

# 1. Working with HTML5 and CSS

---

## HTML

HTML stands for Hyper Text Markup Language, which is the most widely used language on Web to develop web pages. HTML was created by Berners-Lee in late 1991 but "HTML 2.0" was the first standard HTML specification which was published in 1995. HTML 4.01 was a major version of HTML and it was published in late 1999. Though HTML 4.01 version is widely used but currently we are having HTML-5 version which is an extension to HTML 4.01, and this version was published in 2012.

## Why to Learn HTML?

HTML is being widely used to format web pages with the help of different tags available in HTML language.

HTML is a MUST for students and working professionals to become a great Software Engineer especially when they are working in Web Development Domain.

Some of the key advantages of learning HTML:

- Create Web site - You can create a website or customize an existing web template if you know HTML well.
- Become a web designer - If you want to start a career as a professional web designer, HTML and CSS designing is a must skill.
- Understand web - If you want to optimize your website, to boost its speed and performance, it is good to know HTML to yield best results.
- Learn other languages - Once you understand the basic of HTML then other related technologies like javascript, php, or angular are become easier to understand.

## Hello World using HTML.

```
<!DOCTYPE html>
<html>
  <head>
    <title>This is document title</title>
  </head>
  <body>
    <h1>This is a heading</h1>
    <p>Hello World!</p>
  </body>
</html>
```

## Applications of HTML

As mentioned before, HTML is one of the most widely used languages over the web. I'm going to list few of them here:

- Web pages development - HTML is used to create pages which are rendered over the web. Almost every page of web is having html tags in it to render its details in browser.
- Internet Navigation - HTML provides tags which are used to navigate from one page to another and is heavily used in internet navigation.
- Responsive UI - HTML pages now-a-days works well on all platforms, mobile, tabs, desktop or laptops owing to responsive design strategy.
- Offline support HTML pages once loaded can be made available offline on the machine without any need of internet.
- Game development- HTML5 has native support for rich experience and is now useful in gaming development arena as well.

## BASIC TAGS

### Heading Tags

Any document starts with a heading. You can use different sizes for your headings. HTML also has six levels of headings, which use the elements `<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>`, and `<h6>`. While displaying any heading, browser adds one line before and one line after that heading.

#### Example:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Heading Example</title>
  </head>
  <body>
    <h1>This is heading 1</h1>
    <h2>This is heading 2</h2>
    <h3>This is heading 3</h3>
    <h4>This is heading 4</h4>
    <h5>This is heading 5</h5>
    <h6>This is heading 6</h6>
  </body>
</html>
```

This will produce the following result –

**This is heading 1**

**This is heading 2**

**This is heading 3**

**This is heading 4**

**This is heading 5**

**This is heading 6**

## Paragraph Tag

The `<p>` tag offers a way to structure your text into different paragraphs. Each paragraph of text should go in between an opening `<p>` and a closing `</p>` tag as shown below in the example –

### Example

```
<!DOCTYPE html>
<html>
  <head>
    <title>Paragraph Example</title>
  </head>
  <body>
    <p>Here is a first paragraph of text.</p>
    <p>Here is a second paragraph of text.</p>
    <p>Here is a third paragraph of text.</p>
  </body>
</html>
```

This will produce the following result –

Here is a first paragraph of text.

Here is a second paragraph of text.

Here is a third paragraph of text.

## Line Break Tag

Whenever you use the **<br />** element, anything following it starts from the next line. This tag is an example of an **empty** element, where you do not need opening and closing tags, as there is nothing to go in between them.

The **<br />** tag has a space between the characters **br** and the forward slash. If you omit this space, older browsers will have trouble rendering the line break, while if you miss the forward slash character and just use **<br>** it is not valid in XHTML.

### Example

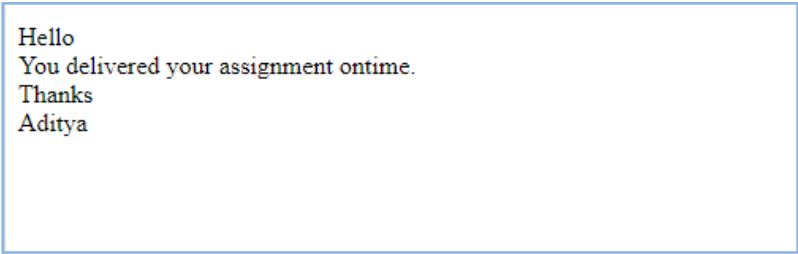
```
<!DOCTYPE html>
<html>

  <head>
    <title>Line Break Example</title>
  </head>

  <body>
    <p>Hello<br />
      You delivered your assignment ontime.<br />
      Thanks<br />
      Aditya</p>
  </body>

</html>
```

This will produce the following result –



```
Hello
You delivered your assignment ontime.
Thanks
Aditya
```

## Centering Content

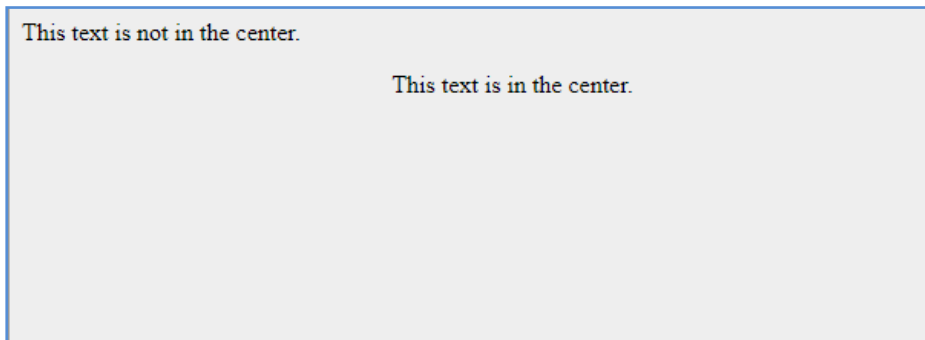
You can use **<center>** tag to put any content in the center of the page or any table cell.

### Example

```
<!DOCTYPE html>
<html>
  <head>
    <title>Centring Content Example</title>
  </head>
  <body>
    <p>This text is not in the center.</p>
    <center>
      <p>This text is in the center.</p>
    </center>
  </body>
</html>
```

```
</body>
</html>
```

This will produce following result –



## Horizontal Lines

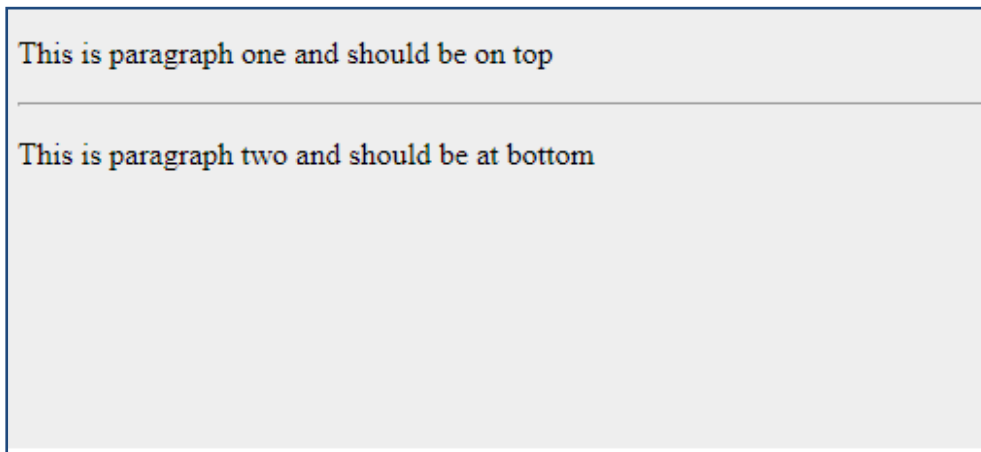
Horizontal lines are used to visually break-up sections of a document. The **<hr>** tag creates a line from the current position in the document to the right margin and breaks the line accordingly.

For example, you may want to give a line between two paragraphs as in the given example below –

### Example

```
<!DOCTYPE html>
<html>
  <head>
    <title>Horizontal Line Example</title>
  </head>
  <body>
    <p>This is paragraph one and should be on top</p>
    <hr />
    <p>This is paragraph two and should be at bottom</p>
  </body>
</html>
```

This will produce the following result –



Again `<hr />` tag is an example of the **empty** element, where you do not need opening and closing tags, as there is nothing to go in between them.

The `<hr />` element has a space between the characters **hr** and the forward slash. If you omit this space, older browsers will have trouble rendering the horizontal line, while if you miss the forward slash character and just use `<hr>` it is not valid in XHTML

## Preserve Formatting

Sometimes, you want your text to follow the exact format of how it is written in the HTML document. In these cases, you can use the preformatted tag `<pre>`.

Any text between the opening `<pre>` tag and the closing `</pre>` tag will preserve the formatting of the source document.

### Example

```
<!DOCTYPE html>
<html>
  <head>
    <title>Preserve Formatting Example</title>
  </head>
  <body>
    <pre>
      function testFunction( strText ){
        alert (strText)
      }
    </pre>
  </body>
</html>
```

This will produce the following result –

```
function testFunction( strText ){
  alert (strText)
}
```

Try using the same code without keeping it inside `<pre>...</pre>` tags

```
function testFunction( strText ){ alert (strText) }
```

## Nonbreaking Spaces

Suppose you want to use the phrase "12 Angry Men." Here, you would not want a browser to split the "12, Angry" and "Men" across two lines –

An example of this technique appears in the movie "12 Angry Men."

In cases, where you do not want the client browser to break text, you should use a nonbreaking space entity **&nbsp;**; instead of a normal space. For example, when coding the "12 Angry Men" in a paragraph, you should use something similar to the following code –

### Example

```
<!DOCTYPE html>
<html>

  <head>
    <title> Nonbreaking Spaces Example</title>
  </head>

  <body>
    <p>An example of this technique appears in the movie
"12&nbsp;Angry&nbsp;Men."</p>
  </body>

</html>
```

This will produce the following result –

## 1.1 Concepts of CSS

### 1.1.1 Adding CSS

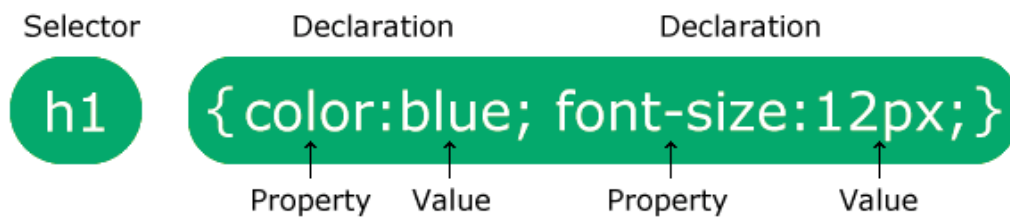
CSS is an acronym for Cascading Style Sheets. It is a presentation definition language that is used to control the layout of HTML document by attaching styles.

It determines how the elements in our HTML documents are displayed and formatted. It is designed to separate the content of a web page from the presentation of that content.

It enables us to make all pages of our website look similar and consistent (font, color, etc.). It allows us to make site-wide formatting changes from a single location (rather than having to edit each page individually).

## CSS Syntax:

A CSS rule consists of a selector and a declaration block.



The selector is usually the HTML element you want to style.

-> Each statement consists of a property and a value.

-> The properties the style attribute you want to change. Each property has a value.

Multiple CSS declarations are separated with semicolons, and declaration blocks are surrounded by curly braces.

**Example:**

```
p {  
  color: red;  
  text-align: center;  
}
```

### Example Explained

- **p** is a selector in CSS (it points to the HTML element you want to style: <p>).
- **color** is a property, and **red** is the property value
- **text-align** is a property, and **center** is the property value

## Applying CSS Code:

There are three ways you can apply CSS .

1. In-Line CSS
2. Internal/ Embedded CSS
3. External CSS

### 1. In-Line CSS:

- The Inline style is specific to the tag itself.
- The inline style uses the HTML "style" attribute to style a specific tag.
- This is not recommended, as every CSS change has to be made in every tag that has the inline style applied to it.



- The Inline style is good for one an individual CSS change that you do not use repeatedly through the site.
- An inline style sheet has the highest priority. It will override styles declared in an internal style sheet, an external style sheet, and a web browsers default styles.

**Example:**

```
<html>
  <head>
    <title>Inline CSS Example</title>
  </head>
<body style="background-color: pink">
  <p>This is a Home Page. </p>
  <p style="color:red; font-size:18px">This is a paragraph!</p>
</body>
</html>
```

**2. Internal/ Embedded CSS:**

- An internal style sheet holds the CSS code for the webpage in the head section of the particular file via the style tag.
 

```
<style type="text/css"></style>
```
- Internal Styles are also called “Embedded” Styles.
- This makes it easy to apply styles like classes or id's in order to reuse the code.
- The drawback of using an internal style sheet is that changes to the internal style sheet only affect the page the code is inserted into.
- An internal style sheet has the second highest priority. It will override styles declared in an external style sheet and a web browsers default styles.

**Example:**

```
<html>
  <head>
    <title>Internal CSS Example</title>
    <style type="text/css">
      body
      {
        background-color: pink;
      }
      p
      {
        color: green
      }
    </style>
  </head>
  <body>
    <p>This is a Welcome page</p>
  </body>
</html>
```

### 3. External CSS:

- The External Stylesheet is a .css file that you link your website to.
- This makes it so that whatever you change in the .css sheet, will affect every page in your website.
- This prevents you from having to make many code changes in each page.
- This is for "global" site changes.

A connection of this type can be created with one line of HTML code :

```
<link rel="stylesheet" type="text/css" href="style.css" />
```

#### Name of file: style.css

```
body
{
    background-color: red;
}
```

#### Example:

```
<html>
  <head>
    <title>External Css Example</title>
    <link rel="stylesheet" type="text/css" href="style.css" />
  </head>
  <body>
    This page is going to implement external CSS example.
  </body>
</html>
```

### 1.1.2 HTML Links and attribute. (\_self, \_blank, \_parent, \_top)

#### <a> tag:

The <a> tag defines a hyperlink, which is used to link from one page to another.

#### HTML Links - Syntax

```
<a href="url">Link text</a>
```

The most important attribute of the <a> element is the href attribute, which indicates the link's destination.

The *link text* is the part that will be visible to the reader.

Clicking on the link text, will send the reader to the specified URL address.

#### Example:

```
<a href="https://www.vnsgu.ac.in">Visit VNsgu Site!</a>
```

By default, links will appear as follows in all browsers:

- An unvisited link is underlined and blue
- A visited link is underlined and purple
- An active link is underlined and red

## The target Attribute

We have used **target** attribute in our previous example. This attribute is used to specify the location where linked document is opened.

Following are the possible options –

1\_blank: Opens the linked document in a new window or tab.

2\_self: Opens the linked document in the same frame.

3\_parent: Opens the linked document in the parent frame.

4\_top: Opens the linked document in the full body of the window.

5targetframe: Opens the linked document in a named target frame.

### Absolute URL in href:

points to another web site (like href="http://www.example.com/default.htm")

### Relative URL in href:

points to a file within a web site (like href="default.htm")

### <img> tag:

The <img> tag is used to embed an image in an HTML page. Images are not technically inserted into a web page; images are linked to web pages. The <img> tag creates a holding space for the referenced image.

### Attributes:

- src - Specifies the path to the image.
- alt - Specifies an alternate text for the image, if the image for some reason cannot be displayed.
- style- Specifies the inline css to apply on image.
- width - Specifies the width of an image.
- Height - Specifies the height of an image.

### Example:

```

```

## 1.2 HTML Forms

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	<b>action</b> Backend script ready to process your passed data.
2	<b>method</b> Method to be used to upload data. The most frequently used are GET and POST methods.
3	<b>target</b> Specify the target window or frame where the result of the script will be displayed. It takes values like <code>_blank</code> , <code>_self</code> , <code>_parent</code> etc.
4	<b>enctype</b> You can use the enctype attribute to specify how the browser encodes the data before it sends it to the server. Possible values are – <b>application/x-www-form-urlencoded</b> – This is the standard method most forms use in simple scenarios. <b>multipart/form-data</b> – 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>
```

## Attributes

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

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

## 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 but type attribute is set to **password**.

### Example

Here is a basic example of a single-line password input used to take user password –

```
<!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	<b>type</b> Indicates the type of input control and for password input control it will be set to <b>password</b> .
2	<b>name</b> Used to give a name to the control which is sent to the server to be recognized and get the value.
3	<b>value</b> This can be used to provide an initial value inside the control.
4	<b>size</b> Allows to specify the width of the text-input control in terms of characters.
5	<b>maxlength</b> 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>
```

### Attributes

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

Sr.No	Attribute & Description
1	<b>name</b> Used to give a name to the control which is sent to the server to be recognized and get the value.
2	<b>rows</b> Indicates the number of rows of text area box.
3	<b>cols</b> 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>
```

## Attributes

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

Sr.No	Attribute & Description
1	<b>type</b> Indicates the type of input control and for checkbox input control it will be set to <b>checkbox</b> .
2	<b>name</b> Used to give a name to the control which is sent to the server to be recognized and get the value.
3	<b>value</b> The value that will be used if the checkbox is selected.
4	<b>checked</b> Set to <i>checked</i> 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>
```

### Attributes

Following is the list of attributes for radio button.

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

### Attributes

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

Sr.No	Attribute & Description
1	<b>name</b> Used to give a name to the control which is sent to the server to be recognized and get the value.
2	<b>size</b> This can be used to present a scrolling list box.
3	<b>multiple</b> 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	<b>value</b> The value that will be used if an option in the select box box is selected.

2	<b>selected</b> Specifies that this option should be the initially selected value when the page loads.
3	<b>label</b> 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>
```

## 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	<b>submit</b> This creates a button that automatically submits a form.
2	<b>reset</b> This creates a button that automatically resets form controls to their initial values.
3	<b>button</b> This creates a button that is used to trigger a client-side script when the user clicks that button.
4	<b>image</b> This creates a clickable button but we can use an image as background of the button.

### Example

```
<!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>
```

### 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 passed current page.

### Example

```
<!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>
</html>
```

## HTML Media:

### Video:

The HTML <video> element is used to show a video on a web page.

#### Example:

```
<video width="320" height="240" controls>  
  <source src="movie.mp4" type="video/mp4">  
  <source src="movie.ogg" type="video/ogg">  
Your browser does not support the video tag.  
</video>
```

#### How it Works

- The controls attribute adds video controls, like play, pause, and volume.
- It is a good idea to always include width and height attributes. If height and width are not set, the page might flicker while the video loads.
- The <source> element allows you to specify alternative video files which the browser may choose from. The browser will use the first recognized format.
- The text between the <video> and </video> tags will only be displayed in browsers that do not support the <video> element.

#### Autoplay attribute:

To start a video automatically, use the autoplay attribute:

#### Example:

```
<video width="320" height="240" autoplay muted>  
  <source src="movie.mp4" type="video/mp4">  
  <source src="movie.ogg" type="video/ogg">  
Your browser does not support the video tag.  
</video>
```

#### Muted attribute:

Add muted after autoplay to let your video start playing automatically (but muted):

#### Example:

```
<video width="320" height="240" autoplay muted>  
  <source src="movie.mp4" type="video/mp4">  
  <source src="movie.ogg" type="video/ogg">  
Your browser does not support the video tag.  
</video>
```

## Audio:

To play an audio file in HTML, use the <audio> element:

### Example:

```
<audio controls>  
<source src="horse.ogg" type="audio/ogg">  
<source src="horse.mp3" type="audio/mpeg">  
Your browser does not support the audio element.  
</audio>
```

### How it works:

- The controls attribute adds audio controls, like play, pause, and volume.
- The <source> element allows you to specify alternative audio files which the browser may choose from. The browser will use the first recognized format.
- The text between the <audio> and </audio> tags will only be displayed in browsers that do not support the <audio> element.

Autoplay and muted attribute of audio tag worked same as in video tag.

```
<audio controls autoplay muted>  
<source src="horse.ogg" type="audio/ogg">  
<source src="horse.mp3" type="audio/mpeg">  
Your browser does not support the audio element.  
</audio>
```