HTTP

HTTP (Hypertext Transfer Protocol) is the set of rules for transferring files (text, graphic images, sound, video, and other multimedia files) on the World Wide Web.

Url

A **Uniform Resource Locator** (**URL**), colloquially termed a **web address**,[1] is a reference to a web resource that specifies its location on a computer network and a mechanism for retrieving it.

Most web browsers display the URL of a web page above the page in an address bar. A typical URL could have the form `http://www.example.com/index.html`, which indicates a protocol (`http`), a hostname (`www.example.com`), and a file name (`index.html`).

WWW

World wide web is virtual space where all the servers are connected.

Eg:-shopping mall where everything is available which you want to buy and roads are the internet define how to reach to shopping mall.

www is governed by rules laws & regulations that is hyper text transfer protocol.

Web Technology

Web technology refers to the means by which computers communicate with each other using markup languages and multimedia packages. It gives us a way to interact with hosted information, like websites. **Web technology** involves the use of hypertext markup language (HTML) and cascading style sheets (CSS).

1. Browsers

Browsers are the interpreters of the web. They request information and then when they receive it, they show us on the page in a format we can see and understand.

- Google Chrome - Currently, the most popular browser brought to you by Google
- Safari - Apple's web browser1
- Firefox - Open-source browser supported by the Mozilla Foundation
- Internet Explorer - Microsoft's browser. You will most often hear web developers complain about this one.

2. HTML

HTML is a markup language. It provides the structure of a website so that web browsers know what to show.

3. CSS

CSS is a Cascading Style Sheet. CSS let's web designers change colors, fonts, animations, on the web. They make the web look good.

4. Programming Languages

Programming languages are ways to communicate to computers and tell them what to do. There are many different programming languages just like there are many different lingual languages (English, Spanish, French, Chinese, etc). One is not better than the other. Developers typically are just proficient at a couple so they promote those more than others. Below are just some of the languages and links to their homepages

- Javascript - used by all web browsers, Meteor, and lots of other1 frameworks

- Coffeescript - is a kind of “dialect” of javascript. It is viewed as simpler and easier on your eyes as a developer but it complies (converts) back into javascript
- Python -used by the Django framework and used in a lot of mathematical calculations
- Ruby - used by the Ruby on Rails framework
- PHP - used by Wordpress
- Go - newer language, built for speed.
- Objective-C - the programming language behind iOS (your iPhone), lead by Apple
- Swift - Apple’s newest programming language
- Java - Used by Android (Google) and a lot of desktop applications.

5. Frameworks

Frameworks are built to make building and working with programming languages easier. Frameworks typically take all the difficult, repetitive tasks in setting up a new web application and either does them for you or make them very easy for you to do.

- Meteor - a full-stack (front and back end) javascript framework
- Node.js - a server-side javascript framework
- Ruby on Rails - a full-stack framework built using ruby
- Django - a full-stack framework built using python
- Ionic - a mobile framework
- Phonegap / Cordova - a mobile framework that exposes native api’s of iOS and Android for use when writing javascript
- Bootstrap - a UI (user interface) framework for building with HTML/CSS/Javascript
- Foundation - a UI framework for building with HTML/CSS/Javascript
- Wordpress - a CMS (content management system) built on PHP. Currently, about 20% of all websites run on this framework
- Drupal - a CMS framework built using PHP.
- .NET - a full-stack framework built by Microsoft2
- Angular.js - a front-end javascript framework.
- Ember.js - a front-end javascript framework.
- Backbone.js - a front-end javascript framework.

6. Libraries

Libraries are groupings of code snippets to enable a large amount of functionality without having to write it all by yourself. Libraries typically also go through the trouble to make sure the code is efficient and works well across browsers and devices (not always the case, but typically they do).

- jQuery

7. Databases

Databases are where all your data is stored. MongoDB - is an open-sourced NoSQL database and is currently the only database supported by Meteor.

- Redis

- PostgreSQL
- MySQL –
-
- Oracle .
- SQL Server

8. Client (or Client-side)

A client is one user of an application. It's you and me when we visit http://google.com. Client's can be desktop computers, tablets, or mobile devices. There are typically multiple clients interacting with the same application stored on a server.

9. Server (or Server-side)

A server is where the application code is typically stored. Requests are made to the server from clients, and the server will gather the appropriate information and respond to those requests.

10. Front-end

The front-end is comprised of HTML, CSS, and Javascript. This is how and where the website is shown to users.

11. Back-end

The back-end is comprised of your server and database. It's the place where functions, methods, and data manipulation happens that you don't want the clients to see.

12. Protocols

Protocols are standardized instructions for how to pass information back and forth between computers and devices.

- HTTP - This protocol is how each website gets to your browser. Whenever you type a website like "http://google.com" this protocol requests the website from google's server and then receives a response with the HTML, CSS, and javascript of the website.

13. API

An API is an application programming interface. It is created by the developer of an application to allow other developers to use some of the application's functionality without sharing code.

HTML

What is HTML?

- Stands for Hypertext Markup Language.
- Most documents that appear on the World Wide Web were written in HTML.
- HTML is a markup language, not a programming language. In fact, the term HTML is an acronym that stands for Hypertext Markup Language.
- We can apply this markup language to your pages to display text, images, sound and movie files, and almost any other type of electronic information.
- We use the language to format documents and link them together, regardless of the type of computer with which the file was originally created.

The major points of HTML are given below:

- HTML stands for Hyper Text Markup Language.
- HTML is used to create web pages.
- HTML is widely used language on the web.
- We can create static website by HTML only.

HTML Example

Example:-

```
1. <!DOCTYPE>
2. <html>
3. <body>
4. <h1>Write Your First Heading</h1>
5. <p>Write Your First Paragraph.</p>
6. </body>
7. </html>
```

Brief History of HTML

Tim Berners-Lee is known as *father of HTML*. The first available description of HTML was a document called "HTML Tags" proposed by Tim in late 1991.

ersion	Year
TML	1991
TML 2.0	1995
TML 3.2	1997
TML 4.01	1999
HTML	2000
TML5	2014

What is HTML

HTML is stands for Hyper Text Markup Language. Let's see what is Hyper Text and what is Markup Language?

Hyper Text: Hyper Text simply means "Text within Text". A text has a link within it, is a hypertext. Every time when you click on a word which brings you to a new webpage, you have clicked on a hypertext.

Markup language: A markup language is used to make text more interactive and dynamic. It can turn a text into images, tables, links etc.

Follow the four steps below to create your first web page with Notepad or TextEdit.

Step 1: open text editor.

Step 2: Write Some HTML

Step 3: Save the HTML Page

Step 4: View the HTML Page in Your Browser

An HTML document is made of many HTML tags and each HTML tag contains different content.

Example:-

```
1. <!DOCTYPE html>
2. <html>
3. <body>
4. <h1>Write Your First Heading</h1>
5. <p>Write Your First Paragraph.</p>
6. </body>
7. </html>
```

Description of HTML Example

- The `<!DOCTYPE html>` declaration defines this document to be HTML5
- The `<html>` element is the root element of an HTML page
- The `<head>` element contains meta information about the document
- The `<title>` element specifies a title for the document
- The `<body>` element contains the visible page content
- The `<h1>` element defines a large heading
- The `<p>` element defines a paragraph

```
<html>
<head>
<title>Page title</title>
</head>
<body>
<h1>This is a heading</h1>
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>
</body>
</html>
```

HTML Elements

HTML Tags

- A web browser reads an HTML document top to bottom, left to right.
- Each time the browser finds a tag, it is displayed accordingly (paragraphs look like paragraphs, tables look like tables, etc).
- Tags have 3 major parts: opening tag(s), content(s), and closing tag(s).
- Recall that a completed tag is termed an element.

- An element consists of three basic parts: an opening tag, the element's content, and finally, a closing tag.

  * - opening paragraph tag*
  ElementContent-paragraphwords

  * - closing tag*

- Every (web) page requires four critical elements: the html, head, title, and body elements.

2. <html> Element…</html>

- <html> begins and ends each and every web page.
- Its purpose is to encapsulate all the HTML code and describe the HTML document to the web browser.

  <html></html>

3. <head> Element

- The <head> element is "next" as they say. As long as it falls somewhere between your <html> tag and your web page content (<body>).
- The head functions "behind the scenes." Tags placed within the head element are not directly displayed by web browsers.
- We will be placing the <title> element here.
- Other elements used for scripting (JavaScript) and formatting (CSS) will eventually be introduced and you will have to place them within your head element.

  <html>

  <head>

  </head>

  </html>

4. The <title> Element

- Place the <title> tag within the <head> element to title your page.
- The words you write between the opening and closing <title></title> tags will be displayed at the top of a viewer's browser.

<html><head><title>My WebPage!</title></head></html>

5. The <body> Element

- The <body> element is where all content is placed. (Paragraphs, pictures, tables, etc).
- The body element will encapsulate all of your webpage's viewable content.

<html>

<head><title>My WebPage!</title></head>

<body>

Hello World! All my content goes here!

</body>

</html>

1. Paragraph Tag

- The tag defines a paragraph. Using this tag places a blank line above and below the text of the paragraph.

* Avoid losing floppy disks with important school... *

* For instance, let's say you had a HUGE school... *

2. HTML - Headings 1:6

- A heading in HTML is just what we might expect, a title or subtitle.
- By placing text inside of <h1> (heading) tags, the text displays bold and the size of the text depends on the number of heading (1-6).
- Headings are numbered 1-6, with 1 being the largest heading and 6 being the smallest.

<html><body>

<h1>Thisisheading1</h1><h2>Thisisheading2</h2><h3>Thisisheading3</h3>

<h4>Thisisheading4</h4><h5>Thisisheading5</h5><h6>Thisisheading6</h6>

</body></html>

3. Line Breaks

- Line breaks are different then most of the tags we have seen so far. A line break ends the line you are

currently on and resumes on the next line

HTML 5

, we will learn HTML 5 tags such as audio tag, video tag, canvas tag, HTML svg, HTML geolocation, HTML drag and drop etc.

Note:-

1. its supports multimedia without flash player.

2. Creating drawing in your web page using canvas ,svg and js code.

3. Support to work with server and database.

4. It helps to create API

5. Fully supports css3 and become more powerfull.

HTML5 Document

The following tags have been introduced for better structure –

- **section** – This tag represents a generic document or application section. It can be used together with h1-h6 to indicate the document structure.
- **article** – This tag represents an independent piece of content of a document, such as a blog entry or newspaper article.
- **aside** – This tag represents a piece of content that is only slightly related to the rest of the page.
- **header** – This tag represents the header of a section.
- **footer** – This tag represents a footer for a section and can contain information about the author, copyright information, etc.
- **nav** – This tag represents a section of the document intended for navigation.
- **dialog** – This tag can be used to mark up a conversation.
- **figure** – This tag can be used to associate a caption together with some embedded content, such as a graphic or video.

Web Browsers

The purpose of a web browser (Chrome, IE, Firefox, Safari) is to read HTML documents and display them.

The browser does not display the HTML tags, but uses them to determine how to display the document:

Features of HTML

1) It is a very **easy and simple** language. It can be easily understood and modified.

2) It is very easy to make **effective presentation** with HTML because it has a lot of *formatting tags*.

3) It is a **markup language** so it provides a flexible way to design web pages along with the text.

4) It facilitates programmers to add **link** on the web pages (by *html anchor tag*) , so it enhances the interest of browsing of the user.

5) It is **platform-independent** because it can be displayed on any platform like Windows, Linux and Macintosh etc.

6) It facilitates the programmer to add **Graphics, Videos, and Sound** to the web pages which makes it more attractive and interactive.

HTML Tags

HTML tags contain three main parts: opening tag, content and closing tag. But some HTML tags are unclosed tags.

When a web browser reads an HTML document, browser reads it from top to bottom and left to right. HTML tags are used to create HTML documents .Each HTML tags have different properties.

The HTML **element** is everything from the start tag to the end tag:

Syntax

<tag> content </tag>

Unclosed HTML Tags

Some HTML tags are not closed, for example br hr and img.

**
 Tag**: br stands for break line, it breaks the line of the code.

<hr> Tag: hr stands for Horizontal Rule. This tag is used to put a line across the webpage.

HTML Meta Tags

DOCTYPE, title, link, meta and style

Nested HTML Elements

HTML elements can be nested (elements can contain elements).

All HTML documents consist of nested HTML elements.

HTML Attributes

- All HTML elements can have **attributes**
- Attributes provide **additional information** about an element
- Attributes are always specified in **the start tag**
- Attributes usually come in name/value pairs like: **name="value"**

The src Attribute

HTML images are defined with the **<img>** tag.

The filename of the image source is specified in the **src** attribute:

Eg:-

```
<img src="img_girl.jpg" width="500" height="600">
```

The style Attribute

The style attribute is used to specify the styling of an element, like color, font, size etc.

```
<p style="color:red">I am a paragraph</p>
```

The title Attribute

Here, a **title** attribute is added to the **<p>** element. The value of the title attribute will be displayed as a tooltip when you mouse over the paragraph:

Eg:-

```
<p title="I'm a tooltip">This is a paragraph.</p>
```

width and height Attributes

Images in HTML have a set of size attributes, which specifies the width and height of the image:

Eg:

```
<img src="img_girl.jpg" width="500" height="600">
```

HTML Formatting

HTML Formatting is *a process of formatting text for better look and feel*. There are many formatting tags in HTML. These tags are used to make text bold, italicized, or underlined.

1) Bold Text

If you write anything within ..... element, is shown in bold letters.

See this example:

** **Write Your First Paragraph in bold text.** **

2) Italic Text

If you write anything within ..... element, is shown in italic letters.

See this example:

** **Write Your First Paragraph in italic text.** **

3) HTML Marked formatting

If you want to mark or highlight a text, you should write the content within <mark>.....</mark>.

See this example:

<h2> I want to put a **<mark>** Mark**</mark>** on your face**</h2>**

4) Underlined Text

If you write anything within <u>.....</u> element, is shown in underlined text.

See this example:

```
<p> <u>Write Your First Paragraph in underlined text.</u></p>
```

5) Strike Text

Anything written within <strike>.....</strike> element is displayed with strikethrough. It is a thin line which cross the statement.

See this example:

```
<p> <strike>Write Your First Paragraph with strikethro
ugh</strike>.</p>
```

6) Monospaced Font

If you want that each letter has the same width then you should write the content within <tt>.....</tt> element.

Example:-

```
<p>Hello <tt>Write Your First Paragraph in monospaced font.</tt></p>
```

7) Superscript Text

If you put the content within ^{.....} element, is shown in superscript ; means it is displayed half a character's height above the other characters.

See this example:

** **Hello **^{**Write Your First Paragraph in superscrip t.**} **

8) Subscript Text

If you put the content within _{.....} element, is shown in subscript ; means it is displayed half a character's height below the other characters.

See this example:

** **Hello **_{**Write Your First Paragraph in subscript.**} **

9) Deleted Text

Anything that puts within ~~.....~~ is displayed as deleted text.

See this example:

** **Hello **~~**Delete your first paragraph.**~~ **

10) Inserted Text

Anything that puts within <ins>.....</ins> is displayed as inserted text.

See this example:

** ~~**Delete your first paragraph.**~~<ins>**Write another paragraph.**</ins></ p>**

11) Larger Text

If you want to put your font size larger than the rest of the text then put the content within <big>.....</big>. It increase one font size larger than the previous one.

See this example:

** **Hello **<big>**Write the paragraph in larger font.**</big> **

12) Smaller Text

If you want to put your font size smaller than the rest of the text then put the content within <small>.....</small>tag. It reduces one font size than the previous one.

See this example:

** **Hello **<small>**Write the paragraph in smaller font.**</small> **

HTML Anchor

The **HTML anchor tag** defines *a hyperlink that links one page to another page*. The "href" attribute is the most important attribute of the HTML a tag.

href attribute of HTML anchor tag

The href attribute is used to define the address of the file to be linked. In other words, it points out the destination page

The syntax of HTML anchor tag is given below.

```
<a href = "..........."> Link Text </a>
```

Appearance of HTML anchor tag

An **unvisited link** is displayed underlined and blue.

A **visited link** displayed underlined and purple.

An **active link** is underlined and red.

HTML Ordered List | HTML Numbered List

HTML Ordered List or Numbered List displays elements in numbered format. The HTML ol tag is used for ordered list. There can be different types of numbered list:

- Numeric Number (1, 2, 3)
- Capital Roman Number (I II III)
- Small Roman Number (i ii iii)
- Capital Alphabet (A B C)
- Small Alphabet (a b c)

To represent different ordered lists, there are 5 types of attributes in <ol> tag.

Type	Description
Type "1"	This is the default type. In this type, the list items are numbered with numbers.
Type "I"	In this type, the list items are numbered with upper case roman numbers.
Type "i"	In this type, the list items are numbered with lower case roman numbers.
Type	In this type, the list items are numbered with upper

"A"	case letters.
Type "a"	In this type, the list items are numbered with lower case letters.

Example:-

```
<ol type="I">
 <li>HTML</li>
 <li>Java</li>
 <li>JavaScript</li>
 <li>SQL</li>
</ol>
```

HTML Description List | HTML Definition List

HTML Description List or Definition List displays elements in definition form like in dictionary. The <dl>, <dt> and <dd> tags are used to define description list.

The 3 HTML description list tags are given below:

1. **<dl> tag** defines the description list.
2. **<dt> tag** defines data term.
3. **<dd> tag** defines data definition (description).

```
<dl>
  <dt>HTML</dt>
  <dd>is a markup language</dd>
  <dt>Java</dt>
  <dd>is a programming language and platform</d
d>
```

```
 <dt>JavaScript</dt>
 <dd>is a scripting language</dd>
  <dt>SQL</dt>
  <dd>is a query language</dd>
</dl>
```

Output:

```
HTML
        is a markup language
Java
        is a programming language and platform
JavaScript
        is a scripting language
SQL
        is a query language
```

HTML Table

HTML table tag is used to display data in tabular form (row * column). There can be many columns in a row.

HTML tables are used to manage the layout of the page e.g. header section, navigation bar, body content, footer section etc. But it is recommended to use div tag over table to manage the layout of the page .

HTML Table Tags

Tag	Description
<table>	It defines a table.

<tr>	It defines a row in a table.
<th>	It defines a header cell in a table.
<td>	It defines a cell in a table.

HTML Table with Border

There are two ways to specify border for HTML tables.

1. By border attribute of table in HTML
2. By border property in CSS

---

1) HTML Border attribute

You can use border attribute of table tag in HTML to specify border. But it is not recommended now.

```
<table border="1">
<tr><th>First_Name</th><th>Last_Name</th><th>Marks</th></tr>
<tr><td>Sonoo</td><td>Jaiswal</td><td>60</td></tr>
<tr><td>James</td><td>William</td><td>80</td></tr>
<tr><td>Swati</td><td>Sironi</td><td>82</td></tr>
<tr><td>Chetna</td><td>Singh</td><td>72</td></tr>
</table>
```

2) CSS Border property

It is now recommended to use border property of CSS to specify border in table.

```
<style>
table, th, td {
```

```
   border: 1px solid black;
}
</style>
```

HTML Table with cell padding

You can specify padding for table header and table data by two ways:

1. By cellpadding attribute of table in HTML
2. By padding property in CSS

```
<style>
table, th, td {
   border: 1px solid pink;
   border-collapse: collapse;
}
th, td {
   padding: 10px;
}
</style>
```

HTML Form

An **HTML form** is *a section of a document* which contains controls such as text fields, password fields, checkboxes, radio buttons, submit button, menus etc.

An HTML form facilitates the user to enter data that is to be sent to the server for processing.

---

ag	Description

Why use HTML Form

HTML forms are required if you want to collect some data from the site visitor.

For example: If a user want to purchase some items on internet, he/she must fill the form such as shipping address and credit/debit card details so that item can be sent to the given address.

HTML Form Syntax

```
1. <form action="server url" method="get|post">
2.  //input controls e.g. textfield, textarea, radiobutton, button
3. </form>
```

HTML Form Tags

Let's see the list of HTML 5 form tags.

<form>	It defines an HTML form to enter inputs by the used side.
<input>	It defines an input control.
<textarea>	It defines a multi-line input control.
<label>	It defines a label for an input element.
<fieldset>	It groups the related element in a form.
<legend>	It defines a caption for a <fieldset> element.
<select>	It defines a drop-down list.
<optgroup>	It defines a group of related options in a drop-down list.
<option>	It defines an option in a drop-down list.
<button>	It defines a clickable button.

The Submit Button

<input type="submit"> defines a button for **submitting** the form data to a **form-handler**.

The form-handler is typically a server page with a script for processing input data.

The form-handler is specified in the form's **action** attribute:

HTML Form Elements

The <input> Element

The most important form element is the **<input>** element.

The <input> element can be displayed in several ways, depending on the **type** attribute.

Example:-

```
<input name="firstname" type="text">
```

The <select> Element

The **<select>** element defines a **drop-down list**:

Example:-

```
<select name="cars">
  <option value="volvo">Volvo</option>
  <option value="saab">Saab</option>
  <option value="fiat">Fiat</option>
  <option value="audi">Audi</option>
</select>
```

The **<option>** elements defines an option that can be selected.

By default, the first item in the drop-down list is selected.

To define a pre-selected option, add the **selected** attribute to the option:

```
<option value="fiat" selected>Fiat</option>
```

Visible Values:

Use the **size** attribute to specify the number of visible values:

Example:-

```
<select name="cars" size="3">
  <option value="volvo">Volvo</option>
  <option value="saab">Saab</option>
  <option value="fiat">Fiat</option>
  <option value="audi">Audi</option>
</select>
```

Allow Multiple Selections:

Use the **multiple** attribute to allow the user to select more than one value:

Example:-

```
<select name="cars" size="4" multiple>
  <option value="volvo">Volvo</option>
  <option value="saab">Saab</option>
  <option value="fiat">Fiat</option>
  <option value="audi">Audi</option>
</select>
```

The <textarea> Element

The **<textarea>** element defines a multi-line input field (**a text area**):

Example:-

```
<textarea name="message" rows="10" cols="30">
this is text area</textarea>
```

The **rows** attribute specifies the visible number of lines in a text area.

The **cols** attribute specifies the visible width of a text area.

This is how the HTML code above will be displayed in a browser:

You can also define the size of the text area by using CSS:

Example:-

```
<textarea name="message" style="width:200px; height:600px">
The cat was playing in the garden.
</textarea>
```

The <button> Element

The **<button>** element defines a clickable **button**:

```
<button type="button" onclick="alert('Hello World!')">Click
Me!</button>
```

HTML5 Form Elements

HTML5 added the following form elements:

- <datalist>

HTML5 <datalist> Element

The **<datalist>** element specifies a list of pre-defined options for an <input> element.

Users will see a drop-down list of the pre-defined options as they input data.

The **list** attribute of the <input> element, must refer to the **id** attribute of the <datalist> element.

```
<form action="/action_page.php">
  <input list="browsers">
  <datalist id="browsers">
    <option value="Internet Explorer">
    <option value="Firefox">
    <option value="Chrome">
    <option value="Opera">
```

```
    <option value="Safari">
  </datalist>
</form>
```

Grouping Form Data with <fieldset>

The **<fieldset>** element is used to group related data in a form.

The **<legend>** element defines a caption for the <fieldset> element.

Example:-

```
<form action="/action_page.php">
  <fieldset>
    <legend>Personal information:</legend>
    First name:<br>
    <input type="text" name="firstname" value="Mickey"><br>
    Last name:<br>
    <input type="text" name="lastname" value="Mouse"><br><br>
    <input type="submit" value="Submit">
  </fieldset>
</form>
```

HTML Input Types

input Type Text

<input type="text"> defines a **one-line text input field**

Input Type Password

<input type="password"> defines a **password field**:

Input Type Submit

<input type="submit"> defines a button for **submitting** form data to a **form-handler**.

The form-handler is typically a server page with a script for processing input data.

The form-handler is specified in the form's **action** attribute:

Input Type Reset

<input type="reset"> defines a **reset button** that will reset all form values to their default values:

Input Type Radio

<input type="radio"> defines a **radio button**.

Radio buttons let a user select ONLY ONE of a limited number of choices:

Input Type Checkbox

<input type="checkbox"> defines a **checkbox**.

Checkboxes let a user select ZERO or MORE options of a limited number of choices.

Input Type Button

<input type="button"> defines a **button**:

HTML5 Input Types

HTML5 added several new input types:

- color
- date
- datetime-local
- email
- month
- number
- range
- search
- tel
- time
- url
- week

Input Type Color

The **<input type="color">** is used for input fields that should contain a color.

Input Type Date

The **<input type="date">** is used for input fields that should contain a date.

Input Type Datetime-local

The **<input type="datetime-local">** specifies a date and time input field, with no time zone.

Input Type Email

The **<input type="email">** is used for input fields that should contain an e-mail address.

Input Type Month

The **<input type="month">** allows the user to select a month and year.

Input Type Number

The **<input type="number">** defines a **numeric** input field.

You can also set restrictions on what numbers are accepted.

The following example displays a numeric input field, where you can enter a value from 1 to 5:

Example:-

```
<form>
  Quantity (between 1 and 5):
  <input type="number" name="quantity" min="1" max="5">
</form>
```

Input Type Search

The **<input type="search">** is used for search fields (a search field behaves like a regular text field).

Input Type Tel

The **<input type="tel">** is used for input fields that should contain a telephone number.

Input Type Time

The **<input type="time">** allows the user to select a time (no time zone).

Input Type Url

The **<input type="url">** is used for input fields that should contain a URL address.

Input Type Week

The **<input type="week">** allows the user to select a week and year.

HTML Div Tag

The **HTML tag** is used *to group the large section of HTML elements together*.

We know that every tag has a specific purpose e.g. p tag is used to specify paragraph, <h1> to <h6> tag are used to specify headings but the tag is just like a container unit which is used to encapsulate other page elements and divides the HTML documents into sections.

The div tag is generally used by web developers to group HTML elements together and apply CSS styles to many elements at once. For example: If you wrap a set of paragraph elements into a div element so you can take the advantage of CSS styles and apply font style to all paragraphs at once instead of coding the same style for each paragraph element.

```
<div style="border:1px solid pink;padding:20px;font-size:20px">

<p>, Here you get tutorials on latest technologies.</p>

<p>This is second paragraph</p>

</div>
```

Difference between HTML div tag and span tag

div tag	span tag
HTML div is a **block** element.	HTML span is an **inline** element
HTML div element is used to **wrap large sections of elements**.	HTML span element is used to **wrap small portion of texts, image** etc.

HTML <frame> Tag.

HTML File Paths

A file path describes the location of a file in a web site's folder structure.

File paths are used when linking to external files like:

- Web pages
- Images
- Style sheets
- JavaScripts

Absolute File Paths

An absolute file path is the full URL to an internet file:

Relative File Paths

A relative file path points to a file relative to the current page.

In this example the file path points to a file in the images folder located at the root of the current web:

```
<img src="/images/picture.jpg" alt="Mountain">
```

In this example the file path points to a file in the images folder located in the current folder:

Example

```
<img src="images/picture.jpg" alt="Mountain">
```

In this example the file path points to a file in the images folder located in the folder one level above the current folder:

Example

```
<img src="../images/picture.jpg" alt="Mountain">
```

Best Practice

It is a best practice to use relative file paths (if possible).

When using relative file paths, your web pages will not be bound to your current base URL. All links will work on your own computer (localhost) as well as on your current public domain and your future public domains.

HTML Table - Adding a Caption

To add a caption to a table, use the **<caption>** tag:

```
<table style="width:100%">
  <caption>Monthly savings</caption>
  <tr>
    <th>Month</th>
    <th>Savings</th>
  </tr>
  <tr>
    <td>January</td>
    <td>$100</td>
```

```
  </tr>
  <tr>
    <td>February</td>
    <td>$50</td>
  </tr>
</table>
```

Note: The <caption> tag must be inserted immediately after the <table> tag.

What is Multimedia?

Multimedia comes in many different formats. It can be almost anything you can hear or see.

Examples: Images, music, sound, videos, records, films, animations, and more.

Web pages often contain multimedia elements of different types and formats.

In this chapter you will learn about the different multimedia formats.

Browser Support

The first web browsers had support for text only, limited to a single font in a single color.

Later came browsers with support for colors and fonts, and images!

Audio, video, and animation have been handled differently by the major browsers. Different formats have been supported, and some formats require extra helper programs (plug-ins) to work.

Hopefully this will become history. HTML5 multimedia promises an easier future for multimedia.

Multimedia Formats

Multimedia elements (like audio or video) are stored in media files.

The most common way to discover the type of a file, is to look at the file extension.

Multimedia files have formats and different extensions like: .swf, .wav, .mp3, .mp4, .mpg, .wmv, and .avi.

Common Video Formats

MP4 is the new and upcoming format for internet video.

MP4 is recommended by YouTube.

MP4 is supported by Flash Players.

MP4 is supported by HTML5.

Format	File	Description
MPEG	.mpg .mpeg	MPEG. Developed by the Moving Pictures Expert Group. The first popular video format on the web. Used to be supported by all browsers, but it is not supported in HTML5 (See MP4).
AVI	.avi	AVI (Audio Video Interleave). Developed by Microsoft. Commonly used in video cameras and TV hardware. Plays well on Windows computers, but not in web browsers.

WMV	.wmv	WMV (Windows Media Video). Developed by Microsoft. Commonly used in video cameras and TV hardware. Plays well on Windows computers, but not in web browsers.
QuickTime	.mov	QuickTime. Developed by Apple. Commonly used in video cameras and TV hardware. Plays well on Apple computers, but not in web browsers. (See MP4)
RealVideo	.rm .ram	RealVideo. Developed by Real Media to allow video streaming with low bandwidths. It is still used for online video and Internet TV, but does not play in web browsers.
Flash	.swf .flv	Flash. Developed by Macromedia. Often requires an extra component (plug-in) to play in web browsers.
Ogg	.ogg	Theora Ogg. Developed by the Xiph.Org Foundation. Supported by HTML5.
WebM	.webm	WebM. Developed by the web giants, Mozilla, Opera, Adobe, and Google. Supported by HTML5.
MPEG-4 or MP4	.mp4	MP4. Developed by the Moving Pictures Expert Group. Based on QuickTime. Commonly used in newer video cameras and TV hardware. Supported by all HTML5 browsers. Recommended by YouTube.

Only MP4, WebM, and Ogg video are supported by the HTML5 standard.

HTML Audio Tag

Audio on the Web

Before HTML5, audio files could only be played in a browser with a plug-in (like flash).

The HTML5 `<audio>` element specifies a standard way to embed audio in a web page.

HTML audio tag is used to define sounds such as music and other audio clips. Currently there are three supported file format for HTML 5 audio tag.

1. mp3
2. wav
3. ogg

HTML5 supports <video> and <audio> controls. The Flash, Silverlight and similar technologies are used to play the multimedia items.

This table defines that which web browser supports which audio file format.

Browser	mp3	wav
Internet Explorer	yes	no
Google Chrome	yes	yes
Mozilla Firefox	yes*	yes
Opera	no	yes
Apple Safari	yes	yes

Attributes of HTML Audio Tag

There is given a list of HTML audio tag.

Attribute	Description
controls	It defines the audio controls which is displayed with play/pause buttons.
autoplay	It specifies that the audio will start playing as soon as it is ready.
loop	It specifies that the audio file will start over again, every time when it is completed.
muted	It is used to mute the audio output.
preload	It specifies the author view to upload audio file when the page loads.
src	It specifies the source URL of the audio file.

HTML Audio Tag Attribute Example

Here we are going to use controls, autoplay, loop and src attributes of HTML audio tag.

```
<audio controls autoplay loop>
 <source src="koyal.mp3" type="audio/mpeg"></audio>
```

MIME Types for HTML Audio format

The available MIME type HTML audio tag is given below.

Audio Format	MIME Type

mp3	audio/mpeg
ogg	audio/ogg
wav	audio/wav

HTML Video Tag

HTML 5 supports <video> tag also. The HTML video tag is used for streaming video files such as a movie clip, song clip on the web page.

Currently, there are three video formats supported for HTML video tag:

1. mp4
2. webM
3. ogg

Let's see the table that defines which web browser supports video file format.

Browser	mp4	webM
Internet Explorer	yes	no
Google Chrome	yes	yes
Mozilla Firefox	yes	yes
Opera	no	yes
Apple Safari	yes	no

HTML Video Tag Example

Let's see the code to play mp4 file using HTML video tag.

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

Let's see the example to play ogg file using HTML video tag.

```
<video controls>
  <source src="movie.ogg" type="video/ogg">
 Your browser does not support the html video tag.
</video>
```

HTML Progress Tag

HTML <progress> tag is used to display the progress of a task. It provides an easy way for web developers to create progress bar on the website. It is mostly used to show the progress of a file uploading on the web page.

The HTML progress tag is new in HTML5 so you must use new browsers.

Attributes of HTML Progress Tag

HTML <progress> tag supports the global and event attributes as well as 2 specific attributes.

Tag	Description
value	It defines that how much work the task has been completed.

max	It defines that how much work the task requires in total.

HTML Progress Tag Example

Let's see HTML progress example without attribute.

```
<progress></progress>
```

Styling Progress Bar

You can apply CSS code on progress tag.

```
progress{
  width: 300px;
  height: 30px;
}
```

```
<!DOCTYPE>
<html>
<head>
<style>
progress{width:300px;height:30px}
</style>
</head>
```

```
<body>
<progress value="50" max="100"></progress>
</body>
</html>
```

Marquee HTML

The **Marquee HTML** tag is a non-standard HTML element which is used to scroll a image or text horizontally or vertically.

In simple words, you can say that it scrolls the image or text up, down, left or right automatically.

Marquee tag was first introduced in early versions of Microsoft's Internet Explorer. It is compared with Netscape's blink element.

Marquee HTML Example

<marquee>This is an example of html marquee **</marquee>**

HTML Canvas Tag

The **HTML canvas element** provides HTML a bitmapped surface to work with. It is used to draw graphics on the web page.

The **HTML 5 <canvas> tag** is used to draw graphics using scripting language like JavaScript.

The <canvas> element is only a container for graphics, you must need a scripting language to draw the graphics. The <canvas> element allows for dynamic and scriptable rendering of 2D shapes and bitmap images.

It is a low level, procedural model that updates a bitmap and does not have a built-in scene. There are several methods in canvas to draw paths, boxes, circles, text and add images.

How to create a HTML canvas?

A canvas is a rectangle like area on an HTML page. It is specified with canvas element. By default, the <canvas> element has no border and no content, it is like a container.

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

HTML 5 Canvas Tag Example

```
<canvas id="myCanvas1" width="300" height="100" style="border:2px solid;">

Your browser does not support the HTML5 canvas tag.

</canvas>
```

Note: It is always necessary to specify the id attribute and the height & width attribute to define the size of the canvas. You can have multiple canvas elements on one HTML page.

HTML Canvas Tag with JavaScript

A canvas is a two dimensional grid.

Coordinates (0,0) defines the upper left corner of the canvas. The parameters (0,0,200,100) is used for fillRect() method. This parameter will fill the rectangle start with the upper-left corner (0,0) and draw a 200 * 100 rectangle.

```
<canvas id="myCanvas" width="250" height="150" style="border:1px solid #c3c3c3;">

Your browser does not support the HTML5 canvas tag.
```

```
</canvas>

<script>

var c = document.getElementById("myCanvas");

var cctx = c.getContext("2d");

ctx.fillStyle = "#FF0000";

ctx.fillRect(0,0,200,100);

</script>
```

Drawing Circle on Canvas

If you want to draw a circle on the canvas, you can use the arc() method:

arc(x, y, r, start, stop)

To sketch circle on HTML canvas, use one of the ink() methods, like stroke() or fill().

```
<canvas id="myCanvasCircle" width="200" height="100" style="border:1px solid #d3d3d3;
">

Your browser does not support the HTML5 canvas tag.</canvas>

<script>

var c = document.getElementById("myCanvasCircle");

var cctx = c.getContext("2d");

ctx.beginPath();

ctx.arc(95,50,40,0,2*Math.PI);

ctx.stroke();

</script>
```

Drawing text on canvas

There are property and methods used for drawing text on the canvas.

font property: It is used to define the font property for the text.

fillText(text,x,y) method: It is used to draw filled text on the canvas. It looks like bold font.

strokeText(text,x,y) method: It is also used to draw text on the canvas, but the text is unfilled.

Let's see **fillText()** method example.

```
<canvas id="myCanvasText1" width="300" height="100" style="border:1px solid #d3d3d3;
">

Sorry! Your browser does not support the HTML5 canvas tag.</canvas>

<script>

var c = document.getElementById("myCanvasText1");

var cctx = c.getContext("2d");

ctx.font = "30px Arial";

ctx.fillText("Hello JavaTpoint",10,50);

</script>
```

Let's see **strokeText()** method example.

```
<canvas id="myCanvasText2" width="300" height="100" style="border:1px solid #d3d3d3;">

 Sorry!Upgrade your browser. It does not support the HTML5 canvas tag.</canvas>

<script>

var c = document.getElementById("myCanvasText2");

var cctx = c.getContext("2d");

ctx.font = "30px Arial";

ctx.strokeText("Hello JavaTpoint",10,50);
```

```
</script>
```

HTML Iframes

An iframe is used to display a web page within a web page.

Iframe Syntax

An HTML iframe is defined with the **<iframe>** tag:

```
<iframe src="URL"></iframe>
```

The **src** attribute specifies the URL (web address) of the inline frame page.

Iframe - Set Height and Width

Use the **height** and **width** attributes to specify the size of the iframe.

The attribute values are specified in pixels by default, but they can also be in percent (like "80%")

```
<iframe src="demo_iframe.htm" height="200" width="300"></iframe>
```

Iframe - Remove the Border

By default, an iframe has a border around it.

To remove the border, add the **style** attribute and use the CSS **border** property:

```
<iframe src=" " style="border:none;"></iframe>
```

Note:-With CSS, you can also change the size, style and color of the iframe's border:

Example:-

```
<iframe src="demo_iframe.htm" style="border:2px solid grey;"></iframe>
```

Iframe - Target for a Link

An iframe can be used as the target frame for a link.

The **target** attribute of the link must refer to the **name** attribute of the iframe:

```
<iframe src="demo_iframe.htm" name="iframe_a"></iframe>

<p><a href=" " target="iframe_a">rajat </a></p>
```