

# 1. WHAT IS HTML?

- HTML stands for **Hyper Text Markup Language**.
- HTML describes the **structure of web pages** using markup.
- HTML is used to provide the content(words, images, audio, video, and so on) to the web pages.
- HTML is a tag based language. They are defined within the **angle brackets**.
- HTML file can be created using a text editor.

Following is a sample HTML code :-

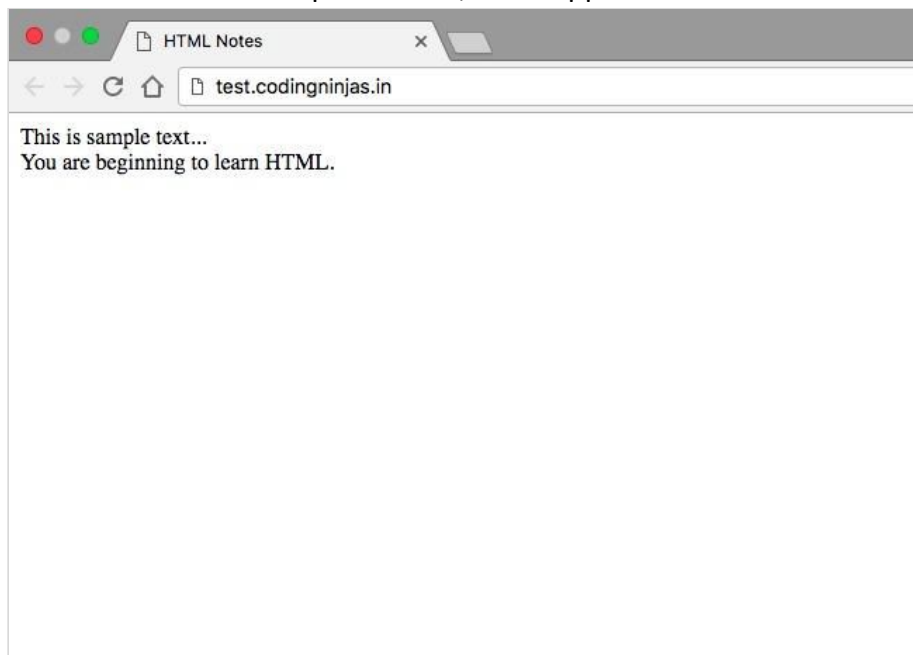
```
<!DOCTYPE html>
<html>
  <head></head>

  <body>
    This is sample text...<br/> You are beginning to learn HTML.
    <!-- We use this syntax to write comments -->
    <!-- Page content and rest of the tags here.... -->
    <!-- This is the actual area that gets shown in the browser -->
  </body>

</html>
```

Remember to save the above file with **".html" extension**.

Start the browser and open the file; it will appear as:



## 2. COMMENTS IN HTML

The comment tag `<!--...-->` is used to insert comments in the source code. Comments are **not displayed in the browsers**.

You can use comments to explain your code, which can help you when you edit the source code at a later date. This is especially useful if you have a lot of code.

## 3. TAGS

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

- HTML tags are surrounded by the **two characters < and >** (They are called **angle brackets**).
- The tag name can either start from an **alphabet** or an **underscore(\_)**.
- The text between the start and end tags is the **element content**.
- Tags with an opening and closing can have **any number of tags within itself**.
- HTML tags are **not case sensitive**, `<p>` means the same as `<P>`.
- HTML tags normally **comes in pairs(container tags)**, i.e. both opening and closing(it is same, just the name of the tag with character '/' in the beginning) tag.
- Eg: `<html>` and `</html>` is a tag that comes in pair.
- Eg: `<br>` does not have a closing tag.

### 3.1. Description of tags used till now

- `<!DOCTYPE html>` tells the browser that the file being displayed is HTML5 page.
- `<html>` `</html>` meant to contain all the html data and is the **start of an HTML document**.
- `<head>` `</head>` provides **information about the document**. It is not displayed in the browser window.
- `<title>` `</title>` provides a **title for the document**.
- `<body>` `</body>` contains all the **things visible** on the web page.

**NOTE:** You might come across "self-closing" tags, whereby a `br` tag, for eg., will look like `"<br/>"` instead of simply `"<br>"`.

#### EXTRA:

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

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

## 4. DOCTYPE

The **DOCTYPE** declaration describes **what version of HTML the page is written in**. It is the very first thing in your HTML document that you see in every web page. It is written at the top of every page before the <html> tag.

The doctype declaration is not an HTML tag. It is the one recommended by HTML5.

The syntax for doctype is: **<!DOCTYPE html>**.

## 5. 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: this is an HTML element: `<b>This text is bold</b>`

- The HTML element starts with an opening tag: `<b>`
- The content of the HTML element is: *This text is bold*
- The HTML element ends with a closing tag: `</b>`

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

### 5.1. Paragraphs

Paragraphs are **blocks of text** separated from each other by some space. They are defined using the **<p> and </p>** tags. When p element ends, the *next element appears in 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 `p`-tag, new lines are appended on the current line. This happens because the **spacing of text doesn't matter to the browser**.

## 5.2. Headings

These are tags in HTML to mark some content as headings. In fact, there are six different levels of headings **h1**, **h2**, **h3**, **h4**, **h5** and **h6**. Among them *h1* defines the largest heading and *h6* defines the smallest 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:

**Heading level 1**

**Heading level 2**

**Heading level 3**

**Heading level 4**

**Heading level 5**

**Heading level 6**

### 5.3. BR Tag

**<br>** tag is used to introduce a **single line break** between the contents. This means that when this tag is used in between a single line, the content after this tag will move to the next line. Do not use it to provide space between block of elements( eg., paragraph and heading).

Eg.,

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

will show the heading as:

**We are studying in  
Coding Ninjas**

## 6. LISTS

Lists are used to **group together related pieces of information** so they are clearly associated with each other and easy to read. Lists are good from a **structural** point of view as they help create a well-structured, more accessible, easy to maintain document.

HTML supports ordered, unordered and definition lists.

### 6.1. Unordered Lists

It is used to group a set of related items in **no particular order**. Unordered lists are used when the numbering of items is not required. By default the items are followed by **bullets**.

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:



HTML provides an interesting feature to change the style of the list item marker. There are 4 types of style 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

**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).

## 6.2. Ordered Lists

It is used to group a set of related items in a **specific order**. Ordered lists are used when the numbering of items is required. By default the items are followed by **numerical numbering**.

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

```
<!DOCTYPE html>
<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:

## Lists

1. first item
2. second item
3. third item

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 list items will be numbered with numbers (default)
- A. **type="A"** - The list items will be numbered with uppercase letters
- a. **type="a"** - The list items will be numbered with lowercase letters
- I. **type="I"** - The list items will be numbered with uppercase roman numbers
- i. **type="i"** - The list items will be numbered with 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:

### 6.3. Description Lists

A definition list is not a list of items. This is a list of terms and explanation of the terms.

A definition list starts with the `<dl>` tag. Each definition - list term starts with the `<dt>` tag. Each definition - list definition starts with the `<dd>` tag.

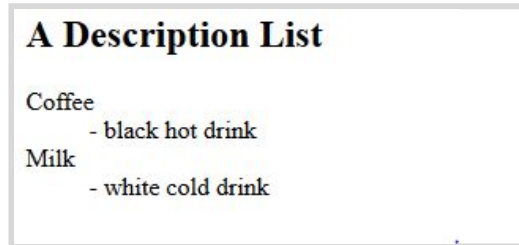
Description lists are very specific in use as compared to ordered and unordered lists and hence are very less used. But whenever, a structure like a list of terms and their description is required, the description lists are the perfect elements.

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:



## 7. NESTING ELEMENT

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

Eg:

```
<ul>
  <li>first item</li>
  <li>second item
    <!-- Look, the closing </li> tag is not placed here! -->
    <ul>
      <li>second item first subitem</li>
      <li>second item second subitem
        <!-- Same for the second nested unordered list! -->
        <ul>
          <li>second item second subitem first sub-subitem</li>
          <li>second item second subitem second sub-subitem</li>
          <li>second item second subitem third sub-subitem</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, but we should be careful in nesting elements inside each other and should only do for something meaningful.*

## 8. IMAGES IN HTML

With HTML you can also display images in a document. In HTML, images are defined with the **<img>** tag.

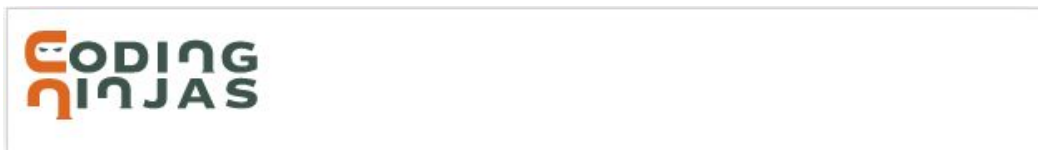
To display an image on a page, you need to use the **src** attribute. Src stands for "source". The value of the src attribute is the **URL of image** you want to display on your page.

The syntax of defining an image:

```

```

The image will now be displayed on the page like:



Some points you need to know:

- <img> 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.

### 8.1. The ALT Attribute

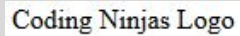
The **alt** attribute or **alternate text** tells the reader what he or she is missing on a page if the browser can't load images. The browser will then display the alternate text instead of the image.

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

```

```

The text would be seen now as:



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

## 8.2. Height and Width

The height and width of an image can be set directly by using the **height="value"** and **width="value"** attributes. By default, the value provided is in pixels.

Eg.,

```

```

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

## 9. ATTRIBUTES

Attributes can provide **additional information** about the HTML elements on your page and **control their behaviour**.

Eg: `<tag_name attribute_name="value_value">Content Enclosed</tag_name>`.

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. Attribute values should always be enclosed in quotes. **Double style quotes (" ")** are the most common, but **single style quotes (' ')** are also allowed.
- iv. In some rare situations, like when the *attribute value itself contains quotes*, it is necessary to use single quotes: **name='John "ShotGun" Nelson'** and vice-versa.

## 10. ANCHOR TAG

The `<a>` tag defines a **hyperlink**, which is used to link from one page to another.

You have seen that clicking on a link opens a new page may be on the same page or another. These web pages are connected using links. They give us the ability to go to a different web page without each time entering its URL. This kind of links are **external links** i.e. they help in connecting to external web pages.

Links can also be **internal** which means that they will be linking the content within the same page. Eg: link to the top of the page or any link to any specific content on the page.

By default, links will appear as follows in all browsers:

- An **unvisited** link is underlined and blue
- A **visited** link is underlined and purple
- An **active** link is underlined and red

## 10.1. href Attribute

The most important attribute of the <a> element is the **href** attribute, which indicates the **link's destination**. In other words, the href attribute is used to address the document to link to.

Eg:

```
<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](http://www.codingninjas.in/students/assignments).

An anchor can point to any resource on the Web: an HTML page, an image, a sound file, a movie, etc. These all are known as **external links**.

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

## 10.2. Relative and Absolute Linking

**Relative linking** is used to **specify local links**, i.e. link to files inside the root folder.

**Absolute linking** is used to **specify outside links**, i.e. URL of the web pages.

Relative links works relative to the page. So, when a user clicks a relative link, the browser looks for the location of the file relative to the current page.

Four situation arises in this case:-

- **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.
- **File is present in the subfolder** - In this case, the name of the file provided is preceded with 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.
- **File is present somewhere in the parent folder** - In this case, to move one folder above use '../'. Eg: `<a href="../relativeFile.html">Click Me</a>`, will move to the parent folder and look for the file inside it.
- **File is present in another subfolder of parent folder** - This case covers above two cases. Eg: `<a href="../subfolder/relativeFile.html">Click Me</a>`, will move to the parent folder, then to folder named 'subfolder' and look for the file inside it.

Absolute links provides complete web address of the web page where you want to go. Eg: `<a href="https://www.codingninjas.in/">Click Me</a>`, will make the browser directly go to the specified URL.

### 10.3. 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 into the current tab itself. This is the default.
- **\_blank**: load the URL into a new tab or browser window.
- **\_parent**: load the URL into the parent browsing context. If there is no parent, this behaves same as **\_self**.
- **\_top**: load the URL into the top-level browsing context. If there is no parent, this behaves same as **\_self**.

The line below will open the document in a new browser window:

```
<a href="http://www.codingninjas.in/students/assignments"
target="_blank">Coding Ninjas</a>
```

**NOTE:** By default, linked page will be displayed in the **current browser window**.