Introduction to HTML

What is HTML?

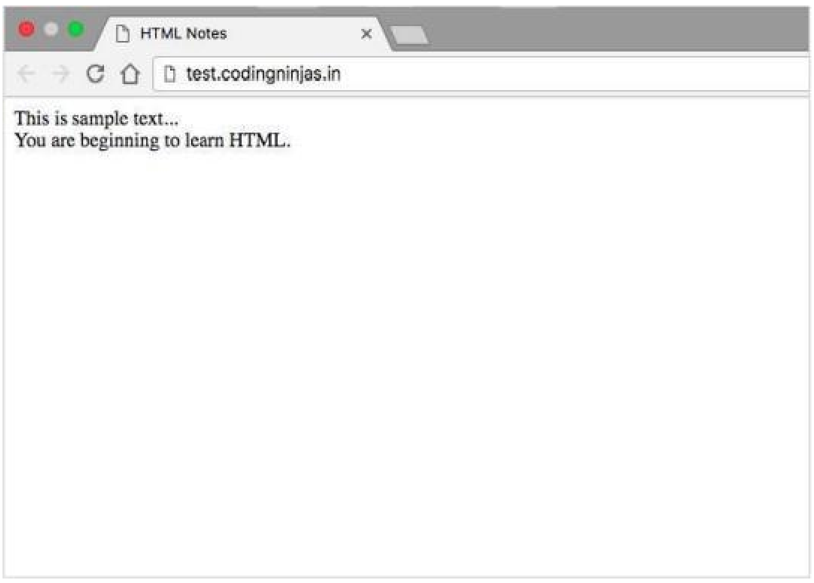
HTML is the most basic building foundation of any web page. Follow along with this guide to get started.

- 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.**



This is sample text...
You are beginning to learn HTML.

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 "**<br/>**" instead of simply "**<br>**".*

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.

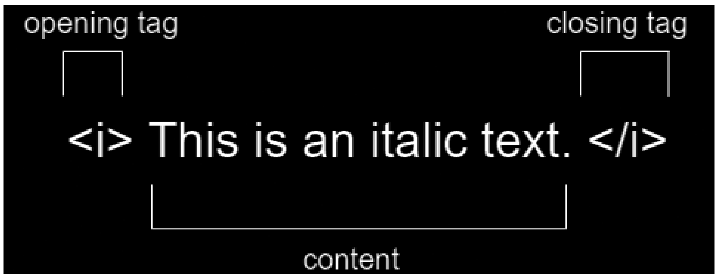
It appears before the <html> tag at the top of every page.

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	**<p> CONTENT </p>**	Paragraph tag
2	**<h1> CONTENT </h1>**	Heading tag
3	** **	Break tag - to enter into a new line

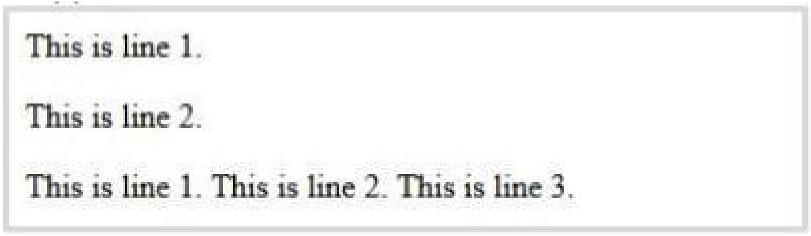
Paragraphs

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

They are defined using the **<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:

```html
<!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.***

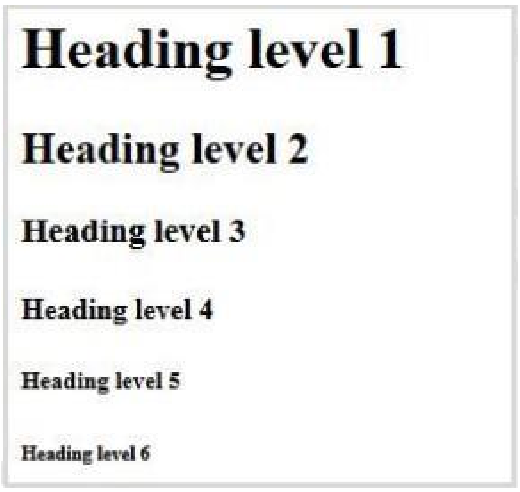
Headings

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

```html
<!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


Heading level 1

Heading level 2

Heading level 3

Heading level 4

Heading level 5

Heading level 6

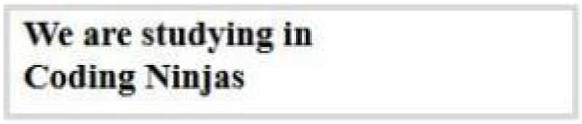
BR Tag

<br> 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


We are studying in
Coding Ninjas

LISTS

Lists are used to **group different pieces of information together** so that they are **easily linked** and **easy to read**.

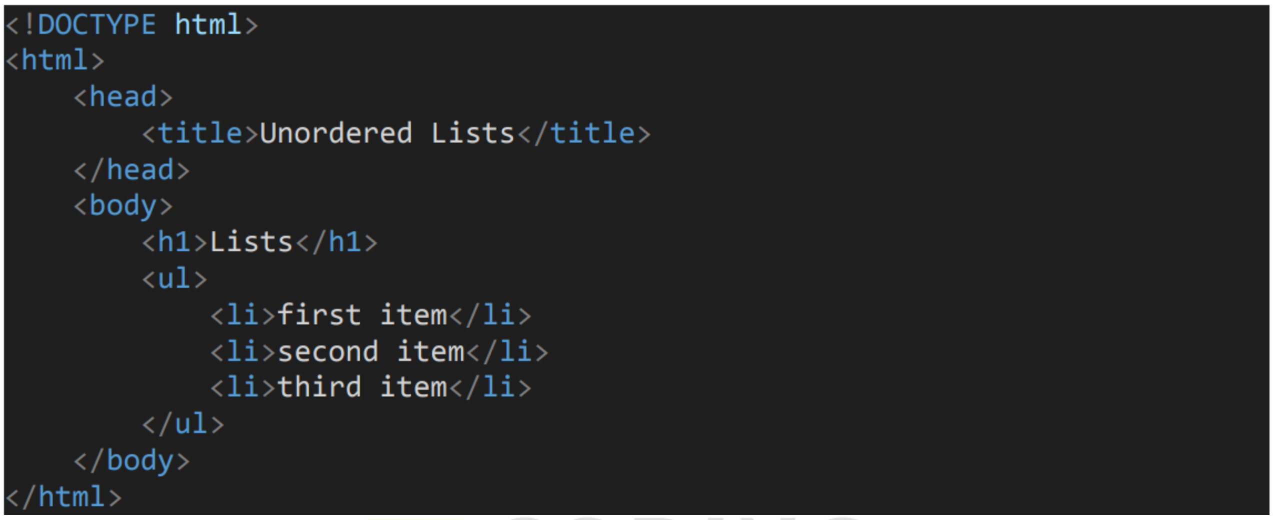
Lists help construct a **well-structured**, more open, and **easy-to-maintain** document from a structural standpoint.

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

Unordered Lists

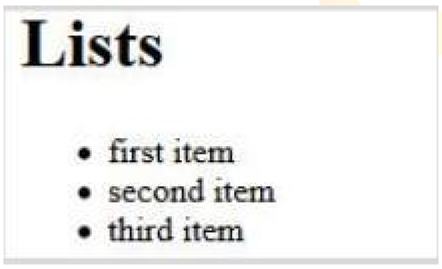
It's used to group a group of similar objects that aren't arranged in any specific order.
Where the counting of objects isn't necessary, unordered lists are used.
Bullets are used by default to separate the items.

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:



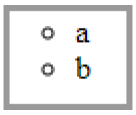
```
Lists

  • first item
  • second item
  • third item
```

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



```
∘ a
∘ b
```

***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

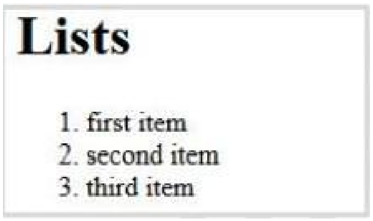
It is used in a certain order to group a number of related items.

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:



```
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 numbering will contain **numbers** (default)

A. ***type="A"*** - the numbering will contain **uppercase letters**

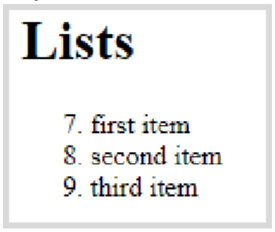
a. ***type="a"*** - the numbering will contain **lowercase letters**

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.**

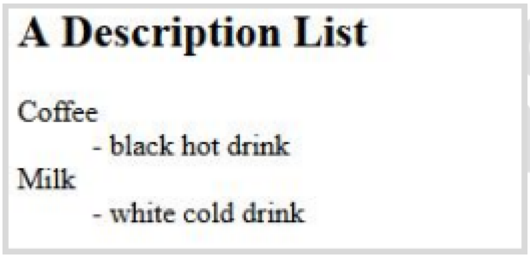
Tag	Description
<dl> tag	to **start** a definition list.
<dt> tag	to **begin** each definition - list term.
<dd> tag	to **begin each definition - list definition.**

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:

```
<!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:

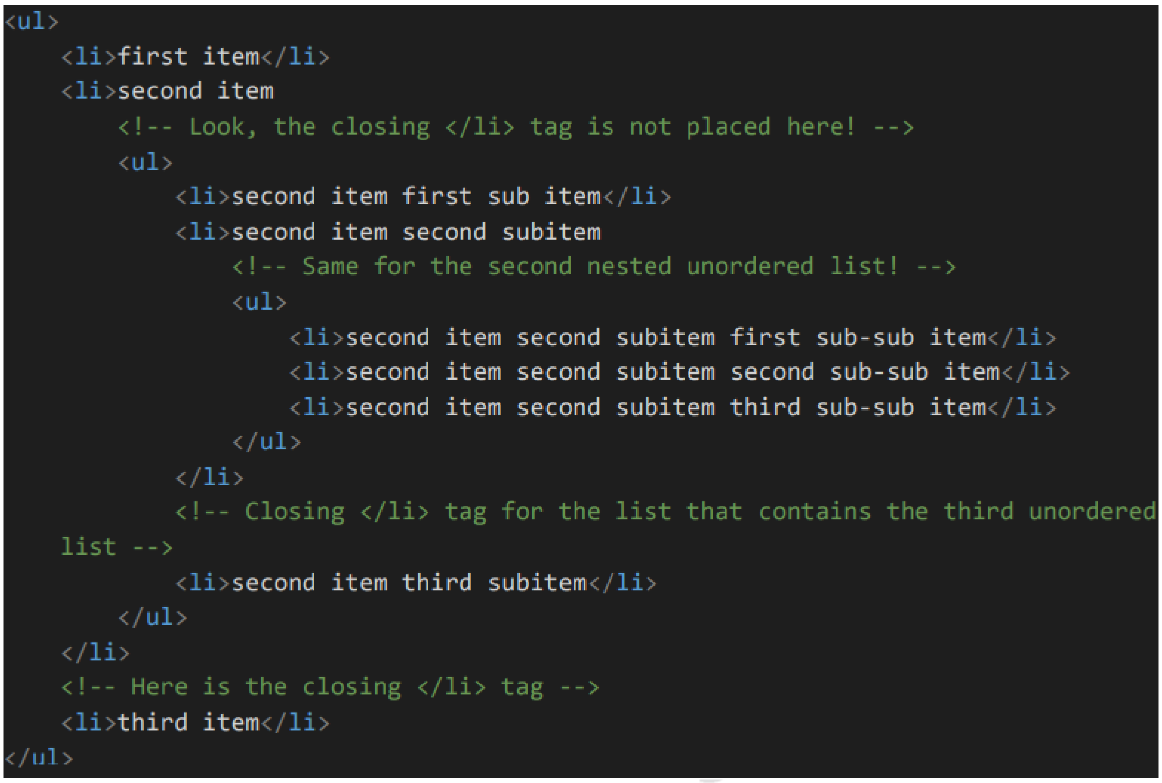


```
A Description List

Coffee
        - black hot drink
Milk
        - white cold drink
```

NESTING ELEMENTS

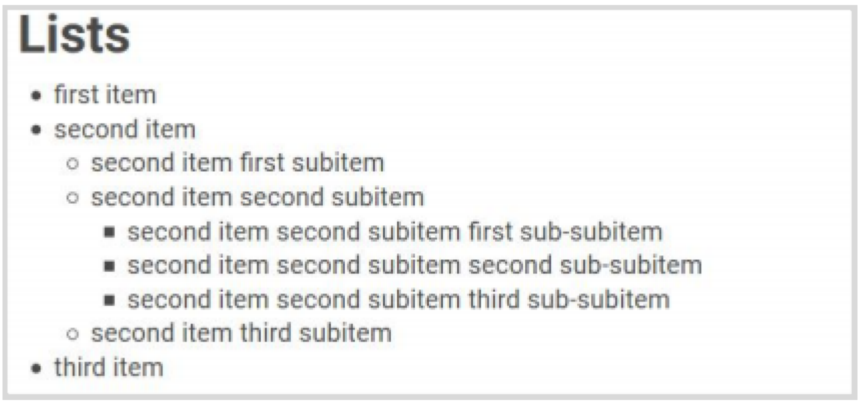
HTML elements can be nested i.e. **elements can contain elements.** Actually, all HTML documents consist of nested HTML elements

Eg:



```
<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

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.

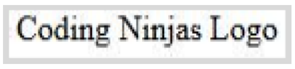
The ALT Attribute

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


Coding Ninjas Logo

***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.

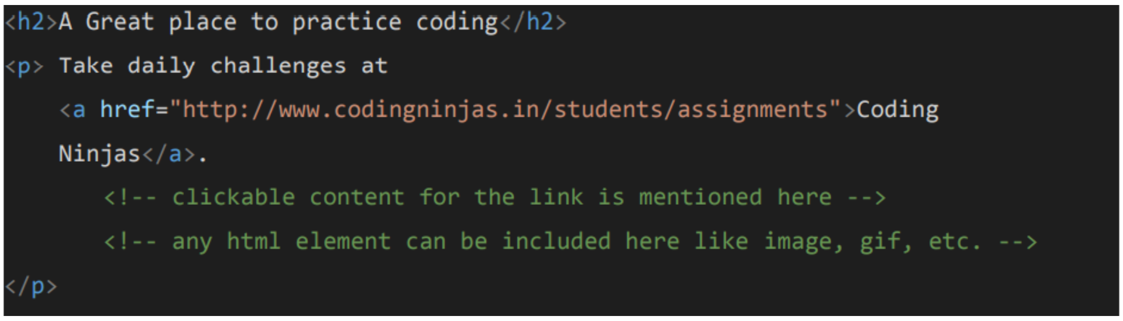
In all browsers, by default, links appear as follows:

- 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

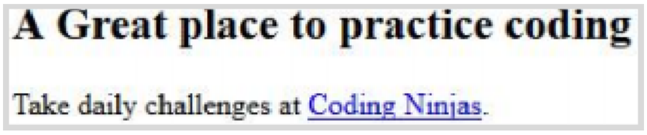
The **<a>** element's most significant attribute, **href**, indicates **the destination of the link**. T his means that the href attribute is used to **refer to the document** that is **linked to the relation.**

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:



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 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: <a href="../relativeFile.html"> Click Me</a>, 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: <a href="../subfolder/relativeDateien.html"> Click Me</a> 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:: <a href="https://1603.de/"> Click Me</a> 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

File Paths

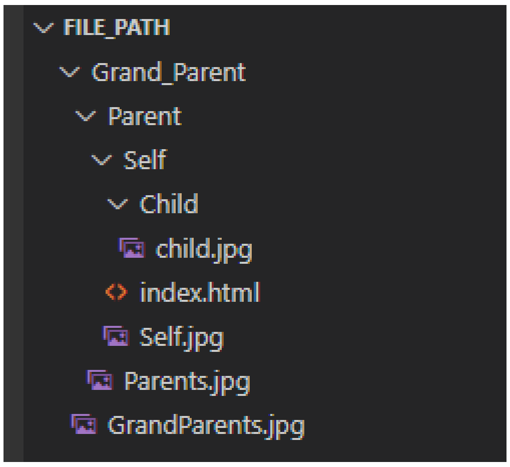
1. **What is File Path?**
   A File path specifies the location of a file on that web site's folder structure.

   File paths are used when we link to external files, like:
   - Images
   - Web pages
   - Style sheets
   - Java Scripts

   They are of two types:
   I. Relative file Path
   II. Absolute file Path

1. **What is Relative File Path?**
   A relative file path points to the location of files in the root folder of a particular web project with reference to the current working file.

   Assuming we have a project with the structure below:



```
FILE_PATH
  Grand_Parent
    Parent
      Self
        Child
          child.jpg
        index.html
        Self.jpg
      Parents.jpg
    GrandParents.jpg
```

Consider these scenarios;

To display self.jpg in a section of our index.html file, We have to assign the image's file path to the **src** attribute of an image tag.

Scenario 1:

- **If File present in the Same folder, use ./ delimiter:** this delimiter points to the current folder. You can use this to specify the relative file path when the file we want to link is in the same folder.

  E.g. In this case, the index.html file and self.jpg are both in the same folder, `Self`.So the relative path for self.jpg looks like this:

  ```
  <img src="./Self.jpg" alt="My pic" />
  ```

  or

  ```
  <img src="Self.jpg" alt="My pic" />
  ```

Scenario 2:

- **The file present in the subfolder:** In this case, the name of the file provided is preceded by the folder's name according to the hierarchy.

  E.g. The child.jpg file is in the child folder of the `Self` folder. So the relative file path to the child.jpg looks like this:

  ```
  <img src="./Child/child.jpg" alt="My Child's Image" />
  ```

Scenario 3:

- **If the file is present somewhere in the parent folder, use ../ delimiter:** This delimiter points one folder up the parent folder of the current working file.

  E.g.: The parents.jpg file is one level up parent folder of the `self` folder. So the relative file path to the parents.jpg looks like this:

  ```
  <img src="../Parents.jpg" alt="My parent's Image" />
  ```

Scenario 4:

- **If the file is present in the grandparent folder, use ../../ delimiter:** This delimiter points to two folders up of the current working file.

  E.g., The grandParents.jpg is two levels up of the `self` folder. So the relative file path to the grandParents.jpg looks like this:

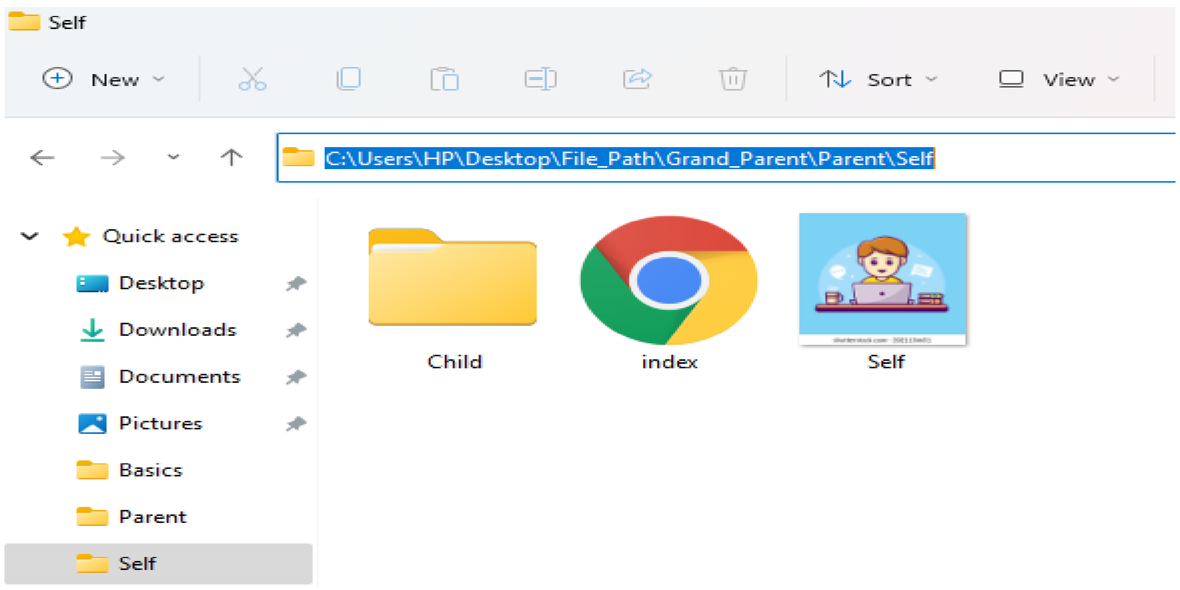
```
<img src="../../GrandParents.jpg" alt="My GrandParent's Image" />
```

Summary:

- **./** this means current folder. You can use this to specify the relative file path, and the file you want to link to is in the same folder.
- **./child/** this means go to the child folder from the current folder. You can use this to the relative file path, and the file you want to link to is in the child folder of the current folder.
- **../** this means to go one level up from the current folder. You can use this to specify a relative file path, and the file you want to link is one level up from the current folder.
- **../../** this means go two levels up from the current folder. You can use this to specify the relative file path when the file you want to link to two levels up of the parent folder of the current folder.

2. **What is an Absolute File Path?**
   An absolute path means the file path always starts from the root element or root folder and includes all the folder lists to locate the file.

   Example: **C:\Users\HP\Desktop\File_Path\Grand_Parent\Parent\Self**


C:\Users\HP\Desktop\File_Path\Grand_Parent\Parent\Self

Also, Absolute file paths are usually used to specify the location of files over the internet. These paths always include a domain name as part of the link.

Example:

```
<img
src="https://images.unsplash.com/photo-1453928582365-b6ad33cbcf64
?ixlib=rb-1.2.1&ixid=MnwxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8
&auto=format&fit=crop&w=873&q=80" alt="Laptop image">
```

Note: Don't use the Absolute File path that starts from your computer's root folder because it only works on your computer if you want to link a static file from your computer to your project.

The best practice would be to first move the static file into your project folder and then specify the relative path property.

HTML Boilerplate

What is a boilerplate?

Boilerplate code in HTML is required to set up the basic web page structure. This web page structure is the same for all web pages in general.

```
<html>
    <head>
      <title></title>
    </head>
    <body>
        Body Content
    </body>
</html>
```

Automatic boilerplate generation in different Text Editors

Text editors can produce HTML boilerplate for us. The basic boilerplate generated (as shown above) will always stay the same across all text editors.

Although different text editors may produce some extra code as well apart from the basic boilerplate as shown above. For example, boilerplate produced by Atom text editor is:

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
   <head>
     <meta charset="utf-8">
     <title></title>
   </head>
   <body>
```

```
    </body>
</html>
```

- **<!DOCTYPE html>** indicates the current version of HTML i.e., HTML5

- **<html lang="en" dir="ltr">**
  1. **<u>lang</u>** specifies the language of the content.
     lang="en" means language is English
     lang="es" means language is Spanish
     And so on.
     But we'll be using English.
     Usage of lang attribute helps the browser to display the text in the desired language.

  2. **<u>dir</u>** specifies the direction of text in which it'll be displayed
     dir="ltr" means left to right
     dir="rtl" means right to left

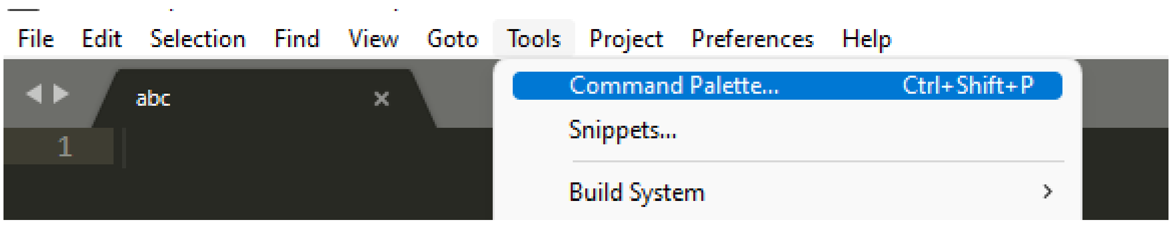
- **<meta charset="utf-8">**
  Declaring the "charset" as "UTF-8", tells the browser to use the UTF-8 character encoding, which is a method of converting the typed characters into machine-readable code.

Notes:

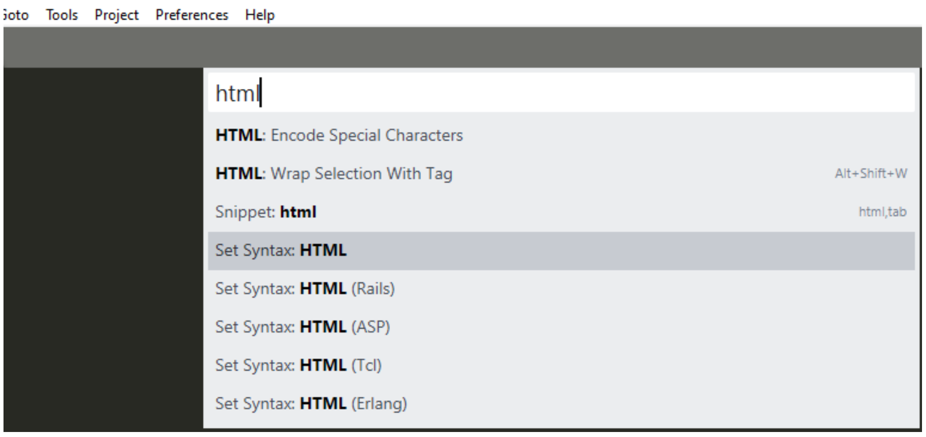
- *You need NOT cram and learn these. But as a software developer, you should be aware of this kind of HTML code and its meaning.*
- *There are some attributes that browsers consider implicitly by default if you don't mention them explicitly. For example, the value of **dir** attribute is **"ltr"** (left to right) by default.*

Generating HTML Boilerplate in Sublime Text

1. go to Tools → Command Palette

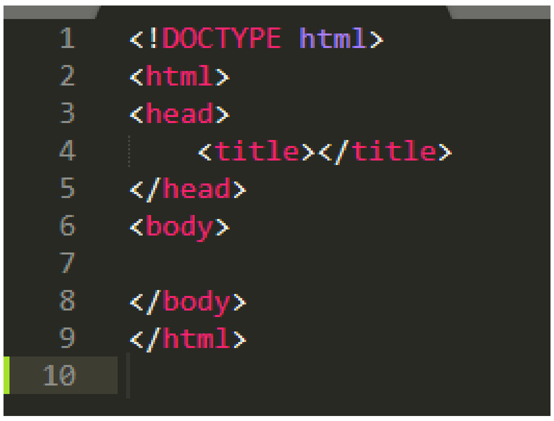


2. Type ‘HTML’ then select ‘Set Syntax: HTML’



3. Type in the text editor: <html

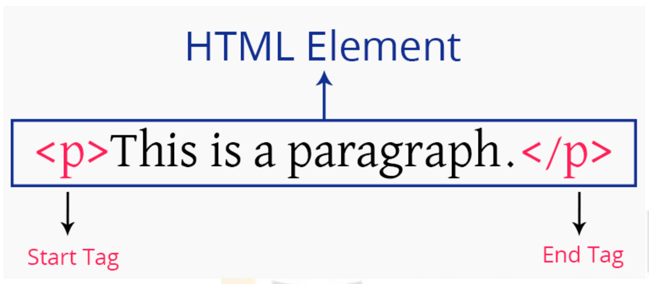
Then press enter and your boilerplate will be ready!



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

</body>
</html>

```

HTML Elements and Tags

- **A webpage can be seen as a collection of HTML elements.** For example, a web page is made up of various headings, paragraphs, images, links, etc. All of these can be seen as HTML elements.
- **HTML elements are represented using HTML tags.**
- **An HTML element is a** collection of an opening tag, a closing tag, the content within these two tags, and its attributes. (You'll learn about attributes in further lectures).



- Although some elements don't require a closing tag and hence no content in between. These are called **empty elements.**

  They are made up of one tag which is referred to as a **self-closing tag.** For example, <br> tag.

- **Generic HTML element structure**
  <tagName> CONTENT </tagName>

  For example,
  <h1> Hello World <h1>
  Here <h1> is the opening tag, </h1> is the closing tag, 'Hello World' is the content. All of these together represent an HTML element 'heading'.

Notes:

- ***Nested HTML elements:*** *The content of an HTML element can be plain text, image, or can even be other HTML elements. An HTML element can include several other HTML elements as its content.*

For example,

```
<div>
  <p>This is a paragraph element inside a div element</p>
</div>
```

- But some HTML elements are not directly seen on the webpage but they do exist as they hold some special meaning.
  For example, <link> tag is used to link an HTML file with other files such as CSS, js files, etc. You'll learn more examples in further lectures!

CONTENT SECTIONING

Content sectioning elements allow you to organize the document content into logical pieces. Use the sectioning elements to create a broad outline for your page content, including header and footer navigation, and heading elements to identify sections of

<base>	**Specifies the base URL to use for all relative URLs in a document. There can be only one such element in a document.**
<head>	Contains machine-readable information (metadata) about the document, like its title, scripts, and style sheets.
<link>	Specifies relationships between the current document and an external resource. This element is most commonly used to link to CSS but is also used to establish site icons (both "favicon" style icons and icons for the home screen and apps on mobile devices) among other things.
<meta>	Represents metadata that cannot be represented by other HTML meta-related elements, like <base>, <link>, <script>, <style> and <title>.
<style>	Contains style information for a document or part of a document. It contains CSS, which is applied to the contents of the document containing this element.
<title>	**Defines the document's title that is shown in a browser's title bar or a page's tab. It only contains text; tags within the element are ignored.**

TEXT CONTENT

Use HTML text content elements to organize blocks or sections of content placed between the opening <body> and closing </body> tags. Important for accessibility and SEO, these elements identify the purpose or structure of that content.

<blockquote>	Indicates that the enclosed text is an extended quotation. Usually, this is rendered visually by indentation. A URL for the source of the quotation may be given using the cite attribute, while a text representation of the source can be given using the <cite> element.
<dd>	Provides the description, definition, or value for the preceding term (<dt>) in a description list (<dl>).
<div>	The generic container for flow content. It has no effect on the content or layout until styled in some way using CSS (e.g., styling is directly applied to it, or some kind of layout model like flexbox is applied to its parent element).
<dl>	Represents a description list. The element encloses a list of groups of terms (specified using the <dt> element) and descriptions (provided by <dd> elements). Common uses for this element are to implement a glossary or to display metadata (a list of key-value pairs).
<dt>	Specifies a term in a description or definition list, and as such must be used inside a <dl> element. It is usually followed by a <dd> element; however, multiple <dt> elements in a row indicate several terms that are all defined by the immediate next <dd> element.
<figcaption>	Represents a caption or legend describing the rest of the contents of its parent <figure> element.
<figure>	Represents self-contained content, potentially with an optional caption, which is specified using the <figcaption> element. The figure, its caption, and its contents are referenced as a single unit.
<hr>	Represents a thematic break between paragraph-level elements: for example, a change of scene in a story, or a shift of topic within a section.
<li>	Represents an item in a list. It must be contained in a parent element: an ordered list (<ol>), an unordered list (<ul>), or a menu (<menu>). In menus and unordered lists, list items are usually displayed using bullet

	points. In ordered lists, they are usually displayed with an ascending counter on the left, such as a number or letter.
<menu>	A semantic alternative to <ul>, but treated by browsers (and exposed through the accessibility tree) as no different than <ul>. It represents an unordered list of items (which are represented by <li> elements).
<ol>	Represents an ordered list of items — typically rendered as a numbered list.
<p>	Represents a paragraph. Paragraphs are usually represented in visual media as blocks of text separated from adjacent blocks by blank lines and/or first-line indentation, but HTML paragraphs can be any structural grouping of related content, such as images or form fields.
<pre>	Represents preformatted text which is to be presented exactly as written in the HTML file. The text is typically rendered using a non-proportional, or monospaced, font. Whitespace inside this element is displayed as written.
<ul>	Represents an unordered list of items, typically rendered as a bulleted list.

INLINE TEXT SEMANTIC

Use the HTML inline text semantic to define the meaning, structure, or style of a word, line, or any arbitrary piece of text.

<a>	Together with its href attribute, creates a hyperlink to web pages, files, email addresses, locations within the current page, or anything else a URL can address.
<abbr>	Represents an abbreviation or acronym.
<b>	Used to draw the reader's attention to the element's contents, which are not otherwise granted special importance. This was formerly known as the Boldface element, and most browsers still draw the text in boldface. However, you should not use <b> for styling text or granting importance. If you wish to create boldface text, you should use the CSS font-weight property. If you wish to indicate an element is of special importance, you should use the strong element.
<bdi>	Tells the browser's bidirectional algorithm to treat the text it contains in isolation from its surrounding text. It's particularly useful when a website dynamically inserts some text and doesn't know the directionality of the text being inserted.
<bdo>	Overrides the current directionality of text, so that the text within is rendered in a different direction.
<br>	Produces a line break in text (carriage-return). It is useful for writing a poem or an address, where the division of lines is significant.
<cite>	Used to mark up the title of a cited creative work. The reference may be in an abbreviated form according to context-appropriate conventions related to citation metadata.
<code>	Displays its contents styled in a fashion intended to indicate that the text is a short fragment of computer code. By default, the content text is displayed using the user agent's default monospace font.
<data>	Links a given piece of content with a machine-readable translation. If the content is time- or date-related, the<time> element must be used.
<dfn>	Used to indicate the term being defined within the context of a definition phrase or sentence. The ancestor <p> element, the <dt>/<dd> pairing, or the nearest section ancestor of the <dfn> element, is considered to be the

	definition of the term.
<em>	Marks text that has stress emphasis. The <em> element can be nested, with each nesting level indicating a greater degree of emphasis.
<i>	Represents a range of text that is set off from the normal text for some reason, such as idiomatic text, technical terms, and taxonomical designations, among others. Historically, these have been presented using italicized type, which is the original source of the <i> naming of this element.
<kbd>	Represents a span of inline text denoting textual user input from a keyboard, voice input, or any other text entry device. By convention, the user agent defaults to rendering the contents of a <kbd> element using its default monospace font, although this is not mandated by the HTML standard.
<mark>	Represents text which is marked or highlighted for reference or notation purposes due to the marked passage's relevance in the enclosing context.
<q>	Indicates that the enclosed text is a short inline quotation. Most modern browsers implement this by surrounding the text in quotation marks. This element is intended for short quotations that don't require paragraph breaks; for long quotations use the <blockquote> element.
<rp>	Used to provide fall-back parentheses for browsers that do not support the display of ruby annotations using the <ruby> element. One <rp> element should enclose each of the opening and closing parentheses that wrap the <rt> element that contains the annotation's text.
<rt>	Specifies the ruby text component of a ruby annotation, which is used to provide pronunciation, translation, or transliteration information for East Asian typography. The <rt> element must always be contained within a <ruby> element.
<ruby>	Represents small annotations that are rendered above, below, or next to base text, usually used for showing the pronunciation of East Asian characters. It can also be used for annotating other kinds of text, but this usage is less common.
<s>	Renders text with a strikethrough, or a line through it. Use the <s> element to represent things that are no longer relevant or no longer accurate. However, <s> is not appropriate when indicating document edits; for that, use the del and ins elements, as appropriate.
<samp>	Used to enclose inline text which represents sample (or quoted) output from a computer program. Its contents are typically rendered using the browser's

	default monospaced font (such as Courier or Lucida Console).
<small>	Represents side-comments and small print, like copyright and legal text, independent of its styled presentation. By default, it renders text within it one font size smaller, such as from small to x-small.
<span>	A generic inline container for phrasing content, which does not inherently represent anything. It can be used to group elements for styling purposes (using the class or id attributes), or because they share attribute values, such as lang. It should be used only when no other semantic element is appropriate. <span> is very much like a div element, but div is a block-level element whereas a <span> is an inline-level element.
<strong>	Indicates that its contents have strong importance, seriousness, or urgency. Browsers typically render the contents in bold type.
<sub>	Specifies inline text which should be displayed as subscript for solely typographical reasons. Subscripts are typically rendered with a lowered baseline using smaller text.
<sup>	Specifies inline text which is to be displayed as superscript for solely typographical reasons. Superscripts are usually rendered with a raised baseline using smaller text.
<time>	Represents a specific period in time. It may include the datetime attribute to translate dates into machine-readable format, allowing for better search engine results or custom features such as reminders.
<u>	Represents a span of inline text which should be rendered in a way that indicates that it has a non-textual annotation. This is rendered by default as a simple solid underline but may be altered using CSS.
<var>	Represents the name of a variable in a mathematical expression or a programming context. It's typically presented using an italicized version of the current typeface, although that behavior is browser-dependent.
<wbr>	Represents a word break opportunity—a position within text where the browser may optionally break a line, though its line-breaking rules would not otherwise create a break at that location.

Extra Resources

All HTML elements have certain attributes associated with them. Some attributes are specific to a certain HTML element whereas others are not. Here you can find attributes of HTML elements you have learned so far.

- **Global Attributes:** They are attributes that can be applied to all HTML elements. Although they may or may not show any effect on some elements.

  <h1> to <h6> tags, and <p> tag only include the Global Attributes.

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

- **All about <img> tag:** including all the image formats that HTML supports, the reasons for an image not to load in the browser, and all the attributes of the image element.

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

- **All about <a> tag attributes:**

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

Important:

- You can always look into the HTML documentation for more information whenever you need it.

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

- As a web developer, while building web applications you'll always come across scenarios that you had not encountered before. So if you get stuck at a point, it's very important to know, where to get help from?
  Most probably you'll be able to find solutions to your queries, but if not, you can post your questions here and connect with your fellow developers on these platforms to seek help.
  https://stackoverflow.com/
  https://www.quora.com/