**What is Internet?**

* A Global network of networks.
* It’s just a bunch of computers connected together.
* Internet is Global network of interconnected computers that communicate via TCP/IP Network of networks.

**What is Web?**

* The World Wide Web is an information system where documents and other resources are available over the Internet.
* Documents are transferred via HTTP.

**Server:**

* A Computer that satisfy requests on the Web.
* Server is also used to refer to the software running on the computer.

**Client:**

* The computer that accesses a server

The basis of all web interactions is someone asking for information, and receiving information. In order to ask for and receive any information, we need two players - the asker and the producer. In basic web interactions, the ‘asker’ is a **client** and the ‘producer’ is a **server**. Clients send **Requests** to Servers asking for some kind of information. Upon receiving a Request, Servers send **Responses** back to the Client.

The **Internet** is the network between devices that allows clients and servers to exchange this information. **HTTP** is a set of rules for how this exchange of information happens. Clients and Servers adhere to these rules to ensure that they understand each other’s Requests and Responses.

**The Request and Response Cycle:**

* First a user gives a client a URL, the client builds a **request** for information (or resources) to be generated by a server. When the server receives that request, it uses the information included in the request to build a **response** that contains the requested information. Once built, that response is sent back to the client in the requested format, to be rendered to the user.

**Frontend vs. Backend:**

The front end is what users see and interact with and the backend is how everything works. Each side needs to communicate and operate effectively with the other as a single unit to improve the website’s functionality.

* The front end is the part of the website users can see and interact with such as the graphical user interface (GUI) and the command line including the design, navigating menus, texts, images, videos, etc. The backend, on the contrary, is part of the website users cannot see and interact with.
* The visual aspects of the website that can be seen and experienced by users are frontend. On the other hand, everything that happens in the background can be attributed to the backend.

**Front end Development:**

The part of a website that the user interacts with directly is termed the front end. It is also referred to as the ‘client side of the application. It includes everything that users experience directly: text colors and styles, images, graphs and tables, buttons, colors, and a navigation menu. HTML, CSS, and JavaScript are the languages used for Front End development. Responsiveness and performance are the two main objectives of the Front End. The developer must ensure that the site is responsive i.e. it appears correctly on devices of all sizes no part of the website should behave abnormally irrespective of the size of the screen.

**Front End Languages:**

The front-end portion is built by using some languages which are discussed below:

* [**HTML**](https://www.geeksforgeeks.org/html/)**:** HTML stands for Hypertext Markup Language. It is used to design the front-end portion of web pages using a markup language. HTML is a combination of Hypertext and Markup language. Hypertext defines the link between web pages.
* [**CSS**](https://www.geeksforgeeks.org/css/)**:** Cascading Style Sheets fondly referred to as CSS is a simply designed language intended to simplify the process of making web pages presentable. CSS allows you to apply styles to web pages
* [**JavaScript**](https://www.geeksforgeeks.org/javascript/)**:** JavaScript is a famous scripting language used to create magic on sites to make the site interactive for the user. It is used to enhance the functionality of a website to run cool games and web-based software. Applicable in both front-end and back-end.

### ****Front-End Frameworks and Libraries:****

* [**AngularJS**](https://www.geeksforgeeks.org/angularjs/)**:** AngularJs is a JavaScript open-source front-end framework that is mainly used to develop single-page web applications(SPAs). It is a continuously growing and expanding framework which provides better ways for developing web applications. It changes the static HTML to dynamic HTML. It is an open-source project which can be free. It extends HTML attributes with Directives, and data is bound with HTML.
* [**React.js**](https://www.geeksforgeeks.org/reactjs-tutorials/)**:** React is a declarative, efficient, and flexible JavaScript library for building user interfaces. ReactJS is an open-source, component-based front-end library responsible only for the view layer of the application. It is maintained by Facebook. Moreover, React Js makes Front-end development very easy.
* **Bootstrap:** Bootstrap is a free and open-source tool collection for creating responsive websites and web applications. It is the most popular HTML, CSS, and JavaScript framework for developing responsive, mobile-first websites.
* [**jQuery**](https://www.geeksforgeeks.org/jquery/)**:** jQuery is an open-source JavaScript library that simplifies the interactions between an HTML/CSS document, or more precisely the Document Object Model (DOM), and JavaScript. Elaborating on the terms, jQuery simplifies HTML document traversing and manipulation, browser event handling, DOM animations, Ajax interactions, and cross-browser JavaScript development.
* [**SASS**](https://www.geeksforgeeks.org/sass/)**:** It is the most reliable, mature, and robust CSS extension language. It is used to extend the functionality of an existing CSS of a site including everything from variables, inheritance, and nesting with ease.
* [**Flutter**](https://www.geeksforgeeks.org/flutter-tutorial/)**:**Flutter is an open-source UI development SDK managed by google. It is powered by the Dart programming language. It builds performant and good-looking natively compiled applications for mobile (Ios, Android), web, and desktop from a single code base. The key selling point of flutter is flat development is made easier, more expressive, and flexible with UI and native performance. In march 2021 flutter announce Flutter 2 which upgrades flutter to build release applications for the web, and the desktop is in beta state.
* Some other libraries and frameworks are Semantic-UI, Foundation, Materialize, Backbone.js, Ember.js, etc.

**Back End Development:**

* The backend is the server side of the website. It stores and arranges data, and also makes sure everything on the client side of the website works fine. It is part of the website that you cannot see and interact with. It is the portion of software that does not come in direct contact with the users. The parts and characteristics developed by backend designers are indirectly accessed by users through a front-end application. Activities, like writing APIs, creating libraries, and working with system components without user interfaces or even systems of scientific programming, are also included in the backend.

**Back End Languages**

The back-end portion is built by using some languages which are discussed below:

* [**PHP**](https://www.geeksforgeeks.org/php-tutorials/)**:** PHP is a server-side scripting language designed specifically for web development. Since PHP code is executed on the server side, it is called a server-side scripting language.
* [**C++**](https://www.geeksforgeeks.org/cpp-tutorial/)**:** It is a general-purpose programming language and is widely used nowadays for competitive programming. It is also used as a backend language.
* [**Java**](https://www.geeksforgeeks.org/java/)**:** Java is one of the most popular and widely used programming languages and platforms. It is highly scalable.
* [**Python**](https://www.geeksforgeeks.org/python-programming-language/)**:** Python is a programming language that lets you work quickly and integrate systems more efficiently.
* [**Node.js**](https://www.geeksforgeeks.org/nodejs/)**:** Node.js is an open-source and cross-platform runtime environment for executing JavaScript code outside a browser. You need to remember that NodeJS is not a framework, and it’s not a programming language. Most people are confused and understand it’s a framework or a programming language. We often use Node.js for building back-end services like APIs like Web App or Mobile App. It’s used in production by large companies such as Paypal, Uber, Netflix, Walmart, and so on.

**Back-End Frameworks:**

* [**Express**](https://www.geeksforgeeks.org/express-js/)**–**Express is a Nodejs framework used for backend/server-side development. It is used to build single-page, multi-page, and hybrid web applications. With its help, you can handle multiple different HTTP requests.
* [**Django**](https://www.geeksforgeeks.org/django-tutorial/)– Django is a Python web-based framework, following the model-template-views pattern. It is used to build large and complex web applications. Its features include being fast, secure, and scalable.
* [**Ruby on Rails**](https://www.geeksforgeeks.org/ruby-on-rails-introduction/)– Ruby on Rails is a server-side framework following the model-view-controller architecture pattern. It provides default structures such as web services, web pages, and databases.
* [**Laravel**](https://www.geeksforgeeks.org/laravel/)– Laravel is a web application framework for PHP and is robust. The feature which makes it perfect is reusing the components of different frameworks for creating a web application.
* [**Spring**](https://www.geeksforgeeks.org/introduction-to-spring-framework/)– This server-side framework provides infrastructure support for Java applications. It acts as a support to various frameworks like Hibernate, Struts, EJB, etc. It also has extensions that help in developing Java applications quickly and easily.
* Some more back-end programming/scripting languages are [C#](https://www.geeksforgeeks.org/csharp-programming-language/), [Ruby](https://www.geeksforgeeks.org/ruby-programming-language/), [GO](https://www.geeksforgeeks.org/go-programming-language-introduction/), etc.

**HTML:**

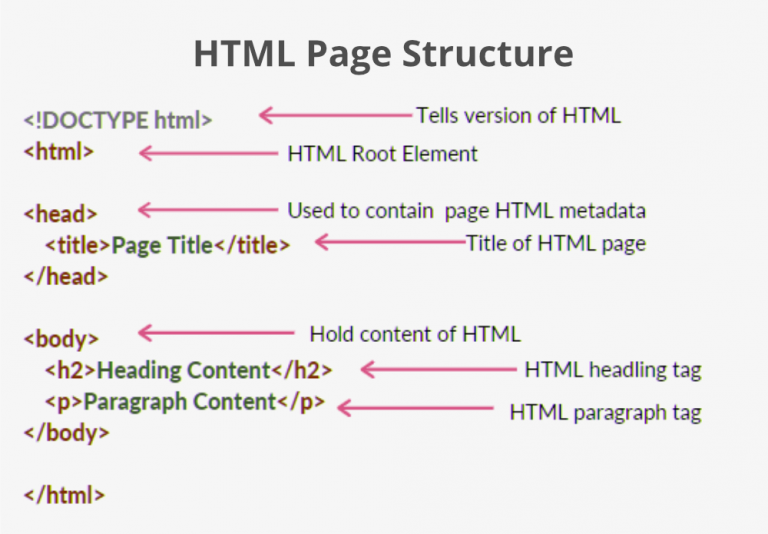
* HTML stands for **HyperText Markup Language**. It is used to design web pages using the markup language. HTML is the combination of Hypertext and Markup language. Hypertext defines the link between the web pages and markup language defines the text document within the tag that define the structure of web pages.

## What is HTML used for ?

**HTML is used to create the structure of web pages** that are displayed on the World Wide Web (www). **It contains Tags and Attributes that are used to design the web pages. Also, we can link multiple pages using Hyperlinks.**

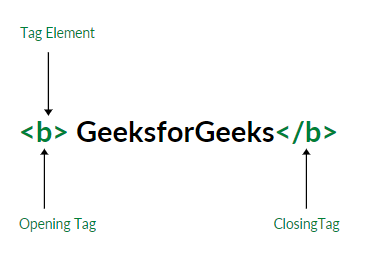
* HTML is a markup language used by the browser to manipulate text, images, and other content, in order to display it in the required format. HTML was created by Tim Berners-Lee in 1991. The first-ever version of HTML was HTML 1.0, but the first standard version was HTML 2.0, published in 1995.

**HTML page structure:**The basic structure of an HTML page is laid out below. It contains the essential building-block elements (i.e. doctype declaration, HTML, head, title, and body elements) upon which all web pages are created.



* [**<!DOCTYPE html>**](https://www.geeksforgeeks.org/html-doctypes/)**:** This is the document type declaration (not technically a tag). It declares a document as being an HTML document. The doctype declaration is not case-sensitive. A doctype or document type declaration is an instruction that tells the web browser about the markup language in which the current page is written. The Doctype is not an element or tag, it lets the browser know about the version of or standard of HTML or any other markup language that is being used in the document.
* [**<html>**](https://www.geeksforgeeks.org/html-html-tag/)**:** This is called the HTML root element. All other elements are contained within it.
* [**<head>**](https://www.geeksforgeeks.org/html-head-tag/#:~:text=The%20tag%20in%20HTML,head%3E%20element%20can%20be%20omitted.)**:** The head tag contains the “behind the scenes” elements for a webpage. Elements within the head aren’t visible on the front-end of a webpage. HTML elements used inside the <head> element include:
* [**<style>**](https://www.geeksforgeeks.org/html-style-tag/) **-**This html tag allows us to insert styling into our webpages and make them appealing to look at with the help of CSS.
* [**<title>**](https://www.geeksforgeeks.org/html-title-tag/) **-**The title is what is displayed on the top of your browser when you visit a website and contains the title of the webpage that you are viewing.
  + [**<base>**](https://www.geeksforgeeks.org/html-base-tag/) **-** It specifies the base URL for all relative URL’s in a document.
  + [**<noscript>**](https://www.geeksforgeeks.org/html-noscript-tag/) **-** Defines a section of HTML that is inserted when the scripting has been turned off in the users browser.
  + [**<script>**](https://www.geeksforgeeks.org/html-script-tag/) **-** This tag is used to add functionality in the website with the help of JavaScript.
  + [**<meta>**](https://www.geeksforgeeks.org/html-meta-tag/#:~:text=The%20tag%20in%20HTML,keywords%2C%20document%20author%2C%20etc.) **-**This tag encloses the meta data of the website that must be loaded every time the website is visited. For eg:- the metadata charset allows you to use the standard UTF-8 encoding in your website. This in turn allows the users to view your webpage in the language of their choice. It is a self closing tag.
  + [**<link>**](https://www.geeksforgeeks.org/html-link-tag/) **-** The ‘link’ tag is used to tie together HTML, CSS, and JavaScript. It is self closing.
* [**<body>**](https://www.geeksforgeeks.org/html-body-tag/#:~:text=The%20tag%20in%20HTML,well%20as%20an%20ending%20tag.)**:** The body tag is used to enclose all the visible content of a webpage. In other words, the body content is what the browser will show on the front-end.

**Elements and Tags:** HTML uses predefined [**tags**](https://www.geeksforgeeks.org/html-html-tag/#:~:text=The%20tag%20in%20HTML,DOCTYPE%3E%20tag.)and [**elements**](https://www.geeksforgeeks.org/html-elements/) which tell the browser how to properly display the content.



An HTML element is a collection of start and end tags with the content inserted in between them.

**Syntax:**

<tagname > Contents... </tagname>

#### HTML Element: The HTML element consists of 3 parts.

* **Opening tag**: It is used to tell the browser where the content material starts.
* **Closing tag**: It is used to tell the browser where the content material ends.
* **Content**: It is the actual content material inside the opening and closing tag.

**Nested HTML Elements:** The HTML element used inside another HTML Element is called nested HTML element.

**Tags:**

1. **Heading Tag:**

An HTML heading tag is used to define the headings of a page. There are six levels of headings defined by HTML. These 6 heading elements are h1, h2, h3, h4, h5, and h6; with h1 being the highest level and h6 being the least.

* **<h1>** is used for the main heading. (Biggest in size)
* **<h2>** is used for subheadings, if there are further sections under the subheadings then the**<h3>** elements are used.
* **<h6>** for the small heading (smallest one).

1. **Paragraph tag:**

The **<p>** tag in HTML defines a paragraph. These have both opening and closing tags. So anything mentioned within **<p>** and **</p>** is treated as a paragraph. Most browsers read a line as a paragraph even if we don’t use the closing tag i.e, </p>, but this may raise unexpected results. So, it is a good convention, and we **must** use the closing tag.

1. **List Tags:**

A list is a record of short pieces of related information or used to display the data or any information on web pages in the ordered or unordered form.

**Supported Tags:**These tags are used in HTML listing.

* [HTML <ul> Tag](https://www.geeksforgeeks.org/html-ul-tag/)
* [HTML <ol> Tag](https://www.geeksforgeeks.org/html-ol-tag/)
* [HTML <dl> Tag](https://www.geeksforgeeks.org/html-dl-tag/)

**i) The HTML Unordered List:**An unordered list starts with the “ul” tag. Each list item starts with the “[li](https://www.geeksforgeeks.org/html-li-tag/)” tag. The list items are marked with bullets i.e small black circles by default.

**Syntax:**

<ul>

<li>Item1</li>

<li>Item2</li>

<li>Item3</li>

</ul>

**attribute:** This tag contains two attributes which are listed below:

* [**compact**](https://www.geeksforgeeks.org/html-ul-compact-attribute/#:~:text=The%20HTML%20%7C%20compact,It%20is%20a%20Boolean%20attribute.)**:** It will render the list smaller.It is used to define the list should be smaller than the normal by reducing the space between the list items and the indentation of the list. It is a Boolean attribute.
* [**type**](https://www.geeksforgeeks.org/html-ul-type-attribute/)**:** It specifies which kind of marker is used in the list.

**ii) HTML Ordered List:**An ordered list starts with the “ol” tag. Each list item starts with the “li” tag. The list items are marked with numbers by default.

**Syntax:**

<ol>

<li>Item1</li>

<li>Item2</li>

<li>Item3</li>

</ol>

**Attributes:**

* [**compact**](https://www.geeksforgeeks.org/html-ol-compact-attribute/)**:** It defines the list should be compacted (compact attribute is not supported in HTML5. Use CSS instead.).
* [**reversed**](https://www.geeksforgeeks.org/html-ol-reversed-attribute/)**:** It defines that the order will be descending.
* [**start**](https://www.geeksforgeeks.org/html-ol-start-attribute/)**:** It defines from which number or alphabet the order will start.
* [**type**](https://www.geeksforgeeks.org/html-ol-type-attribute/)**:** It defines which type(1, A, a, I, and i) of the order you want in your list of numeric, alphabetic, or roman numbers.

**iii) HTML Description List:**A description list is a list of terms, with a description of each term. The [<dl>](https://www.geeksforgeeks.org/html-dl-tag/) tag defines the description list, the <dt> tag defines the term name, and the <dd> tag describes each term.

**Syntax:**

<dl> Contents... </dl>

1. **Anchor Tag:**

It is a connection from one web resource to another. A link has two ends, An anchor and direction. The link starts at the “source” anchor and points to the “destination” anchor, which may be any Web resource such as an image, a video clip, a sound bite, a program, an HTML document or an element within an HTML document.

**Syntax:**

<a href=”url”>Text Link</a>

**Syntax Explanation:**

* **href :** The href attribute is used to specify the destination address of the link used. "href" stands for Hypertext reference.
* **Text link :** The text link is the visible part of the link. It is what the viewer clicks on.

**Internal Links**

An internal link is a type of hyperlink whose target or destination is a resource, such as an image or document, on the same website or domain.

**Self closing tags:**

* A self-closing tag in HTML is a kind of HTML tag that does not need to be closed manually by its closing tag, which means it does not have a separate closing tag as </tag>. Some few self-closing tags are <input/>, <hr/>, <br/>, <img/>, etc.
* Self-closing tags in HTML are sometimes also known as empty tags, void tags, singletons tags, etc. This means that these tags have no content and cannot have any children.

**Closing tags vs. self-closing tags**

|  |  |
| --- | --- |
| **Closing tags** | **Self-closing tags** |
| Closing tags necessitate the use of a closing tag. For example, <p> tag always need </p> | Self-closing tags do not necessitate a closing tag. For example, <input> is a self-closing tag. |
| Closing tags include content. For example, <h1>This is a heading </h1> | Self-closing tags do not contain any content. They are either empty or receive data from attributes to render. |
| Child elements exist in closing tags. | Child elements do not exist in self-closing tags. |

1. **Image Tag:**

There are 2 ways to insert the images into a webpage:

* By providing a full path or address (URL) to access an internet file.
* By providing the file path relative to the location of the current web page file.

The <img> tag is used to add or embed the images to a webpage/website. The “img” tag is an empty tag, which means it can contain only a list of attributes and it has no closing tag.

**Syntax:**

<img src="url" alt="some\_text" width="" height="">

**Attribute:**The ***<img>***tag has following attributes:

* [**src**](https://www.geeksforgeeks.org/html-img-src-attribute/)**:** It is used to specify the path to the image.
* [**alt**](https://www.geeksforgeeks.org/html-img-alt-attribute/)**:** It is used to specify an alternate text for the image. It is useful as it informs the user about what the image means and also due to any network issue if the image cannot be displayed then this alternate text will be displayed.
* [**crossorigin**](https://www.geeksforgeeks.org/html-link-crossorigin-attribute/)**:** It is used to import images from third-party sites that allow cross-origin access to be used with canvas.
* [**height**](https://www.geeksforgeeks.org/html-img-height-attribute/)**:** It is used to specify the height of the image.
* [**width**](https://www.geeksforgeeks.org/html-img-width-attribute/)**:**It is used to specify the width of the image.
* [**ismap**](https://www.geeksforgeeks.org/html-img-ismap-attribute/)**:**It is used to specify an image as a server-side image map.
* [**loading**](https://www.geeksforgeeks.org/html-img-loading-attribute/)**:** It is used to specify whether a browser should defer the loading of images until some conditions are met or load an image immediately.
* [**longdesc**](https://www.geeksforgeeks.org/html-img-longdesc-attribute/)**:**It is used to specify a URL to a detailed description of an image.
* [**referrerpolicy**](https://www.geeksforgeeks.org/html-img-referrerpolicy-attribute/): It is used to specify which referrer information to use when fetching an image i.e. no-referrer, no-referrer-when-downgrade, origin, origin-when-cross-origin, unsafe-url.
* [**sizes**](https://www.geeksforgeeks.org/html-size-attribute/)**:**It is used to specify image sizes for different page layouts.
* [**srcset**](https://www.geeksforgeeks.org/html-source-srcset-attribute/)**:**It is used to specify a list of image files to use in different situations.
* [**usemap**](https://www.geeksforgeeks.org/html-img-usemap-attribute/)**:**It is used to specify an image as a client-side image map.

**Adding Image as a Link:**An image can work as a link with a URL embedded in it. It can be done by using the “img” tag inside an “a” tag. We need to specify the file path in order to render the image on the webpage. File paths are used to link external resources such as images, videos, style sheets, JavaScript, displaying other web pages, etc. To insert a file on a web page its source must be known.

File paths are of two types:

* **Absolute File Paths**: It always contains the root element along with the complete directory list required to locate the file.
* **Relative File Paths:** It is the hierarchical path representation that locates the file or folder on a file system beginning from the current directory.

**Example:**

<a href="<https://ide.geeksforgeeks.org/tryit.php>">

    <img src=

"<https://media.geeksforgeeks.org/wp-content/cdn-uploads/20190710102234/download3.png>"

         alt="GeeksforGeeks logo" />

 </a>

**Block Elements vs. Inline Elements:**

**Block elements:**They consume the entire width available irrespective of their sufficiency. They always start in a new line and have top and bottom margins. It does not contain any other elements next to it.

**Examples of Block elements:**

* [**<h1>-<h6>**](https://www.geeksforgeeks.org/html-heading/)**:**This element is used for including headings of different sizes ranging from 1 to 6.
* [**<div>**](https://www.geeksforgeeks.org/div-tag-html/)**:**This is a container tag and is used to make separate divisions of content on the web page.
* [**<hr>**](https://www.geeksforgeeks.org/html-hr-size-attribute/)**:**This is an empty tag and is used for separating content by horizontal lines.
* [**<li>**](https://www.geeksforgeeks.org/html-li-tag/)**:**This tag is used for including list items of an ordered or unordered list.
* [**<ul>**](https://www.geeksforgeeks.org/html-ul-compact-attribute/)**:**This tag is used to make an unordered list.
* [**<ol>**](https://www.geeksforgeeks.org/html-ol-compact-attribute/)**:**This tag is used to make an ordered list.
* [**<p>**](https://www.geeksforgeeks.org/html-basics/)**:**This tag is used to include paragraphs of content in the webpage.
* [**<table>**](https://www.geeksforgeeks.org/html-table-border-attribute/)**:**This tag is used for including the tables in the webpage when there is a need for tabular data.

**HTML 5 Semantic block elements:**

* [**<header>**](https://www.geeksforgeeks.org/html5-semantics/)**:**This tag is used for including all the main things of the webpage like navbar, logos, and heading of the webpage.
* [**<nav>**](https://www.geeksforgeeks.org/html5-semantics/)**:**This tag helps to navigate through different sections by including different blocks of hyperlinks in the webpage.
* [**<footer>**](https://www.geeksforgeeks.org/html5-semantics/)**:**This contains all information about the authorization, contact, and copyright details of the webpage.
* [**<main>**](https://www.geeksforgeeks.org/html5-semantics/)**:**The main content of the webpage resides in this tag.
* [**<section>**](https://www.geeksforgeeks.org/html5-semantics/)**:**This is used separate different sections in the webpage.
* [**<article>**](https://www.geeksforgeeks.org/html5-semantics/)**:**This tag is used to include different independent articles on the webpage.
* [**<aside>**](https://www.geeksforgeeks.org/html5-semantics/)**:**This tag is used to mention details of the main content aside.

**Inline elements:** Inline elements occupy only enough width that is sufficient to it and allows other elements next to it which are inline. Inline elements don’t start from a new line and don’t have top and bottom margins as block elements have.

Examples of**Inline elements**:

* [**<a>**](https://www.geeksforgeeks.org/html-links/)**:**This tag is used for including hyperlinks in the webpage.
* [**<br>**](https://www.geeksforgeeks.org/html-brgt-tag/)**:**This tag is used for mentioning line breaks in the webpage wherever needed.
* [**<script>**](https://www.geeksforgeeks.org/html-script-tag/)**:**This tag is used for including external and internal JavaScript codes.
* [**<input>**](https://www.geeksforgeeks.org/html-input-tag/)**:**This tag is used for taking input from the users and is mainly used in forms.
* [**<img>**](https://www.geeksforgeeks.org/html-img-tag/)**:**This tag is used for including different images in the webpage to add beauty to the webpage.
* [**<span>**](https://www.geeksforgeeks.org/span-tag-html/)**:** This is an inline container that takes necessary space only.
* [**<b>**](https://www.html.am/html-codes/text/html-bold.cfm)**:** This tag is used in places where bold text is needed.
* [**<label>**](https://www.geeksforgeeks.org/html-label-tag/)**:**The tag in HTML is used to provide a usability improvement for mouse users i.e, if a user clicks on the text within the <label> element, it toggles the control.

**Difference between Inline and Block elements:**

| **Inline Elements** | **Block Elements** |
| --- | --- |
| Inline elements occupy only sufficient width required. | Block Elements occupy the full width irrespective of their sufficiency. |
| Inline elements don’t start in a new line. | Block elements always start in a line. |
| Inline elements allow other inline elements to sit behind. | Block elements doesn’t allow other elements to sit behind |
| Inline elements don’t have top and bottom margin | Block elements have top and bottom margin. |

1. **Hr Tag:**

The <hr> tag in HTML stands for horizontal rule and is used to insert a horizontal rule or a thematic break in an HTML page to divide or separate document sections. The <hr> tag is an empty tag, and it does not require an end tag.

1. **Br Tag:**

The browser does not recognize new lines and paragraph formatting in the text. If you want to start a new line, you need to insert a line break with the help of the <br>. The <br> tag inserts a single carriage return or breaks in the document. This element has no end tag.

**Attributes:**This tag accepts an attribute called **clear**

* **clear:**Which indicates where to begin the next line after the break.

# Subscript and Superscript Tags:

**Subscript:** The <sub> tag is used to add a subscript text to the HTML document. The <sub> tag defines the subscript text. Subscript text appears half a character below the normal line and is sometimes rendered in a smaller font. Subscript text can be used for chemical formulas, like H2O to be written as H2O.  
**Superscript:** The <sup> tag is used to add a superscript text to the HTML document. The <sup> tag defines the superscript text. Superscript text appears half a character above the normal line and is sometimes rendered in a smaller font. Superscript text can be used for footnotes.

**HTML Entities:**

HTML provides some method to display reserved characters. Reserved characters are those characters that are either reserved for HTML or those which are not present in the basic keyboard. For instance, ‘<‘ is reserved in HTML language. Sometimes this character needs to display on the web page which creates ambiguity in code. Along with these are the characters which are normally not present in basic keyboard ( £, ¥, €, © ), etc. HTML provides some Entity names and Entity numbers to use these symbols. Entity names are case-sensitive. Entity number is easy to learn.

**Syntax:**

&entity\_name; or &#entity\_number;

**Note:**

**Entity names are case sensitive**

**Examples:**

**For < , entity is &lt;**

**For >, entity is &gt;**

**For &, entity is &amp;**

**For +, entity is &plus;**

**HTML Semantics:**

* HTML tags are classified in two types.   
    Semantic

Non-Semantic

**Semantic Elements:** Semantic elements have meaningful names which tell about type of content. For example header, footer, table, etc. HTML5 introduces many semantic elements as mentioned below which make the code easier to write and understand for the developer as well as instructs the browser on how to treat them. 

* article
* aside
* details
* figcaption
* figure
* footer
* header
* main
* mark
* nav
* section

1. **Article:** It contains independent content which doesnt require any other context.   
   Example: Blog Post, Newspaper Article etc.
2. **Aside:** It is used to place content in a sidebar i.e. aside the existing content. It is related to surrounding content.
3. **Details and Summary:** “details” defines additional details that the user can hide or view. “summary” defines a visible heading for a “details” element.
4. **Figure and Figcaption:** These are used to add an image in a web page with small description.
5. **Header:** As the name suggests, it is for the header of a section introductory of a page. There can be multiple headers on a page.
6. **Footer:** Footer located at the bottom of any article or document, they can contain contact details, copyright information etc. There can be multiple footers on a page.
7. **Main:** It defines the main content of the document. The content inside main tag should be unique.
8. **Mark:** It is used to highlight the text.
9. **Section:** A page can be split into sections like Introduction, Contact Information, Details etc and each of these sections can be in a different section tag.
10. **nav:** It is used to define a set of navigation links in the form of navigation bar or nav menu.

**Non-Semantic Elements:** Tags like div, span fall under the Non- Semantic categories as their names don’t tell anything about what kind of content is present inside them.  
 **1) div:** It is a block level element or division of a section. It is used as a container.

**2)** **span:** It is an inline element which doesn’t start on a new line and takes up only the necessary width.

**HTML Tables:**

An HTML table is defined with the “table” tag. Each table row is defined with the “tr” tag. A table header is defined with the “th” tag. By default, table headings are bold and centered. A table data/cell is defined with the “td” tag.

**Adding Cells that span many rows in HTML Tables:**To make a cell span more than one row, we must use the rowspan attribute.

**Adding Cells that Span Many Columns in HTML Tables:** To make a cell span more than one column, we must use the colspan attribute.

**thead:**

The <thead> element is used in conjunction with the [<tbody>](https://www.w3schools.com/tags/tag_tbody.asp) and [<tfoot>](https://www.w3schools.com/tags/tag_tfoot.asp) elements to specify each part of a table (header, body, footer).

Browsers can use these elements to enable scrolling of the table body independently of the header and footer. Also, when printing a large table that spans multiple pages, these elements can enable the table header and footer to be printed at the top and bottom of each page.

**Note:** The <thead> element must have one or more [<tr>](https://www.w3schools.com/tags/tag_tr.asp) tags inside.

**tbody:**

The <tbody> tag is used to group the body content in an HTML table.

**Note:** The <tbody> element must have one or more [<tr>](https://www.w3schools.com/tags/tag_tr.asp) tags inside.

**tfoot:**

The <tfoot> tag is used to group footer content in an HTML table.

**HTML Forms:**

<form> is an HTML element to collect input data containing interactive controls. It provides facilities to input text, number, values, email, password, and control fields such as checkboxes, radio buttons, submit buttons, etc., or in other words, form is a container that contains input elements like text, email, number, radio buttons, checkboxes, submit buttons, etc. Forms are generally used when you want to collect data from the user. For example, a user wants to buy a bag online, so he/she has to first enter their shipping address in the address form and then add their payment details in the payment form to place an order.

**Form elements**

These are the following HTML <form> elements:

* **<label>:**It defines label for <form> elements.
* **<input>:**It is used to get input data from the form in various types such as text, password, email, etc by changing its type.
* **<button>:**It defines a clickable button to control other elements or execute a functionality.
* **<select>:**It is used to create a drop-down list.
* **<textarea>:**It is used to get input long text content.
* **<fieldset>:**It is used to draw a box around other form elements and group the related data.
* **<legend>:**It defines a caption for fieldset elements.
* **<datalist>:**It is used to specify pre-defined list options for input controls.
* **<output>:**It displays the output of performed calculations.
* **<option>:**It is used to define options in a drop-down list.
* **<optgroup>:**It is used to define group-related options in a drop-down list.

**Textbox in HTML Form**

In an HTML form,we use the <input> tag by assigning type attribute value to text to input single line input. To define type attribute see the below syntax.

Tip: The default value of the type attribute is “text”.

**Syntax:**

<input type="text" />

Or shorthand for “text” type:

<input />

**Password in an HTML Form**

We can change the type value text to password to get the input password

**Radio Button in an HTML Form**

To create a radio button, we use the <input> tag following by radio type to provide users to choose a limited number of choices.

**Syntax:**

<input type="radio" name="radio\_button\_name" value="radio\_button\_value" />

**Note:**The radio button must have shared the same name to be treated as a group.

**Note:** The value attribute defines the unique value associated with each radio button. The value is not shown to the user, but is the value that is sent to the server on “submit” to identify which radio button that was selected.

**Example:**In this example, we will create a radio button to choose your gender.

<!DOCTYPE html>

<html>

<head>

<title>Page Title</title>

</head>

<body>

<h2>Select your gender</h2>

<form>

<label>Male<input type="radio" name="gender" value="male" />

</label>

<label>Female<input type="radio" name="gender" value="female" />

</label>

</form>

</body>

</html>

**Checkbox in an HTML Form**

To create a checkbox in an HTML form, we use the <input> tag followed by the input type checkbox. It is a square box to tick to activate this. It used to choose more options at a time.

**Syntax:**

<input type="checkbox" name="select\_box\_name" value="select\_box\_value" />

**Note:** the “name” and “value” attributes are used to send the checkbox data to the server.

**Note:**checkbox elements must have shared the same name to be treated as a group.

**Combobox in an HTML Form**

Combobox is used to create a drop-down menu in your form which contains multiple options. So, to create a Combobox in an HTML form, we use the <select> tag with <option> tag. It is also known as a drop-down menu.

**Syntax:**

<select name="select\_box\_name">

<option value="value1">option1</option>

<option value="value2">option2</option>

<option value="value3">option3</option>

</select>

**Note:** the “name” and “value” attributes are used to send the Combobox data to the server.

**Submit button in an HTML Form**

In the HTML form, submit button is used to submit the details of the form to the form handler. A form handler is a file on the server with a script that is used to process input data.

**Syntax:**

<button type="submit">submit</button>

Note: By default, type of button is submit. To make a button, just behave like a button make type=”button.

**Syntax:**

<button type="button">submit</button>

**TextArea in an HTML Form**

In the HTML form, a text area is used to add comments or reviews, or addresses to the form, in other words, the text area is a multi-line text input control. It contains an unlimited number of characters, the text renders in a fixed-width font, and the size of the text area is given by the <rows> and <cols> attributes. To create a text area in the form use the <textarea> tag.

**Syntax:**

<textarea name="textarea\_name">content</textarea>

**Note:** the name attribute is used to reference the textarea data after it is sent to a server.

## Attributes associated with Text area:

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Value** | **Description** |
| [autofocus](https://www.w3schools.com/tags/att_textarea_autofocus.asp) | autofocus | Specifies that a text area should automatically get focus when the page loads |
| [cols](https://www.w3schools.com/tags/att_textarea_cols.asp) | *number* | Specifies the visible width of a text area |
| [dirname](https://www.w3schools.com/tags/att_textarea_dirname.asp) | *textareaname*.dir | Specifies that the text direction of the textarea will be submitted |
| [disabled](https://www.w3schools.com/tags/att_textarea_disabled.asp) | disabled | Specifies that a text area should be disabled |
| [form](https://www.w3schools.com/tags/att_textarea_form.asp) | *form\_id* | Specifies which form the text area belongs to |
| [maxlength](https://www.w3schools.com/tags/att_textarea_maxlength.asp) | *number* | Specifies the maximum number of characters allowed in the text area |
| [name](https://www.w3schools.com/tags/att_textarea_name.asp) | *text* | Specifies a name for a text area |
| [placeholder](https://www.w3schools.com/tags/att_textarea_placeholder.asp) | *Text* | Specifies a short hint that describes the expected value of a text area |
| [readonly](https://www.w3schools.com/tags/att_textarea_readonly.asp) | readonly | Specifies that a text area should be read-only |
| [required](https://www.w3schools.com/tags/att_textarea_required.asp) | required | Specifies that a text area is required/must be filled out |
| [rows](https://www.w3schools.com/tags/att_textarea_rows.asp) | *number* | Specifies the visible number of lines in a text area |
| [wrap](https://www.w3schools.com/tags/att_textarea_wrap.asp) | hard soft | Specifies how the text in a text area is to be wrapped when submitted in a form |

**HTML Attributes:**

 All HTML elements have attributes that will provide additional information about that particular element. It takes 2 parameters, ie, a *name*& a *value*which define the properties of the element and are placed inside the element tag.

**Points to remember for attributes:**

* Attributes always come in name/value pairs like this: attribute\_name=”value”.
* Attributes are always added to the start tag of an HTML element.
* Attribute values should always be enclosed in quotes. Double style quotes (“ ”) are the most common, but single style quotes (‘ ’) are also allowed.
* In some rare situations, like when the attribute value itself contains quotes, it is necessary to use single quotes: name=’John “ShotGun” Nelson’ and vice-versa.

[**HTML id Attribute**](https://www.geeksforgeeks.org/html-id-attributes/)**:**This attribute is used to provide a unique identification to an element. Situations may arise when we will need to access a particular element that may have a similar name as the others. In that case, we provide different id’s to various elements so that they can be uniquely accessed.

**HTML placeholder Attribute:**

The placeholder attribute specifies a short hint that describes the expected value of an input field (e.g. a sample value or a short description of the expected format).

The short hint is displayed in the input field before the user enters a value.

**Note:** The placeholder attribute works with the following input types: text, search, url, tel, email, and password.

**HTML for attribute:**

When used together with the <label> element, the for attribute specifies which form element a label is bound to.

**HTML Tags vs. Elements vs. Attributes:**

|  |  |  |
| --- | --- | --- |
| **HTML Tags** | **HTML Elements** | **HTML Attributes** |
| HTML tags are used to hold the HTML element. | HTML element holds the content. | HTML attributes are used to describe the characteristic of an HTML element in detail. |
| HTML tag starts with < and ends with > | Whatever written within a HTML tag are HTML elements. | HTML attributes are found only in the starting tag. |
| HTML tags are almost like keywords where every single tag has unique meaning. | HTML elements specifies the general content. | HTML attributes specify various additional properties to the existing HTML element. |