HTML 5 :: Notes

HTML 5 ___ TAG | SYNTAX | REFERENCE

This Short Note Created by Muhammad S.A. Iqbal (Last Update: August 23, 2024)

Essential Toolkits

```
❑ Useful URL:
_> HTML Color Code: https://htmlcolorcodes.com
_> HTML Color Code: https://flatuicolors.com
_> HTML Color Code: Color Names List view
_> HTML Color Code: RGB Color view
_> HTML Color Code: RGBA Color view
_> HTML Color Code: CMYK Color view
_> HTML Color Code: HEX Color view
_> HTML Color Code: HSL Color view
_> HTML Color Code: HSLA Color view

_> Image Maps: https://imagemap.org |Ref.Tutorial view
_> Favicon Icon Tools: https://www.favicon.cc

_> HTML Validation Check: validator.w3.org

❑ Tutorial URL:
_> Tutorial :: HTML > Mozila view
_> Tutorial :: HTML > W3School view

❑ Blog:
_> Character Encodings: Essential concepts view1 view2
```

Table of Notes

Keyboard Shortcut for VS Code

Purpose	Keyboard Shortcut
HTML Basic Syntax Format	Type: “ ! ”; then, Press: TAB
Block or Tag Syntax	Type Tag Name; then, Press: TAB
Duplicate Line	Alt + Shift + ↓
Toggle Block Comment	Alt + Shift + A

—— 「 SECTION [~] :: Introduction 」 ——

> ╰☆☆ `<! DOCTYPE html >` | `<!-- Comments -->` ☆☆╮

What is HTML?

HTML (Hypertext Markup Language) is the standard markup language for creating web pages.

```
<!DOCTYPE html>
<html>
<head>
    <title>Page Title</title>
    <meta charset="utf-8" />
</head>
<body>

    <h1>My First Heading</h1>
    <p>My first paragraph.</p>

</body>
</html>

Example Explained:
- 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 <body> element contains the visible page content
```

❑ HTML TAG & ATTRIBUTES

❑ HTML Tags

The first tag in a pair is the start tag, the second tag is the end tag. e.g.

```
        <h1> Short Notes </h1> ---- example 1
        <br/> ---------------------- example 2
        <!DOCTYPE html> ----------- example 3 (exceptional)
```

❑

HTML Attributes

Attributes provide additional information about HTML elements

- 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"
For Examples:

```
        <a href="www.abc.com"> This is a link </a>

              <img src="abc.jpg" alt="abc.com" width="104" height="142">

Example Explained:
- href keyword is attribute of </a>
- </img> is tag and others (e.g. src/alt/width/height) are attributes.
```

❑ HTML <HEAD />

The HTML `<head>` element is a container for all the head elements. Such as — `<title>`, `<style>`, `<meta>`, `<link>`, `<script>`, and `<base>`.

```
<!DOCTYPE html>
<html lang="en">
  <head>
  <title>Document</title>

  <meta charset="UTF-8" />
  <meta name="description" content="Free Web tutorials">
  <meta name="keywords" content="HTML, CSS, XML, JavaScript">
  <meta name="author" content="John Wick">
  <meta name="viewport" content="width=device-width, height=device-height,
                    initial-scale=1.0, maximum-scale=2.0, minimum-scale=1.0" />
  <meta http-equiv="refresh" content="30">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">

  <!-- ---- Apple Device ----- -->
  <meta name="apple-mobile-web-app-capable" content="yes">
  <meta content="black-translucent" name="apple-mobile-web-app-status-bar-style">

  </head>
  <body>
                  // Code... Block

  </body>
</html>
```

Basic Tag Keys:

✪ Title :: `<title>` : Defines a title in the browser tab. Provides a title for the page when it is added to favorites and display in search engine results.

Basic Meta Tag Keys:

✪ Meta :: `charset =" UTF-8 "` : Define the character set.

✪ Meta :: `name =" description "` : Define a description of your web page.

✪ Meta :: `name =" keyword "` : Define keywords for search engines.

✪ Meta :: `name =" author "` : Define the author of a page.

✪ Meta :: `http-equiv =" refresh "` : Refresh document every 30 seconds.

✪ Meta :: `name =" viewport "` : Control the page's dimensions and scaling. (Browser Responsive)

Apple Specific Meta Tag Keys:

✪ Meta :: `name =" apple-mobile-web-app-capable "` : If `content` is set to `yes`, the web application runs in full-screen mode; otherwise, it does not. The default behavior is to use Safari to display web content.

✪ Meta :: `name =" apple-mobile-web-app-capable "` : This tag has no effect unless you first specify full-screen mode as described in `apple-mobile-web-app-capable`. If `content` is set to `default`, the status bar appears normal. If set to `black-translucent`, the status bar is black and translucent.

❑ Common TAG

> ❑ **HTML 5 ::** `<! DOCTYPE html >`
> The declaration defines this document to be HTML5.

> ❑ **Comment ::** `<!-- Comments -->`
> Comment tags are used to insert comments in the HTML source code.

❑ CSS STYLE in HTML

Cascading Style Sheets (CSS) is used to format the layout of a webpage.

CSS can be added to HTML documents in 3 ways:

1. **Inline** - by using the `style` attribute inside HTML elements
2. **Internal** - by using a `< style >` element in the `<head>` section
3. **External** - by using a `<link>` element to link to an external CSS file

1. ☂ Inline CSS

`Inline CSS Code into HTML Tag`

The HTML `style` attribute is used to add styles to an element, such as color, font, size, and more.

```
Syntax:
                <tagname style="property:value;">

Example:
                <body>
                        <h1 style="background-color:powderblue;">This is a heading</h1>
                        <p style="background-color:tomato;">This is a paragraph.</p>
                </body>
```

2. ☂ Internal CSS

`Internal CSS Code into HTML Page`

An internal CSS is used to define a style for a single HTML page. An internal CSS is defined in the `<head>` section of an HTML page, within a `<style>` element.

```
<!DOCTYPE html>
<html>
    <head>
```

```
        <style>
            body {background-color: powderblue;}
            h1   {color: blue;}
            p    {color: red;}
        </style>
    </head>
<body>
    <h1>This is a heading</h1>
    <p>This is a paragraph.</p>
</body>
</html>
```

3. ☂ External CSS

`External CSS Code for HTML`

An external style sheet is used to define the style for many HTML pages. To use an external style sheet, add a link to it in the `<head>` section of each HTML page:

```
<!DOCTYPE html>
<html>
<head>
      <link rel="stylesheet" href="styles.css">
</head>
<body>
        <h1>This is a heading</h1>
        <p>This is a paragraph.</p>
</body>
</html>

The external style sheet can be written in any text editor.
The file must not contain any HTML code, and must be saved with a .css extension.
Here is what the "styles.css" file looks like:

        body {
          background-color: powderblue;
        }
        h1 {
          color: blue;
        }
        p {
          color: red;
        }
```

❑ HTML Block and Inline Elements

Every HTML element has a default display value, depending on what type of element it is. The two display values are:

1. Block-Level Elements
2. Inline Elements

1. ☂ Block-Level Elements:

A block-level element always starts on a new line and takes up the full width available (stretches out to the left and right as far as it can). example:

```
<div> Hello World! </div>
```

> ❑ For Example :: **The `< div >` Element**
> This element is often used as a container for other HTML elements. The `< div >` element has no required attributes, but `style`, `class` and `id` are common. When used together with CSS, the `< div >` element can be used to style blocks of content. for another example:

```
<div style="background-color:black;color:white;padding:20px;">
  <h2>London</h2>
  <p>London is the capital city of England. </p>
</div>
```

Block level elements in HTML:

`< address >`	`< article >`	`< aside >`	`< blockquote >`	`< canvas >`	`< dd >`	`< div >`
`< dl >`	`< dt >`	`< fieldset >`	`< figcaption >`	`< figure >`	`< footer >`	`< form >`
`< h1 >` - `< h2 >`	`< header >`	`< noscript >`	`< main >`	`< nav >`	`< hr >`	`< li >`
`< section >`	`< table >`	`< tfoot >`	`< ul >` `< ol >`	`< pre >`	`< video >`	`< p >`

2. ☂ Inline Elements:

An inline element does not start on a new line and only takes up as much width as necessary.

```
This is an inline <span> element inside </span> a paragraph.
```

> ❑ For Example :: **The `< span >` Element**
> This element is often used as a container for some text. The `< span >` element has no required attributes, but `style`, `class` and `id` are common. When used together with CSS, the `< span >` element can be used to style parts of the text. for another example:

```
<h1>My <span style="color:red">Important</span> Heading</h1>
```

Inline elements in HTML:

`< acronym >`	`< abbr >`	`< a >`	`< b >`	`< bdo >`	`< big >`	`< br >`
`< button >`	`< cite >`	`< code >`	`< dfn >`	`< img >`	`< em >`	`< i >`
`< input >`	`< label >`	`< object >`	`< output >`	`< map >`	`< kbd >`	`< q >`
`< script >`	`< select >`	`< strong >`	`< small >`	`< samp >`	`< span >`	`< sub >`
`< textarea >`	`< time >`	`< sup >`	`< tt >`	`< var >`		

❑ HTML/CSS Selector & Combinator

In CSS, selectors are used to target the HTML elements on our web pages that we want to style. There are a wide variety of CSS selectors available:

1. CSS Element / Type / Tag Selector
2. CSS ID Selector
3. CSS Class Selector
4. CSS Universal Selector
5. CSS Grouping Selector
6. CSS Attribute Selector
7. Pseudo-Classes and Pseudo-Elements

1. ☂️ Selector :: Element / Type / Tag

The element selector selects HTML elements based on the element name.

```
HTML File Code:

<!DOCTYPE html>
<html>
<head>
        <title>This is HTML Document</title>
        <link rel="stylesheet" href="style.css"/>
</head>
<body>
        <div>
            <p> Lorem ipsum dolor sit amet. </p>
        </div>
</body>
</html>

CSS File Code: (This File name is: style.css)

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

Explain:
Here, all <p> elements on the page will be center-aligned, with a red text color
```

2. ☂️ Selector :: ID

The id selector uses the id attribute of an HTML element to select a specific element.

- The id of an element is unique within a page, so the id selector is used to select one unique element!
- To select an element with a specific id, write a hash (#) character, followed by the id of the element.

> 💡 **Note:** An id name cannot start with a number!

```
HTML File Code:

<!DOCTYPE html>
```

```
<html>
<head>
        <title>This is HTML Document</title>
        <link rel="stylesheet" href="style.css"/>
</head>
<body>
        <div #text1>
            <p> Lorem ipsum dolor sit amet. </p>
        </div>
</body>
</html>


CSS File Code: (This File name is: style.css)

#tex1 {
  text-align: center;
  color: red;
}

Explain:
The CSS rule below will be applied to the HTML element with id="tex1".
```

3. ☂ Selector :: Class

The class selector selects HTML elements with a specific class attribute.

- To select elements with a specific class, write a period (.) character, followed by the class name.

```
Example 1:

HTML File Code:

<!DOCTYPE html>
<html>
<head>
        <title>This is HTML Document</title>
        <link rel="stylesheet" href="style.css"/>
</head>
<body>
        <div class="center">
            <p> Lorem ipsum dolor sit amet. </p>
        </div>
</body>
</html>

CSS File Code: (This File name is: style.css)

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

```
Explain:
In this example all HTML elements with class="center" will be red and center-aligned.
```

```
Example 2:
                    <p class="center large"> This paragraph refers to two classes. </p>

Explain:
In this example the <p> element will be styled according to class="center"
and to class="large"
```

Different Between ID Selector and Class Selector

Key	ID	Class
Syntax	In HTML, for an element, the ID name starts with the "#" symbol followed by a unique name assigned to it.	"class" assigned to an element has its name starts with "." followed by class name.
Selector	Only one ID selector can be attached to an element.	Multiple class selectors can be attached to an element.
Uniqueness	ID is unique in a page and can only apply to at most one element.	The class can be applied to multiple elements so it could be multiple times on a single page.

4. ☂️ Selector :: Universal

The universal selector (*) selects all HTML elements on the page.

```
* {
  text-align: center;
  color: blue;
}

Explain:
The CSS rule below will affect every HTML element on the page.
```

5. ☂️ Selector :: Grouping

The grouping selector selects all the HTML elements with the same style definitions.

```
Example 1:
                        h1, h2, p {
                          text-align: center;
                          color: red;
                        }
Explain:
In this example we have grouped the selectors from this code.
```

```
Example 2.1:
                        h1, .special {
                          color: blue;
                        }
Explain:
I could also combine these into a selector list, by adding a comma between them.

Example 2.2:
```

```
                            h1,
                            .special {
                              color: blue;
                            }

Note:
White space is valid before or after the comma. You may also find the selectors
more readable if each is on a new line.
```

6. ☂️ Selector :: Attribute

CSS `[ attribute ]` Selector

This selector is used to select elements with a specified attribute.

```
a[target] {
  background-color: yellow;
}

Explain:
This example selects all <a> elements with a target attribute.
```

CSS `[ attribute =" value "]` Selector

This selector is used to select elements with a specified attribute and value.

```
a[target="_blank"] {
  background-color: yellow;
}

Explain:
This example selects all <a> elements with a [target="_blank"] attribute.
```

CSS `[ attribute ~=" value "]` Selector

This selector is used to select elements with an attribute value containing a specified word.

```
[title~="flower"] {
  border: 5px solid yellow;
}

Explain:
This example selects all elements with a title attribute that contains
a space-separated list of words, one of which is "flower".
```

CSS `[ attribute ^=" value "]` Selector

This selector is used to select elements with the specified attribute, whose value starts with the specified value.

> 💡 **Note:** The value does not have to be a whole word!

```
a[href^="www."]{
    font-size: 150%;
}
```

```
Explain:
This example selects all elements with a href attribute value that starts with "www.".
```

CSS `[ attribute $=" value "]` Selector

This selector is used to select elements whose attribute value ends with a specified value.

> 💡 **Note:** The value does not have to be a whole word!

```
a[href$=".com"]{
    text-transform: uppercase;
}

Explain:
This example selects all elements with a href attribute value that ends with ".com".
```

CSS `[ attribute *=" value "]` Selector

This selector is used to select elements whose attribute value contains a specified value.

> 💡 **Note:** The value does not have to be a whole word!

```
a[href*="google"]{
    text-transform: lowercase;
}

Explain:
This example selects all elements with a href attribute value that contains "google".
```

CSS `[ attribute |=" value "]` Selector

This selector is used to select elements with the specified attribute, whose value can be exactly the specified value, or the specified value followed by a hyphen (-).

> 💡 **Note:** The value has to be a whole word.

```
a[href|="chrome"]{
    text-transform: uppercase;
}

Explain:
The value has to be a whole word, either alone, like href="www.chrome.com",
or followed by a hyphen( - ), like href="www.chrome.com/download-chrome-browser".
```

7. ☂️ Selector :: Pseudo-Classes and Pseudo-Elements

Pseudo-classes and pseudo-elements in CSS are selectors that allow the styling of specific states or parts of elements.

Pseudo-Classes:

A pseudo-class is used to define a **special state of an element**.

A pseudo-class represents a state of a selector like `:hover`, `:active`, `:last-child`, etc. These start with a single colon(:).

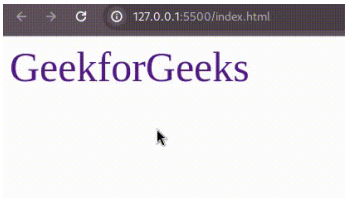

Figure: Pseudo-Class Appearance

```
<!DOCTYPE html>
<html lang="en">

<head>
    <style>
            a{
                font-size: 50px;
                height: 30px;
                text-decoration: none;
            }
            a:hover{
                color: green;
            }
    </style>
</head>

<body>
        <a href="">GeekforGeeks</a>
</body>
</html>

Explain:
:hover Pseudo-class is used to add a special effect to an element when our mouse
pointer is over it. When your mouse enters the box area, its background color changes
from yellow to orange.
```

Pseudo-Elements:

A CSS pseudo-element is used to **style specified parts of an element.**

Similarly, a pseudo-element is used to select virtual elements like `::after`, `::before`, `::first-line`, etc. These start with a double colon(::).

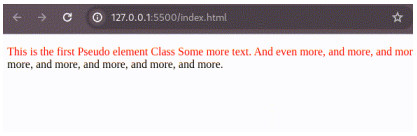

Figure: Pseudo-Elements Appearance

```
<!DOCTYPE html>
<html>
<head>
    <style>
            p::first-line{
                color: red;
            }
```

```
    </style>
</head>

<body>
    <p>
            This is the first Pseudo element Class
            Some more text. And even more, and more,
            and more, and more, and more, and more,
            and more, and more, and more, and more,
            and more, and more.
    </p>
</body>
</html>

Explain:
::first-line Pseudo-element applies styles to the first line of a block-level element.
Note that the length of the first line depends on many factors, including
the width of the element, the width of the document, and the font size of
the text.

Note that only a few properties are applied for first-line
pseudo-element like font properties, color properties, background properties,
word-spacing, letter-spacing, text-decoration, vertical-align, text-transform,
line-height, clear, etc.
```

Difference between Pseudo-Classes and Pseudo-Elements

Pseudo-Classes	Pseudo-Elements
Pseudo-Classes are based on state or user interaction on the element.	Pseudo-elements style the specific part of an element.
Pseudo Classes starts with (‘ : ‘) name.	Pseudo-Elements begins with (‘ :: ‘) double colon.
It can used with multiple selector, allowing condition.	Pseudo Elements mostly used alone and targets the specific parts of an element
Pseudo-classes examples are: `element : hover { color : red ; }`	Pseudo-Element examples are: `elemen t:: first-line { font-size : weight : bold ;}`

❑ HTML/CSS Invalid Selector

When you group selectors in this way, if any selector is syntactically invalid, the whole rule will be ignored.

In the following example, the invalid class selector rule will be ignored, whereas the `< h1 >` would still be styled.

```
Example 1:
                        h1 {
                          color: blue;
                        }

                        ..special {
                          color: blue;
                        }

Example 2:
```

```
                        h1, ..special {
                          color: blue;
                        }

Explain:
Here, double dot (..) invalid to this class.
```

Selector :: Invalid `: invalid`

This selector selects form elements with a value that does not validate according to the element's settings.

> 💡 **Note:** The `:invalid` selector only works for `< form >` elements with limitations. Such as `< form >`, `< fieldset >`, `< input >` or other `< form >` element whose contents fail to validate.

> 📔 **Tip:** Use the `:valid` selector to select form elements with a value that validates according to the element's settings.

```
            input:invalid {
              background-color: ivory;
              border: none;
              outline: 2px solid red;
              border-radius: 5px;
            }
```

❑ HTML/CSS Combinators (as Selector)

CSS combinators are explaining the relationship between two selectors.

There are four different combinators in CSS:

- Descendant selector (space)
- Child selector (>)
- Adjacent sibling selector (+)
- General sibling selector (~)

1. ☂ Combinator :: Descendant Selector

The descendant selector matches all elements that are descendants of a specified element. This selector is used to select all the child elements of the specified tag.

```
<!DOCTYPE html>
<html>

<head>
    <title>Combinator Property</title>
    <style>
            div p{
                color: #009900;
                font-size:32px;
                text-align:center;
```

```
            }
            div {
                text-align:center;
            }
            p {
                text-align:center;
            }
    </style>
</head>

<body>

        <div>Descendant selector property</div>
        <p>GeeksforGeeks</p>
        <div>
            <div>child div content</div>
            <p>G4G</p>
            <p>Descendant selector</p>
        </div>
        <p>Geeks</p>
        <p>Hello</p>

</body>
</html>

Explain:
Selects all <p> elements inside <div> elements.
```

2. ☂ Combinator :: Child Selector (>)

The child selector selects all elements that are the children of a specified element. **This combinator is stricter than the descendant selector** because it selects only the second selector if it has the first selector element as its parent.

```
<!DOCTYPE html>
<html>

<head>
    <title>Combinator Property</title>
    <style>
            div > p{
                color: #009900;
                font-size:32px;
                font-weight:bold;
                margin:0px;
                text-align:center;
            }
            div {
                text-align:center;
            }
            p {
                text-align:center;
            }
    </style>
```

```
</head>

<body>

        <div>Child selector property</div>
        <p>GeeksforGeeks</p>
        <div>
            <div>child div content</div>
            <p>G4G</p>
        </div>
        <p>Geeks</p>
        <p>Hello</p>

</body>
</html>

Explain:
Selects all <p> elements that are children of a <div> element.
```

3. ☂ Combinator :: Adjacent Sibling Selector (+)

The adjacent sibling selector is used to select an element that is directly after another specific element. **This combinator selects only one tag that is just next to the specified tag.**

```
<!DOCTYPE html>
<html>
<head>
    <title>Combinator Property</title>
    <style>
            div + p{
                color: #009900;
                font-size:32px;
                font-weight:bold;
                margin:0px;
                text-align:center;
            }
            div {
                text-align:center;
            }
            p {
                text-align:center;
            }
    </style>
</head>

<body>
        <div>Adjacent sibling selector property</div>
        <p>GeeksforGeeks</p>
        <div>
            <div>child div content</div>
            <p>G4G</p>
        </div>
        <p>Geeks</p>
        <p>Hello</p>
```

```
</body>
</html>

Explain:
selects the first <p> element that are placed immediately after <div> elements.
```

4. ☂️ Combinator :: General Sibling Selector (~)

The general sibling selector selects all elements that are next siblings of a specified element. This can be used to select a group of elements that share the same parent element.

```
<!DOCTYPE html>
<html>

<head>
    <title>Combinator Property</title>
    <style>
            div ~ p{
                color: #009900;
                font-size:32px;
                text-align:center;
            }
            div {
                text-align:center;
            }
    </style>
</head>

<body>

        <div>General sibling selector property</div>
        <p>GeeksforGeeks</p>
        <div>
            <div>child div content</div>
            <p>G4G</p>
        </div>
        <p>Geeks</p>
        <p>Hello</p>

</body>
</html>

Explain:
Selects all <p> elements that are next siblings of <div> elements.
```

❑ JavaScript in HTML

JavaScript makes HTML pages more dynamic and interactive.

👨🏻‍💻 JavaScript Code in HTML Page

```
        <script>
              // Internal JavaScript Code
        </script>
```

🥷 **JavaScript Code from outside source**

```
         <script type="text/javascript" src="#"></script>
```

✍️ Note: The external script file cannot contain the `<script>` tag. Point to the external script file exactly where you would have written the script.

❑ Example: HTML Layout Elements and Techniques

Layout Elements

- `<header>` - Defines a header for a document or a section
- `<div>` - Defines a set of specific part
- `<section>` - Defines a section in a document
- `<aside>` - Defines content aside from the content (like a sidebar)
- `<footer>` - Defines a footer for a document or a section

Example:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width" />
  </head>
  <body>
    <header>
      <img src="lgo.jpg" alt="abc.com" width="104" height="142" />
      <ul>
        <li><a href="">Home</a></li>
        <li><a href="">Service</a></li>
        <li><a href="">About</a></li>
        <li><a href="">Contact</a></li>
      </ul>
    </header>
    <section>
      <div>
        <h1>Tropic: Lorem Ipsum</h1>
        <p>
          Lorem ipsum dolor sit amet, consectetur adipisicing elit. Non dolor
          veniam reprehenderit numquam quia possimus magnam? Atque esse
          perspiciatis itaque veritatis architecto beatae? Exercitationem
          eligendi reiciendis earum maiores nobis! Beatae corrupti saepe
          architecto eaque eum iure illum iste, sint rerum ratione nulla alias
          at sunt, nisi praesentium. Fugit, labore quo.
```

```
        </p>
      </div>
      <aside>
        <h2>Sidebar</h2>
        <ul>
          <li><a href="">Blog Post</a></li>
          <li><a href="">Blog Post</a></li>
          <li><a href="">Blog Post</a></li>
          <li><a href="">Blog Post</a></li>
        </ul>
      </aside>
    </section>
    <footer>
      <p>All Right Reserved @ Iqbal | 2024</p>
    </footer>
  </body>
</html>
```

—— 「 SECTION [A] :: Paragraphs 」 ——

> ╰☆☆ `< p >` | `< pre >` | `< br/ >` | `< hr/ >` | `< h1 >` | `< h2 >` | `< h3 >` | `< h4 >` | `< h5 >` | `< h6 >` `< b >` | `< strong >` | `< i >` | `< em >` | `< u >` | `< mark >` | `< small >` | `< del >` | `< s >` | `< ins >` | `< sub >` | `< sup >` | `< q >` | `< blockquote >` | `< abbr >` | `< address >` | `< cite >` | `< bdo >` ☆☆╮

❑ Paragraphs

`< p >`

HTML paragraphs are defined with the `< p >` tag. e.g.

```
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>
```

❑ Pre-formatted Text

`< pre >`

The HTML `< pre >` tag is used to define pre-formatted text in HTML. It preserves both spaces and line breaks, displaying the text exactly as it appears in the HTML code. e.g.

```
<pre>
  My Bonnie lies over the ocean.
  My Bonnie lies over the sea.
  My Bonnie lies over the ocean.
  Oh, bring back my Bonnie to me.
</pre>
```

```
                    __________Output:
                      My Bonnie lies over the ocean.
                      My Bonnie lies over the sea.
                      My Bonnie lies over the ocean.
                      Oh, bring back my Bonnie to me.
```

❑ Break

`< br/ >`

The HTML break tag `< br/ >` is used to break the line of HTML paragraphs. It creates a line break without starting a new paragraph.

```
            <p>
              The purpose of a web browser Safari is to read HTML documents and displ
ay
              them. <br />
              The browser does not display the HTML tags.
            </p>


            __________Output:
            The purpose of a web browser Safari is to read HTML documents
            and display them.
            The browser does not display the HTML tags.
```

❑ Horizontal Rule

`< hr/ >`

Horizontal line of HTML paragraphs.

```
        <p> The purpose of a web browser Safari is to read HTML documents
        and display them. <hr/> The browser does not display the HTML tags. </p>


    __________Output:
    The purpose of a web browser Safari is to read HTML documents and display them.
    ____________________________________________________________________________
    The browser does not display the HTML tags.
```

❑ Heading

`< h1 >` | `< h2 >` | `< h3 >` | `< h4 >` | `< h5 >` | `< h6 >`

Headings are defined with the `< h1 >` to `< h6 >` tags.

`< h1 >` defines the most important heading. `< h6 >` defines the least important heading.

```
<h1> Heading 1 </h1> ---- <H1> Largest Text and Most Importance between Heading Tag
<h2> Heading 2 </h2> ---- Less than <H1>
<h3> Heading 3 </h3> ---- Less than <H2>
<h4> Heading 4 </h4> ---- Less than <H3>
<h5> Heading 5 </h5> ---- Less than <H4>
<h6> Heading 6 </h6> ---- Less than <H5>


__________Output:
                  Heading 1
                                Heading 2
                                Heading 3
                                Heading 4
                                Heading 5
                                Heading 6
```

❑ Formatting

`< b >` | `< strong >` | `< i >` | `< em >` | `< u >` | `< mark >` | `< small >` | `< del >` | `< s >` | `< ins >` | `< sub >` | `< sup >`

[👨🏽] **Bold Text** :: `< b >` `(__this tag for HTML4)`
The HTML `< b >` element defines **Bold** text, without any extra importance.

```
<b> This text is Bold </b>
```

> **Important Text** (bold text) :: `< strong >` `(__this tag for HTML5)`
> The HTML `< strong >` element defines **strong** text, with added semantic "strong" importance. For Example: `< strong >` `This text is Strong` `< /strong >`

[👨🏽] **Italic Text** :: `< i >` `(__this tag for HTML4)`
The HTML `< i >` element defines *italic* text, without any extra importance.

```
<i> This text is Italic </i>
```

> **Emphasized Text** (Italic text) :: `< em >` `(__this tag for HTML5)`
> The HTML `< em >` element defines *Emphasized* text, with added semantic
> For Example: `< em >` `This text is Emphasized` `< /em >`

[👨🏽] **Underlined Text** :: `< u >` `(__this tag for HTML4)`
The HTML `< u >` element defines Underlined text, with added semantic

```
<u> This text is Underlined </u>
```

> **Inserted Text** (Underlined Text) :: `< ins >` `(__this tag for HTML5)`
> The HTML `< ins >` element defines inserted (added) text.
> For Example: `< ins >` `This text is Emphasized` `< /ins >`

[👨🏻‍💻] **Marked Text** :: `< mark >`

The HTML `< mark >` element defines marked or highlighted text:

```
<h2>HTML<mark>Marked</mark>Formatting</h2>
```

[👨🏻‍💻] **Small Text** :: `< small >` The HTML `< small >` element defines

```
<h2>HTML<small>Small</small>Formatting</h2>
```

[👨🏻‍💻] **Strike Text** :: `< s >` The HTML `< s >` element defines ~~Strike~~ text.

```
<s> The text is strike </s>
```

[👨🏻‍💻] **Delete Text** :: `< del >` The HTML `< del >` element defines ~~deleted~~ (removed) text.

```
<p>My favorite color is <del>blue</del> red.</p>
```

[👨🏻‍💻] **Superscript Text** :: `< sup >` The HTML `< sup >` element defines superscript text.

```
<p>This is <sup>superscript</sup> text.</p>
```

[👨🏻‍💻] **Subscripts Text** :: `< sub >`

The HTML `< sub >` element defines subscript text.

```
<p>This is <sub>subscript</sub> text.</p>
```

❑ Quotes or Quotation

`< q >` | `< blockquote >` | `< abbr >` | `< address >` | `< cite >` | `< bdo >`

[👨🏻‍💻] **Short Quotations** :: `< q >` The HTML `< q >` element defines a short quotation.

```
<p>WWF's goal is to: <q>Build a future where people live in harmony with nature.</q>
</p>


__________Output:
WWF's goal is to: "Build a future where people live in harmony with nature."
```

Quotations

[👨🏻] **Quotations ::** `< blockquate >` The HTML `< blockquate >` element defines a section that is quoted from another source.

```
<p>Here is a quote from WWF's website:</p>
<blockquote cite="http://www.worldwildlife.org/who/index.html">
For 50 years, WWF has been protecting the future of nature. The world's leading conse
rvation organization, WWF works in 100 countries and is supported by 1.2 million memb
ers in the United States and close to 5 million globally.
</blockquote>


__________Output:
Here is a quote from WWF's website:
    For 50 years, WWF has been protecting the future of nature. The world's leading
    conservation organization, WWF works in 100 countries and is supported by 1.2
    million members in the United States and close to 5 million globally.
```

[👨🏻] **Abbreviations ::** `< abbr >` The HTML `< abbr >` element defines an abbreviation or an acronym.

```
<p>The <abbr title="World Health Organization">WHO</abbr> was founded in 1948.</p>
```

[👨🏻] **Contact Information ::** `< address >`
The HTML `< address >` element defines contact information (author/owner) of a document or an article

```
<address>
Written by John Doe. <br>
Visit us at: <br>
Example.com <br>
Box 564, Disneyland <br>
USA <br>
</address>


__________Output:
Written by John Doe.
Visit us at:
Example.com
Box 564, Disneyland
USA
```

[👨🏽‍🦱] **Work Title ::** `< cite >` The HTML `< cite >` element defines the title of a work.

```
<p><cite>The Scream</cite> by Edvard Munch. Painted in 1893.</p>


__________Output:
The Scream by Edvard Munch. Painted in 1893.
```

[👨🏽‍🦱] **Bi-directional Override ::** `< bdo >`
If your browser supports bi-directional `< bdo >` override (bdo), the next line will be written from right to left (rtl):

```
<bdo dir="rtl">This text will be written from right to left</bdo>


__________Output:
This line will be written from right to left
```

❑ Anchor Text and Hyperlink

`< a href =" # ">` `< a href =" # " target =" # ">`

The HTML hyperlink is defined using the `< a >` tag. It allows to create clickable links within web page.

Types of Hyperlink:

```
HTML Link: Hyperlinks
Example Code:
<a href="abc.com">Visit our Website</a>

HTML Link: Local Link
Example Code:
<a href="html_images.php">HTML Images</a>

HTML Link: Global Link
Example Code:
<a href="www.abe.com">HTML Images</a>
```

Attribute of **Hyperlink:**

Attribute	Details
href	Specifies the destination address. It can be an absolute or relative URL, or the name of an anchor.
hreflang	Specifies the language of the resource linked by the href attribute (which must be present with this one). Use language values from BCP 47 for HTML5 and RFC 1766 for HTML 4.
rel	Specifies the relationship between the current document and the linked document. For HTML5, the values must be defined in the specification or registered in the Microformats wiki.
target	Specifies where to open the link, e.g. in a new tab or window.
title	Specifies extra information about a link.

Attribute	Details
download	Specifies that the target will be downloaded when a user clicks on the hyperlink. The value of the attribute will be the name of the downloaded file.

Parameter of Attribute (__for Create **Hyperlink**):

Parameter	Attribute	Details
_blank	target	Opens the linked document in a new window or tab
_self	target	Opens the linked document in the same window/tab (this is default)
_parent	target	Opens the linked document in the parent frame
_top	target	Opens the linked document in the full body of the window
_framename	target	Opens the linked document in a named frame

Example 1: **Link to Another Site**

```
<a href="https://www.example.com/" target="_blank">Visit Example!</a>
<a href="https://www.example.com/" target="_top">HTML tutorial!</a>
<a href="https://www.example.com/" target="_parent">HTML tutorial!</a>
<a href="https://www.example.com/" target="_self">HTML tutorial!</a>

<a href="http://example.com/" rel="external">example site</a>
```

Example 2: **Link to an Anchor or Create Bookmarks**

```
Suppose, I've created a page (page1.html) on many topics. Now, I want to create a
Table of Contents at the top of the page with quick-links to specific sections. e.g.

                <h2 id="Topic1">First topic</h2>
                <p>Content about the first topic</p>

                <h2 id="Topic2">Second topic</h2>
                <p>Content about the second topic</p>

                          --------------------------
Now, we can use the anchor in our table of contents:

                <h1>Table of Contents</h1>
                <a href='#Topic1'>Click to jump to the First Topic</a>
                <a href='#Topic2'>Click to jump to the Second Topic</a>

These anchors are also attached to the web page they're on (page1.html).
So you can link across the site from one page to the other by referencing the page
and anchor name.

Remember, you can always <a href="page1.html#Topic1"> look back in the First Topic
</a> for supporting information.
```

Example 2: **Link to a page on the same site**

```
                    <a href="/example">Text Here</a>

The above example would go to the file example at the root directory (/) of the server.
If this link was on http://example.com, the following two links would bring
the user to the same location.
```

```
                    <a href="/page">Text Here</a>
                    <a href="http://example.com/page">Text Here</a>

Both of the above would go to the page file at the root directory of example.com.
```

Example 3: **Link that Dials a Number**

```
If the value of the href-attribute begins with tel:, your device will dial the number
when you click it.
                                            <a href="tel:11234567890">Call us</a>

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

Example 4: **Link that runs E-mail Client**

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

                                <a href="mailto:example@example.com">Send email</a>
This will put the email address example@example.com as the recipient for
the newly created email.


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

<a href="mailto:abc@example.com?cc=abc1@example.com&bcc=abc2@example.com">Send email</a


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

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

Example 5: **Link that runs Program Code**

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

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

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

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

The return false; is necessary to prevent your page from scrolling to the top when
the link to # is clicked. Make sure to include all code you'd like to run before it,
as returning will stop execution of further code.

for more..
```

Example 6: **Image as a Link**

```
<a href="default.asp">
    <img src="images/a.jpg" alt="HTML tuts" style="width:42px;height:42px;border:0;">
</a>
```

Example 7: **Button as a Link**

```
<button onclick="document.location = 'default.asp'">HTML Tutorial</button>
```

❑ Color

HTML colors are specified with predefined color names, or with RGB, RGBA, HEX, HSL, or HSLA values.

Color Space	Details	Syntax	URL
Color Name	Which Colors Support to HTML. All modern browsers support the following 140 color names.	`red` `blue` `green` `yellow` e..g.	view
RGB	Each parameter (red, green, and blue) defines the intensity of the color with a value between 0 and 255.	`rgb(red, green, blue);`	view
RGBA	RGBA color (red, green, blue and alpha) values are an extension of RGB color values with an alpha channel - which specifies the opacity for a color.	`rgb(red, green, blue, alpha);`	view
CMYK	CMYK colors is a combination of CYAN, MAGENTA, YELLOW , and BLACK. Computer screens display colors using RGB color values. Printers often presents colors using CMYK color values. CMYK is not supported in HTML, but it is suggested as a new standard in CSS4.	`cmyk(cyan, magenta, yellow, black)`	view
HEX	Hexadecimal color values are also supported in all browsers. A hexadecimal color is specified with: **#RRGGBB**. RR (red), GG (green) and BB (blue) are hexadecimal integers between 00 and FF specifying the intensity of the color. You can use upper case or lower case letters to specify hexadecimal values.	`# RR GG BB`	view
HSL	HSL stands for Hue, Saturation and Lightness. 1. Hue is a degree on the color wheel (from 0 to 360): → 0 (or 360) is red → 120 is green → 240 is blue 2. Saturation is a percentage value: 100% is the full color.	`hsl(hue, saturation, lightness)`	view

Color Space	Details	Syntax	URL
	3. Lightness is also a percentage; 0% is dark (black) and 100% is white.		
HSLA	HSLA color values are an extension of HSL color values with an alpha channel - which specifies the opacity for a color. An HSLA color value is specified with: hsla(hue, saturation, lightness, alpha), where the alpha parameter defines the opacity. The alpha parameter is a number between 0.0 (fully transparent) and 1.0 (fully opaque).	`hsla(hue, saturation, lightness, alpha)`	view
Opacity	The CSS opacity property sets the opacity for the whole element (both background color and text will be opaque/transparent). The opacity property value must be a number between 0.0 (fully transparent) and 1.0 (fully opaque).	`color; opacity: 0.0;`	

Examples

```
By Color Names:
<p style="color:red;">Bangladesh</p>

By RGB Color:
<p style="color: rgb(255,0,0);"> Bangladesh </p>

By RGBA Color:
<p style="color: rgba(255, 0, 0, 0.8);"> Bangladesh </p>

By HEX Color:
<p style="color: #FF0000;"> Bangladesh </p>

By HSL Color:
<p style="color: hsl(0, 100%, 50%);"> Bangladesh </p>

By HSLA Color:
<p style="color: hsl(0, 100%, 50%, 0.3);"> Bangladesh </p>

By Opacity:
<p style="background-color: #FF0000; opacity: 0.4;"> Bangladesh </p>
```

—— 「 SECTION [B] :: Image 」 ——

> ☆☆ `< img >` ☆☆

Images can improve the design and the appearance of a web page.

❑ Common Image Formats

Here are the most common image file types, which are supported in all browsers.

Abbreviation	File Format	File Extension
JPEG	Joint Photographic Expert Group image	`.jpg`, `.jpeg`, `.jfif`, `.pjpeg`, `.pjp`
PNG	Portable Network Graphics	`.png`
APNG	Animated Portable Network Graphics	`.apng`
GIF	Graphics Interchange Format	`.gif`
ICO	Microsoft Icon	`.ico`, `.cur`
SVG	Scalable Vector Graphics	`.svg`

❑ HTML Image

`< img >`

The `<img>` tag is empty, it contains attributes only, and does not have a closing tag. The `<img>` tag has two required attributes:

`src` - Specifies the path to the image

`alt` - Specifies an alternate text for the image

```
<img src="img_chania.jpg" alt="Flowers in Chania">

<!-- Images in Another Folder -->
<img src="/image/img_chania.jpg" alt="Flowers in Chania">

<!-- Images on Another Server/Website -->
<img src="https://www.w3schools.com/images/w3schools_green.jpg" alt="W3Schools.com">

Notes on external images:
External images might be under copyright. If you do not get permission to use it,
you may be in violation of copyright laws. In addition, you cannot control
external images; they can suddenly be removed or changed.
```

> 💡 **Note**: When a web page loads, it is the browser, at that moment, that gets the image from a web server and inserts it into the page. Therefore, make sure that the image actually stays in the same spot in relation to the web page, otherwise your visitors will get a broken link icon. **The broken link icon and the `alt` text are shown if the browser cannot find the image.**

❑ Image :: Size (Height & Width, or Style)

The width, height, and style attributes are all valid in HTML.

However, we suggest using the style attribute. It prevents styles sheets from changing the size of images:

```
<!DOCTYPE html>
<html>
<head>
```

```
            <style>
                    img {
                      width: 100%;
                    }
            </style>
</head>
<body>

        <img src="html5.gif" alt="HTML5 Icon" width="128" height="128">

        <img src="html5.gif" alt="HTML5 Icon" style="width:128px;height:128px;">

</body>
</html>
```

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

❑ Image :: Loading

The `loading` attribute specifies whether a browser should load an image immediately or to defer loading of off-screen images until for example the user scrolls near them.

> Add `loading="lazy"` only to images which are positioned below the fold.

Value	Description
eager	Default. Loads an image immediately
lazy	Defer loading of images until some conditions are met

```
<img src="img_chania.jpg" alt="Flowers in Chania" loading="eager">

<img src="img_chania.jpg" alt="Flowers in Chania" loading="lazy">
```

❑ Image :: Maps

The HTML `<map>` tag defines an image map. An image map is an image with clickable areas. The areas are defined with one or more `<area>` tags.

Ref. URL: https://imagemap.org

Ref. Tutorial: https://www.youtube.com/watch?v=K7IkHyUWEso

```
Image Maps
<img src="#" alt="image_name" usemap="#map_name">
<map name="map_name">
  <area shape="rect" coords="x1, y1, x2, y2" alt="image_name" href="#">
```

```
  <area shape="circle" coords="x, y, radius" alt="image_name" href="#">
  <area shape="poly" coords="x, y, radius" alt="image_name" href="#">
</map>


Notes:
Rectangle: x1, y1, x2, y2 specifies the coordinate of top-left (x1, y1)
           and bottom-right (x2, y2) corner of the rectangle.
Circle: x, y, radius specifies the center coordinates (x, y) and
        radius (radius) of circle.
Polygon: x1, y1, x2, y2, .., xn, yn specifies the coordinates of polygon.
         If the first and last coordinate pairs are not the same,
         the browser will add the last coordinate pair to close the polygon.


Example Code:
<img src="2.0.jpg" alt="Workplace" usemap="#workmap" width="400" height="379">
<map name="workmap">
    <area shape="rect" coords="34,44,270,350" alt="Computer" href="google.com">
    <area shape="rect" coords="290,172,333,250" alt="Phone" href="facebook.com">
    <area shape="circle" coords="337,300,44" alt="EarPhone" href="w3schools.com">
</map>
```

❑ Image :: Favicon

Favicons are icons that appear on the tab for a page along with the page's title.

A favicon is a small 16×16 pixel icon that serves as branding for your website. Due to its small size and resolution, the favicon may need to be an even smaller sizer or part of a company's original logo.

```
<!DOCTYPE html>
<html>
<head>
  <title>My Page Title</title>
  <link rel="icon" type="image/x-icon" href="/images/favicon.ico">
</head>
<body>

<h1>This is a Heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

The following table shows the file format support for a favicon image:

Browser	ICO	PNG	GIF	JPEG	SVG
Microsoft Edge	Yes	Yes	Yes	Yes	Yes
Google Chrome	Yes	Yes	Yes	Yes	Yes
Mozila Firefox	Yes	Yes	Yes	Yes	Yes
Opera	Yes	Yes	Yes	Yes	Yes
Safari	Yes	Yes	Yes	Yes	Yes

—— 「 SECTION [C] :: Audio & Video 」 ——

> ╰☆☆ `< Audio >` `< Video >` ☆☆╮

❑ Audio

The `<audio>` tag is used to embed sound content in a document, such as music or other audio streams.

```
<audio src="Faded.mp3" width="500" height="300" controls></audio>
```

Attribute	Value	Description
autoplay	autoplay	Specifies that the audio will start playing as soon as it is ready
controls	controls	Specifies that audio controls should be displayed (such as a play/pause button etc)
loop	loop	Specifies that the audio will start over again, every time it is finished
muted	muted	Specifies that the audio output should be muted
preload	auto metadata none	Specifies if and how the author thinks the audio should be loaded when the page loads
src	*URL*	Specifies the URL of the audio file

❑ Video

The `<video>` tag is used to embed video content in a document, such as a movie clip or other video streams.

```
<video src="infora.mp4" width="500" height="300" controls></video>
```

Attribute	Value	Description
autoplay	autoplay	Specifies that the video will start playing as soon as it is ready
controls	controls	Specifies that video controls should be displayed (such as a play/pause button etc).
height	*pixels*	Sets the height of the video player
loop	loop	Specifies that the video will start over again, every time it is finished
muted	muted	Specifies that the audio output of the video should be muted
poster	*URL*	Specifies an image to be shown while the video is downloading, or until the user hits the play button
preload	auto metadata none	Specifies if and how the author thinks the video should be loaded when the page loads
src	*URL*	Specifies the URL of the video file
width	*pixels*	Sets the width of the video player

—— 「 SECTION [D] :: Table 」 ——

> ╰☆☆ `< table >` ☆☆╮

HTML tables allow web authors to arrange data into rows and columns.

```
<table>
    <tr>
         <th>Name</th>
         <th>Address</th>
         <th>Emp ID</th>
     </tr>
     <tr>
         <td>Shihab</td>
         <td>Bogra</td>
         <td>53528</td>
      </tr>
      <tr>
         <td>AshiQ</td>
         <td>Rangpur</td>
         <td>53658</td>
       </tr>
</table>
```

HTML Table Tags:

Tag	Description
`<table>`	Defines a table
`<th>`	Defines a header cell in a table
`<tr>`	Defines a row in a table
`<td>`	Defines a cell or column in a table

—— 「 SEC [E] :: List 」 ——

> ╰☆☆ `< table >` ☆☆╮

HTML offers web authors three ways for specifying lists of information. All lists must contain one or more list elements. Lists may contain –

Order List: `<ol>` – An ordered list. This will use different schemes of numbers to list your items.

Unorder List: `<ul>` – An unordered list. This will list items using plain bullets.

Definition List: `<dl>` – A definition list. This arranges your items in the same way as they are arranged in a dictionary.

❑ List :: Order

If you are required to put your items in a numbered list instead of bulleted, then HTML ordered list will be used. This list is created by using
`<ol>` tag. The numbering starts at one and is incremented by one for each successive ordered list element tagged with `<li>`.

```
<ol type="A">
     <li>Beetroot</li>
     <li>Ginger</li>
     <li>Potato</li>
     <li>Radish</li>
</ol>

---------------------- Output:
            Beetroot
            Ginger
            Potato
            Radish

The "type" Attribute:
You can use type attribute for <ol> tag to specify the type of numbering you like.
By default, it is a number. Following are the possible options –
<ol type = "1"> - Default-Case Numerals.
<ol type = "I"> - Upper-Case Numerals.
<ol type = "i"> - Lower-Case Numerals.
<ol type = "A"> - Upper-Case Letters.
<ol type = "a"> - Lower-Case Letters.
```

❑ List :: Unorder

An unordered list is a collection of related items that have no special order or sequence. This list is created by using HTML `<ul>` tag. Each item in the list is marked with a bullet.

```
<ul>
      <li>Beetroot</li>
      <li>Ginger</li>
      <li>Potato</li>
      <li>Radish</li>
</ul>

---------------------- Output:
            ⚪ Beetroot
            ⚪ Ginger
            ⚪ Potato
            ⚪ Radish
```

❑ List :: Description

HTML and XHTML supports a list style which is called **definition lists** where entries are listed like in a dictionary or encyclopedia. The definition list is the ideal way to present a glossary, list of terms,
or other name/value list.

```
   <dl>
        <dt>HTML</dt>
```

```
        <dd>This stands for Hyper Text Markup Language</dd>
        <dt>HTTP</dt>
        <dd>This stands for Hyper Text Transfer Protocol</dd>
    </dl>

    ---------------------- Output:
        HTML
            This stands for Hyper Text Markup Language
        HTTP
            This stands for Hyper Text Transfer Protocol
```

—— 「 SECTION [F] :: Embed & Path 」 ——

> ╰☆☆ `< iframe >` ☆☆╮

❑ Embedded :: Iframe

An HTML `<iframe>` is used to display a web page within a web page.

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

Explain Code:
src = The src attribute specifies the URL (web address) of the inline frame page.
```

[👨🏻‍💻] Iframe :: Height & Width

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

```
        <!-- Example 1 -->
        <iframe src="demo_iframe.html" height="200" width="300"></iframe>

        <!-- Example 2 -->
        <iframe src="demo_iframe.htm" style="height:200px; width:300px;"></iframe>
```

[👨🏻‍💻] Iframe :: Remove the Border

By default, an `iframe` has a border around it.

✍️ Note: To remove the border (e.g. “frameborder="0" ”), add the `style` attribute and use the CSS `border` property

```
    <iframe src="demo_iframe.htm" style="border:2px solid red;"></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="https://www.w3schools.com" target="iframe_a">W3Schools.com</a></p>
```

[👨🏽] Iframe :: Some more... Attributes

The following attributes can be used with the <iframe> tag in HTML:

```
            <!-- ----- Example ----- -->
      <iframe
            width="600"
            height="300"
            src="https://www.youtube.com/embed/QyRW73JLJao"
            title="অভিশপ্ত রাজা | Double Gopal | Full Episode"
            allow="accelerometer; autoplay; clipboard-write; encrypted-media;
                  gyroscope; picture-in-picture; web-share"
            allowfullscreen
            >
      </iframe>

Some Attribute for iframe:
width="pixel"
height="pixel"
src="url"
title="type iframe title"
allow="parameters"       --- accelerometer | autoplay | clipboard-write |
                             encrypted-media | gyroscope | picture-in-picture |
                             web-share
allowfullscreen
scrolling="#"            --- auto | yes | no
name="name"
referrerpolicy="value"   --- no-referrer | no-referrer-when-downgrade | origin |
                             origin-when-cross-origin | same-origin | strict-origin |
                             strict-origin-when-cross-origin
sandbox="value"          --- no-values | allow-forms | allow-pointer-lock |
                             allow-popups | allow-same-origin | allow-scripts: |
                             allow-top-navigation
srcdoc="HTML_code"
loading="lazy"           --- lazy | eager
```

❑ File Paths

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

Path	Description
<img src="a.jpg">	“a.jpg” is located in the same folder as the current page
<img src="images/a.jpg">	“a.jpg” is located in the images folder in the current folder
<img src="/images/a.jpg">	“a.jpg” is located in the images folder at the root of the current web
<img src="../images/a.jpg">	“a.jpg” is located in the folder one level up from the current folder

[👨🏽] Absolute File Paths

An absolute file path is the full URL to a file:

```
                <!-- ---- Example-1 ---- -->
        <img src="https://www.abc.com/images/a.jpg" alt="Mountain">

                <!-- ---- Example-2 ---- -->
        <img src="a.jpg">
```

[👨🏻‍💻] Relative File Paths

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

```
                <!-- ---- Example-1 ---- -->
         <img src="images/a.jpg" alt="Mountain">

                <!-- ---- Example-2 ---- -->
        <img src="/images/a.jpg" alt="Mountain">

                <!-- ---- Example-3 ---- -->
        <img src="../images/a.jpg" alt="Mountain">
```

—— 「 SECTION [G] :: Form 」 ——

> ╰☆☆ `< form >` ☆☆╮

An HTML form is used to collect user input. The user input is most often sent to a server for processing.

```
<form action="">
     <input type="text" placeholder="Enter your name">
     <input type="email" placeholder="Enter your email">
     <input type="number" placeholder="Enter your phone number">
     <input type="password" placeholder="Enter your phone password">
     <input type="checkbox"><level for="">Agree</level>
     <input type="checkbox"><level for="">Not Agree</level>
     <input type="radio"><level for="">Condition</level>
     <input type="checkbox" checked disabled><level for="">Condition</level>
     <input type="submit" value="Send Message">
</form>
```

- `<input type=" text ">` - Displays a single-line text input field
- `<input type=" checkbox ">` - Displays a checkbox (for selecting zero or more of many choices)
- The `<input type=" checkbox ">` defines a **checkbox**.
- The `<input type=" radio ">` defines a radio button.
- The `<input type=" submit ">` defines a button for submitting the form data to a form-handler.