Html

# HTML: HyperText Markup Language

**HTML** (HyperText Markup Language) is the most basic building block of the Web. It defines the meaning and structure of web content.

Other technologies besides HTML are generally used to describe a web page's appearance/presentation ([CSS](https://developer.mozilla.org/en-US/docs/Web/CSS)) or functionality/behavior ([JavaScript](https://developer.mozilla.org/en-US/docs/Web/JavaScript)).

"Hypertext" refers to links that connect web pages to one another, either within a single website or between websites. Links are a fundamental aspect of the Web

An HTML element is set off from other text in a document by "tags", which consist of the element name surrounded by "<" and ">". The name of an element inside a tag is case-insensitive.

Internet is managed by icaan-> internet corporation for assigned names and numbers.

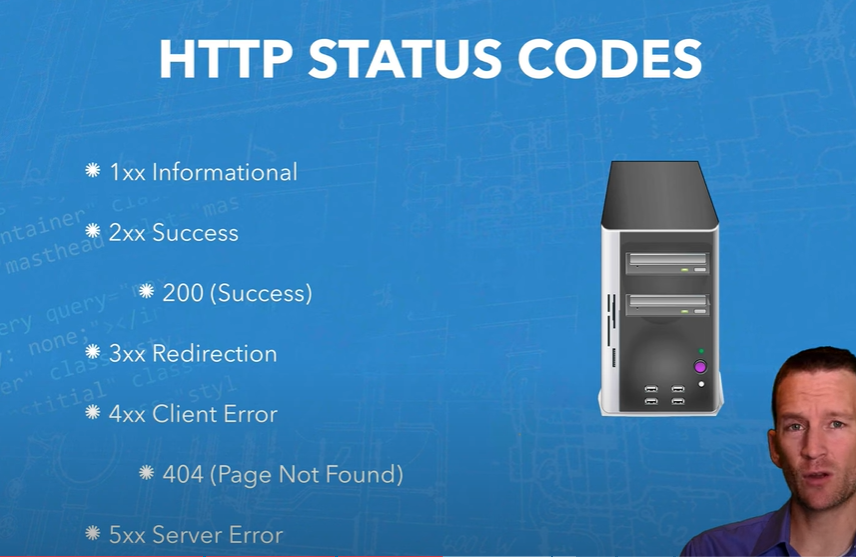
How does the internet work?

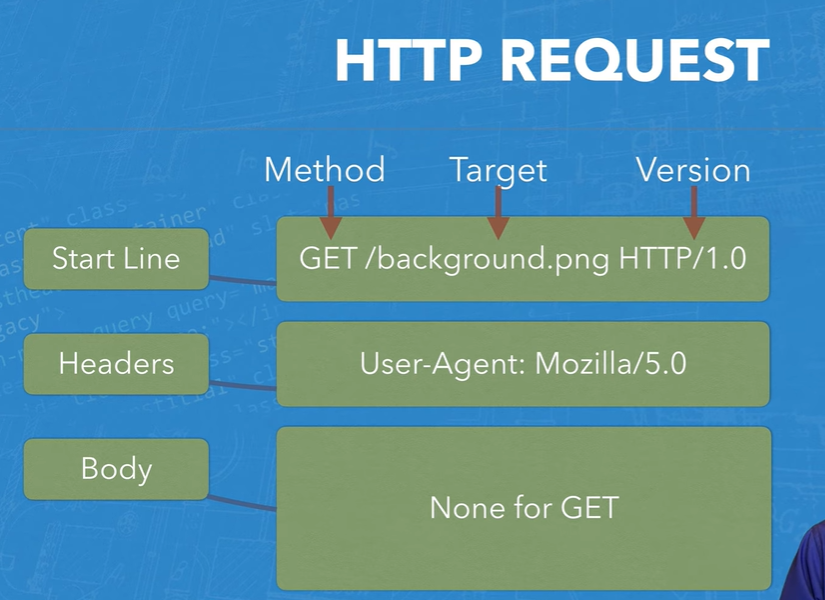
A network switch is a device that connects multiple devices to a local area network (LAN) using a single cable or wire. Network switches are used to connect two or more networks together.

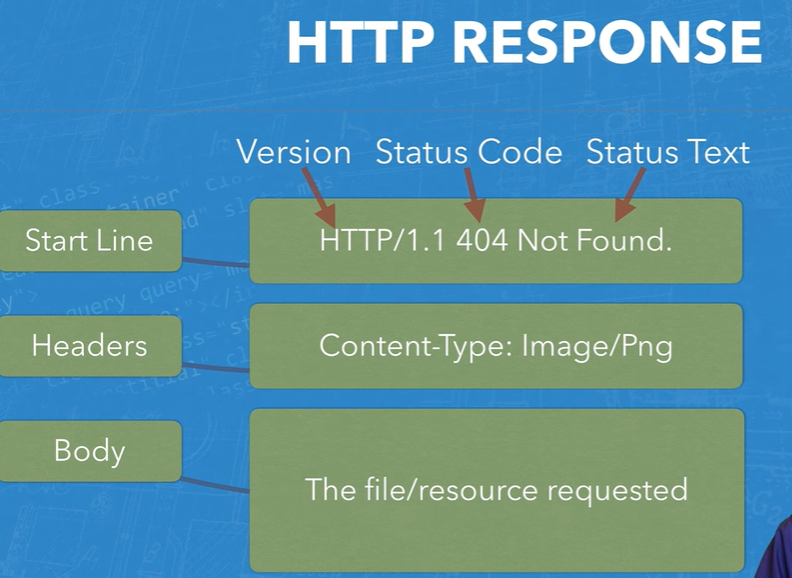


* Cat5 and cat6 cables are used
* CAT 5 is 100 MHz with data transfer at speeds up to 1000 Mbps. CAT 6 is 250 MHz and can get up to 1 Gbps.
* Cat5 and Cat6 cables are Ethernet cables used to connect devices to modems, switches, servers, and network equipment. They are also called network, LAN, or Ethernet cables.
* Cat6 has plastic ribbon in between them.
* Internet is given by ISP internet service provider.
* Submerine cables are fiber optic cables.which transgers the data in the speed of light.









Hypertext Transfer Protocol (HTTP) is a protocol that allows web clients and web servers to communicate with each other. It is the foundation of the World Wide Web and is used to load webpages using hypertext links.



# Basic Networking Commands in PowerShell for Windows:

# 1. Ping

The ping command is a basic yet essential networking command used to test the reachability of a host on an Internet Protocol (IP) network.

# ping google.com

**2. ipconfig**

ipconfig is a handy command that displays all current TCP/IP network configuration values and refreshes Dynamic Host Configuration Protocol (DHCP) and Domain Name System (DNS) settings. It's like asking your computer to spill the beans about its network setup.

Here’s how you can use it:powershellCopy code

ipconfig

#### 5. Hostname

It will show us the hostname of the machine.

C:\Users\user>hostname

Abhishek

#### 6. getmac

It will give you the MAC address of the network interface. People might use this to control which device can connect to the network. Each device has a unique MAC address and it is assigned by the manufacturer, store in the device hardware.

C:\Users\user>getmac

Physical Address Transport Name

=================== ==========================================================

78-AF-08-7D-3B-60 \Device\Tcpip\_{0CC8C576-34EB-446D-83DB-241D11747E39}

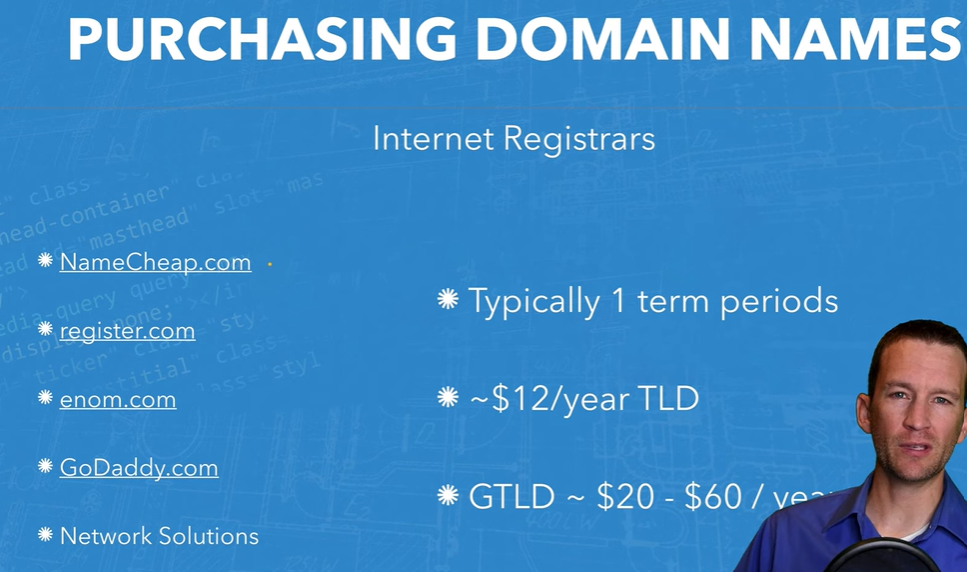
78-AF-08-7D-3B-64 Media disconnected

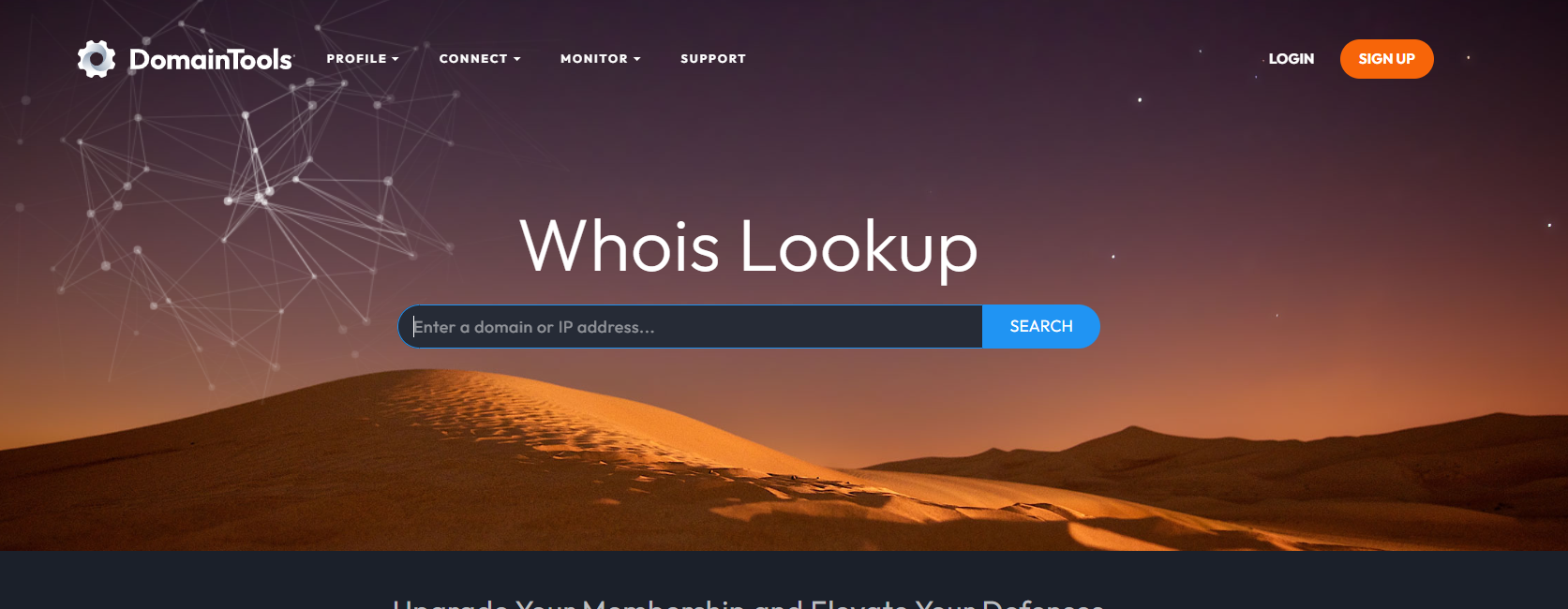
#### 8. nslookup

It stands for name server lookup. When we want to know the IP address of the domain we can use this command. Fact is if we run this command over and over, we will get different IP addresses for a website like google, yahoo, Flipkart because these domains have spread to different machines. You can see in the below image IP address is different for the same domain name.  
the first IP address is: 13.227.188.78  
the second IP address is: 184.31.213.84

#### 9. netstat

It is used for network statics, diagnostics, and analysis. If we are managing a huge college campus network, then this tool is useful because it provides an advanced aspect of the network.





The above website is used to check the domain names .

Whois database the hides the details of the person who is registered the domain name in the server.

# How do you set up a local testing server?

## [Local files vs. remote files](https://developer.mozilla.org/en-US/docs/Learn/Common_questions/Tools_and_setup/set_up_a_local_testing_server#local_files_vs._remote_files)

Throughout most of the learning area, we tell you to just open your examples directly in a browser — this can be done by double-clicking the HTML file, dragging and dropping it into the browser window, or choosing File > Open… and navigating to the HTML file. There are many ways to achieve this.

If the web address path starts with file:// followed by the path to the file on your local hard drive, a local file is being used. In contrast, if you view one of our examples hosted on GitHub (or an example on some other remote server), the web address will start with http:// or https://, to show that the file has been received via HTTP.

## **Using Git with Command Line**

To start using Git, we are first going to open up our Command shell

### **Example**

git --version

git version 2.30.2.windows.1

## **Configure Git**

Now let Git know who you are. This is important for version control systems, as each Git commit uses this information:

### **Example**

git config --global user.name "w3schools-test"

git config --global user.email "test@w3schools.com"

## **Creating Git Folder**

Now, let's create a new folder for our project:

### **Example**

mkdir myproject

cd myproject

## **Initialize Git**

Once you have navigated to the correct folder, you can initialize Git on that folder:

### **Example**

git init

Initialized empty Git repository in /Users/user/myproject/.git/

git status

On branch master

No commits yet

Untracked files:

  (use "git add ..." to include in what will be committed)     index.html nothing added to commit but untracked files present (use "git add" to track)

**Staged** files are files that are ready to be **committed** to the repository you are working on. You will learn more about commit shortly.

For now, we are done working with index.html. So we can add it to the Staging Environment:

### **Example**

git add index.html

## **Git Commit**

Since we have finished our work, we are ready move from stage to commit for our repo.

Adding commits keep track of our progress and changes as we work. Git considers each commit change point or "save point". It is a point in the project you can go back to if you find a bug, or want to make a change.

When we commit, we should **always** include a **message**.

By adding clear messages to each commit, it is easy for yourself (and others) to see what has changed and when.

### **Example**

git commit -m "First release of Hello World!"

[master (root-commit) 221ec6e] First release of Hello World!

3 files changed, 26 insertions(+)

create mode 100644 README.md

create mode 100644 bluestyle.css

create mode 100644 index.html

The commit command performs a commit, and the -m "message" adds a message.

The Staging Environment has been committed to our repo, with the message:  
"First release of Hello World!"

## **Git Commit without Stage**

Sometimes, when you make small changes, using the staging environment seems like a waste of time. It is possible to commit changes directly, skipping the staging environment. The -a option will automatically stage every changed, already tracked file.

git status --short

M index.html

**Note:** Short status flags are:

* ?? - Untracked files
* A - Files added to stage
* M - Modified files
* D - Deleted files

We see the file we expected is modified. So let's commit it directly:

### **Example**

git commit -a -m "Updated index.html with a new line"

[master 09f4acd] Updated index.html with a new line

1 file changed, 1 insertion(+)

## **Git Commit Log**

To view the history of commits for a repository, you can use the log command:

### **Example**

git log

commit 09f4acd3f8836b7f6fc44ad9e012f82faf861803 (HEAD -> master)

Author: w3schools-test

Date: Fri Mar 26 09:35:54 2021 +0100

Updated index.html with a new line

commit 221ec6e10aeedbfd02b85264087cd9adc18e4b26

Author: w3schools-test

Date: Fri Mar 26 09:13:07 2021 +0100

First release of Hello World!

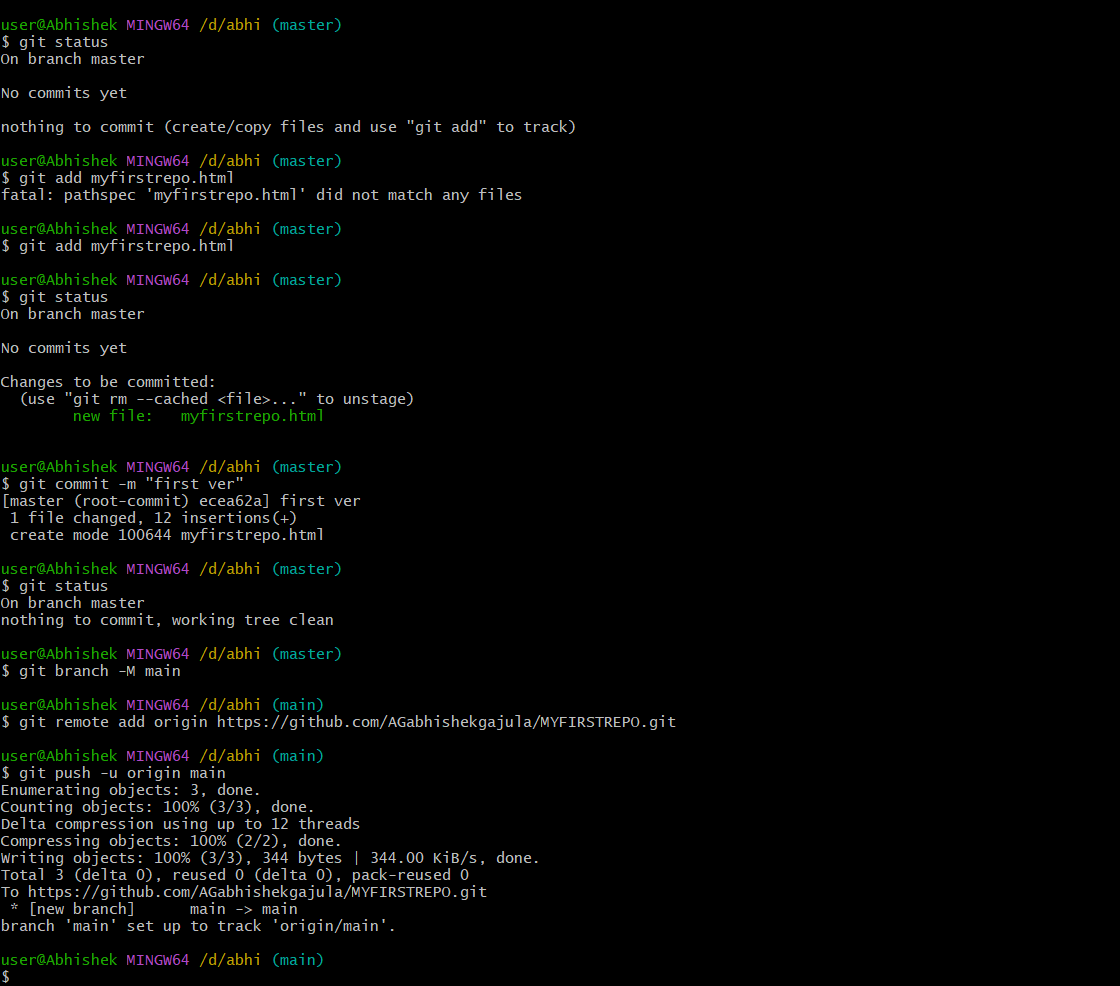
## **Git Help**

If you are having trouble remembering commands or options for commands, you can use Git help.

There are a couple of different ways you can use the help command in command line:

* git command -help -  See all the available options for the specific command
* git help --all -  See all possible commands

 Let's go over the different commands.

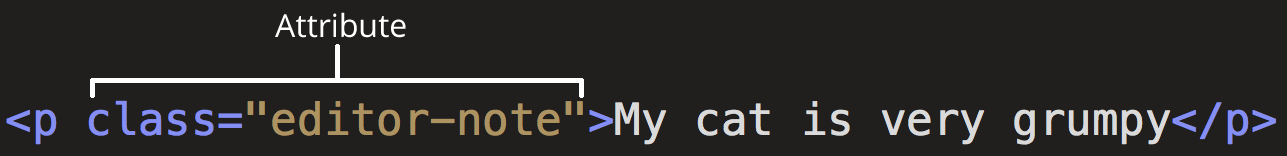


## [An aside on casing and spacing](https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/Dealing_with_files#an_aside_on_casing_and_spacing)

You'll notice that throughout this article, we ask you to name folders and files completely in lowercase with no spaces. This is because:

1. Many computers, particularly web servers, are case-sensitive. So for example, if you put an image on your website at test-site/MyImage.jpg and then in a different file you try to invoke the image as test-site/myimage.jpg, it may not work.
2. Browsers, web servers, and programming languages do not handle spaces consistently. For example, if you use spaces in your filename, some systems may treat the filename as two filenames. Some servers will replace the spaces in your filenames with "%20" (the character code for spaces in URLs), resulting in all your links being broken. It's better to separate words with hyphens, rather than underscores: my-file.html vs. my\_file.html.

* Html
* Attributes:



[src](https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/Getting_started#src)

The src attribute is a **required** attribute that specifies the location of the image. For example: src="https://raw.githubusercontent.com/mdn/beginner-html-site/gh-pages/images/firefox-icon.png".

[alt](https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/Getting_started#alt)

The alt attribute specifies a text description of the image. For example: alt="The Firefox icon".

[width](https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/Getting_started#width)

The width attribute specifies the width of the image with the unit being pixels. For example: width="300".

[height](https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/Getting_started#height)

The height attribute specifies the height of the image with the unit being pixels. For example: height="300".

## [Anatomy of an HTML document](https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/Getting_started#anatomy_of_an_html_document)

Individual HTML elements aren't very useful on their own. Next, let's examine how individual elements combine to form an entire HTML page:

HTMLCopy to Clipboard

<!doctype html>

<html lang="en-US">

<head>

<meta charset="utf-8" />

<title>My test page</title>

</head>

<body>

<p>This is my page</p>

</body>

</html>

Here we have:

1. <!DOCTYPE html>: The doctype. When HTML was young (1991-1992), doctypes were meant to act as links to a set of rules that the HTML page had to follow to be considered good HTML. Doctypes used to look something like this:

HTMLCopy to Clipboard

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

More recently, the doctype is a historical artifact that needs to be included for everything else to work right. <!DOCTYPE html> is the shortest string of characters that counts as a valid doctype. That is all you need to know!

1. <html></html>: The [<html>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/html) element. This element wraps all the content on the page. It is sometimes known as the root element.
2. <head></head>: The [<head>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/head) element. This element acts as a container for everything you want to include on the HTML page, **that isn't the content** the page will show to viewers. This includes keywords and a page description that would appear in search results, CSS to style content, character set declarations, and more. You will learn more about this in the next article of the series.
3. <meta charset="utf-8">: The [<meta>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/meta) element. This element represents metadata that cannot be represented by other HTML meta-related elements, like [<base>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/base), [<link>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/link), [<script>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/script), [<style>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/style) or [<title>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/title). The [charset](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/meta#charset) attribute specifies the character encoding for your document as UTF-8, which includes most characters from the vast majority of human written languages. With this setting, the page can now handle any textual content it might contain. There is no reason not to set this, and it can help avoid some problems later.
4. <title></title>: The [<title>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/title) element. This sets the title of the page, which is the title that appears in the browser tab the page is loaded in. The page title is also used to describe the page when it is bookmarked.
5. <body></body>: The [<body>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/body) element. This contains all the content that displays on the page, including text, images, videos, games, playable audio tracks, or whatever else.

### [Whitespace in HTML](https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/Getting_started#whitespace_in_html)

No matter how much whitespace you use inside HTML element content (which can include one or more space characters, but also line breaks), the HTML parser reduces each sequence of whitespace to a single space when rendering the code.

## [Entity references: Including special characters in HTML](https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/Getting_started#entity_references_including_special_characters_in_html)

In HTML, the characters <, >,",', and & are special characters. They are parts of the HTML syntax itself. So how do you include one of these special characters in your text? For example, if you want to use an ampersand or less-than sign, and not have it interpreted as code.

| **Literal character** | **Character reference equivalent** |
| --- | --- |
| < | &lt; |
| > | &gt; |
| " | &quot; |
| ' | &apos; |
| & | &amp; |
| * emmets |  |

## 

## [HTML comments](https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/Getting_started#html_comments)

To write an HTML comment, wrap it in the special markers <!-- and -->

<p>I'm not inside a comment</p>

<!-- <p>I am!</p> -->

the head's job is to contain [metadata](https://developer.mozilla.org/en-US/docs/Glossary/Metadata) about the document.

## [Adding custom icons to your site](https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/The_head_metadata_in_HTML#adding_custom_icons_to_your_site)

A favicon can be added to your page by:

1. Saving it in the same directory as the site's index page, saved in .ico format (most also support favicons in more common formats like .gif or .png)
2. Adding the following line into your HTML's [<head>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/head) block to reference it:

HTMLCopy to Clipboard

<link rel="icon" href="favicon.ico" type="image/x-icon" />

## [Applying CSS and JavaScript to HTML](https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/The_head_metadata_in_HTML#applying_css_and_javascript_to_html)

* The [<link>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/link) element should always go inside the head of your document. This takes two attributes, rel="stylesheet", which indicates that it is the document's stylesheet, and href, which contains the path to the stylesheet file:

HTMLCopy to Clipboard

<link rel="stylesheet" href="my-css-file.css" />

* The [<script>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/script) element should also go into the head, and should include a src attribute containing the path to the JavaScript you want to load, and defer, which basically instructs the browser to load the JavaScript after the page has finished parsing the HTML.
* <script src="my-js-file.js" defer></script>
* Text formatting tags:

| **Tags** | **Descriptions** |
| --- | --- |
| [<i>](https://www.geeksforgeeks.org/html-i-tag/) | Showcases italicized text. |
| [<small>](https://www.geeksforgeeks.org/html-small-tag/) | Renders text in a smaller font size. |
| [<ins>](https://www.geeksforgeeks.org/html-ins-tag/) | Highlights added or inserted text. |
| [<sub>](https://www.geeksforgeeks.org/html-sub-tag/) | Creates subscript text. |
| [<strong>](https://www.geeksforgeeks.org/html-strong-tag/) | Emphasizes text with importance, often in bold. |
| [<b>](https://www.geeksforgeeks.org/html-b-tag/) | Displays text in a bold format. |
| [<mark>](https://www.geeksforgeeks.org/html-mark-tag/) | Accentuates text with a background highlight. |
| [<del>](https://www.geeksforgeeks.org/html-del-tag/) | Strikes through text to signify deletion. |
| [<em>](https://www.geeksforgeeks.org/html-em-tag/) | Adds emphasis to text, commonly styled as italic. |
| [<sup>](https://www.geeksforgeeks.org/html-sup-tag/) | Formats text as superscript. |

# HTML Paragraphs

## **HTML Paragraphs**

The HTML <p> element defines a paragraph.

A paragraph always starts on a new line, and browsers automatically add some white space (a margin) before and after a paragraph.

## **HTML Horizontal Rules**

The <hr> tag defines a thematic break in an HTML page, and is most often displayed as a horizontal rule.

The <hr> tag is an empty tag, which means that it has no end tag.

## **HTML Line Breaks**

The HTML <br> element defines a line break.

The <br> tag is an empty tag, which means that it has no end tag.

## **The HTML <pre> Element**

The HTML <pre> element defines preformatted text.

The text inside a <pre> element is displayed in a fixed-width font (usually Courier), and it preserves both spaces and line breaks:

|  |  |
| --- | --- |
| **Tag** | **Description** |
| [<p>](https://www.w3schools.com/tags/tag_p.asp) | Defines a paragraph |
| [<hr>](https://www.w3schools.com/tags/tag_hr.asp) | Defines a thematic change in the content |
| [<br>](https://www.w3schools.com/tags/tag_br.asp) | Inserts a single line break |
| [<pre>](https://www.w3schools.com/tags/tag_pre.asp) | Defines pre-formatted text |

## **HTML List Tags**

Here is the list of all **lists tags HTML:**

| **Tag** | **Description** |
| --- | --- |
| [<ul>](https://www.geeksforgeeks.org/html-ul-tag/) | **Defines an unordered list** |
| [<ol>](https://www.geeksforgeeks.org/html-ol-tag/) | **Defines an ordered list** |
| [<li>](https://www.geeksforgeeks.org/html-li-tag/) | **Defines a list item** |
| [<dl>](https://www.geeksforgeeks.org/html-dl-tag/) | **Defines a description list** |
| [<dt>](https://www.geeksforgeeks.org/html-dt-tag/) | **Defines a term in a description list** |
| [<dd>](https://www.geeksforgeeks.org/html-dd-tag/) | **Details the term in a description list** |

## **The HTML Unordered List or Bulleted List**

The[unordered list](https://www.geeksforgeeks.org/html-unordered-lists/) items are marked with bullets. It is also known as bulleted lists. An unordered list starts with the **<ul> tag**. Each list item starts with the <li> tag.

**Syntax:**

<ul> list of items </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.
* [**type**](https://www.geeksforgeeks.org/html-ul-type-attribute/)**:** It specifies which kind of marker is used in the list.

# Creating hyperlinks

# A basic link is created by wrapping the text or other content inside an [<a>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/a) element and using the [href](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/a" \l "href) attribute, also known as a **Hypertext Reference**, or **target**, that contains the web address.

<p>

I'm creating a link to

<a href="https://www.mozilla.org/en-US/">the Mozilla homepage</a>.

</p>

# As mentioned before, almost any content can be made into a link, even [block-level elements](https://developer.mozilla.org/en-US/docs/Glossary/Block/CSS). If you want to make a heading element a link then wrap it in an anchor (<a>)

<a href="https://developer.mozilla.org/en-US/">

<h1>MDN Web Docs</h1>

</a>

<p>

Documenting web technologies, including CSS, HTML, and JavaScript, since 2005.

</p>

# If you have an image you want to make into a link, use the [<a>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/a) element to wrap the image file referenced with the [<img>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/img) element

<a href="https://developer.mozilla.org/en-US/">

<img src="mdn\_logo.svg" alt="MDN Web Docs" />

</a>

### [Absolute versus relative URLs](https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/Creating_hyperlinks#absolute_versus_relative_urls)

**absolute URL**: Points to a location defined by its absolute location on the web, including [protocol](https://developer.mozilla.org/en-US/docs/Glossary/Protocol) and [domain name](https://developer.mozilla.org/en-US/docs/Glossary/Domain_name).

**relative URL**: Points to a location that is relative to the file you are linking from, more like what we looked at in the previous section.

HTML Image

The <img> tag is used to embed an image in an HTML page.

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

The <img> tag has two required attributes:

* src - Specifies the path to the image
* alt - Specifies an alternate text for the image, if the image for some reason cannot be displayed

**Note:** Also, always specify the width and height of an image. If width and height are not specified, the page might flicker while the image loads.

# HTML Block and Inline Elements

A block-level element always starts on a new line, and the browsers automatically add some space (a margin) before and after the element.

A block-level element always takes up the full width available (stretches out to the left and right as far as it can).

[<address>](https://www.w3schools.com/tags/tag_address.asp)

[<article>](https://www.w3schools.com/tags/tag_article.asp)

[<aside>](https://www.w3schools.com/tags/tag_aside.asp)

[<blockquote>](https://www.w3schools.com/tags/tag_blockquote.asp)

[<canvas>](https://www.w3schools.com/tags/tag_canvas.asp)

[<dd>](https://www.w3schools.com/tags/tag_dd.asp)

[<div>](https://www.w3schools.com/tags/tag_div.asp)

[<dl>](https://www.w3schools.com/tags/tag_dl.asp)

[<dt>](https://www.w3schools.com/tags/tag_dt.asp)

[<fieldset>](https://www.w3schools.com/tags/tag_fieldset.asp)

[<figcaption>](https://www.w3schools.com/tags/tag_figcaption.asp)

[<figure>](https://www.w3schools.com/tags/tag_figure.asp)

[<footer>](https://www.w3schools.com/tags/tag_footer.asp)

[<form>](https://www.w3schools.com/tags/tag_form.asp)

[<h1>-<h6>](https://www.w3schools.com/tags/tag_hn.asp)

[<header>](https://www.w3schools.com/tags/tag_header.asp)

[<hr>](https://www.w3schools.com/tags/tag_hr.asp)

[<li>](https://www.w3schools.com/tags/tag_li.asp)

[<main>](https://www.w3schools.com/tags/tag_main.asp)

[<nav>](https://www.w3schools.com/tags/tag_nav.asp)

[<noscript>](https://www.w3schools.com/tags/tag_noscript.asp)

[<ol>](https://www.w3schools.com/tags/tag_ol.asp)

[<p>](https://www.w3schools.com/tags/tag_p.asp)

[<pre>](https://www.w3schools.com/tags/tag_pre.asp)

[<section>](https://www.w3schools.com/tags/tag_section.asp)

[<table>](https://www.w3schools.com/tags/tag_table.asp)

[<tfoot>](https://www.w3schools.com/tags/tag_tfoot.asp)

[<ul>](https://www.w3schools.com/tags/tag_ul.asp)

[<video>](https://www.w3schools.com/tags/tag_video.asp)

## **Inline Elements**

An inline element does not start on a new line.

An inline element only takes up as much width as necessary.

This is a <span> element inside a paragraph.

[<a>](https://www.w3schools.com/tags/tag_a.asp)

[<abbr>](https://www.w3schools.com/tags/tag_abbr.asp)

[<acronym>](https://www.w3schools.com/tags/tag_acronym.asp)

[<b>](https://www.w3schools.com/tags/tag_b.asp)

[<bdo>](https://www.w3schools.com/tags/tag_bdo.asp)

[<big>](https://www.w3schools.com/tags/tag_big.asp)

[<br>](https://www.w3schools.com/tags/tag_br.asp)

[<button>](https://www.w3schools.com/tags/tag_button.asp)

[<cite>](https://www.w3schools.com/tags/tag_cite.asp)

[<code>](https://www.w3schools.com/tags/tag_code.asp)

[<dfn>](https://www.w3schools.com/tags/tag_dfn.asp)

[<em>](https://www.w3schools.com/tags/tag_em.asp)

[<i>](https://www.w3schools.com/tags/tag_i.asp)

[<img>](https://www.w3schools.com/tags/tag_img.asp)

[<input>](https://www.w3schools.com/tags/tag_input.asp)

[<kbd>](https://www.w3schools.com/tags/tag_kbd.asp)

[<label>](https://www.w3schools.com/tags/tag_label.asp)

[<map>](https://www.w3schools.com/tags/tag_map.asp)

[<object>](https://www.w3schools.com/tags/tag_object.asp)

[<output>](https://www.w3schools.com/tags/tag_output.asp)

[<q>](https://www.w3schools.com/tags/tag_q.asp)

[<samp>](https://www.w3schools.com/tags/tag_samp.asp)

[<script>](https://www.w3schools.com/tags/tag_script.asp)

[<select>](https://www.w3schools.com/tags/tag_select.asp)

[<small>](https://www.w3schools.com/tags/tag_small.asp)

[<span>](https://www.w3schools.com/tags/tag_span.asp)

[<strong>](https://www.w3schools.com/tags/tag_strong.asp)

[<sub>](https://www.w3schools.com/tags/tag_sub.asp)

[<sup>](https://www.w3schools.com/tags/tag_sup.asp)

[<textarea>](https://www.w3schools.com/tags/tag_textarea.asp)

[<time>](https://www.w3schools.com/tags/tag_time.asp)

[<tt>](https://www.w3schools.com/tags/tag_tt.asp)

[<var>](https://www.w3schools.com/tags/tag_var.asp)

# HTML Div Element

The <div> element is by default a block element, meaning that it takes all available width, and comes with line breaks before and after.

# HTML <span> Tag

The <span> tag is an inline container used to mark up a part of a text, or a part of a document.

The <span> tag is easily styled by CSS or manipulated with JavaScript using the class or id attribute.

The <span> tag is much like the [<div>](https://www.w3schools.com/tags/tag_div.asp) element, but <div> is a block-level element and <span> is an inline element.

# HTML <hr> Tag

The <hr> tag defines a thematic break in an HTML page (e.g. a shift of topic).

The <hr> element is most often displayed as a horizontal rule that is used to separate content (or define a change) in an HTML page.

# HTML Semantic Elements

Semantic elements = elements with a meaning.

## **What are Semantic Elements?**

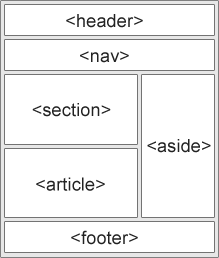
A semantic element clearly describes its meaning to both the browser and the developer.

Examples of **non-semantic** elements: <div> and <span> - Tells nothing about its content.

Examples of **semantic** elements: <form>, <table>, and <article> - Clearly defines its content.

## **Semantic Elements in HTML**

Many web sites contain HTML code like: <div id="nav"> <div class="header"> <div id="footer"> to indicate navigation, header, and footer.

In HTML there are some semantic elements that can be used to define different parts of a web page:

* <article>
* <aside>
* <details>
* <figcaption>
* <figure>
* <footer>
* <header>
* <main>
* <mark>
* <nav>
* <section>
* <summary>
* <time>
* Uses of semantic tags :
  1. Meaningful and layout
  2. Seo friendly
  3. Readable
  4. **HTML <section> Element**

The <section> element defines a section in a document.

According to W3C's HTML documentation: "A section is a thematic grouping of content, typically with a heading."

Examples of where a <section> element can be used:

* Chapters
* Introduction
* News items
* Contact information

A web page could normally be split into sections for introduction, content, and contact information.

* 1. **HTML <article> Element**

The <article> element specifies independent, self-contained content.

An article should make sense on its own, and it should be possible to distribute it independently from the rest of the web site.

Examples of where the <article> element can be used:

* Forum posts
* Blog posts
* User comments
* Product cards
* Newspaper articles

you will find HTML pages with <section> elements containing <article> elements, and <article> elements containing <section> elements.

* 1. **HTML <header> Element**

The <header> element represents a container for introductory content or a set of navigational links.

A <header> element typically contains:

* one or more heading elements (<h1> - <h6>)
* logo or icon
* authorship information
  1. **HTML <footer> Element**

The <footer> element defines a footer for a document or section.

A <footer> element typically contains:

* authorship information
* copyright information
* contact information
* sitemap
* back to top links
* related documents
  1. **HTML <nav> Element**

The <nav> element defines a set of navigation links.

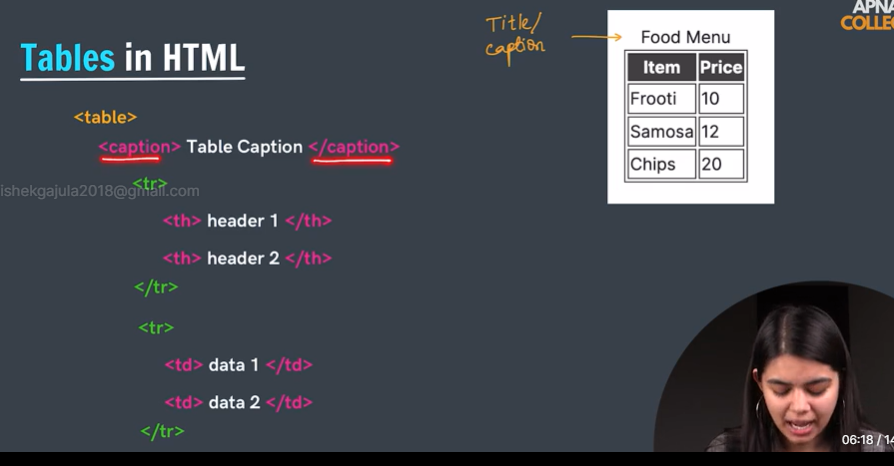
* 1. **HTML <aside> Element**

The <aside> element defines some content aside from the content it is placed in (like a sidebar).

The <aside> content should be indirectly related to the surrounding content.

Note: refer the beginner guide to html

* Tables in html:



* Table semantics



* Colspan: To make a cell span over multiple columns, use the colspan attribute

 The value of the colspan attribute represents the number of columns to span.

* Rowspan: To make a cell span over multiple rows, use the rowspan attribute

The value of the rowspan attribute represents the number of rows to span.

* Forms in html:

**HTML Forms** employ the <form> element to gather input data with interactive controls. It encompasses various input types such as text, numbers, email, password, checkboxes, radio buttons, and submit buttons. Essentially, it’s a container for diverse input elements facilitating user interaction.

**Syntax:**

<form>

<!--form elements-->

</form>

<form action="*URL*">  
The action attribute specifies where to send the form-data when a form is submitted.

## **Form Elements**

| **Elements** | **Descriptions** |
| --- | --- |
| [**<label>**](https://www.geeksforgeeks.org/html-label-tag/) | It defines labels for <form> elements. |
| [**<input>**](https://www.geeksforgeeks.org/html-input-tag/) | It is used to get input data from the form in various types such as text, password, email, etc by changing its type. |
| [**<button>**](https://www.geeksforgeeks.org/button-tag-vs-input-typebutton-attribute/) | It defines a clickable button to control other elements or execute a functionality. |
| [**<select>**](https://www.geeksforgeeks.org/html-select-tag/) | It is used to create a drop-down list. |
| [**<textarea>**](https://www.geeksforgeeks.org/html-textarea-name-attribute/) | It is used to get input long text content. |
| [**<fieldset>**](https://www.geeksforgeeks.org/html-fieldset-form-attribute/) | It is used to draw a box around other form elements and group the related data. |
| [**<legend>**](https://www.geeksforgeeks.org/html-legend-tag/) | It defines a caption for fieldset elements |
| [**<datalist>**](https://www.geeksforgeeks.org/html-datalist-tag/) | It is used to specify pre-defined list options for input controls. |
| [**<output>**](https://www.geeksforgeeks.org/html-output-tag/) | It displays the output of performed calculations. |
| [**<option>**](https://www.geeksforgeeks.org/html-option-tag/) | It is used to define options in a drop-down list. |
| [**<optgroup>**](https://www.geeksforgeeks.org/html-optgroup-tag/) | It is used to define group-related options in a drop-down list. |

## Commonly Used Input Types in HTML Forms

<input type=”text”>

| **Input Type** | **Description** |
| --- | --- |
| [<input type=”t**ext**“>](https://www.geeksforgeeks.org/html-input-typetext/) | Defines a one-line text input field |
| [<input type=”p**assword**“>](https://www.geeksforgeeks.org/html-input-typepassword/) | Defines a password field |
| [<input type=”s**ubmit**“>](https://www.geeksforgeeks.org/html-input-typesubmit/) | Defines a submit button |
| [<input type=”r**eset**“>](https://www.geeksforgeeks.org/html-input-typereset/) | Defines a reset button |
| [<input type=”r**adio**“>](https://www.geeksforgeeks.org/html-input-typeradio/) | Defines a radio button |
| [<input type=”e**mail**“>](https://www.geeksforgeeks.org/html-input-typeemail/) | Validates that the input is a valid email address. |
| [<input type=”n**umber**“>](https://www.geeksforgeeks.org/html-input-typenumber/) | Allows the user to enter a number. You can specify min, max, and step attributes for range. |
| [<input type=”c**heckbox**“>](https://www.geeksforgeeks.org/html-input-typecheckbox/) | Used for checkboxes where the user can select multiple options. |
| [<input type=”d**ate**“>](https://www.geeksforgeeks.org/html-input-typedate/) | Allows the user to select a date from a calendar. |
| [<input type=”t**ime**“>](https://www.geeksforgeeks.org/html-input-typetime/) | Allows the user to select a time. |
| [<input type=”f**ile**“>](https://www.geeksforgeeks.org/html-input-typefile/) | Allows the user to select a file to upload. |

* ATTRIBUTES:

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).

<input placeholder="*text*">

<!-- label are used to combine two entities -->

    enter your name:<input type="text" placeholder="username" name="username"><!--name or value acts as variable in the backend to access the value-->

    <br>

    <!-- normal approach for label -->

    <label >your fav colour

    <input type="colour" placeholder="colour" >

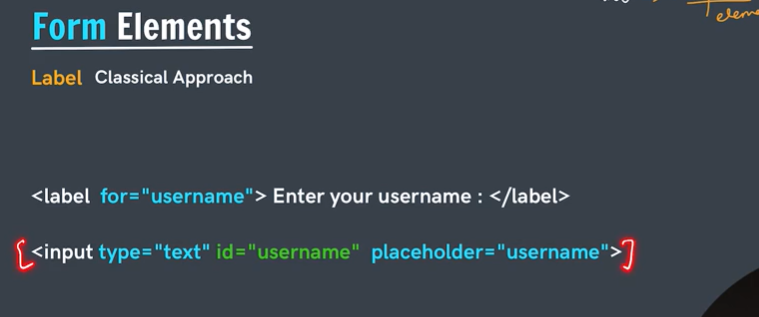
    </label>

    <!--classical approach for label -->

    <br>

    <label for="time">enter your birth time</label>

    <input type="time" placeholder="time" id="time">

    <div> 

# HTML <button> Tag

**HTML <button> tag** defines a clickable button within a form or standalone. It can contain text or images and triggers an action when clicked, typically associated with JavaScript functions or form submissions.

* It is important to always specify the **type** attribute for a button element to inform the browser what type of button it is.
* <button type = "button"> Click Me </button>

the type attribute specifies the type of button.

<button type="button|submit|reset">

* Submit: Submits the current form data
* Reset: Resets data in the current form
* Button: Just a button

Note: if button(submit type) is in the form then when it is clicked the will automatically submitted.

We can also create buttons using input tag using type=”button”

# HTML name Attribute

This name attribute can be used to reference the element in a JavaScript.

For a <form> element, the name attribute is  used as a reference when the data is submitted.

Usually the data in the form is transferred in the form name and value

Eg:name=value

<input type="checkbox" id="age" name="age" value="18">

# HTML<input type="checkbox">

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

The checkbox is shown as a square box that is ticked (checked) when activated.

Checkboxes are used to let a user select one or more options of a limited number of choices.

# <input type="checkbox" id="vehicle1" name="vehicle1" value="Bike"> <label for="vehicle1"> I have a bike</label><br> <input type="checkbox" id="vehicle2" name="vehicle2" value="Car"> <label for="vehicle2"> I have a car</label><br>

# HTML <input type=”radio”>

# HTML **<input type=”radio”>** creates a radio button input field allowing users to select one option from multiple choices. Only one radio button in a group can be selected at a time.

# **Note:** The “**value**” attribute in radio buttons assigns a distinct identifier for each option. While not visible to users, this value is sent to the server upon submission, helping identify the selected radio button.

# Note: grouping of radio buttons and checkboxes can be done by giving same name attribute to all radio or checkboxes . and if we want to pass the name of radio button or checkbox button clicked by the user then we should use value attribute.

favrouite food items

            <input type="checkbox" name="fooditem" id="fooditem1" value="biryani">

            <label for="fooditem1">biryani</label>

            <input type="checkbox" name="fooditem" id="fooditem2" value="paneer">

            <label for="foodite2">paneer</label>

            <input type="checkbox" name="fooditem" id="fooditem3" value="burger">

            <label for="fooditem3">burger</label>

<input type="radio" name="food" id="foodname1" value="biryani">

            <label for="foodname1">biryani</label>

            <input type="radio" name="food" id="foodname2" value="paneer">

            <label for="foodname2">paneer</label>

            <input type="radio" name="food" id="foodname3" value="burger">

            <label for="foodname3">burger</label>

# HTML <select> Tag

The **<select> tag** in HTML is used to create a drop-down list. The <select> tag contains <option> tag to display the available option of the drop-down list.

**Note:** The <select> tag is used in a form to receive user responses.

<select>  
 <option>  
 </option>  
 ...  
</select>

<!DOCTYPE html>

<**html**>

<**body**>

    <**h2**>Welcome To GeeksforGeeks</**h2**>

    <**label** for="Brands">Choose a Brand:</**label**>

    <**select** name="Brands" id="Brands">

        <**optgroup** label="Tech Brands">

            <**option** value="Google">Google</**option**>

            <**option** value="Apple">Apple</**option**>

        </**optgroup**>

        <**optgroup** label="Automative Brands">

            <**option** value="Tesla">Tesla</**option**>

            <**option** value="audi">Audi</**option**>

        </**optgroup**>

        <**optgroup** label="Entertainment Brand">

            <**option** value="Disney">Disney</**option**>

        </**optgroup**>

    </**select**>

</**body**>

</**html**>

# Example program for forms:

* <!DOCTYPE html>
* <html lang="en">
* <head>
* <meta charset="UTF-8">
* <meta name="viewport" content="width=device-width, initial-scale=1.0">
* <title>forms</title>
* </head>
* <body>

* <form action="/action">
* <!-- label are used to combine two entities -->
* enter your name:<input type="text" placeholder="username" name="username"><!--name or value acts as variable in the backend to access the value-->
* <br>
* <!-- normal approuch for label -->
* <label >your favrouite colour
* <input type="color" placeholder="colour" >
* </label>
* <!--classical approuch for label -->
* <br>
* <label for="time">enter yout birth time</label>
* <input type="time" placeholder="time" id="time">
* <div>
* password:<input type="password" placeholder="password">
* </div>
* phone number:<input type="number">
* <br>
* <!-- buttons are three types 1.submit button 2.buttuo type 3.reset -->
* <button type="submit">submits</button>
* <button type="reset">clear</button>
* <button type="button">namaste</button>
* <!-- buttons using input tag but not recommended -->
* <div>
* <input type="submit">
* <input type="button" value="hi">
* </div>
* <hr>
* <!-- in radio button we can select only on button when they are grouped -->
* <!-- grouping can be done by using name attribute by giving same name
* example in below : name=food same for all-->
* <div>
* <!-- check boxex can also be gouped but we can select multiple -->
* <input type="checkbox" id="age" name="age" value="18">
* <label for="age">I AM 18+ &1F603;</label>
* <button type="submit">submit</button>
* favrouite food items
* <input type="checkbox" name="fooditem" id="fooditem1" value="biryani">
* <label for="fooditem1">biryani</label>
* <input type="checkbox" name="fooditem" id="fooditem2" value="paneer">
* <label for="foodite2">paneer</label>
* <input type="checkbox" name="fooditem" id="fooditem3" value="burger">
* <label for="fooditem3">burger</label>
* </div>
* <div>
* <input type="radio" name="food" id="foodname1" value="biryani">
* <label for="foodname1">biryani</label>
* <input type="radio" name="food" id="foodname2" value="paneer">
* <label for="foodname2">paneer</label>
* <input type="radio" name="food" id="foodname3" value="burger">
* <label for="foodname3">burger</label>
* <button>submit</button>
* </div>
* <hr>
* <div>
* <label for="state">select your state</label>
* <select name="state" id="state">
* <option value="none" selected>select your state</option>
* <option value="telangana">telangana</option>
* <option value="andhara">andhara</option>
* <option value="kerala">kerala</option>
* </select>
* </div>
* <hr>
* <div>
* <label for="age">select your age:</label>
* <input type="range" min="0" max="18" step="2" id="age" name="age">
* </div>
* <hr>
* <div>
* <label for="feedback">please submit your feedback</label>
* <textarea name="feedback" id="feedback" id="feedback" placeholder="enter here"></textarea>
* </div>
* <label for="food">please submit your feedback on food</label>
* <textarea name="feedback on food" id="food" id="food" rows="20" cols="40" placeholder="enter here"></textarea>
* </form>


* </body>
* </html>