# Introduction to HTML

What is HTML

Okay, so, in order to jump into web development, you'll need to begin by learning to write HTML, which is the prime ingredient of most websites. Follow along with this guide to get started.

- HTML is the Hyper Text Markup Language.
- HTML is used to create an overall website structure.
- HTML is used to provide content (words, images, audio, video, etc.) to web pages.
- HTML is a language based on tags. They are defined in the angle brackets.
- A text editor such as Visual Studio Code or ATOM can be used to create an HTML file.

Following is a sample HTML code :

```html
<!DOCTYPE html>
<html>
    <head></head>
    <body>
        This is sample text...<br/> You are beginning to learn HTML.
    </body>
</html>
```

Create a file in any text editor like Visual Studio Code and ATOM, and type the above code and save it as a **.html file.**
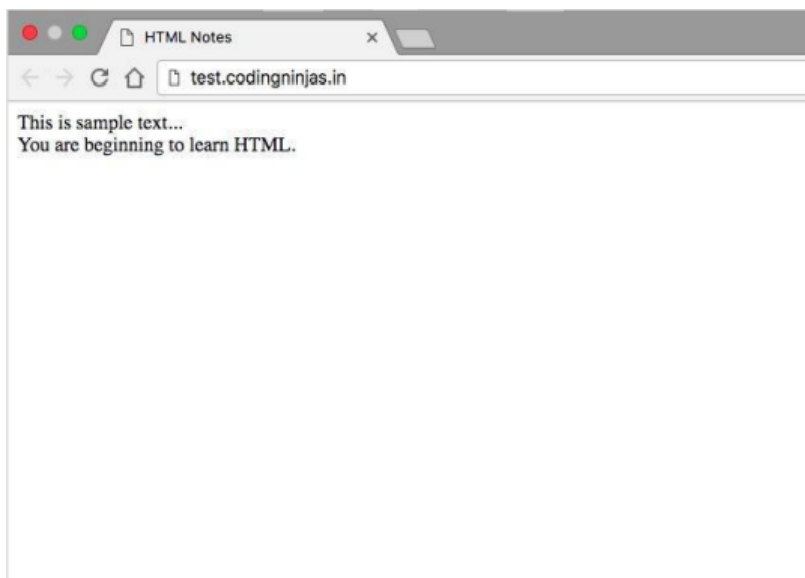
Comments in HTML

Comments are very important for a developer. Comments are **ignored** by the browser so, they will not **show up along with other text.** Generally, developers use comments to **explain their code to other developers** and to **mark something important** which needs editing. You can write a comment by placing the comment text between **<!-- --> tags.**

For example:

```
<!-- This is comment -->
```

TAGS

Tags define all elements of the document, i.e. they give meaning to the plain text of HTML.

- Tags **define** all document elements, i.e. the HTML plain text gives meaning.
- Two characters < and > surround HTML tags (They are called **angle brackets)**.
- The name of the tag can either begin with an **alphabet** or an **underscore(_)**.
- The element contents displayed between the **start** and **end tags** are **displayed**.
- Tags that have an **opening** and **closing** can have **any number of tags within them**.
- The **<H1>** and **<h1>** tags in HTML have the **same meaning,** i.e. tags are **not case sensitive.**
- The HTML tags are **usually available in pairs,** i.e. opening and closing (it's the same, with the tag name '/' at the beginning) tag.
  Eg: <html> and </html> is a tag that comes in pairs and <hr> does not have a closing tag.

Description of tags used till now:

- **<!DOCTYPE html>** tells the browser that the file to be displayed is the HTML5 page.
- **<html> </html>** is meant to contain all the HTML data and is the starting point for an HTML document.
- **<head> </head>** provides information on the document. This is not shown in the browser window.
- **<title> </title>** provides the title of the document
- All of the elements visible on the web page are contained in **<body></body>.**

NOTE: *There are also "self-closing" tags, whereby a br tag, for eg., will look like "**
" instead of simply "
**".*

To get the list of all valid tags in HTML5, visit:

https://developer.mozilla.org/en-US/docs/Web/HTML/Element

DOCTYPE

The **DOCTYPE** declaration specifies the **HTML version used to create the page**. It's the **first thing** you see on every web page when you open your HTML document. It appears before the <html> tag at the top of every page. The declaration of doctype is **not** an HTML tag. It's the **one HTML5 recommends.**

<!DOCTYPE html> is the syntax for doctype.

HTML ELEMENTS

Now we'll see different HTML elements. Elements are the things that actually make up the web page. Tags just define the beginning and end of the elements. Everything that a web page includes is an HTML element.
Eg: `<i>This text is italic</i>`
- The HTML element starts with a opening tag: <i>
- The content of the HTML element is: This text is italic.
- The HTML element ends with an closing tag: </i>

The basic elements used till now have been briefly described below:

Paragraphs

Paragraphs are **blocks of text** separated from each other by some space. They are defined using the **<p> and </p>** tags. When the p element ends, the next element appears in the next line.

Eg: here's a sample of code for <p> tag:

```
<!DOCTYPE html>
<html>
    <head>
        <title>p tag</title>
```

```
    </head>
    <body>
        <p>This is line 1.</p>
        <p>This is line 2.</p>
        <!-- trying to format the text without using p-tag -->
        This is line 1. This is line 2. This is line 3.
    </body>
</html>
```

It appears on a web browser like this:

This is line 1.

This is line 2.

This is line 1. This is line 2. This is line 3.
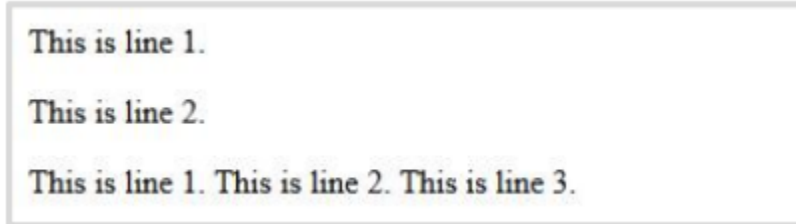
NOTE: *When formatting without a p-tag, new lines are appended on the current line.*

*This happens because the **spacing of text doesn't matter to the browser.***

Headings

These are HTML tags that are used to indicate that some content should be treated as **headings**. The headings are divided into six levels: **h1, h2, h3, h4, h5**, and **h6.** Among them, **h1** is the **highest** level heading and **h6** is the lowest level heading.

Eg: here's a sample of code for H tags

```
<!DOCTYPE html>
<html>
    <head>
        <title>Heading Levels</title>
    </head>
    <body>
        <h1>Heading level 1</h1>
        <h2>Heading level 2</h2>
        <h3>Heading level 3</h3>
        <h4>Heading level 4</h4>
        <h5>Heading level 5</h5>
        <h6>Heading level 6</h6>
    </body>
</html>
```

The content appears as:

BR Tag

**
 tag** can be used to make a **single line split** between the contents of the tab. This means that when this tag is used between a single line, the **contents of after this tag will pass to the next line**. Do not use it to allow space between a block of elements (eg., paragraph and heading).

Eg.,

```
<h3>We are studying in<br>Coding Ninjas</h3>
```

will show the heading as:

LISTS

Lists are used to **group different pieces of information together** so that they are **easily linked** and **easy to read**. Lists help construct a **well-structured**, more open, and **easy-to-maintain** document from a structural standpoint.

There are three types of lists to pick from: **ordered**, **unordered**, and **description lists**.
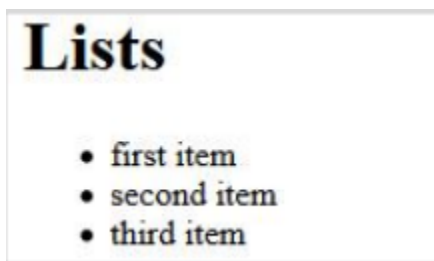
Unordered Lists

It's used to group a group of similar objects that aren't arranged in any specific order. Where the counting of objects isn't necessary, unordered lists are used. Bullets are used by default to separate the items.

They are defined using the <ul> tag. Eg:

```
<!DOCTYPE html>
<html>
    <head>
        <title>Unordered Lists</title>
    </head>
    <body>
        <h1>Lists</h1>
        <ul>
            <li>first item</li>
            <li>second item</li>
            <li>third item</li>
        </ul>
    </body>
</html>
```

The output is as follows:

Lists

- first item
- second item
- third item

HTML provides an interesting feature to change the style of the list item marker. There are

There are 4 types of styles in unordered lists:

- ● *type="disc"* - sets the list item marker to a bullet (default)
- ○ *type="circle"* - sets the list item marker to a circle
- ■ *type="square"* - sets the list item marker to a square
   type="none" - the lists items will not be marked

For example, to create a list with *type=circle:*

```
<ul type="circle">
    <li>a</li>
    <li>b</li>
```

```
</ul>
```
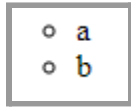
NOTE: *The above styles can be produced by using the* *'type'* *attribute. However, this attribute is now* ***not supported in HTML5*** *and you now need to change the style using CSS(we will learn later about it).*
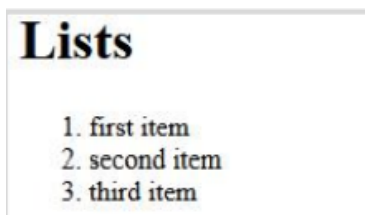
Ordered Lists

It is used in a certain order to group a number of related items. When the n**umbering of items is necessary**, ordered lists are used. By default, **numerical numbers** follow the items.

They are defined using the **<ul>** tag. Eg:

```
<html>
    <head>
        <title>Unordered Lists</title>
    </head>
    <body>
        <h1>Lists</h1>
        <ol>
            <li>first item</li>
            <li>second item</li>
            <li>third item</li>
        </ol>
    </body>
</html>
```

The output is as follows:

Similarly, like the unordered lists, there are also different types of ways to number the ordered lists using the **'type'** attribute:
1.   *type="1"* - the numbering will contain **numbers** (default)
A.   *type="A"* - the numbering will contain **uppercase letters**
a.   *type="a"* - the numbering will contain **lowercase letters**
I.   *type="I"* - the numbering will contain **uppercase roman numbers**
i.   *type="i"* - the numbering will contain **lowercase roman numbers**

Now, *what if you want to change the starting numbering of the lists?*

HTML has got the solution for it: the **'start'** attribute. So, if we change <ol> to **<ol start="7" >**, you will now see the output as:

Lists

7. first item
8. second item
9. third item
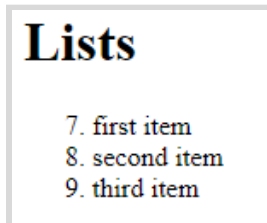
Description Lists

A list of definitions is **not the same** as a list of items. This is a **collection of items with an explanation.**

The **<dl> tag** is used to **start** a definition list. The **<dt>** tag is used to **begin** each definition - list term. The **<dd> tag** is used to **begin each definition - list definition.**

In comparison to ordered and unordered lists, description lists are **very specific in their application** and thus are **rarely used**. However, whenever a structure such as a list of terms and their descriptions is required, description lists are ideal.

Eg:

```
<!DOCTYPE html>
<html>
    <head>
        <title>Description Lists</title>
    </head>
    <body>
        <h2>Description List</h2>
        <dl>
            <dt>Coffee</dt>
            <dd>- black hot drink</dd>
            <dt>Milk</dt>
            <dd>- white cold drink</dd>
        </dl>
    </body>
</html>
```

The output is as follows:

A Description List

Coffee
    - black hot drink
Milk
    - white cold drink

NESTING ELEMENTS

HTML elements can be nested i.e. **elements can contain elements.** Actually, all HTML documents consist of nested HTML elements

Eg:

```html
<ul>
    <li>first item</li>
    <li>second item
        <!-- Look, the closing </li> tag is not placed here! -->
        <ul>
            <li>second item first sub item</li>
            <li>second item second subitem
                <!-- Same for the second nested unordered list! -->
                <ul>
                    <li>second item second subitem first sub-sub item</li>
                    <li>second item second subitem second sub-sub item</li>
                    <li>second item second subitem third sub-sub item</li>
                </ul>
            </li>
            <!-- Closing </li> tag for the list that contains the third unordered
    list -->
            <li>second item third subitem</li>
        </ul>
    </li>
    <!-- Here is the closing </li> tag -->
    <li>third item</li>
</ul>
```

This will give the output as:

Lists

- first item
- second item
  - second item first subitem
  - second item second subitem
    - second item second subitem first sub-subitem
    - second item second subitem second sub-subitem
    - second item second subitem third sub-subitem
  - second item third subitem
- third item

NOTE: *There is no limitation to the depth of nested lists. Although it is true for all paired/container tags, we should be careful in nesting elements inside each other and should only do something meaningful.*

IMAGES IN HTML

You can also **show pictures on a document** with HTML. The **<img> tag** is **used to specify images in HTML.**

You **must use the src attribute** to **display an image** on a page. **Src** is the **'source**.' The src's value is the **URL of the picture that you wish to show on your page.**

The url may be **relative** or **absolute**.

The syntax of defining an image:

```
<img src="images/logo.png">
```

We wrote '**images/logo**,' as you can see, in the **src attribute**. This is a **relative url** example. **Relative** and **absolute** urls are available. After sometime, we'll discuss them later.

The image will now be displayed on the page like:

Some points you need to know:

- tag is a self-closing tag which means that it doesn't contain the closing tag.
- The src tag can contain both relative and absolute paths, as well as internet image links.
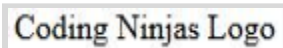
The ALT Attribute

The **alt attribute** or **alternative text** will inform the reader **what it lacks on a page when the browser cannot load pictures**. Instead of the picture, the **browser displays the alternative text.**

Now, we can use the **alt** attribute as:

```
<img src="images/logo.png" alt="Coding Ninjas image">
```

The text would be seen now as:

Coding Ninjas Logo

NOTE: *It is a good practice to include the "**alt**" attribute for each image on a page*

Height and Width

You may **explicitly specify the height and width** of an image with the attributes **height="value"** and **width="value."** The value is given in **pixels by default**.

Eg.,

```
<img src="images/logo.png" alt="Coding Ninjas image" height="500"
width="500">
```

This will fix the height and width of the image to 500px(pixel). There is an alternate for height and width attribute in CSS. We can come to this later.

*NOTE: The value provided should be in **numerical** form. Pixel is a unit of measurement, to set the dimensions of the image.*

ATTRIBUTES

More HTML elements may be supplied and their actions can be controlled by **attributes** on your page. For example:;

```
<img src="images/logo.png" alt="Coding Ninjas image" height="500"
width="500">
```

In above example,
- The *src* attribute is used to define the url/source link of the image file, which is *images/logo.png* in this case.

- The *alt* attribute defines the alternate text which will be shown if the image doesn't load due to any reason; **Coding Ninjas image** will be shown as alternate text.

- The *height* attribute can be used to set the image's height in pixels.

- Similarly, *width* attribute is used to set the width of the image in pixels.

Some points to remember:

I.   Attributes always come in name/value pairs like this: **attribute_name="value".**

II.   Attributes are always added to the **start tag** of an HTML element.

III.   The attribute value should always be mentioned in double quotes (" ") or in single quotes(' ').

IV.   If the attribute value contains quotes itself, then it is necessary to use single quotes, like: **name='John "Ghostemane" Doe'**

ANCHOR TAG

The **<a> tag** sets the connection to link between pages. This tag defines a link.

You may have noticed that by clicking on a link, a **new page appears** that can be on the **same page** or **another**. These web pages' links are **linked together**. They enable us to **navigate to another website without having to type in its URL each time**. These are **external links,** which allow you to **connect to other websites.**

The links may be **internal**, too, so that the c**ontents of the same page are linked**.
Eg: link to the page top or link to a particular page material.

In all browsers, by default, links appear as follows:
- An **unvisited** link is **underlined** and **blue**
- A **visited** link is highlighted in **purple** and **underlined**.
- A link that is **active** is highlighted in **red**.

Href Attribute

The **<a>** element's most significant attribute, **href**, indicates **the destination of the link**. This means that the href attribute is used to **refer to the document** that is **linked to the relation.**

Eg;

```html
<h2>A Great place to practice coding</h2>
<p> Take daily challenges at
    <a href="http://www.codingninjas.in/students/assignments">Coding
    Ninjas</a>.
      <!-- clickable content for the link is mentioned here -->
      <!-- any html element can be included here like image, gif, etc. -->
</p>
```

You will see this:

A Great place to practice coding

Take daily challenges at Coding Ninjas.

An anchor tag may indicate any **web-based resource**: **an HTML page, a picture, a sound file, a video, etc.** Both of these are referred to as **external connections.**

*NOTE: You need to remember that here also, we can provide the **relative url** of a file as a value to href attribute. Eg: **href="/home/myPC/Documents/test.html".***

Relative and Absolute Linking

For specifying **local links**, **relative links are used**, i.e. **links to root files**. **Absolute links** are used in **external links,** i.e. the web pages' **URL**.

Relative page-related links feature. So the browser searches for file locations in relation to the current page when a user clicks the relative connection.

Four situation arises in this case:-

- *The file is present in the same folder* - In this case, the name of the file is provided. Eg: `<a href="relativeFile.html">Click Me</a>`, will look for the file inside the same folder.
- *The file is present in the subfolder* - In this case, the name of the file provided is preceded by the folder names according to hierarchy.
  Eg: `<a href="subfolder/down/relativeFile.html">Click Me</a>`, will move to the 'subfolder' folder, then to 'down' folder and look for the file inside it.
- *The file is present somewhere in the parent folder -* In this case, to move one folder above use '../'. For instance: Click Me, step into the parent folder and search for the inside address.

- ***The file is present in another subfolder of the parent folder*** - This case covers the above two cases.

  For instance: Click Me to switch to the parent folder, then to the 'subfolder' folder, and search for the inside of the folder.

The full web address of the web page you want to go is indicated by absolute links. eg:: <al href="https://1603.de/" Eg: Click Me to guide the browser to the URL you like.

target Attribute

With the **target** attribute, you can define where the **linked document will be opened**.

The target attribute has the following values:

- **_self**: Load the URL to the current tab on its own. This is the default setting.
- **_blank**: Load the URL to the new tab or browser window.
- **_parent**: Load the URL in the context of the parent browsing. It's the same thing as itself if there are no parents.
- **_top**: Load URLs in the high-level context of browsing. It is the same thing if there is no parent.

In a new browser window the following line opens the document:

```
<a href= <a href= "http://www.codingninjas.in/pupils/attributions
"objective=" blank" "> > > >
Ninjas</a> coding
```

NOTE: *By default, the linked page will be displayed in the **current browser window.*** It can be cumbersome to remember all this HTML code. This HTML cheatsheet can be downloaded from the link below to refer to all HTML tags and attributes without the following:

https://html.com/wp-content/uploads/html-cheat-sheet.pdf