Introduction to HTML

What is HTML?

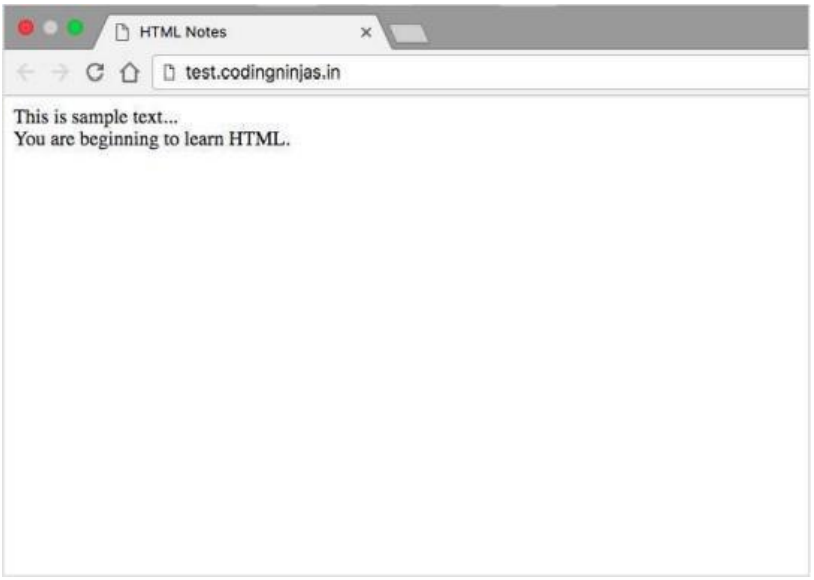
HTML is the most basic building foundation of any web page. Follow along with this guide to get started.

- HTML is the HyperText 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, Sublime, or ATOM can be used to create an HTML file.
- You can download sublime text editor from here: https://www.sublimetext.com/3

Following is a sample HTML code

```
<!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

Generally, developers use comments to **explain their code to other developers** or to **mark something important** which needs editing. Comments are **ignored** by the browser **hence won't be seen on the webpage along with other content.**

You can write a comment by placing the comment text between **<! > tags.**
For example:

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

Comments can't be nested, which means a comment can't be put inside another comment.

TAGS

Tags are used to represent HTML elements. These can be seen as keywords that define how a web browser will format and display the website content.

- Tags **define** all document elements, i.e. they give meaning to the plain HTML text.
- 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 are displayed between the **start** and **end tags**
- 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.

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

Description of tags used till now:

	Tag	Description
1	**<!DOCTYPE html>**	Specifies that HTML version 5 is used to create the web page
2	**<html> </html>**	Root container for all other HTML elements of the web page (including the head tag)s
3	**<head> </head>**	The <head> element is a container for metadata(data about data) Metadata typically define the document title, character set, styles, scripts, and other meta information.
4	**<title> </title>**	Provides the title of the document and is displayed in the tab in the browser
5	**<body></body>**	Contains all of the elements visible on the web page

EXTRA:

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

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

These can be explored whenever required while making a website.

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.

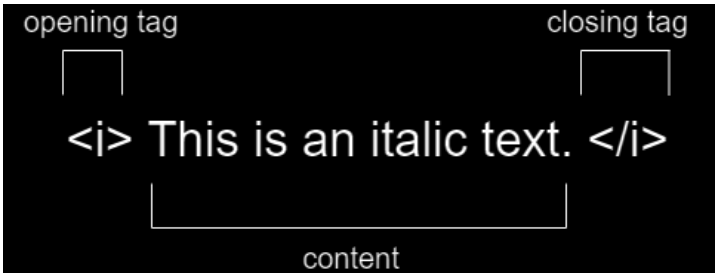
It appears before the <html> tag at the top of every page.

The declaration of doctype is **not** an HTML tag. It's the **one HTML5 recommends.**

<!DOCTYPE html> is the syntax for doctype.

HTML ELEMENTS

HTML Elements are the things that actually make up the web page. Tags ~~just~~ define the beginning and end of HTML elements. A web page can be seen as a collection of HTML elements.



The basic elements used till now have been briefly described below

	HTML Element	**Description**
1	** CONTENT **	Paragraph tag
2	** CONTENT **	Heading tag
3	** **	Break tag - to enter into a new line

Paragraphs

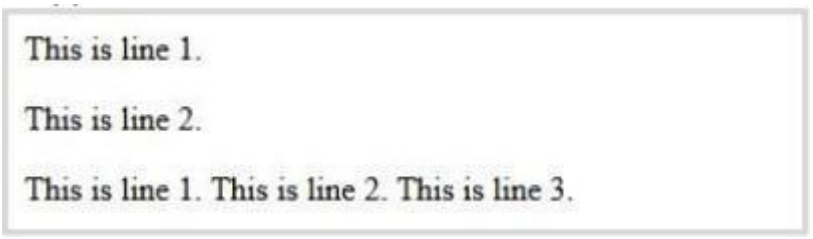
Paragraphs are **blocks of text** separated from each other by some space.

They are defined using the ** ** and ** ** tags. When the p element ends, the next element appears in the next line.

Eg: here's a sample of code for 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:



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.***

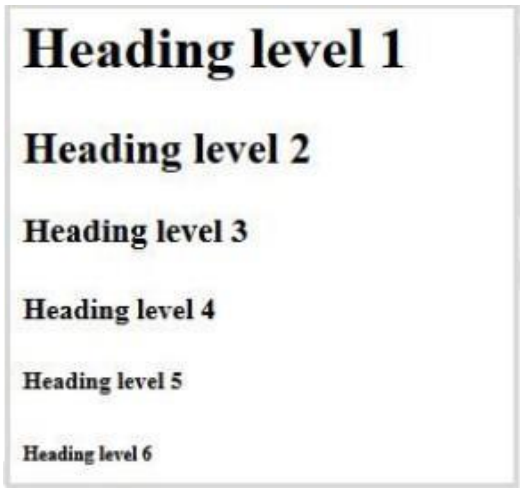
Headings

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 a



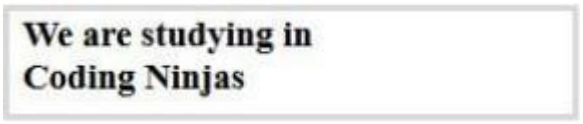
BR Tag

**
 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 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**.

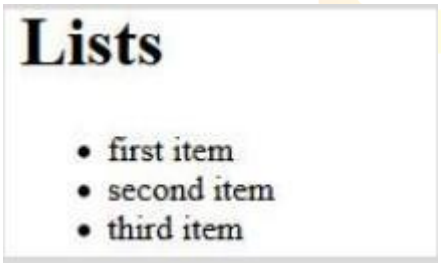
Unordered Lists

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:

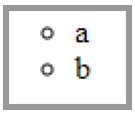


HTML provides an interesting feature to change the style of the list item marker.
There are 4 types of **styles in unordered lists:**

- ***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

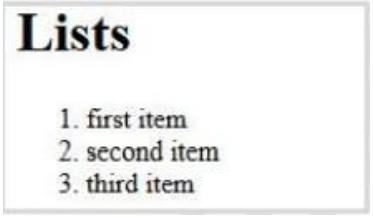
It is used in a certain order to group a number of related items.

When the n**umbering of items is necessary**, ordered lists are used. By default, **numerical numbers** follow the items.

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

```
<html>
    <head>
        <title>  ordered 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

Description Lists

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

Tag	Description
<dl> tag	to **start** a definition list.
<dt> tag	to **begin** each definition - list term.
<dd> tag	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:

```html
<!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:

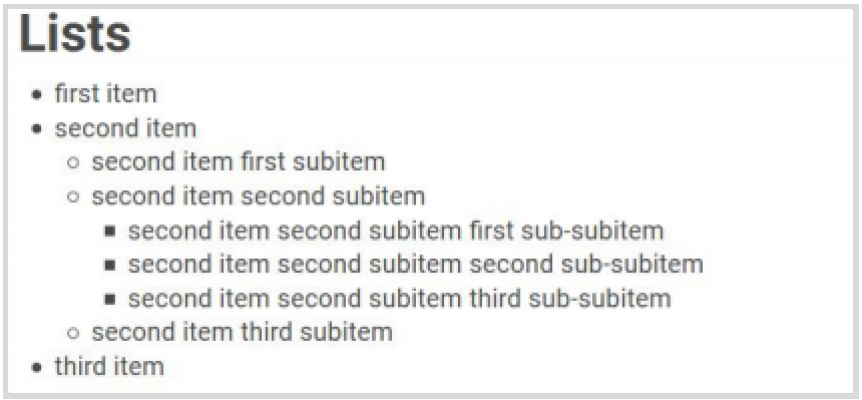
NESTING ELEMENTS

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 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



***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

The **<img> tag** is **used to specify images in HTML.**

Attributes of Image Element

Attribute	Description
src	to specify the source of the image
alt	to specify alternate text (it gets displayed if due to some issue image doesn't get displayed)

The src Attribute

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.
We'll discuss them later.

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.
The image will now be displayed on the page like

Some points you need to know:

- image 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.

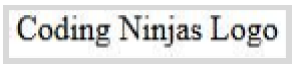
The ALT Attribute

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



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

The Height and Width Attributes

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 alternative for height and width attributes 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

HTML elements can have certain characteristics associated with them which are called attributes. These attributes provide more information about the element For example:

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

In the 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; the **Coding Ninjas image** will be shown as alternate text.
- The ***height*** attribute can be used to set the image's height in pixels.
- Similarly, the ***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 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. 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**. Links can either be external or internal.

External Links	Internal Links
Enables navigation between different web pages without having to type its URL each time. Helps to connect to other websites.	Enables navigation within a web page. 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**. T his means that the href attribute is used to **refer to the document** that is **linked to the relation.**

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:

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 page's **URL**. The browser searches for file locations concerning 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 to is indicated by absolute links.
eg:: Click Me to guide the browser to the URL you like.

The Target Attribute

The **target** attribute defines **where the linked document will be opened**. It 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