HTML5

What is HTML?

HTML Stands for HyperText Markup Language,

- **HyperText stands for Link between web pages.**
- **Markup Language means Text between tags that define the structure.**

HTML is a markup language that is used to create web pages. It defines how the web page looks and how to display content with the help of elements. It forms or defines the structure of our Web Page; thus, it forms or defines the structure of our Web Page. We must remember to save your file with .html extension.

Applications of HTML

HTML is used for various purposes. Let us take a look at them

1. **Web Pages Development**
   HTML is famously used for creating web pages on the world wide web. Every web page contains a set of HTML tags and hyperlinks which are used to connect other pages. Every page on the internet is written using HTML.
2. **Navigating the Internet**
   Navigating on the internet would have been quite a tedious task without HTML. The anchor tags of HTML allow us to link pages and navigate easily. Imagine our life without anchor tags, you would literally have to enter URL every time. Using anchor tags, you can also navigate within a webpage.
3. **Embedding Images and Videos**
   HTML allows us to embed images and videos with ease and gives us features to adjust height, position and even rendering type. You can adjust controls, thumbnails, timestamps and much more for videos. Earlier this was done using Flash and HTML has made it easier with the help of **<video>** tag.
4. **Clinet-side storage**
   HTML5 has made client-side storage possible using localStorage and IndexD due to which we no longer need to reply on Cookies. Both of these tactics have their own set of rules and characteristics. String-based hash-table storage is provided by localStorage. Its API is straightforward, with setItem, getItem, and removeItem functions available to developers. On the other hand, IndexDB is a larger and more capable client-side data store. With the user's permission, the IndexDB database can be enlarged.
5. **Game development**
   Although you cannot create complex high-end video games with HTML, the **<canvas>** element of HTML can be used to make 2D and 3D games using CSS and JavaScript which can be run on browsers.

6. **Data entry support**
   With the usage of new HTML5 standards in all the latest browsers, developers can simply add the tags for required fields, text, data format, etc. and get the data. HTML5 now has several new attributes for data-entry and validation purposes.
7. **Interacting with Native APIs**
   With the help of HTML, you can interact with your Operating system. With this feature, you can easily drag files onto a web page to upload, full-screen a video, and much more.

Features Of HTML

- The learning curve is very easy (easy to modify)
- Creating effective presentations
- Adding Links wherein we can add references
- Can display documents on platforms like Mac, Windows, Linux, etc
- Adding videos, graphics, and audios making it more attractive
- Case insensitive language

HTML Editor

- Simple editor: Notepad
- Notepad++
- Atom
- VS Code
- Best editor: Sublime Text

HTML Skeleton

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

The <!DOCTYPE> Declaration
The <!DOCTYPE> declaration represents the document type, and helps browsers to display web pages correctly.
It must only appear once, at the top of the page (before any HTML tags).
The <!DOCTYPE> declaration is not case sensitive.
Syntax:

```
<!DOCTYPE html>
```

<html>
Root element which acts as a container to hold all the code
The browser should know that this is an HTML document
Permitted content: One head tag followed by one body tag

<body>

- Everything written here will be displayed in the browser
- Contains text, images, links that can be achieved through tags
- Examples:
- ○ This is our first paragraph.
- ○ Go To Google
- ○ <img src="photo.jpg">

HTML Comment

- Comments don't render on the browser
- Helps to understand our code better and makes it readable.
- Helps to debug our code
- Three ways to comment:

1. Single line
2. Multiple lines
3. Comment tag //Supported by IE

HTML Elements

- Elements are created using tags
- Elements are used to define the semantics
- It can be nested and empty.

New Features inHTML5
Here's a list of HTML5 Features, which are explained in detail below:

- Semantic Elements
- Audio and Video Support
- Canvas Elements
- Geolocation API
- Local Storage
- Responsive Images
- Web Workers
- Drag and Drop API
- Form Enhancements
- Web Sockets
- Micro Data
- Cross Document Messaging

Semantic Elements:

What are Semantic Elements?

A semantic element clearly describes its meaning to both the browser and the developer.
Examples of non-semantic elements: and - Tells nothing about its content.
Examples of semantic elements: <form>, <table>, and <article> - Clearly defines its content.

HTML5 features include native audio and video support without the need for Flash.
The HTML5 <audio> and <video> tags make it simple to add media to a website. You need to set src attribute to identify the media source and include a controls attribute so the user can play and pause the media.

Embedding Video

Here is the simplest form of embedding a video file in your webpage −

```
<video src = "foo.mp4"  width = "300" height = "200" controls>
  Your browser does not support the <video> element.
</video>
```

The current HTML5 draft specification does not specify which video formats browsers should support in the video tag. But most commonly used video formats are −

- 1. Ogg
- 2. mpeg4
- You can use <source> tag to specify media along with media type and many other attributes.
- A video element allows multiple source elements and browser will use the first recognized format −

Example:

```
<!DOCTYPE HTML>

<html>

  <body>

    <video  width = "300" height = "200" controls autoplay>

      <source src = "/html5/foo.ogg" type ="video/ogg" />
```

```
      <source src = "/html5/foo.mp4" type = "video/mp4" />
      </video>

  </body>
</html>
```

3. Canvas Elements:

What is HTML Canvas?

The HTML <canvas> element is used to draw graphics, on the fly, via JavaScript.

The <canvas> element is only a container for graphics. You must use JavaScript to actually draw the graphics.

Canvas has several methods for drawing paths, boxes, circles, text, and adding images.

Canvas Examples

A canvas is a rectangular area on an HTML page. By default, a canvas has no border and no content.

The markup looks like this:

```
<canvas id="myCanvas" width="200" height="100"></canvas>
```

Note: Always specify an id attribute (to be referred to in a script), and a width and height attribute to define the size of the canvas. To add a border, use the style attribute.

Here is an example of a basic, empty canvas:

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

<canvas id="myCanvas" width="200" height="100" style="border:1px solid #000000;">
```

Hi, This is a Canvas tag.

</canvas>

</body>

</html>

The following tags (elements) have been introduced in HTML5 –

Tags (Elements)	Description
<article>	Represents an independent piece of content of a document, such as a blog entry or newspaper article
<aside >	Represents a piece of content that is only slightly related to the rest of the page.
<audio>	Defines an audio file.
<canvas>	This is used for rendering dynamic bitmap graphics on the fly, such as graphs or games.
<command>	Represents a command the user can invoke.
<datalist>	Together with the a new list attribute for input can be used to make comboboxes

<details>	**Represents additional information or controls which the user can obtain on demand**
<embed>	**Defines external interactive content or plugin.**
<figure>	**Represents a piece of self-contained flow content, typically referenced as a single unit from the main flow of the document.**
<footer>	**Represents a footer for a section and can contain information about the author, copyright information, et cetera.**
<header>	**Represents a group of introductory or navigational aids.**
<hgroup>	**Represents the header of a section.**
<keygen>	**Represents control for key pair generation.**
<mark>	**Represents a run of text in one document marked or highlighted for reference purposes, due to its relevance in another context.**
<meter>	**Represents a measurement, such as disk usage.**
<nav>	**Represents a section of the document intended for navigation.**
<output>	**Represents some type of output, such as from a calculation done through scripting.**

<progress>	**Represents a completion of a task, such as downloading or when performing a series of expensive operations.**
<ruby>	**Together with <rt> and <rp> allow for marking up ruby annotations.**
<section>	**Represents a generic document or application section**
<time>	**Represents a date and/or time.**
<video>	**Defines a video file.**
<wbr>	**Represents a line break opportunity.**

New Attributes in HTML 5:

Some attributes are defined globally and can be used on any element, while others are defined for specific elements only. All attributes have a name and a value and look like as shown below in the example.
Following is the example of an HTML5 attributes which illustrates how to mark up a div element with an attribute named class using a value of "example" –
...

Attributes may only be specified within start tags and must never be used in end tags.
HTML5 attributes are case insensitive and may be written in all uppercase or mixed case, although the most common convention is to stick with lowercase.

Standard Attributes

The attributes listed below are supported by almost all the HTML 5 tags.

Attribute	Options	Function

accesskey	User Defined	Specifies a keyboard shortcut to access an element.
align	right, left, center	Horizontally aligned tags
background	URL	Places an background image behind an element
bgcolor	numeric, hexidecimal, RGB values	Places a background color behind an element
class	User Defined	Classifies an element for use with Cascading Style Sheets.
contenteditable	true, false	Specifies if the user can edit the element's content or not.
contextmenu	Menu id	Specifies the context menu for an element.
data-XXXX	User Defined	Custom attributes. Authors of a HTML document can define their own attributes. Must start with "data-".
draggable	true,false, auto	Specifies whether or not a user is allowed to drag an element.
height	Numeric Value	Specifies the height of tables, images, or table cells.
hidden	hidden	Specifies whether element should be visible or not.

id	User Defined	Names an element for use with Cascading Style Sheets.
item	List of elements	Used to group elements.
itemprop	List of items	Used to group items.
spellcheck	true, false	Specifies if the element must have it's spelling or grammar checked.
style	CSS Style sheet	Specifies an inline style for an element.
subject	User define id	Specifies the element's corresponding item.
tabindex	Tab number	Specifies the tab order of an element.
title	User Defined	"Pop-up" title for your elements.
valign	top, middle, bottom	Vertically aligns tags within an HTML element.
width	Numeric Value	Specifies the width of tables, images, or table cells.

For a complete list of HTML5 Tags and related attributes, please check our reference to HTML5 Tags.

Custom Attributes

A new feature being introduced in HTML 5 is the addition of custom data attributes.
A custom data attribute starts with data- and would be named based on your requirement.
Here is a simple example –

```
<div class = "example" data-subject = "physics" data-level = "complex">
   ...
</div>
```

The above code will be perfectly valid HTML5 with two custom attributes called *datasubject* and *data-level*. You would be able to get the values of these attributes using JavaScript APIs or CSS in a similar way as you get for standard attributes.

Microdata:

Microdata is a standardized way to provide additional semantics in your web pages.

Microdata lets you define your own customized elements and start embedding custom properties in your web pages. At a high level, microdata consists of a group of name-value pairs.

The groups are called items, and each name-value pair is a property. Items and properties are represented by regular elements.

Example:

- **To create an item, the itemscope attribute is used.**
- **To add a property to an item, the itemprop attribute is used on one of the item's descendants.**

```
<html>
 <body>
   <div itemscope>
     <p>My name is <span itemprop = "name">Zara</span>.</p>
   </div>
    <div itemscope>
     <p>My name is <span itemprop = "name">Nuha</span>.</p>
   </div>
  </body>
</html>
```

HTML Forms & Controls

HTML Forms are required, when you want to collect some data from the site visitor. For example, during user registration you would like to collect information such as name, email address, credit card, etc.

A form will take input from the site visitor and then will post it to a back-end application such as CGI, ASP Script or PHP script etc. The back-end application will perform required processing on the passed data based on defined business logic inside the application.

There are various form elements available like text fields, textarea fields, drop-down menus, radio buttons, checkboxes, etc.

The HTML <form> tag is used to create an HTML form and it has following syntax −

```
<form action = "Script URL" method = "GET|POST">

   form elements like input, textarea etc.

</form>
```

Form Attributes

Apart from common attributes, following is a list of the most frequently used form attributes −

Sr.No	Attribute & Description
1	action Backend script ready to process your passed data.
2	method Method to be used to upload data. The most frequently used are GET and POST methods.
3	target Specify the target window or frame where the result of the script will be displayed. It takes values like _blank, _self, _parent etc.

4	enctype You can use the enctype attribute to specify how the browser encodes the data before it sends it to the server. Possible values are − application/x-www-form-urlencoded − This is the standard method most forms use in simple scenarios. mutlipart/form-data − This is used when you want to upload binary data in the form of files like image, word file etc.

HTML Form Controls

There are different types of form controls that you can use to collect data using HTML form −

- Text Input Controls
- Checkboxes Controls
- Radio Box Controls
- Select Box Controls
- File Select boxes
- Hidden Controls
- Clickable Buttons
- Submit and Reset Button

Text Input Controls

There are three types of text input used on forms −

- Single-line text input controls − This control is used for items that require only one line of user input, such as search boxes or names. They are created using HTML <input> tag.
- Password input controls − This is also a single-line text input but it masks the character as soon as a user enters it. They are also created using HTMl <input> tag.
- Multi-line text input controls − This is used when the user is required to give details that may be longer than a single sentence. Multi-line input controls are created using HTML <textarea> tag.

Single-line text input controls

This control is used for items that require only one line of user input, such as search boxes or names. They are created using HTML <input> tag.

Example

Here is a basic example of a single-line text input used to take first name and last name −

```
<!DOCTYPE html>

<html>

  <head>

    <title>Text Input Control</title>

  </head>

  <body>

    <form >

      First name: <input type = "text" name = "first_name" /><br>

    Last name: <input type = "text" name = "last_name" />

    </form>

  </body>

</html>

<!DOCTYPE html>

<html>

  <head>

    <title>Password Input Control</title>

  </head>

  <body>

    <form >

      User ID : <input type = "text" name = "user_id" />

      <br>

      Password: <input type = "password" name = "password" />

    </form>
```

```
  </body>

</html>
```

Attributes

Following is the list of attributes for <input> tag for creating password field.

Sr.No	Attribute & Description
1	type Indicates the type of input control and for password input control it will be set to password.
2	name Used to give a name to the control which is sent to the server to be recognized and get the value.
3	value This can be used to provide an initial value inside the control.
4	size Allows to specify the width of the text-input control in terms of characters.
5	maxlength Allows to specify the maximum number of characters a user can enter into the text box.

Multiple-Line Text Input Controls

This is used when the user is required to give details that may be longer than a single sentence.
Multi-line input controls are created using HTML <textarea> tag.
Example
Here is a basic example of a multi-line text input used to take item description −

```
<!DOCTYPE html>

<html>

<head>

<title>Multiple-Line Input Control</title>

  </head>

    <body>

     <form>

       Description: <br />

       <textarea rows = "5" cols = "50" name = "description">

         Enter description here...

       </textarea>

 </form>

  </body>

</html>
```

This will produce the following result −

Attributes

Following is the list of attributes for <textarea> tag.

Sr.No	Attribute & Description

1	name Used to give a name to the control which is sent to the server to be recognized and get the value.
2	rows Indicates the number of rows of text area box.
3	cols Indicates the number of columns of text area box

Checkbox Control: -Checkboxes are used when more than one option is required to be selected. They are also created using HTML <input> tag but type attribute is set to checkbox.

Example

Here is an example HTML code for a form with two checkboxes.

```
<!DOCTYPE html>

<html>

  <head>

    <title>Checkbox Control</title>

  </head>

  <body>

    <form>

      <input type = "checkbox" name = "maths" value = "on"> Maths

      <input type = "checkbox" name = "physics" value = "on"> Physics

    </form>

  </body>
```

</html>

This will produce the following result −

Attributes

Following is the list of attributes for <checkbox> tag.

Sr.No	Attribute & Description
1	type Indicates the type of input control and for checkbox input control it will be set to checkbox..
2	name Used to give a name to the control which is sent to the server to be recognized and get the value.
3	value The value that will be used if the checkbox is selected.
4	checked Set to *checked* if you want to select it by default.

Radio Button Control

Radio buttons are used when out of many options, just one option is required to be selected. They are also created using HTML <input> tag but type attribute is set to radio.
Example
Here is example HTML code for a form with two radio buttons −

```
<!DOCTYPE html>

<html>

  <head>
```

```
    <title>Radio Box Control</title>
  </head>
  <body>
    <form>
      <input type = "radio" name = "subject" value = "maths"> Maths
      <input type = "radio" name = "subject" value = "physics"> Physics
    </form>
  </body>
</html>
```

This will produce the following result −

Attributes

Following is the list of attributes for radio button.

Sr.No	Attribute & Description
1	type Indicates the type of input control and for checkbox input control it will be set to radio.
2	name Used to give a name to the control which is sent to the server to be recognized and get the value.
3	value The value that will be used if the radio box is selected.

4	checked Set to *checked* if you want to select it by default.

Select Box Control

A select box, also called drop down box which provides option to list down various options in the form of drop down list, from where a user can select one or more options.

Example

Here is example HTML code for a form with one drop down box

```
<!DOCTYPE html>
<html>

  <head>

    <title>Select Box Control</title>

  </head>

  <body>

    <form>

      <select name = "dropdown">

        <option value = "Maths" selected>Maths</option>

        <option value = "Physics">Physics</option>

      </select>

    </form>

  </body>

</html>
```

This will produce the following result −

Attributes

Following is the list of important attributes of <select> tag −

Sr.No	Attribute & Description

1	name Used to give a name to the control which is sent to the server to be recognized and get the value.
2	size This can be used to present a scrolling list box.
3	multiple If set to "multiple" then allows a user to select multiple items from the menu.

Following is the list of important attributes of <option> tag −

Sr.No	Attribute & Description
1	value The value that will be used if an option in the select box box is selected.
2	selected Specifies that this option should be the initially selected value when the page loads.
3	label An alternative way of labeling options

File Upload Box

If you want to allow a user to upload a file to your web site, you will need to use a file upload box, also known as a file select box. This is also created using the <input> element but type attribute is set to file.

Example

Here is example HTML code for a form with one file upload box −

```
<!DOCTYPE html>

<html>

  <head>

    <title>File Upload Box</title>

  </head>

  <body>

    <form>

      <input type = "file" name = "fileupload" accept = "image/*" />

    </form>

  </body>

</html>
```

This will produce the following result −

Attributes

Following is the list of important attributes of file upload box −

Sr.No	Attribute & Description
1	name Used to give a name to the control which is sent to the server to be recognized and get the value.
2	accept Specifies the types of files that the server accepts.

Button Controls

There are various ways in HTML to create clickable buttons. You can also create a clickable button using <input>tag by setting its type attribute to button. The type attribute can take the following values −

Sr.No	Type & Description
1	submit This creates a button that automatically submits a form.
2	reset This creates a button that automatically resets form controls to their initial values.
3	button This creates a button that is used to trigger a client-side script when the user clicks that button.
4	image This creates a clickable button but we can use an image as background of the button.

Example

Here is example HTML code for a form with three types of buttons −

```
<!DOCTYPE html>

<html>

  <head>

    <title>File Upload Box</title>

  </head>

  <body>

    <form>

      <input type = "submit" name = "submit" value = "Submit" />

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

```
      <input type = "button" name = "ok" value = "OK" />

      <input type = "image" name = "imagebutton" src = "/html/images/logo.png" />

    </form>

  </body>

</html>
```

This will produce the following result −

Hidden Form Controls

Hidden form controls are used to hide data inside the page which later on can be pushed to the server. This control hides inside the code and does not appear on the actual page. For example, following hidden form is being used to keep current page number. When a user will click next page then the value of hidden control will be sent to the web server and there it will decide which page will be displayed next based on the current page.

Example

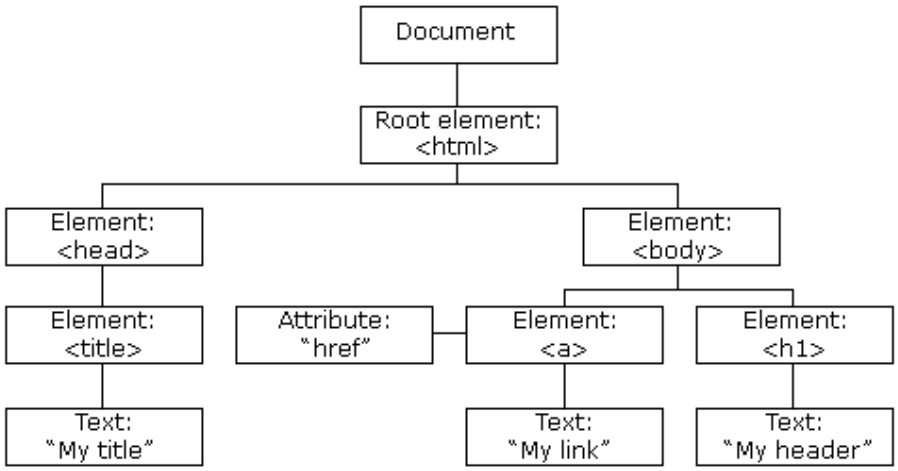
Here is example HTML code to show the usage of hidden control −

```
<!DOCTYPE html>

<html>

  <head>  <title>File Upload Box</title>

</head>

  <body>

    <form>

      <p>This is page 10</p>

      <input type = "hidden" name = "pagename" value = "10" />

      <input type = "submit" name = "submit" value = "Submit" />

      <input type = "reset" name = "reset"  value = "Reset" />

 </form>

  </body>
```

The HTML DOM (Document Object Model)

When a web page is loaded, the browser creates a Document Object Model of the page.

The HTML DOM model is constructed as



a tree of Objects: