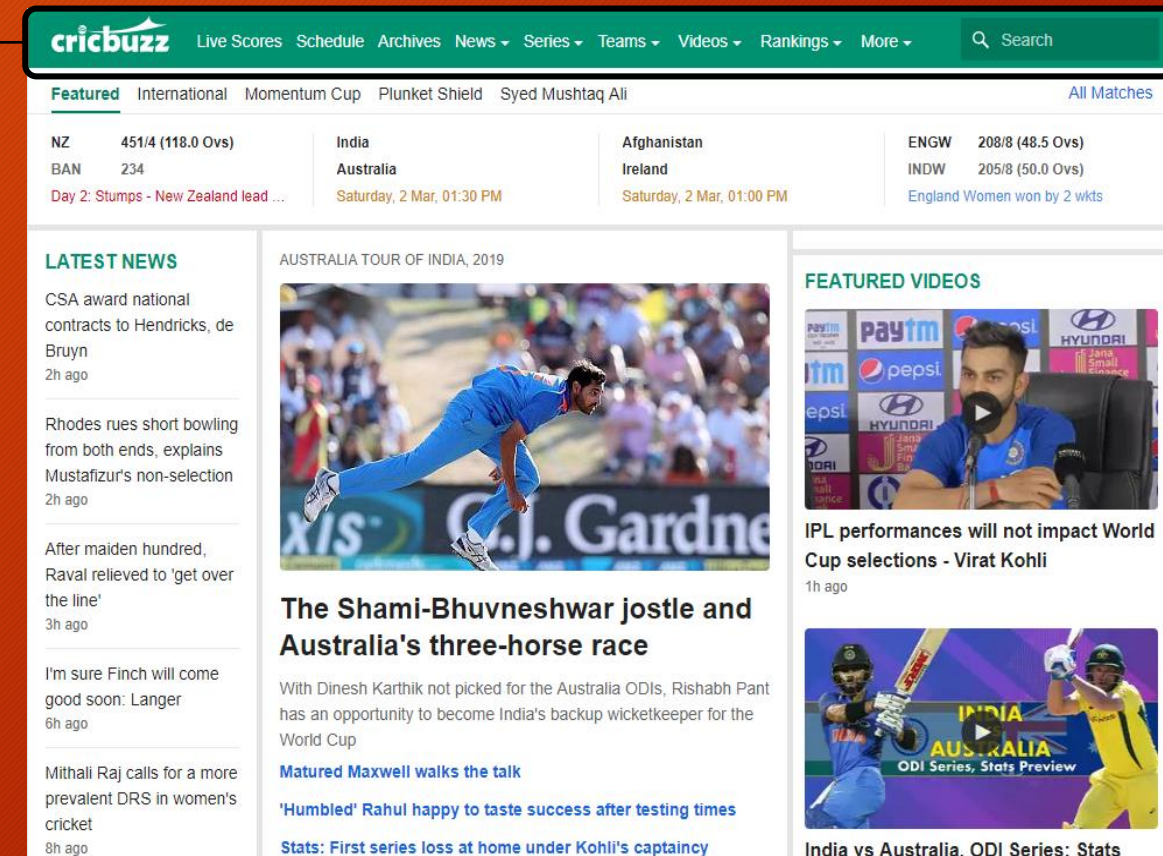    - **Footer** containing some author and copyright information

# Semantic Elements

- Defining Page Structure

  - `<header>` Element

    - It is used to include introductory content or navigational aids that are specific to any single section of a pge

    - It is usually placed at the top of a page or section

    - It can contains title, logo, search bar, menu bar etc.

    - E.g. Cricbuzz website header

# Semantic Elements

- Defining Page Structure

    - `<main>` Element

        - Represents the dominant content inside the `<body>` of a document

        - Consists of content that is directly related to the central topic of a document

        - A document must not have more than one `<main>` element

        - It doesn't contribute to the document's outline
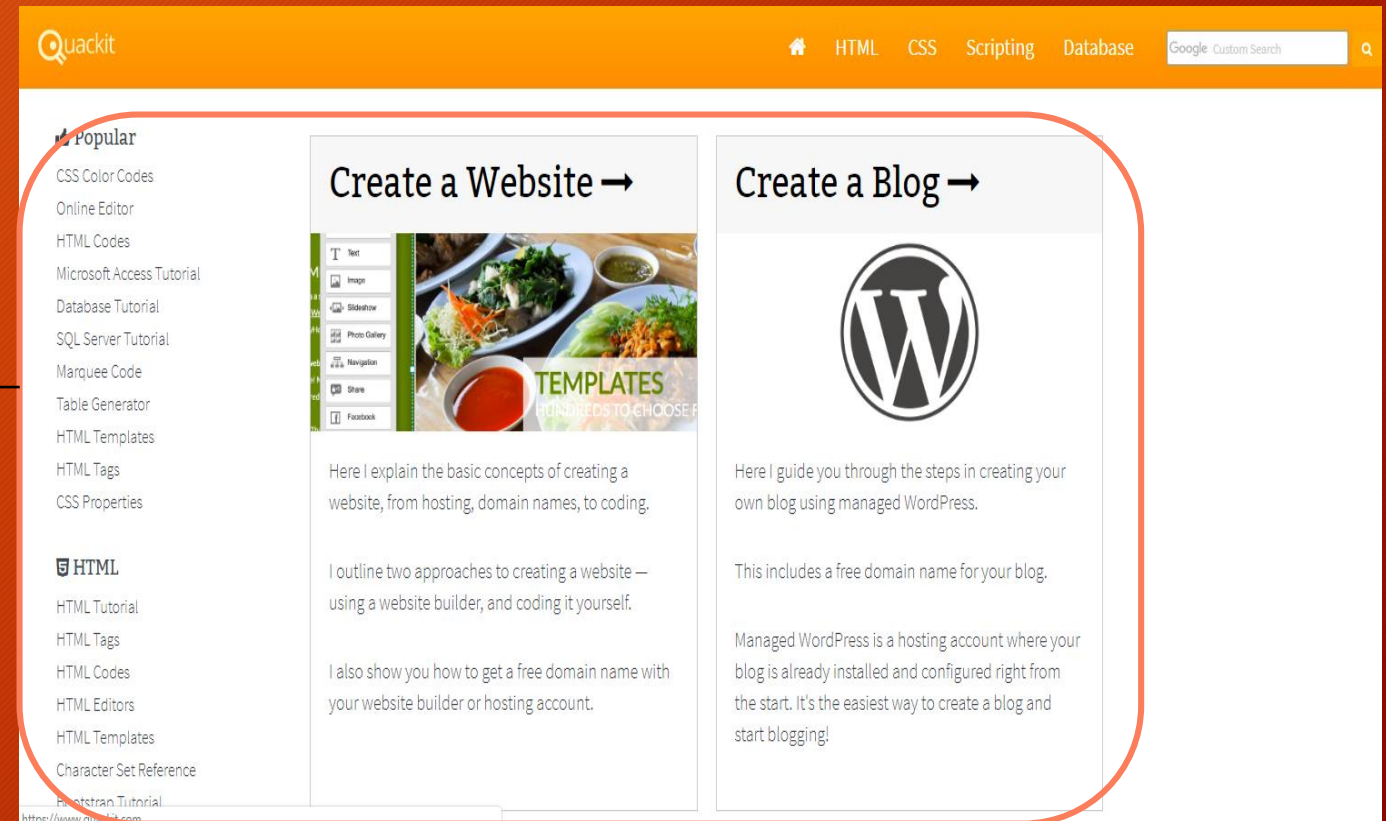
# Semantic Elements

- Defining Page Structure

  - `<section>` Element

    - It defines a generic section of a document or application

    - It is a thematic grouping of content, typically with a heading

    - E.g.

      - Individual sections of a tabbed interface

      - Various sections of an online news site; articles would be grouped into sections covering sports, economic news & world affairs

# Semantic Elements

- Defining Page Structure

  - `<article>` Element

    - It represents a complete composition in a document, page or website

    - It can be independently distributable or reusable

    - It is a single piece of content that can stand on its own

    - E.g.
      - A forum post
      - A magazine or newspaper article (in syndication)
      - A blog entry
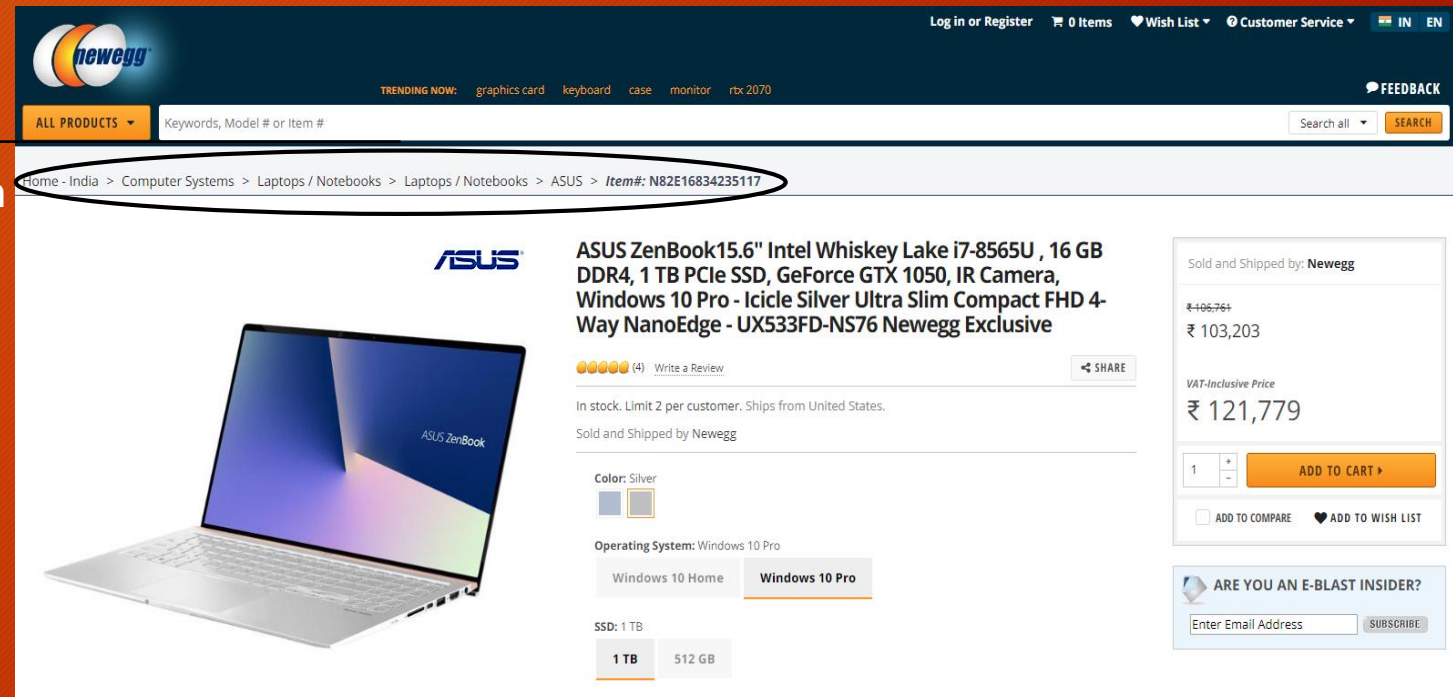      - A user-submitted comment on a blog entry or article

# Semantic Elements

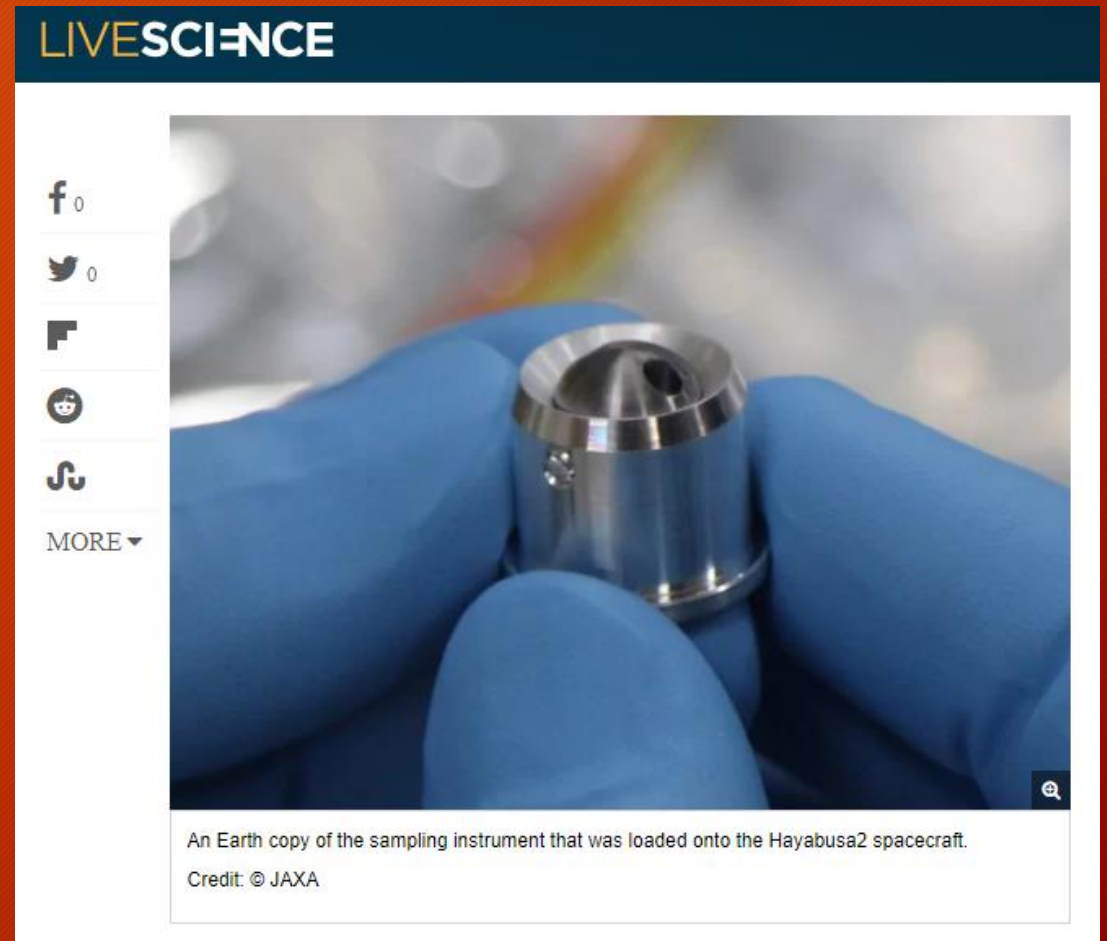- Defining Page Structure

  - `<nav>` Element

    - It represents a group of navigation links

    - Most commonly used for wrapping an unordered list of links

    - Appropriate for pagination or *breadcrumb links*

    - E.g. menus | table of contents | indexes

# Semantic Elements

- Defining Page Structure

  - `<figure>` element

    - Represents self-contained content (with an optional caption using `<figcaption>`)

    - It is usually an image, illustration, diagram or a code snippet that is referenced in the main flow of a document

    - Figure, its caption and its contents are referenced as a single unit



An Earth copy of the sampling instrument that was loaded onto the Hayabusa2 spacecraft.
Credit: © JAXA

# Semantic Elements

- Defining Page Structure

  - `<aside>` Element

    - Represents a portion of a document whose content is only indirectly related to the documents main content

    - Usually presented as sidebars or call-out boxes

    - Don't use `<aside>` element to tag parenthesized text, as this kind of text is considered part of the main flow

# Semantic Elements
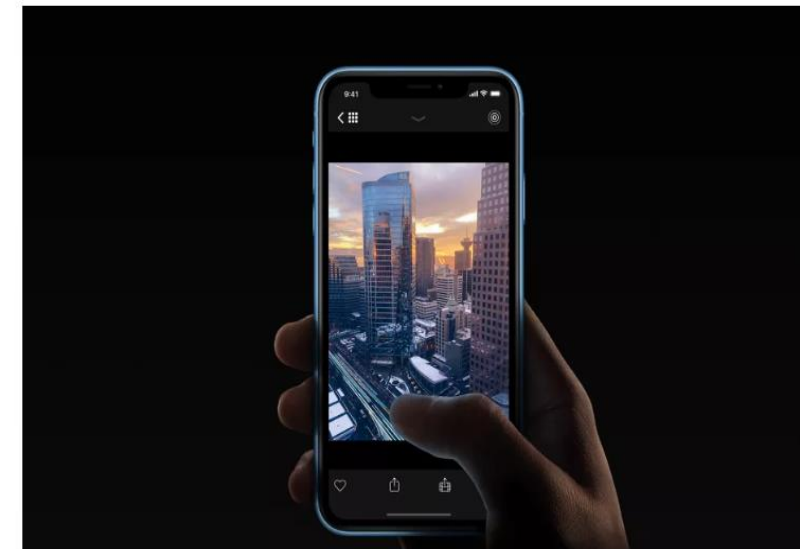
- Defining Page Structure

  - `<footer>` Element

    - Represents a footer for the section of content that is its nearest ancestor

    - The section could be entire document, or it could be a section, article, or aside element

    - Often contain copyright information, related links, author info etc

# Form Validations

## *Required Attribute*

- The simplest type of form validation is the ***required*** attribute

- It's a Boolean attribute that requires the user to enter data before submitting to the server

- E.g.

```
<tr>
    <td>Enter Name</td>
    <td>
        <input type="text" required="required">
    </td>
</tr>
```

Enter Name [                    ]

Please fill out this field.

# Form Validations

*Autocomplete attribute*

- Allows if the field has the property to remember the similar values entered by the user earlier

- Predicts the value & provide suggestions to the user

- Works with the following <input> elements: text, search url, email, range, color etc.

- Syntax

```
<input autocomplete="on|off">
```

# Form Validations

*Autocomplete attribute*

- E.g.

# Form Validations

- INPUT Types

  - `<input type = "email">`

    - Browser validates the email address in a limited fashion

    - In safari, Chrome & Firefox, you need to enter at least *@-.- for the input to be accepted

```
<tr>
    <td>Enter Email</td>
    <td>
        <input type="email" autofocus placeholder="Enter Email">
    </td>
</tr>
```

Enter Email | @gmail.com

! Please enter a part followed by '@'. '@gmail.com' is incomplete.
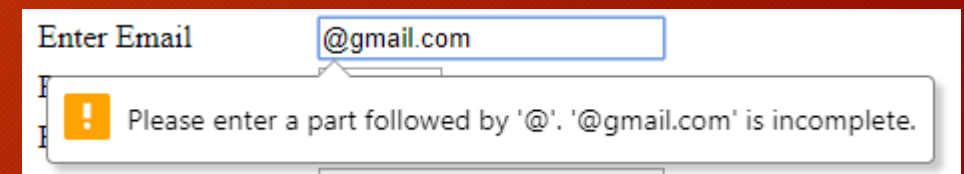
# Form Validations

- INPUT Types

  - `<input type = "number">`

    - It defines a field for entering a number

    - The *min, max* attributes defines the minimum and maximum value allowed

```
<tr>
    <td>Enter Age</td>
    <td>
        <input type="number" min="18" max="50">
    </td>
</tr>
```

Enter Age | 18

Min value is 18

# Form Validations

- INPUT Types

  - `<input type = "url">`

    - Defines a field for entering a valid url

    - Input is automatically validated before the form can be submitted

```html
<!DOCTYPE html>
<html>
<body>

<form action="/action_page.php">
  Add your homepage: <input type="url" name="homepage"><br>
  <input type="submit">
</form>

<p><b>Note:</b> The type="url" is not supported in Internet Explorer 9 and earlier versions.
</p>

</body>
</html>
```

Add your homepage: google.com
Submit
⚠ Please enter a URL.

**Note:** The type="url"          t Explorer 9 and earlier versions.

# Form Validations

- INPUT Types

  - `<input type = "range">`

    - defines a control for entering a number whose exact value is not important (like a slider control)

    - Default range is 0 to 100

    - You can set restrictions on what numbers are accepted with different attributes

      - Min, max, step & value

# Form Validations

- INPUT Types

    - Range E.g.

# Form Validations

- INPUT Types

  - Date & Time

    - E.g.

```
<tr>
    <td>Enter DOB</td>
    <td>
        <input type="date">
    </td>
    <td>
        <input type="time">
    </td>
    <td>
        <input type="datetime-local">
    </td>
</tr>
```

Enter DOB     dd-----yyyy      --:-- --    dd-----yyyy --:-- --

# Form Validations

- INPUT Type

  - <input type = "color">

  - E.g.

# Form Validations

- Regular Expressions

  - A regular expression (regex) is a pattern that can be used to match character combinations in text strings

  - Pattern attribute is commonly used to input regex

  - Visit: regexr.com

  - E.g.

```
<tr>
    <td>Enter Phone Number</td>
    <td>
        <input type="text" title="Only digits are allowed, length will be 10" pattern="[0-9]{10}" placeholder="Enter ph no">
    </td>
</tr>
```

# Multimedia

- Audio Element

  - The audio element is used to embed audio in a web page

  - To play an audio file in HTML, we use the <audio> tag

| Attributes | Description |
|---|---|
| src | Specifies the URL of the media resource |
| controls | Media control features like play/pause will be displayed |
| loop | Causes the media to play in a loop |
| autoplay | Media will play automatically on page load |
| muted | Media will play in muted state |

# Multimedia

- Audio Element

  - Syntax:

    **Specifies location of audio file**

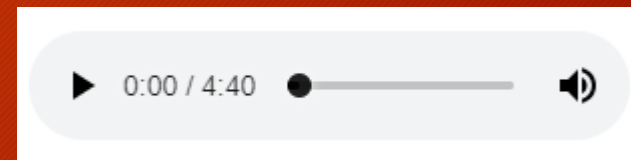    **Specifies whether or not to display player controls**

    <audio src = *"Link of audio file"* controls = "controls">

  - E.g.

```
<audio controls controlslist="nodownload">
    <source src="Flute.mp3"> </source>
</audio>
```

► 0:00 / 4:40 ●———————— 🔊

# Multimedia

- Video Element

  - It specifies a video such as a movie clip or other video streams that is embedded in a web page

| Attributes | Description |
|------------|-------------|
| src | Specifies the URL of the media resource |
| controls | Media control features like play/pause will be displayed |
| loop | Causes the media to play in a loop |
| autoplay | Media will play automatically on page load |
| muted | Media will play in muted state |
| width | Specifies the width of the image in pixels |
| height | Specifies the height of the image in pixels |
| poster | Representative frame for the video till video is played |

# Multimedia

- Video Element

  - Syntax
    
    Specifies the location of a video file
    
    Specifies whether or not to display player controls
    
    <video src = "*Link of video file*" controls = "*controls*>

  - E.g.

```
<video controls width="50%" height="300" poster="Images/PM.png">
    <source src="Blockchain.MKV"></source>
    <track kind="subtitles" label="en" src=""></track>

</video>
```

NPTEL ONLINE CERTIFICATION
https://onlinecourses.nptel.ac.in

0:00 / 31:18

# Thank You