


# HTML-5

## Notes By

# Naveen

# HTML Introduction

1. **HTML** is short for **Hypertext Markup Language**.
2. HTML is used to create electronic documents (called web pages) that are displayed on the web browser.
3. Each page contains a series of connections to other pages called hyperlinks.
4. HTML code ensures the proper formatting of text and images for your Internet browser.
5. Without HTML, a browser would not know how to display text as elements or load images or other elements.
6. To write the program in HTML we use Elements. we need to write element in angular brackets i.e, <, >.
   **Example :** <html>,<body>,<h1>,...
7. In HTML all elements are pre-defined by W3c. for reference look into the URL : https://www.w3.org/
   **Example :**
   <html>,<body>,<h1>,... are valid
   <naveen><sathya><python>,.... are invalid
8. Every opened element must be closed. To close the element we use '/' slash in closing element.
   **Example :**
   <html> --- Opened Element.
   </html> --- Closed Element.

9. Html Elements are not case sensitive it means we can use lower case letters or upper case letters.

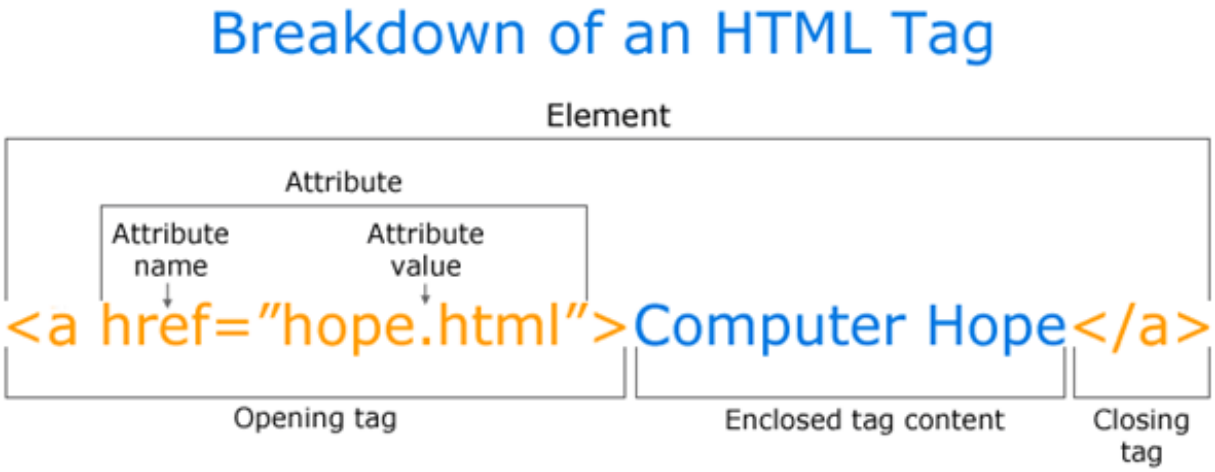
   **Example :**

   <HTML> --- Opened Element in Capital

   </html> --- Closed Element in small, will not give any error

**Note:** It's Recommended to use small letters for open close elements.

## HTML tag look like




## How to create and view HTML

Because HTML is a Markup language, it can be created and viewed in any text editor as long as it's saved with a ***.htm*** **or** ***.html*** file extension.

However, most find it easier to design and create web pages in HTML using an HTML editor.

Once the HTML file is created, it can be viewed locally or uploaded to a **web server** to be viewed online using a **browser.**

## List of HTML editors

- **Amaya** - An open-source HTML editor from the W3C that allows for WYSIWYG editing.
- **Arachnophilia Editor** - Free HTML editor that allows you to do almost everything including frames, drag and drop, CGI, Java, and much more.
- **BBEdit** - Innovative HTML editor for Macintosh users.
- **Bluefish** - An open-source HTML editor, mainly designed for programmers and developers, with support for numerous programming languages.
- **CoffeeCup HTML editor** - Free HTML editor with option for $49 full version.
- **Dreamweaver** - Excellent and widely used HTML editor from the company Adobe.
- **Flux** - Flux is an advanced, non-template based HTML5 web design application.
- **HTML Kit** - An HTML editor that can create web pages, as well as edit other types of text files.

- **Microsoft Expression Web** - An easy and now free to use program that allows you to do everything within a WYSIWYG interface. Expression Web also includes SEO (search engine optimization) tools and the ability to preview in multiple web browsers.
- **RapidWeaver** - Apple Mac all-in-one web design software.
- **WordPad or Notepad** - Included with Windows, these programs or any text editor can create web pages by using HTML and saving the file as a .htm or .html file. While this may be harder initially it is free and does not require any download.

**Note: Click on the blue link to open that web site.**

## Which file extensions are used with HTML?

HTML files use either the **.htm or .html** file extension.

Older versions of Windows (Windows 3.x) only allow three-letter file extensions, so they used .htm instead of .html.

However, both file extensions have the same meaning, and either may be used today.

## Is HTML a programming language?

No. HTML is not a programming language; it's a markup language.

**Markup** means it is a tag used to define the layout or style of how a text is displayed.

## What is a Web Page ?

A **web page** or **webpage** is a document, commonly written in HTML, that is viewed in an Internet browser.

## What is a Website ?

A **site** or **website** is a collection of web pages that are related and accessed by visiting the home page of the website using a browser.

Example : https://www.sathyatech.com/

## What is the difference between static and Dynamic Website ?

A website, or individual web page, can be **static or dynamic**.

A **static website** contains information that does not change. It remains the same, or static, for every viewer of the site.

A **dynamic website** contains information that changes, depending on the viewer, the time of the day, the time zone, the viewer's native language.

## What is Domain ?

When referring to an Internet address or name,
a **domain** or **domain name** is the location of a website.

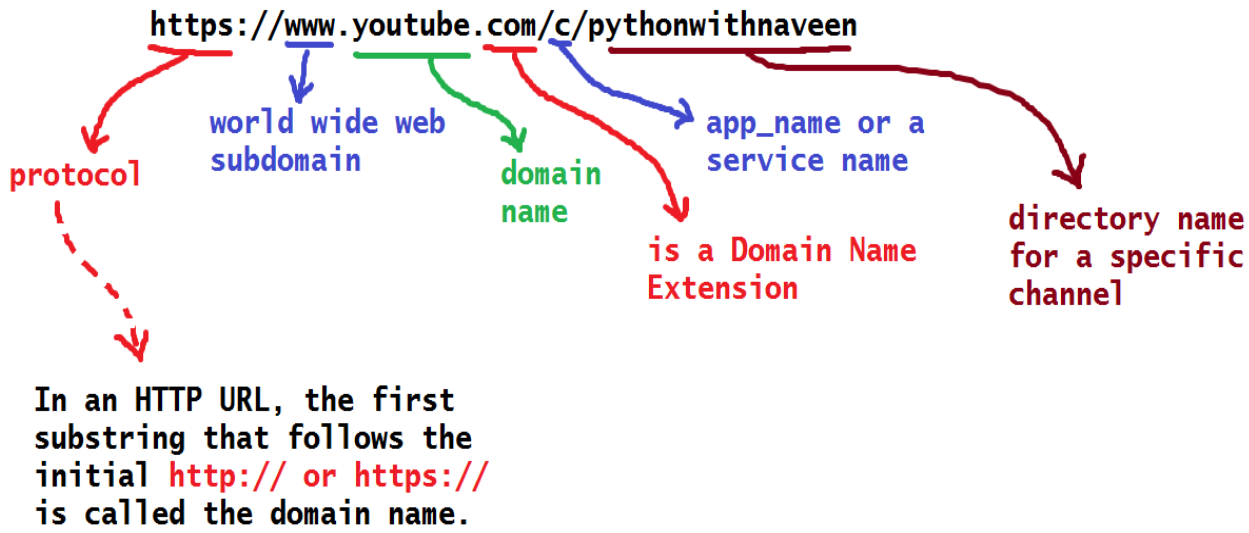
For example, the domain name "google.com" points to the IP address "216.58.216.164".

Generally, it's easier to remember a name rather than a long string of numbers.

A domain name can be a maximum of sixty three characters with one character minimum, and is entered after the protocol in the URL.



## Example :



## http:// or https://

The "http" stands for **"Hypertext Transfer Protocol"**.

An "https" stands for **"Hypertext Transfer Protocol Secure"** and indicates that information transmitted over HTTP is encrypted and secure.

## About URL

Also known as a **web address**, a **URL** (**Uniform Resource Locator**) is a form of URI and a standardized naming convention for addressing a website.

### Example :

1) https://www.sathyatech.com/

2) `https://www.google.com/search?q=python+with+naveen`

In the Second Example

? ---> The question mark in a URL separates the URL from all the parameters or variables.

In the example above, the parameter being sent is *q=python+with+naveen*.

The "q" is a **variable name**, and the " *python+with+naveen*" is the value being sent to that variable.

**Note :** No spaces are allowed in a URL, the space is encoded as %20. In many scripts, a **+** (plus) is also used to represent a space.

## List of Internet browsers

- Google Chrome
- Microsoft Internet Explorer
- Mozilla Firefox
- Opera
- Apple Safari
- Amazon Silk

# HTML Elements

## <!DOCTYPE>

Defines the HTML version used in the document. In this case it is HTML5. Doctypes - short for 'document type'

## <html>

Opens the page. No markup should come after the closing tag (**</html>**). The **lang** attribute declares the primary language of the page (en for English).

## <head>

Opens the head section, which does not appear in the main browser window but mainly contains information *about* the HTML document, called *metadata*. It can also contain imports from external stylesheets and scripts. The closing tag is **</head>**.

## <meta>

Gives the browser some metadata about the document. The charset attribute declares the character encoding. Modern HTML documents should always use UTF-8, even though it is not a requirement. In HTML, the **<meta>** tag does not require a closing tag.

**UTF-8** (8-bit Unicode Transformation Format)

## <title>

The title of the page. Text written between this opening and the closing tag (**</title>**) will be displayed on the title bar of the browser.

## <body>

Opens the part of the document displayed to users, i.e. all the visible or audible content of a page. No content should be added after the closing tag **</body>**.

## Headings

HTML provides not only plain paragraph tags, but six separate header tags to indicate headings of various sizes and thicknesses.

**<h1>**Heading 1**</h1>**

**<h2>**Heading 2**</h2>**

**<h3>**Heading 3**</h3>**

**<h4>**Heading 4**</h4>**

**<h5>**Heading 5**</h5>**

**<h6>**Heading 6**</h6>**

## Paragraphs

The HTML  element defines a paragraph:

This is a paragraph.

This is another paragraph.

## other

**<br>** Inserts a single line break

**<pre>** Defines pre-formatted text

## Highlighting

The **<mark>** element is new in HTML5 and is used to mark or highlight text in a document.

** ** Hello students this is **<mark>**Naveen**</mark>** from Sathya ** **

## Bold, Italic, and Underline

### Bold Text

To bold text, use the **** or **** tags:

****Bold Text Here****

or

****Bold Text Here****

### Italic Text

To italicize text, use the **** or **** tags:

****Italicized Text Here**** (emphasized)

or

****Italicized Text Here****

### Underlined Text

** **Hello students this is **<u>**Naveen**</u>** from Sathya.** **

## Abbreviation

To mark some expression as an abbreviation, use **<abbr>** tag:

** **I like to write **<abbr** title="Hypertext Markup Language"**>** HTML **</abbr> **

## Inserted, Deleted, or Stricken

**<ins>**Python**</ins>**

**<del>**Java**</del>**

**<s>**Dot Net**</s>**

## Superscript and Subscript

To offset text either upward or downward you can use the tags **<sup>** and **<sub>**.

To create superscript: **<sup>**superscript here**</sup>**

To create subscript: **<sub>**subscript here**</sub>**

# Anchors and Hyperlinks

## href

Specifies the destination address. It can be an absolute or relative URL, or the name of an anchor.

Anabsolute URL is the complete URL of a website like https://www.youtube.com/c/pythonwithnaveen

A relative URL points to another directory and/or document inside the same website, e.g. /about-us/ points to the directory "about-us" inside the root directory (/). When pointing to another directory without explicitly specifying the document, web servers typically return the document "index.html" inside that directory.

## hreflang

Specifies the language of the resource linked by the href attribute.

## target

Specifies where to open the link, e.g. in a new tab or window. Possible values are _blank, _self, _parent, _top.

## title

Specifies extra information about a link. The information is most often shown as a tooltip text when the cursor moves over the link.

## download

Specifies that the target will be downloaded when a user clicks on the hyperlink. The value of the attribute will be the name of the downloaded file. There are no restrictions on allowed values, and the browser will automatically detect the correct file extension and add it to the file (.img, .pdf, etc.).

## Link to another site

This is the basic use of the **<a>** (**a**nchor element) element:

**<a** href="https://www.youtube.com/c/pythonwithnaveen/**>** Link to my channel **</a>**

## Link to an anchor

Anchors can be used to jump to specific tags with in an HTML page.

```
<h2 id="Topic1">First topic</h2>
<p>Content about the first topic</p>
<h2 id="Topic2">Second topic</h2>
<p>Content about the second topic</p>

<a href='#Topic1'>Click to jump to the First Topic</a>
<a href='#Topic2'>Click to jump to the Second Topic</a>
```

## Link that dials a number

If the value of the href-attribute begins with tel:, your device will dial the number when you click it.

This works on mobile devices or on computers/tablets running software – like Skype or FaceTime – that can make phone calls.

```
<a href="tel:9052492329">Call us</a>
```

Most devices and programs will prompt the user in some way to confirm the number they are about to dial.

## Link that runs JavaScript

Simply use the javascript: protocol to run the text as JavaScript instead of opening it as a normal link:

```
<a href="javascript:myFunction();">Run Code</a>
```

You can also achieve the same thing using the onclick attribute:

```
<a href="#" onclick="myFunction(); return false;">Run Code</a>
```

### Example

```
<script type="text/javascript">
   function display() {
      alert("This is Naveen")
   }
</script>

<a href="javascript:display();">Run Code</a>
```

## email client

### Basic usage

If the value of the href-attribute begins with mailto: it will try to open an email client on click:

```
<a href="mailto:pythonwithnaveen@gmail.com">Send email</a>
```

### Cc and Bcc

You can also add addresses for cc- or bcc-recipients using the following syntax:

```
<a href="mailto:
pythonwithnaveen@gmail.com?cc=john@example.com&bcc=jane@
example.com">Send email</a>
```

### Subject and body text

You can populate the subject and body for the new email as well:

```
<a href="mailto:
pythonwithnaveen@gmail.com?subject=Example+subject&body=Me
ssage+text">Send email</a>
```

# Lists

HTML offers three ways for specifying lists: **ordered lists, unordered lists, and description lists**.

## Ordered List

An ordered list can be created with the **<ol>** tag and each list item can be created with the **<li>** tag as in the

**<ol>**

**<li>**Python**</li>**

**<li>**Django**</li>**

**<li>**Rest-API**</li>**

**</ol>**

- **<ol** start="3"**>** Will start from 3rd number.
- **<li** value="3"**></li>** Also used to start from 3rd number.

## Unordered List

An unordered list can be created with the **<ul>** tag and each list item can be created with the **<li>.**

**<ul>**

**<li>**Python**</li>**

**<li>**Django**</li>**

**<li>**Rest-API **</li>**

**</ul>**

# Tables

The HTML **<table>** element allows web developer to display tabular data (such as text, images, links, other tables, etc.) in a two dimensional table with rows and columns of cells.

## Simple Table

```
<table>
<tr>
<th>Heading 1/Column 1</th>
<th>Heading 2/Column 2</th>
</tr>
<tr>
<td>Row 1 Data Column 1</td>
<td>Row 1 Data Column 2</td>
</tr>
<tr>
<td>Row 2 Data Column 1</td>
<td>Row 2 Data Column 2</td>
</tr>
</table>
```

**Note :** Table cells can span multiple columns or rows using the colspan and rowspan attributes. These attributes can be applied to <th> and <td> elements.

## Table with thead, tbody, tfoot, and caption

HTML also provides the tables with the **<thead>**, **<tbody>**, **<tfoot>**, and **<caption>** elements.

```
<table>
<caption>Table Title</caption>
<thead>
<tr>
<th>Header content 1</th>
<th>Header content 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Body content 1</td>
<td>Body content 2</td>
</tr>
</tbody>
<tfoot>
```

**<tr>**

**<td>**Footer content 1**</td>**

**<td>**Footer content 2**</td>**

**</tr>**

**</tfoot>**

**</table>**

## Creating comments

HTML comments can be used to leave notes to yourself or other developers about a specific point in code. They can be initiated with <!-- and concluded with -->, like so:

*<!-- I'm an HTML comment! -->*

## Images

### Creating an image

To add an image to a page, use the image tag.

Image tags (img) do not have closing tags. The two main attributes you give to the img tag are src, the image source and alt, which is alternative text describing the image.

**<img** src="images/hello.png" alt="Hello World"**>**

You can also get images from a web URL:

**<img**
src="https://www.facebook.com/photo.php?fbid=10156373983694058&set=gm.2529142424069080&type=3&theater&ifg=1"**>**

# Input Control Elements

## class

Indicates the Class of the input

## id

Indicates the ID of the input

## type

I dentifies the type of input control to display. Acceptable values are hidden, text, tel, url, email, password, date, time, number, range, color, checkbox, radio, file, submit, image, reset, and button.

Defaults to text if not specified, if the value is invalid, or if the browser does not support the type specified.

## name

Indicates the name of the input

## disabled

Boolean value that indicates the input should be disabled. Disabled controls cannot be edited, are not sent on form submission, and cannot receive focus.

## checked

When the value of the type attribute is radio or checkbox, the presence of this Boolean attribute indicates that the control is selected by default otherwise it is ignored.

## multiple

**HTML5** Indicates multiple files or values can be passed (Applies only to file,select and email type inputs )

## placeholder

**HTML5** A hint to the user of what can be entered in the control.

## autocomplete

**HTML5** Indicates whether the value of the control can be automatically completed by the browser.

```
<form action="#" autocomplete="on">
  First name: <input type="text" id="fname" name="fname">
  <input type="submit">
</form>
```

## readonly

Boolean value that indicates the input is not editable. Readonly controls are still sent on form submission, but will not receive focus.

## required

**HTML5** Indicates a value must be present or the element must be checked in order for the form to be submitted

## autofocus

The **<input>** element should get the focus when page loads.

### value

Specifies the value of **<input>** element.

### step

The step attribute specifies the legal number intervals. It works with the following input types: number, range, date, datetime-local, month, time and week.

## Fields

### Text

The most basic input type and the default input if no type is specified. All the characters can be entered into this.

**<input** type="text">

The default width of a text field input is 20 characters.

**<input** type="text" size="20">

### Checkbox and Radio Buttons

**<input** type="checkbox">

**<input** type="radio">

**Example of Grouping checkbox**

**<input** type="radio" name="color" **> Male**

**<input** type="radio" name="color" **> Female**

**<input** type="radio" name="color" **> Others**

## Input Validation

HTML input validation is done automatically by the browser based on special attributes on the input element.

The validation only occurs when attempting to submit the form, so all restricted inputs must be inside a form in order for validation to occur (unless you're using JavaScript).

## validation.

Some newer input types (like email, url, tel, date and many more ) are automatically validated and do not require your own validation constraints.

## Required

Use the required attribute to indicate that a field must be completed in order to pass validation.

**<input** required**>**

## Minimum / Maximum Length

Use the minlength and maxlength attributes to indicate length requirements.

**<input** minlength="3"**>**

**<input** maxlength="15"**>**

**<input** minlength="3" maxlength="15"**>**

## Specifying a range

Use min and max attributes to restrict the range of numbers a user can input into an input of type number or range

Marks: <**input** type="number" size="6" name="marks" min="0" max="100" />

Subject Feedback: <**input** type="range" size="2" name="feedback" min="1" max="5" />

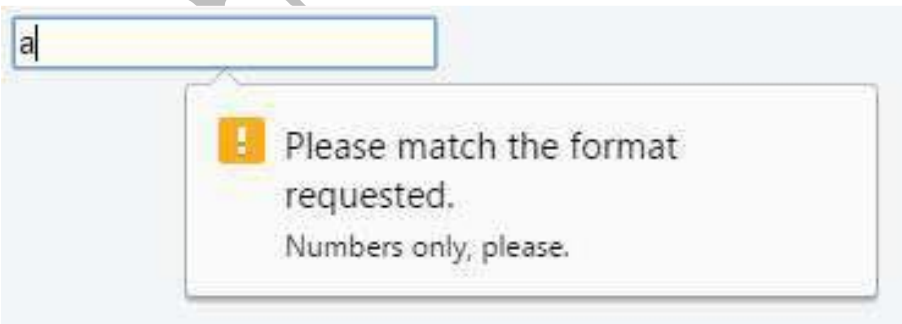
## Match a Pattern

For more control, use the pattern attribute to specify any regular expression that must be matched in order to pass validation. You can also specify a title, which is included in the validation message if the field doesn't pass.

**<input** pattern="\d*" title="Numbers only, please."**>**

Here's the message shown in Google Chrome version 51 when attempting to submit the form with an invalid value

inside this field:




Not all browsers display a message for invalid patterns, although there is full support among most used modern browsers.

## Accept File Type

For input fields of type file, it is possible to accept only certain types of files, such as videos, images, audios, specific file extensions, or certain media types.

**<input** type="file" accept="image/*" title="Only images are allowed"**>**

Multiple values can be specified with a comma

**<input** type="file" accept="image/*,.rar,application/zip"**>**

**Note:** Adding novalidate attribute to the form element or formnovalidate attribute to the submit button, prevents

validation on form elements. For example:

**<form>**

**<input** type="text" name="name" required**>**

**<input** type="submit" value="Publish"**>** *<!-- form will be validated -->*

**<input** type="submit" value="Save" formnovalidate**>** *<!-- form will NOT be validated -->*

**</form>**

## Password

**<input type="password" name="password">**

**<input type="password" name="password" placeholder="Password">**

## File

**<input** type="file" name="fileSubmission"**>**

File inputs allow users to select a file from their local filesystem for use with the current page.

<**form** action="#" method="post" enctype="multipart/form-data">

Select file to upload:

<**input** type="file" name="f1" id="fileSubmission">

<**input** type="submit" value="Upload your file" name="submit">

</**form**>

## Multiple files

Adding the multiple attribute the user will be able to select **more than one** file:

**<input** type="file" name="f2" id="fileSubmission" multiple**>**

## Accept Files

Accept attribute specifies the types of files that user can select. E.g. .png, .gif, .jpeg.

**<input** type="file" name="f3" accept="image/x-png,image /gif,image/jpeg" **/>**

## Button

**<input** type="button" value="Button Text"**>**

without submitting the form

**<input** type="submit" value="Submit">

Which submits the form.

**<input** type="reset" value="Reset">

When clicked, resets all inputs in the form.

## Hidden

**<input** type="hidden" name="inputName" value="inputValue">

A hidden input won't be visible to the user, but its value will be sent to the server when the form is submitted.

## Tel

**<input** type="tel" value="+8400000000">

A telephone number.

## Email

**<form>**

**<label>**E-mail: **<label>**

**<input** type="email" name="email">

**</form>**

E-mail address can be automatically validated when submitted depending on browser support.

## Number

**<input** type="number" value="0" name="quantity">

## Range

<input type="range" min="" max="" step="" />

A control for entering a number whose exact value is not important.

| Attribute | Description | Default value |
|---|---|---|
| min | Minimum value for range | 0 |
| max | Maximum value for range | 100 |
| step | Amount to increase by on each increment. | 1 |

## Week

<input type="week" />

Dependent on browser support, a control will show for entering a week-year number and a week number.

## Url

**<input** type="url" name="Homepage"**>**

This is used for input fields that should contain a URL address.

Depending on browser support, the url field can be automatically validated when submitted.

## Time

**<input** type="time" **/>**

The time input marks this element as accepting a string representing a time.

## Date

**<input** type="date" **/>**

A date picker will pop up on screen for you to choose a date. This is not supported in Firefox or Internet Explorer

## DateTime-Local

**<input** type="datetime-local" **/>**

Dependent on browser support, a date and time picker will pop up on screen for you to choose a date and time.

## Month

**<input** type="month" **/>**

Dependent on browser support, a control will show to pick the month.

**Note**: For all HTML Patterns : http://html5pattern.com/

# Forms

## Submitting

### The Action Attribute

The action attribute defines the action to be performed when the form is submitted. If you leave it blank, it will send it to the same file

<**form** action=" ">

### The Method Attribute

The method attribute is used to define the HTTP method of the form which is either GET or POST.

<**form** action="action.php" method="get">

<**form** action="action.php" method="post">

### Uploading Files

Images and files can be uploaded/submitted to server by setting **enctype** attribute of form tag to **multipart/formdata.**

### Example

<**form** method="post" enctype="multipart/form-data" action=" ">

<**input** type="file" name="pic" />

<**input** type="submit" value="Upload" />

</**form**>

## Grouping a few input fields

While designing a form, you might like to group a few input fields into a group to help organise the form layout.

## Example

```
<form>
<fieldset>
<legend>1st field set:</legend>
Field one:  <input type="text"><br>
Field two:  <input type="text"><br>
</fieldset><br>
<fieldset>
<legend>2nd field set:</legend>
Field three: <input type="text"><br>
Field four: <input type="text"><br>
</fieldset><br>
<input type="submit" value="Submit">
</form>
```

# Div Element

The  element usually has no specific semantic meaning by itself, simply representing a division, and is typically used for grouping within an HTML document.

As such, each  is best described by its contents.

 Hello! This is a paragraph. 

## Nesting

** ** inside another ** **. This is usually referred to as "nesting" elements and allows for further dividing elements into subsections or aid developers with CSS styling.

The **<div** class="outer-div"**>** is used to group together two **<div** class="inner-div"**>** elements; each containing a ** ** element.

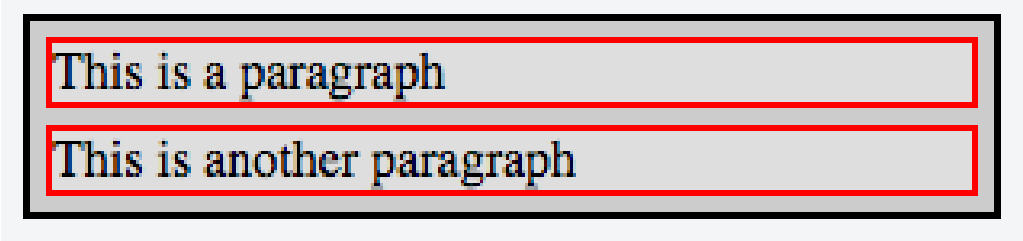
## Example

**<div** class="outer-div"**>**

**<div** class="inner-div"**> **This is a paragraph** **

**<div** class="inner-div"**> **This is another paragraph** **

** **




Note : After CSS.

## About Label

```
<label>
    <input type="checkbox" name="Cats">
    I like Cats!
</label>
```

## Example

```
<form action="#" method="POST">
<label>Username:</label>
<input id="username" type="text"/>
<label >Password:</label>
<input id="pass" type="password" name="pass" />
<input type="submit" name="submit" />
</form>
```

# Media Elements

## Audio

HTML5 provides a new standard for embedding an audio file on a web page. You can embed an audio file to a page using the **<audio>** element:

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

## Video

You can embed also a video to a webpage using the **<video>** element:

**<video** width="500" height="700" controls**>**

**<source** src="video.mp4" type="video/mp4"**>**

Your browser does not support the video tag.

**</video>**

# Selection Menu Controls

## Select Menu

The **<select>** element generates a drop-down menu from which the user can choose an option.

**<select >**

**<option** value="1"**>**One**</option>**

**<option** value="2"**>**Two**</option>**

**<option** value="3"**>**Three**</option>**

**<option** value="4"**>**Four**</option>**

**</select>**

**For Multi-option Selection : <select** multiple**></select>**

**Selecting an option by default :** <option selected>One option</option>

## Option Groups

You can group your options within a selection menu by using the **<optgroup>** element.

```
<select>
  <option>Milk</option>
  <optgroup label="Fruits">
    <option>Bananas</option>
    <option>Strawberries</option>
  </optgroup>
  <optgroup label="Vegetables" disabled>
    <option>Carrots</option>
    <option>Zucchini</option>
  </optgroup>
</select>
```

## Datalist

The **<datalist>** tag specifies a list of pre-defined options for an **<input>** element. It provide an "autocomplete" feature on **<input>** elements. Users will see a drop-down list of options as they write.

## Example

```
<input list="Languages">
<datalist id="Languages">
  <option value="PHP">
  <option value="Perl">
  <option value="Python">
  <option value="Django">

</datalist>
```

# HTML Event Attributes

## HTML Form Events

**onblur**

Fires the moment that the element loses focus

**onchange**

Fires the moment when the value of the element is changed

**oncontextmenu**

Script to be run when a context menu is triggered

**onfocus**

Fires the moment when the element gets focus

**oninput**

Script to be run when an element gets user input

**oninvalid**

Script to be run when an element is invalid

**onreset**

Fires when the Reset button in a form is clicked

**onselect**

Fires after some text has been selected in an element

**onsubmit**

Fires when a form is submitted

## Keyboard Events

### onkeydown

Fires when a user is pressing a key

### onkeypress

Fires when a user presses a key

### onkeyup

Fires when a user releases a key