

HTML5

Contents

HTML5.....	1
New Features.....	5
The DOCTYPE	6
Character Encoding.....	6
The <script> tag.....	6
The <link> tag.....	6
HTML5 Elements	7
HTML5 Attributes.....	7
Building blocks of HTML.....	7
Syntax	8
Example:	8
The building blocks	9
HTML5 Document.....	9
Standard Attributes	12
Custom Attributes	13
HTML Formatting.....	13
1) Bold Text.....	15
HTML Heading	15
Heading no. 1.....	16
Heading no. 2	16
Heading no. 3	16
HTML Paragraph.....	17
Space inside HTML Paragraph	18
How to Use and <hr> tag with paragraph?	19
HTML Phrase tag.....	20
1. Text Abbreviation tag	21
Example	21
2. Marked tag:	21
Example	21
3. Strong text:	22
Example	22
4. Emphasized text	23
Example	23

5. Definition tag:	23
Example	23
6. Quoting text:	24
Example	24
7. Short Quotations:	25
Example:	25
8. Code tags.....	25
9. Keyboard Tag	26
10. Address tag.....	26
HTML Anchor	27
href attribute of HTML anchor tag	27
Specify a location for Link using target attribute	27
Example:	27
Appearance of HTML anchor tag	28
HTML Image	28
Attributes of HTML img tag.....	29
Use of height and width attribute with img tag	29
Example:	30
Use of alt attribute	30
How to get image from another directory/folder?	31
Use tag as a link	31
Example:	31
HTML Table	32
HTML Table Tags.....	33
HTML Table Example	33
HTML Table with Border	34
1) HTML Border attribute	34
2) CSS Border property	35
HTML Table with cell padding.....	35
HTML Table width:.....	36
Example:	36
HTML Table with colspan	38
HTML Table with rowspan	39
HTML table with caption	40
Styling HTML table even and odd cells	40
HTML Lists	41
HTML Ordered List or Numbered List.....	41
HTML Unordered List or Bulleted List.....	42

HTML Description List or Definition List	42
HTML Nested List	43
HTML Form	45
Why use HTML Form	45
HTML Form Syntax	45
HTML Form Tags	46
HTML 5 Form Tags	46
HTML <form> element.....	46
HTML <input> element	47
Example:	47
HTML TextField Control	47
HTML <textarea> tag in form	48
Label Tag in Form.....	49
HTML Password Field Control.....	50
HTML 5 Email Field Control	50
Radio Button Control.....	51
Checkbox Control	52
Submit button control	53
HTML <fieldset> element:	54
HTML Form Example	54
HTML Form Example	55
HTML Classes.....	56
Class Attribute in HTML.....	56
Defining an HTML class	57
Example:	57
Example 1:.....	57
Another Example with different class name.....	58
Example:	58
Class Attribute in JavaScript.....	58
Example:	58
Multiple Classes	59
Example:	59
Same class with Different Tag.....	60
Example:	60
HTML Id Attribute	61
HTML id with CSS	61
Example:	61
Difference between HTML Class and ID.....	61

Example:	61
HTML id with JavaScript.....	62
Example:	62
HTML Id Attribute	63
HTML id with CSS	63
Example:	63
Difference between HTML Class and ID.....	64
Example:	64
HTML id with JavaScript.....	65
Example:	65
HTML tag	65
Syntax.....	65
Example	66
Attribute:	67
Tag-specific attributes:.....	67
Global attribute:.....	67
Event attribute:	67
HTML Div Tag	67
Difference between HTML div tag and span tag	68
HTML div example: Login Form.....	68
HTML iframes	69
Iframe Syntax.....	69
Set Width and Height of iframe	69
Example: (Pixels)	69
Example: (Percentage)	69
Example:	70
Remove the border of iframe	70
Example:	70
Example:	70
Iframe Target for a link	71
Example:	71
Embed YouTube video using iframe	72
Example:	72
Attributes of <iframe>.....	73

HTML5 is the next major revision of the HTML standard superseding HTML 4.01, XHTML 1.0, and XHTML 1.1. HTML5 is a standard for structuring and presenting content on the World Wide Web.

HTML5 is a cooperation between the World Wide Web Consortium (W3C) and the Web Hypertext Application Technology Working Group (WHATWG).

The new standard incorporates features like video playback and drag-and-drop that have been previously dependent on third-party browser plug-ins such as Adobe Flash, Microsoft Silverlight, and Google Gears.

New Features

HTML5 introduces a number of new elements and attributes that can help you in building modern websites. Here is a set of some of the most prominent features introduced in HTML5.

- **New Semantic Elements** – These are like <header>, <footer>, and <section>.
- **Forms 2.0** – Improvements to HTML web forms where new attributes have been introduced for <input> tag.
- **Persistent Local Storage** – To achieve without resorting to third-party plugins.
- **WebSocket** – A next-generation bidirectional communication technology for web applications.
- **Server-Sent Events** – HTML5 introduces events which flow from web server to the web browsers and they are called Server-Sent Events (SSE).
- **Canvas** – This supports a two-dimensional drawing surface that you can program with JavaScript.
- **Audio & Video** – You can embed audio or video on your webpages without resorting to third-party plugins.
- **Geolocation** – Now visitors can choose to share their physical location with your web application.
- **Microdata** – This lets you create your own vocabularies beyond HTML5 and extend your web pages with custom semantics.
- **Drag and drop** – Drag and drop the items from one location to another location on the same webpage.

The HTML 5 language has a "custom" HTML syntax that is compatible with HTML 4 and XHTML1 documents published on the Web, but is not compatible with the more esoteric SGML features of HTML 4.

HTML 5 does not have the same syntax rules as XHTML where we needed lower case tag names, quoting our attributes, an attribute had to have a value and to close all empty elements.

HTML5 comes with a lot of flexibility and it supports the following features –

- Uppercase tag names.
- Quotes are optional for attributes.
- Attribute values are optional.
- Closing empty elements are optional.

The DOCTYPE

DOCTYPEs in older versions of HTML were longer because the HTML language was SGML based and therefore required a reference to a DTD.

HTML 5 authors would use simple syntax to specify DOCTYPE as follows –

```
<!DOCTYPE html>
```

The above syntax is case-insensitive.

Character Encoding

HTML 5 authors can use simple syntax to specify Character Encoding as follows –

```
<meta charset = "UTF-8">
```

The above syntax is case-insensitive.

The <script> tag

It's common practice to add a type attribute with a value of "text/javascript" to script elements as follows –

```
<script type = "text/javascript" src = "scriptfile.js"></script>
```

HTML 5 removes extra information required and you can use simply following syntax –

```
<script src = "scriptfile.js"></script>
```

The <link> tag

So far you were writing <link> as follows –

```
<link rel = "stylesheet" type = "text/css" href = "stylefile.css">
```

HTML 5 removes extra information required and you can simply use the following syntax –

```
<link rel = "stylesheet" href = "stylefile.css">
```

HTML5 Elements

HTML5 elements are marked up using start tags and end tags. Tags are delimited using angle brackets with the tag name in between.

The difference between start tags and end tags is that the latter includes a slash before the tag name.

Following is the example of an HTML5 element –

```
<p>...</p>
```

HTML5 tag names are case insensitive and may be written in all uppercase or mixed case, although the most common convention is to stick with lowercase.

Most of the elements contain some content like <p>...</p> contains a paragraph. Some elements, however, are forbidden from containing any content at all and these are known as void elements. For example, **br**, **hr**, **link**, **meta**, etc.

Here is a complete list of [HTML5 Elements](#).

HTML5 Attributes

Elements may contain attributes that are used to set various properties of an element.

Some attributes are defined globally and can be used on any element, while others are defined for specific elements only. All attributes have a name and a value and look like as shown below in the example.

Following is the example of an HTML5 attribute which illustrates how to mark up a div element with an attribute named class using a value of "example" –

```
<div class = "example">...</div>
```

Attributes may only be specified within start tags and must never be used in end tags.

HTML5 attributes are case insensitive and may be written in all uppercase or mixed case, although the most common convention is to stick with lowercase.

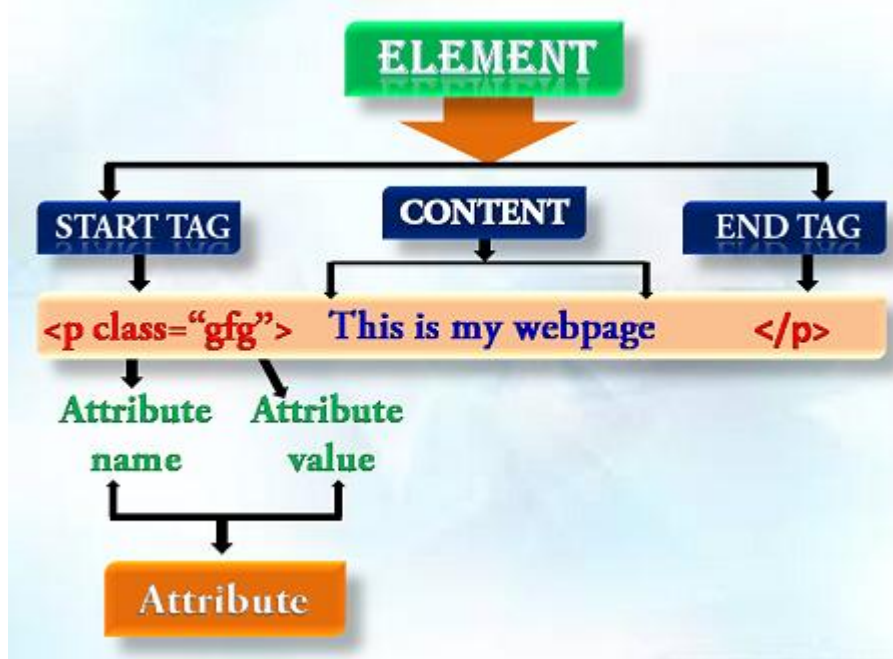
Building blocks of HTML

An HTML document consist of its basic building blocks which are:

- **Tags:** An HTML tag surrounds the content and apply meaning to it. It is written between < and > brackets.
- **Attribute:** An attribute in HTML provides extra information about the element, and it is applied within the start tag. An HTML attribute contains two fields: name & value.

Syntax

1. `<tag name attribute_name= " attr_value"> content </ tag name>`
 - **Elements:** An HTML element is an individual component of an HTML file. In an HTML file, everything written within tags are termed as HTML elements.



Example:

1. `<!DOCTYPE html>`
2. `<html>`
3. `<head>`
4. `<title>The basic building blocks of HTML</title>`
5. `</head>`
6. `<body>`
7. `<h2>The building blocks</h2>`
8. `<p>This is a paragraph tag</p>`
9. `<p style="color: red">The style is attribute of paragraph tag</p>`
10. `The element contains tag, attribute and content`
11. `</body>`
12. `</html>`

Output:

The building blocks

This is a paragraph tag

The style is attribute of paragraph tag

The element contains tag, attribute and content

HTML5 Document

The following tags have been introduced for better structure –

- **section** – This tag represents a generic document or application section. It can be used together with h1-h6 to indicate the document structure.
- **article** – This tag represents an independent piece of content of a document, such as a blog entry or newspaper article.
- **aside** – This tag represents a piece of content that is only slightly related to the rest of the page.
- **header** – This tag represents the header of a section.
- **footer** – This tag represents a footer for a section and can contain information about the author, copyright information, et cetera.
- **nav** – This tag represents a section of the document intended for navigation.
- **dialog** – This tag can be used to mark up a conversation.
- **figure** – This tag can be used to associate a caption together with some embedded content, such as a graphic or video.

The markup for an HTML 5 document would look like the following

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<meta charset = "utf-8">
```

```
<title>...</title>
```

```
</head>
```

```
<body>
```

```
<header>...</header>
```

```
<nav>...</nav>

<article>

  <section>

    ...

  </section>

</article>

<aside>...</aside>

<footer>...</footer>

</body>
</html>
```

```
<!DOCTYPE html>

<html>

  <head>

    <meta charset = "utf-8">

    <title>...</title>

  </head>

  <body>

    <header role = "banner">

      <h1>HTML5 Document Structure Example</h1>

      <p>This page should be tried in safari, chrome or Mozilla.</p>

    </header>

    <nav>
```

```
<ul>

  <li><a href = "https://www.tutorialspoint.com/html">HTML Tutorial</a></li>

  <li><a href = "https://www.tutorialspoint.com/css">CSS Tutorial</a></li>

  <li><a href = "https://www.tutorialspoint.com/javascript">
    JavaScript Tutorial</a></li>

</ul>

</nav>

<article>

  <section>

    <p>Once article can have multiple sections</p>

  </section>

</article>

<aside>

  <p>This is aside part of the web page</p>

</aside>

<footer>

  <p>Created by <a href = "https://tutorialspoint.com/">Tutorials Point</a></p>

</footer>

</body>

</html>
```

As explained in the previous chapter, elements may contain attributes that are used to set various properties of an element.

Some attributes are defined globally and can be used on any element, while others are defined for specific elements only. All attributes have a name and a value and look like as shown below in the example.

Following is the example of an HTML5 attributes which illustrates how to mark up a div element with an attribute named class using a value of "example" –

```
<div class = "example">...</div>
```

Attributes may only be specified within **start tags** and must never be used in **end tags**.

HTML5 attributes are case insensitive and may be written in all uppercase or mixed case, although the most common convention is to stick with lowercase.

Standard Attributes

The attributes listed below are supported by almost all the HTML 5 tags.

Attribute	Options	Function
accesskey	User Defined	Specifies a keyboard shortcut to access an element.
align	right, left, center	Horizontally aligns tags
background	URL	Places an background image behind an element
bgcolor	numeric, hexidecimal, RGB values	Places a background color behind an element
class	User Defined	Classifies an element for use with Cascading Style Sheets.
contenteditable	true, false	Specifies if the user can edit the element's content or not.
contextmenu	Menu id	Specifies the context menu for an element.
data-XXXX	User Defined	Custom attributes. Authors of a HTML document can define their own attributes. Must start with "data-".
draggable	true,false, auto	Specifies whether or not a user is allowed to drag an element.
height	Numeric Value	Specifies the height of tables, images, or table cells.
hidden	hidden	Specifies whether element should be visible or not.

id	User Defined	Names an element for use with Cascading Style Sheets.
item	List of elements	Used to group elements.
itemprop	List of items	Used to group items.
spellcheck	true, false	Specifies if the element must have it's spelling or grammar checked.
style	CSS Style sheet	Specifies an inline style for an element.
subject	User define id	Specifies the element's corresponding item.
tabindex	Tab number	Specifies the tab order of an element.
title	User Defined	"Pop-up" title for your elements.
valign	top, middle, bottom	Vertically aligns tags within an HTML element.
width	Numeric Value	Specifies the width of tables, images, or table cells.

Custom Attributes

A new feature being introduced in HTML 5 is the addition of custom data attributes.

A custom data attribute starts with **data-** and would be named based on your requirement. Here is a simple example –

```
<div class = "example" data-subject = "physics" data-level = "complex">
...
</div>
```

The above code will be perfectly valid HTML5 with two custom attributes called *datasubject* and *data-level*. You would be able to get the values of these attributes using JavaScript APIs or CSS in similar way as you get for standard attributes.

HTML Formatting

HTML Formatting is a process of formatting text for better look and feel. HTML provides us ability to format text without using CSS. There are many formatting tags in HTML. These tags are used to make text bold, italicized, or underlined. There are almost 14 options available that how text appears in HTML and XHTML.

In HTML the formatting tags are divided into two categories:

- Physical tag: These tags are used to provide the visual appearance to the text.
- Logical tag: These tags are used to add some logical or semantic value to the text.

NOTE: There are some physical and logical tags which may give same visual appearance, but they will be different in semantics.

Here, we are going to learn 14 HTML formatting tags. Following is the list of HTML formatting text.

Element name	Description
	This is a physical tag, which is used to bold the text written between it.
	This is a logical tag, which tells the browser that the text is important.
<i>	This is a physical tag which is used to make text italic.
	This is a logical tag which is used to display content in italic.
<mark>	This tag is used to highlight text.
<u>	This tag is used to underline text written between it.
<tt>	This tag is used to appear a text in teletype. (not supported in HTML5)
<strike>	This tag is used to draw a strikethrough on a section of text. (Not supported in HTML5)
<sup>	It displays the content slightly above the normal line.
<sub>	It displays the content slightly below the normal line.

<code></code>	This tag is used to display the deleted content.
<code><ins></code>	This tag displays the content which is added
<code><big></code>	This tag is used to increase the font size by one conventional unit.
<code><small></code>	This tag is used to decrease the font size by one unit from base font size.

1) Bold Text

HTML `` and `` formatting elements

The HTML `` element is a physical tag which display text in bold font, without any logical importance. If you write anything within `.....` element, is shown in bold letters.

See this example:

1. `<p> `Write Your First Paragraph in bold text.`</p>`

Output:

Write Your First Paragraph in bold text.

The HTML `` tag is a logical tag, which displays the content in bold font and informs the browser about its logical importance. If you write anything between `???????.`, is shown important text.

See this example:

1. `<p>`This is an important content``, and this is normal content`</p>`

HTML Heading

A HTML heading or HTML h tag can be defined as a title or a subtitle which you want to display on the webpage. When you place the text within the heading tags `<h1>.....</h1>`, it is displayed on the browser in the bold format and size of the text depends on the number of heading.

There are six different HTML headings which are defined with the `<h1>` to `<h6>` tags, from highest level h1 (main heading) to the least level h6 (least important heading).

h1 is the largest heading tag and h6 is the smallest one. So h1 is used for most important heading and h6 is used for least important.

Headings in HTML helps the search engine to understand and index the structure of web page.

Note: The main keyword of the whole content of a webpage should be display by h1 heading tag.

See this example:

1. `<h1>Heading no. 1</h1>`
2. `<h2>Heading no. 2</h2>`
3. `<h3>Heading no. 3</h3>`
4. `<h4>Heading no. 4</h4>`
5. `<h5>Heading no. 5</h5>`
6. `<h6>Heading no. 6</h6>`

Output:

Heading no. 1

Heading no. 2

Heading no. 3

Heading no. 4

Heading no. 5

Heading no. 6

Heading elements (h1....h6) should be used for headings only. They should not be used just to make text bold or big.

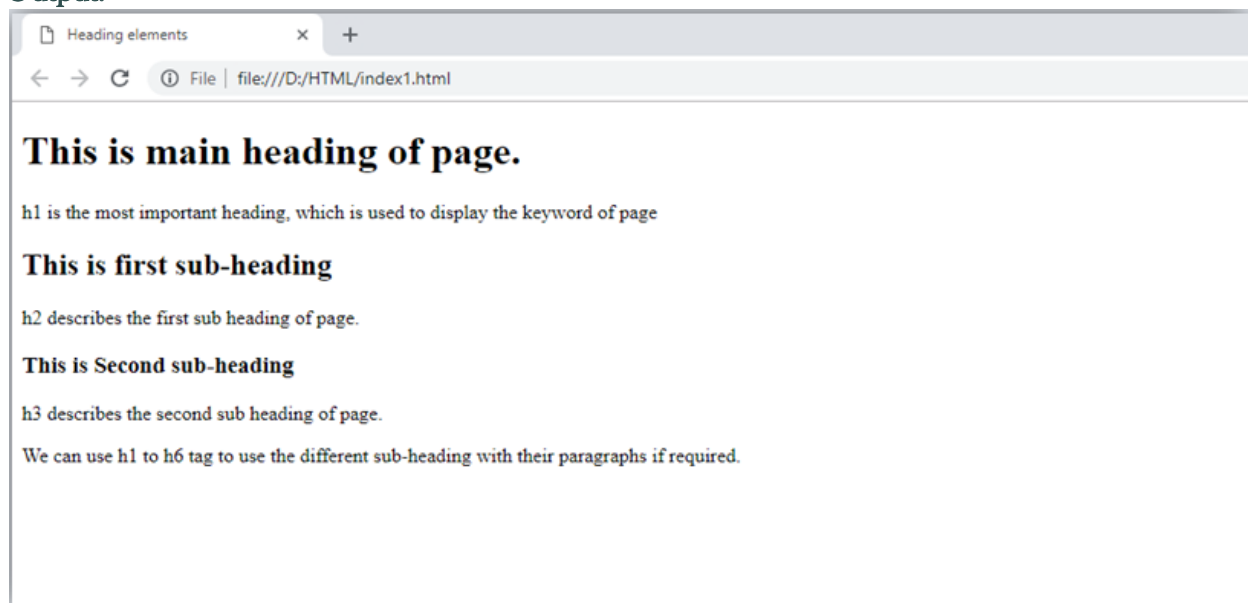
- HTML headings can also be used with nested elements. Following are different codes to display the way to use heading elements.

Example:

1. `<!DOCTYPE html>`
2. `<html>`
3. `<head>`
4. `<title>Heading elements</title>`
5. `</head>`
6. `<body>`
7. `<h1>This is main heading of page. </h1>`

8. `<p>`h1 is the most important heading, which is used to display the keyword of page `</p>`
9. `<h2>`This is first sub-heading`</h2>`
10. `<p>`h2 describes the first sub heading of page. `</p>`
11. `<h3>`This is Second sub-heading`</h3>`
12. `<p>`h3 describes the second sub heading of page.`</p>`
13. `<p>`We can use h1 to h6 tag to use the different sub-heading with their paragraphs if
14. required.
15. `</p>`
16. `</body>`
17. `</html>`

Output:



HTML Paragraph

HTML paragraph or HTML p tag is used to define a paragraph in a webpage. Let's take a simple example to see how it work. It is a notable point that a browser itself add an empty line before and after a paragraph. An HTML `<p>` tag indicates starting of new paragraph.

Note: If we are using various `<p>` tags in one HTML file then browser automatically adds a single blank line between the two paragraphs.

See this example:

1. `<p>`This is first paragraph.`</p>`
2. `<p>`This is second paragraph.`</p>`
3. `<p>`This is third paragraph.`</p>`

Output:

This is first paragraph.

This is second paragraph.

This is third paragraph.

Space inside HTML Paragraph

If you put a lot of spaces inside the HTML p tag, browser removes extra spaces and extra line while displaying the page. The browser counts number of spaces and lines as a single one.

1. `<p>`
2. I am
3. going to provide
4. you a tutorial on HTML
5. and hope that it will
6. be very beneficial for you.
7. `</p>`
8. `<p>`
9. Look, I put here a lot
10. of spaces but I know, Browser will ignore it.
11. `</p>`
12. `<p>`
13. You cannot determine the display of HTML`</p>`
14. `<p>`because resized windows may create different result.
15. `</p>`

Output:

I am going to provide you a tutorial on HTML and hope that it will be very beneficial for you.

Look, I put here a lot of spaces but I know, Browser will ignore it.

You cannot determine the display of HTML

because resized windows may create different result.

As you can see, all the extra lines and unnecessary spaces are removed by the browser.

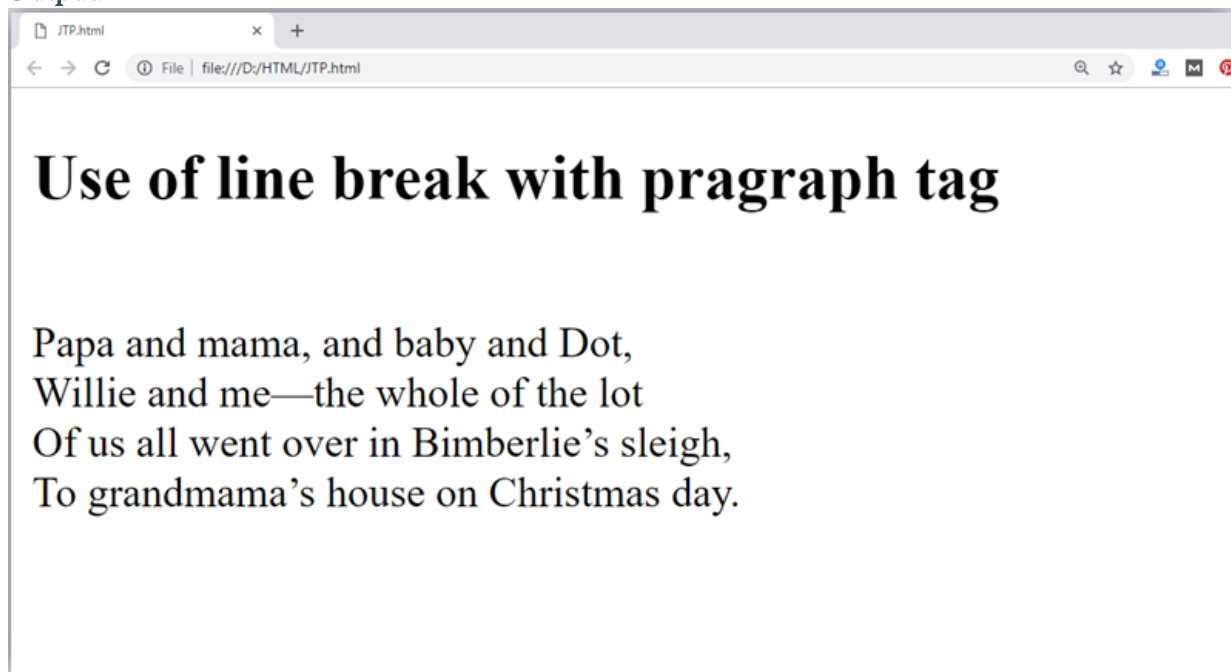
How to Use
 and <hr> tag with paragraph?

An HTML
 tag is used for line break and it can be used with paragraph elements. Following is the example to show how to use
 with <p> element.

Example:

1. <!DOCTYPE html>
2. <html>
3. <head>
4. </head>
5. <body>
6. <h2> Use of line break with paragraph tag</h2>
7. <p>
Papa and mama, and baby and Dot,
8.
Willie and me?the whole of the lot
9.
Of us all went over in Bimberlie's sleigh,
10.
To grandmama's house on Christmas day.
11. </p>
12. </body>
13. </html>

Output:



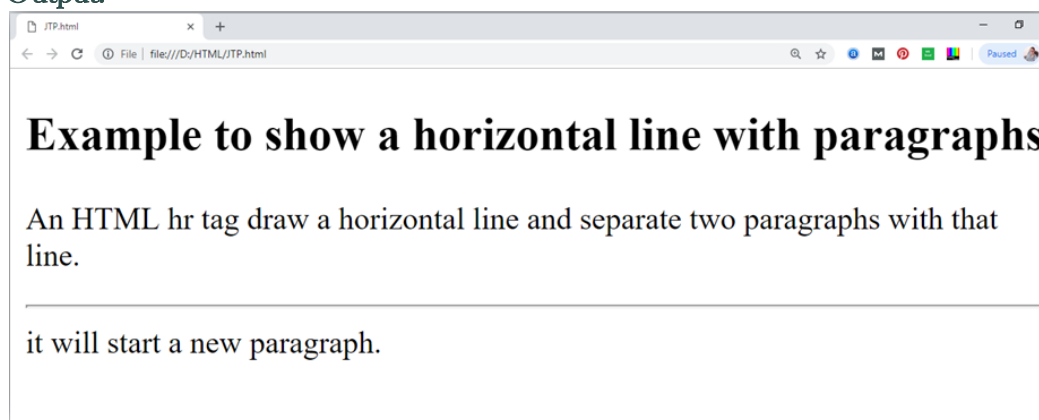
An HTML <hr> tag is used to apply a horizontal line between two statements or two paragraphs. Following is the example which is showing use of <hr> tag with paragraph.

Example:

1. <!DOCTYPE html>

2. `<html>`
3. `<head>`
4. `</head>`
5. `<body>`
6. `<h2>` Example to show a horizontal line with paragraphs`</h2>`
7. `<p>` An HTML hr tag draw a horizontal line and separate two paragraphs with that line.`<hr>`
it will start a new paragraph.
8. `</p>`
9. `</body>`
10. `</html>`

Output:



HTML Phrase tag

The HTML phrase tags are special purpose tags, which defines the structural meaning of a block of text or semantics of text. Following is the list of phrase tags, some of which we have already discussed in HTML formatting.

- Abbreviation tag : `<abbr>`
- Acronym tag: `<acronym>` (not supported in HTML5)
- Marked tag: `<mark>`
- Strong tag: ``
- Emphasized tag : ``
- Definition tag: `<dfn>`
- Quoting tag: `<blockquote>`
- Short quote tag : `<q>`
- Code tag: `<code>`
- Keyboard tag: `<kbd>`
- Address tag: `<address>`

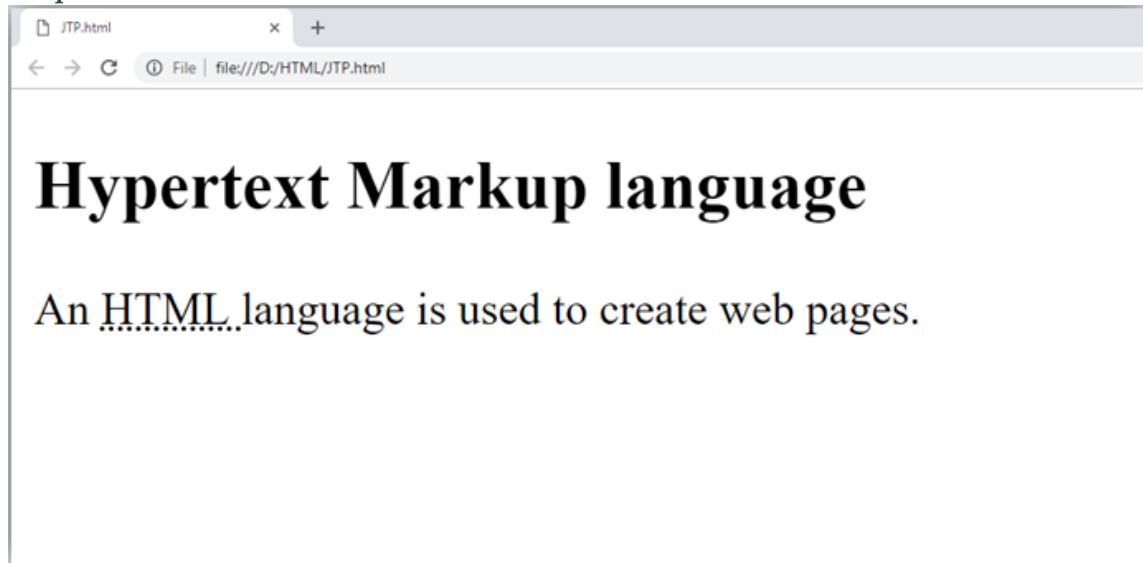
1. Text Abbreviation tag

This tag is used to abbreviate a text. To abbreviate a text, write text between <abbr> and </abbr> tag.

Example

1. `<p>An <abbr title = "Hypertext Markup language">HTML </abbr>language is used to create web pages. </p>`

Output:



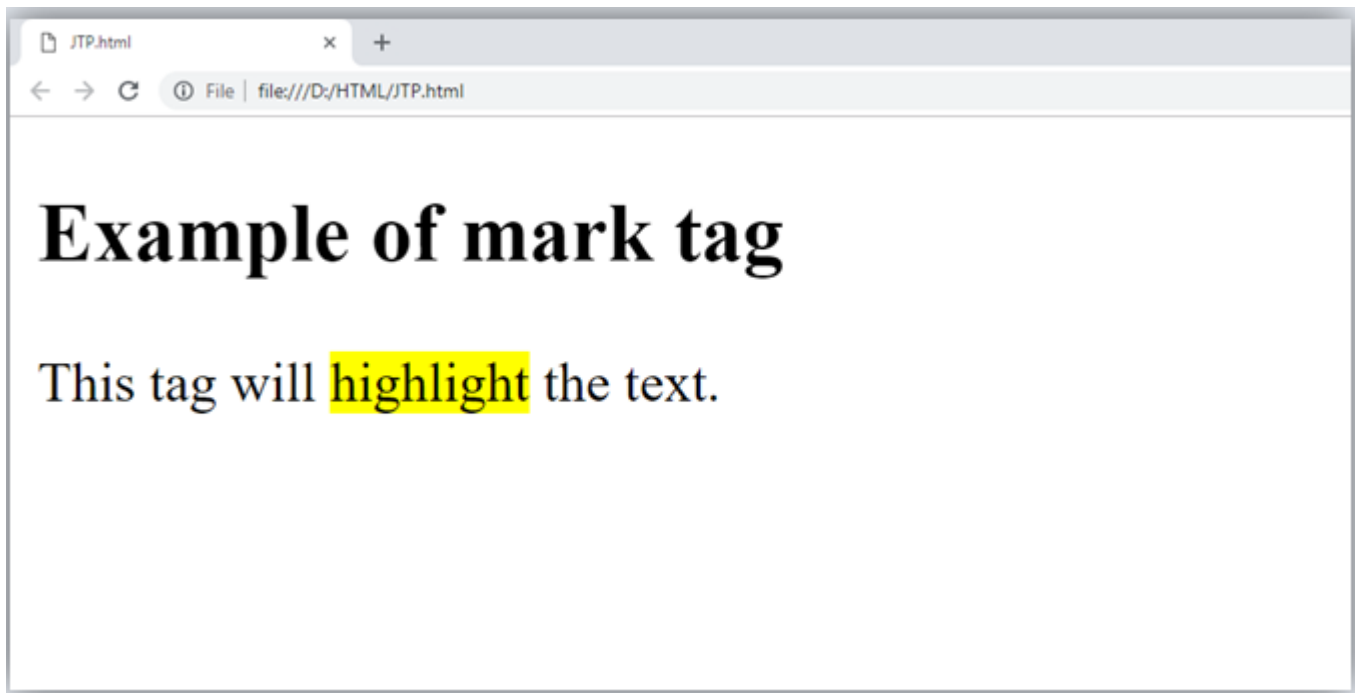
2. Marked tag:

The content written between <mark> and </mark> tag will show as yellow mark on browser. This tag is used to highlight a particular text.

Example

1. `<p>This tag will <mark>highlight</mark> the text.</p>`

Output:



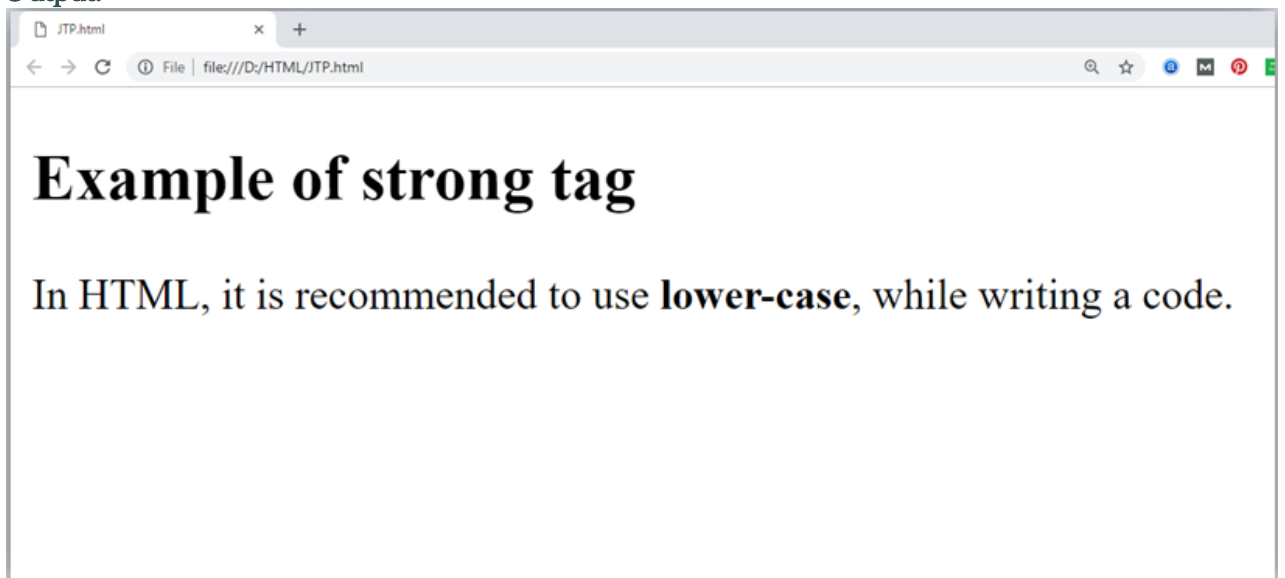
3. Strong text:

This tag is used to display the important text of the content. The text written between and will be displayed as important text.

Example

1. <p>In HTML it is recommended to use lower-case, while writing a code. </p>

Output:



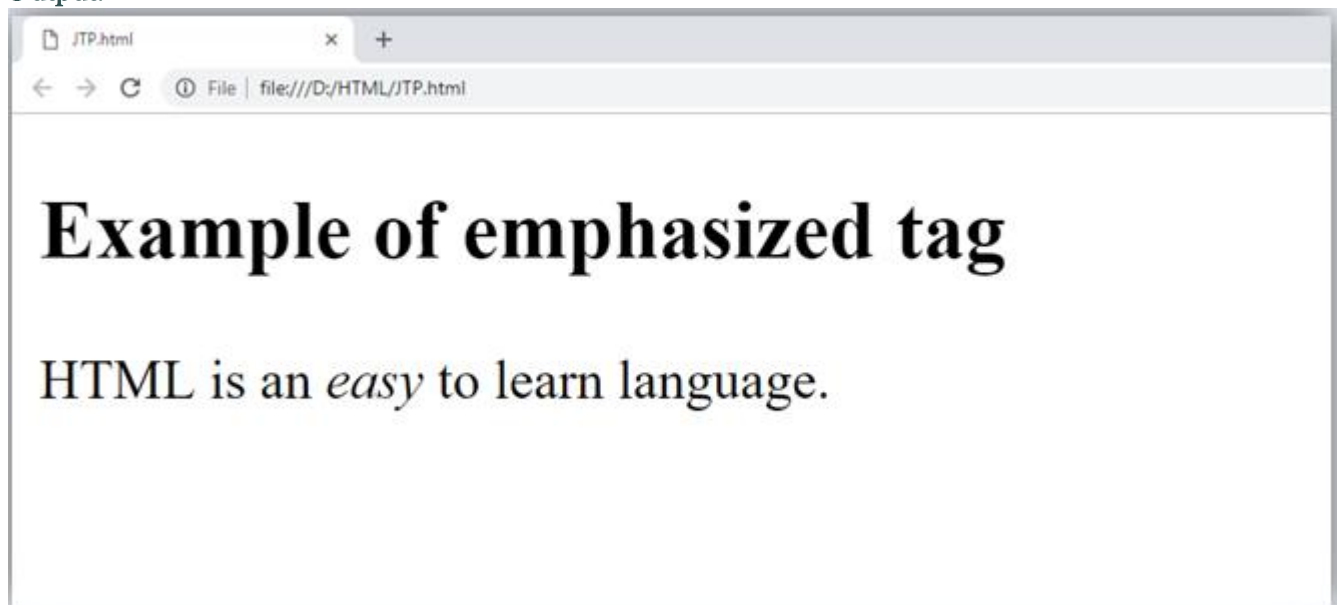
4. Emphasized text

This tag is used to emphasize the text, and displayed the text in italic form. The text written between `` and `` tag will italicized the text.

Example

1. `<p>`HTML is an ``easy ``to learn language.`</p>`

Output:



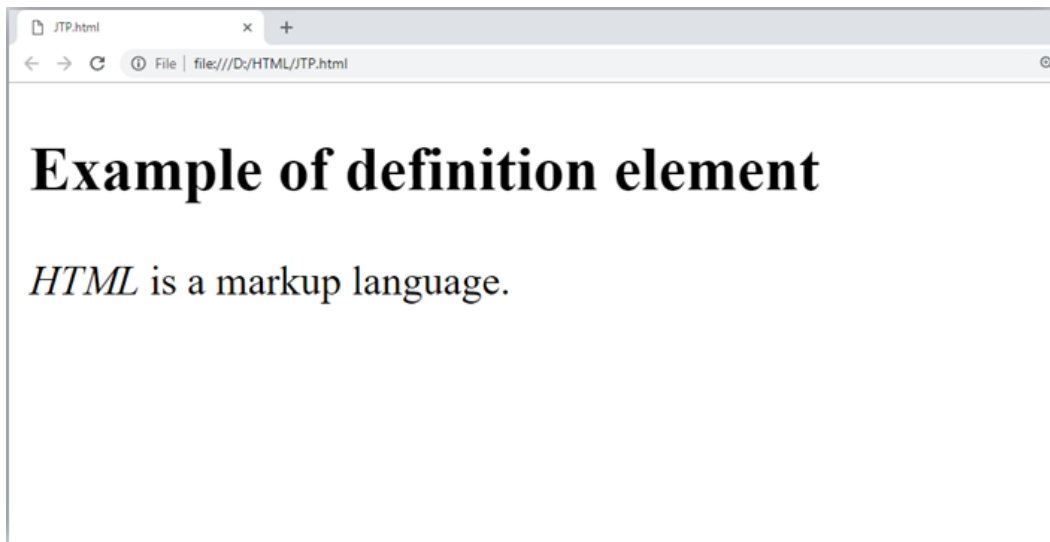
5. Definition tag:

When you use the `<dfn>` and `</dfn>` tags, it allow to specify the keyword of the content. Following is the example to show how to definition element.

Example

1. `<p>``<dfn>`HTML `</dfn>` is a markup language. `</p>`

Output:



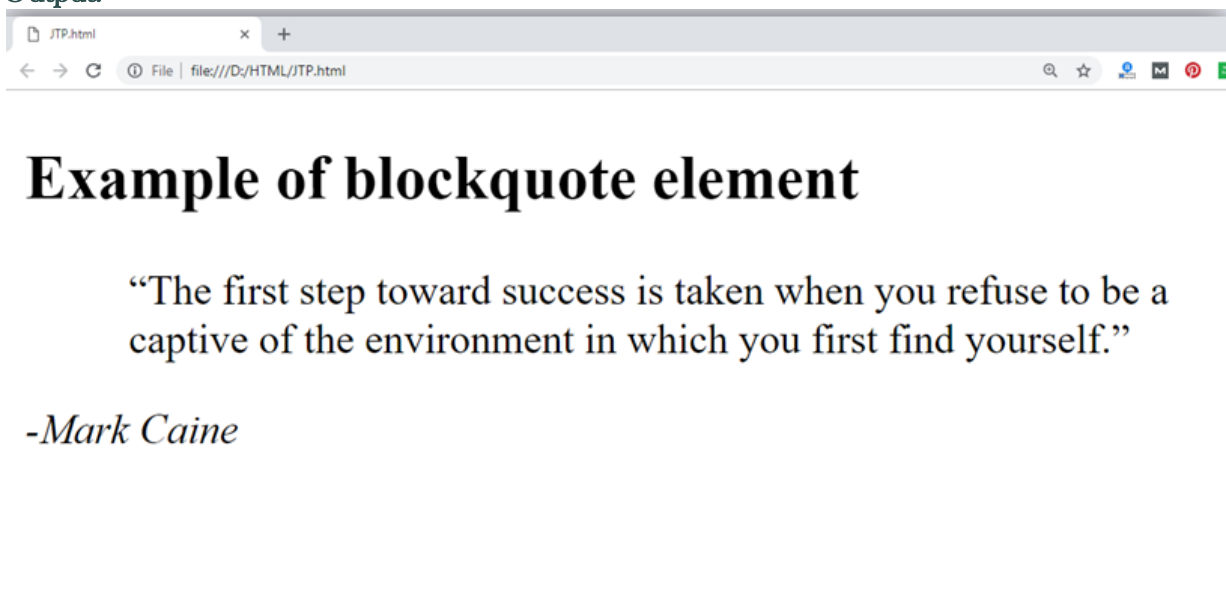
6. Quoting text:

The HTML `<blockquote>` element shows that the enclosed content is quoted from another source. The Source URL can be given using the `cite` attribute, and text representation of source can display using `<cite>` `</cite>` element.

Example

1. `<blockquote cite="https://www.keepinspiring.me/famous-quotes/"><p>?The first step toward success is taken when you refuse to be a captive of the environment in which you first find yourself.?</p></blockquote>`
2. `<cite>-Mark Caine</cite>`

Output:



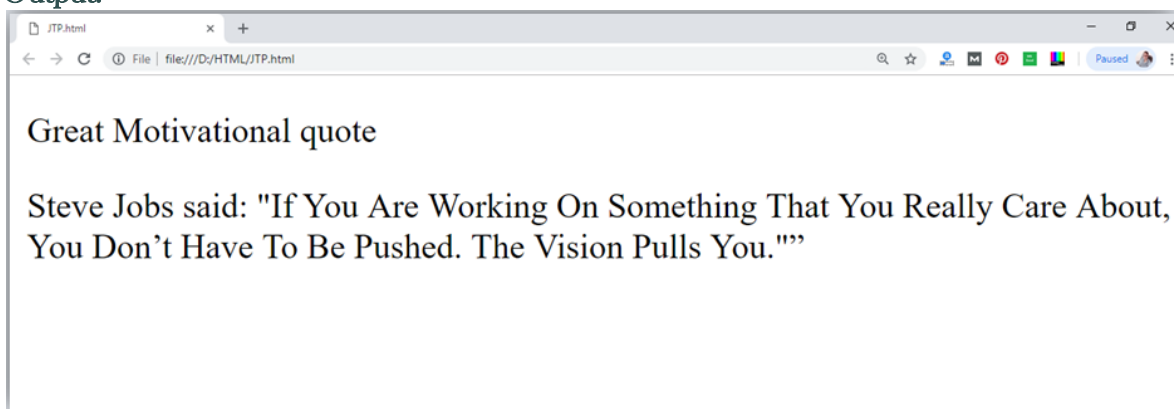
7. Short Quotations:

An HTML `<q> </q>` element defines a short quotation. If you will put any content between `<q> </q>`, then it will enclose the text in double quotes.

Example:

1. `<p>Steve Jobs said: <q>If You Are Working On Something That You Really Care About, You Don't Have To Be Pushed. The Vision Pulls You.</q>?</p>`

Output:

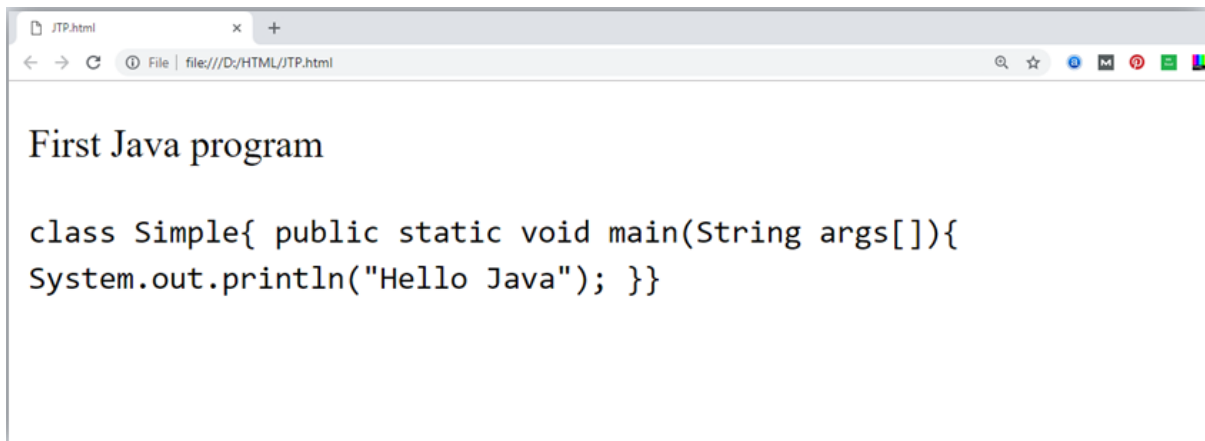


8. Code tags

The HTML `<code> </code>` element is used to display the part of computer code. It will display the content in monospaced font.

1. `<p>First Java program</p>`
2. `<p><code>class Simple{ public static void main(String args[]){`
3. `System.out.println("Hello Java"); }} </code>`
4. `</p>`

Output:

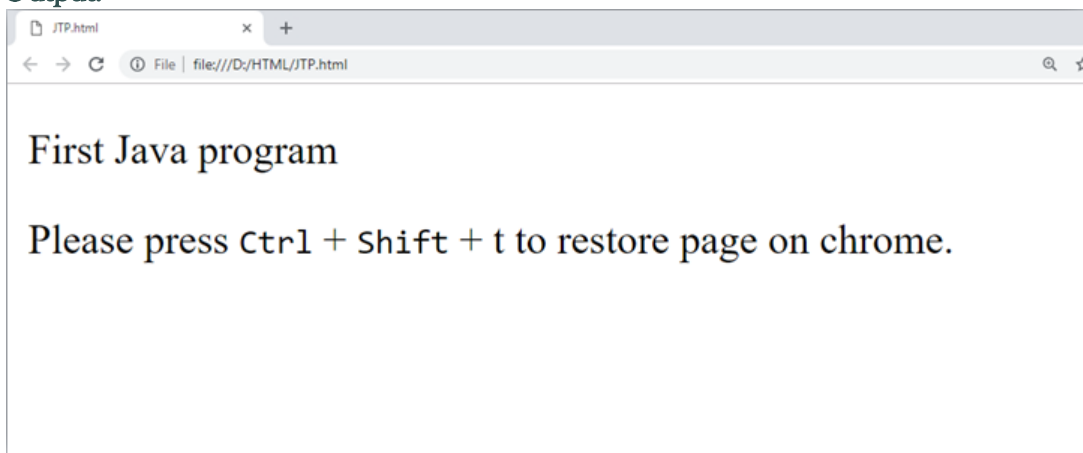


9. Keyboard Tag

In HTML the keyboard tag, `<kbd>`, indicates that a section of content is a user input from keyboard.

1. `<p>`Please press `<kbd>`Ctrl`</kbd>` + `<kbd>`Shift`</kbd>` + t`<kbd>``</kbd>` to restore page on chrome.`</p>`

Output:



10. Address tag

An HTML `<address>` tag defines the contact information about the author of the content. The content written between `<address>` and `</address>` tag, then it will be displayed in italic font.

1. `<address>` You can ask your queries by contact us on ``example123@newdomain.com``
2. `
` You can also visit at: `
`58 S. Garfield Street. Villa Rica, GA 30187.
3. `</address>`

Output:



HTML Anchor

The **HTML anchor tag** defines a *hyperlink that links one page to another page*. It can create hyperlink to other web page as well as files, location, or any URL. The "href" attribute is the most important attribute of the HTML a tag, and which links to destination page or URL.

href attribute of HTML anchor tag

The href attribute is used to define the address of the file to be linked. In other words, it points out the destination page.

The syntax of HTML anchor tag is given below.

```
<a href = "....."> Link Text </a>
```

Let's see an example of HTML anchor tag.

1. `Click for Second Page`

Specify a location for Link using target attribute

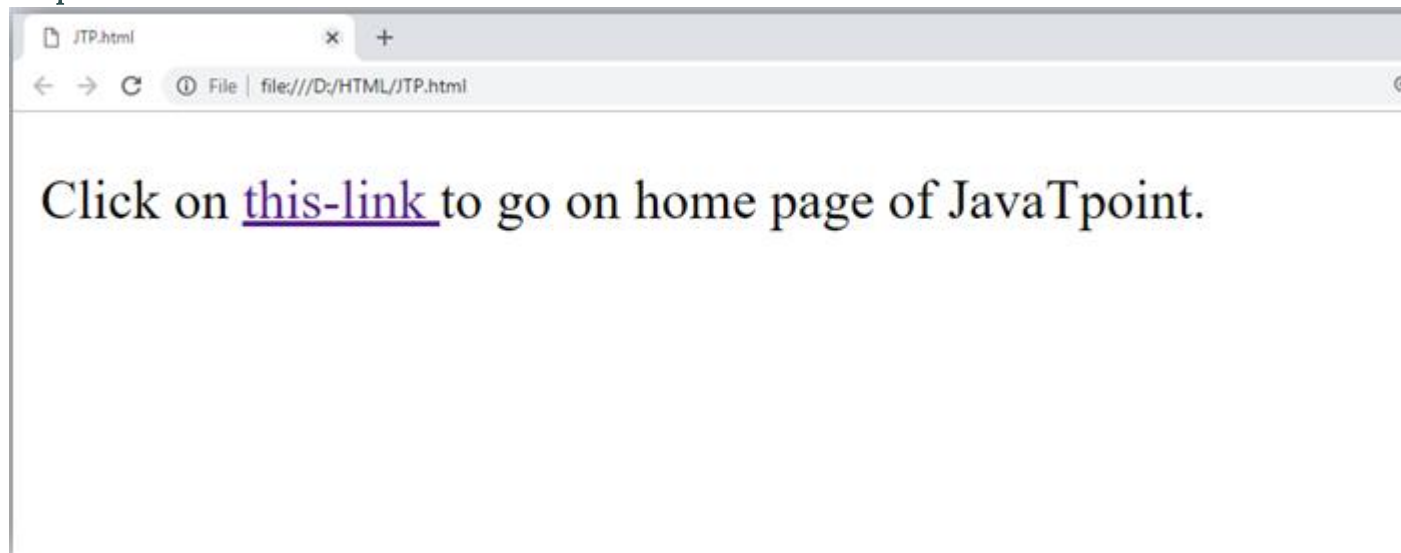
If we want to open that link to another page then we can use target attribute of <a> tag. With the help of this link will be open in next page.

Example:

1. `<!DOCTYPE html>`
2. `<html>`
3. `<head>`
4. `<title></title>`
5. `</head>`

6. `<body>`
7. `<p>`Click on `` this-link ``to go on home page of JavaTpoint.`</p>`
8. `</body>`
9. `</html>`

Output:



Note:

- The **target** attribute can only use with href attribute in anchor tag.
- If we will not use target attribute then link will open in same page.

Appearance of HTML anchor tag

An **unvisited link** is displayed underlined and blue.

A **visited link** displayed underlined and purple.

An **active link** is underlined and red.

HTML Image

HTML img tag is used to display image on the web page. HTML img tag is an empty tag that contains attributes only, closing tags are not used in HTML image element.

Let's see an example of HTML image.

1. `<h2>`HTML Image Example`</h2>`
2. ``

Output:



Attributes of HTML img tag

The src and alt are important attributes of HTML img tag. All attributes of HTML image tag are given below.

1) src

It is a necessary attribute that describes the source or path of the image. It instructs the browser where to look for the image on the server.

The location of image may be on the same directory or another server.

2) alt

The alt attribute defines an alternate text for the image, if it can't be displayed. The value of the alt attribute describe the image in words. The alt attribute is considered good for SEO prospective.

3) width

It is an optional attribute which is used to specify the width to display the image. It is not recommended now. You should apply CSS in place of width attribute.

4) height

It h3 the height of the image. The HTML height attribute also supports iframe, image and object elements. It is not recommended now. You should apply CSS in place of height attribute.

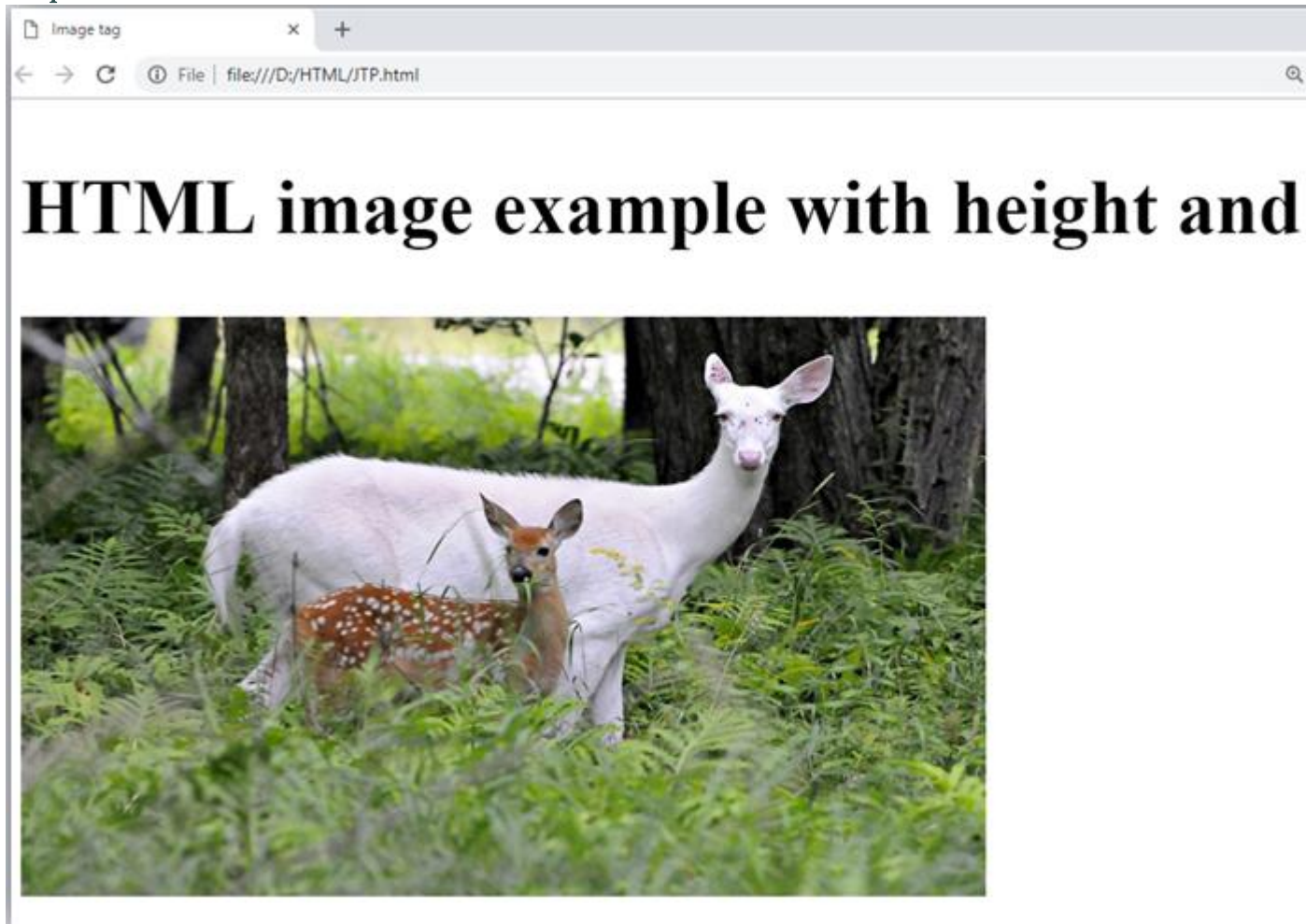
Use of height and width attribute with img tag

You have learnt about how to insert an image in your web page, now if we want to give some height and width to display image according to our requirement, then we can set it with height and width attributes of image.

Example:

1. ``

Output:



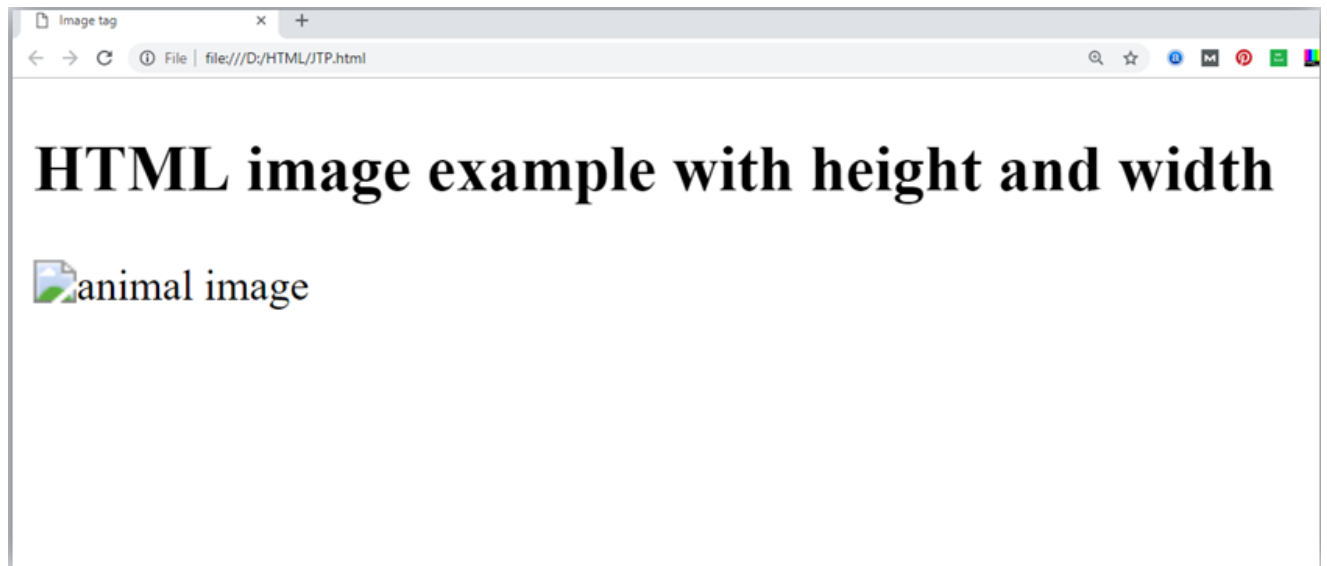
Note: Always try to insert the image with height and width, else it may flicker while displaying on webpage.

Use of alt attribute

We can use alt attribute with `` tag. It will display an alternative text in case if image cannot be displayed on browser. Following is the example for alt attribute:

1. ``

Output:



How to get image from another directory/folder?

To insert an image in your web, that image must be present in your same folder where you have put the HTML file. But if in some case image is available in some other directory then you can access the image like this:

1. ``

In above statement we have put image in local disk E----->images folder----->animal.png.

Note: If src URL will be incorrect or misspell then it will not display your image on web page, so try to put correct URL.

Use tag as a link

We can also link an image with other page or we can use an image as a link. To do this, put tag inside the <a> tag.

Example:

1. ``

Output:



HTML Table

HTML table tag is used to display data in tabular form (row * column). There can be many columns in a row.

We can create a table to display data in tabular form, using `<table>` element, with the help of `<tr>`, `<td>`, and `<th>` elements.

In Each table, table row is defined by `<tr>` tag, table header is defined by `<th>`, and table data is defined by `<td>` tags.

HTML tables are used to manage the layout of the page e.g. header section, navigation bar, body content, footer section etc. But it is recommended to use div tag over table to manage the layout of the page .

HTML Table Tags

Tag	Description
<code><table></code>	It defines a table.
<code><tr></code>	It defines a row in a table.
<code><th></code>	It defines a header cell in a table.
<code><td></code>	It defines a cell in a table.
<code><caption></code>	It defines the table caption.
<code><colgroup></code>	It specifies a group of one or more columns in a table for formatting.
<code><col></code>	It is used with <code><colgroup></code> element to specify column properties for each column.
<code><tbody></code>	It is used to group the body content in a table.
<code><thead></code>	It is used to group the header content in a table.
<code><tfooter></code>	It is used to group the footer content in a table.

HTML Table Example

Let's see the example of HTML table tag. Its output is shown above.

1. `<table>`
2. `<tr><th>First_Name</th><th>Last_Name</th><th>Marks</th></tr>`

3. `<tr><td>Sonoo</td><td>Jaiswal</td><td>60</td></tr>`
4. `<tr><td>James</td><td>William</td><td>80</td></tr>`
5. `<tr><td>Swati</td><td>Sironi</td><td>82</td></tr>`
6. `<tr><td>Chetna</td><td>Singh</td><td>72</td></tr>`
7. `</table>`

Output:

First_Name	Last_Name	Marks
Sonoo	Jaiswal	60
James	William	80
Swati	Sironi	82
Chetna	Singh	72

In the above html table, there are 5 rows and 3 columns = $5 * 3 = 15$ values.

HTML Table with Border

There are two ways to specify border for HTML tables.

1. By border attribute of table in HTML
 2. By border property in CSS
-

1) HTML Border attribute

You can use border attribute of table tag in HTML to specify border. But it is not recommended now.

1. `<table border="1">`
2. `<tr><th>First_Name</th><th>Last_Name</th><th>Marks</th></tr>`
3. `<tr><td>Sonoo</td><td>Jaiswal</td><td>60</td></tr>`
4. `<tr><td>James</td><td>William</td><td>80</td></tr>`
5. `<tr><td>Swati</td><td>Sironi</td><td>82</td></tr>`
6. `<tr><td>Chetna</td><td>Singh</td><td>72</td></tr>`
7. `</table>`

Output:

First_Name	Last_Name	Marks
Sonoo	Jaiswal	60
James	William	80

Swati	Sironi	82
Chetna	Singh	72

2) CSS Border property

It is now recommended to use border property of CSS to specify border in table.

1. `<style>`
2. table, th, td {
3. border: 1px solid black;
4. }
5. `</style>`

You can collapse all the borders in one border by border-collapse property. It will collapse the border into one.

1. `<style>`
2. table, th, td {
3. border: 2px solid black;
4. border-collapse: collapse;
5. }
6. `</style>`

Output:

Name	Last Name	Marks
Sonoo	Jaiswal	60
James	William	80
Swati	Sironi	82
Chetna	Singh	72

HTML Table with cell padding

You can specify padding for table header and table data by two ways:

1. By cellpadding attribute of table in HTML
2. By padding property in CSS

The cellpadding attribute of HTML table tag is obsolete now. It is recommended to use CSS. So let's see the code of CSS.

1. `<style>`
2. table, th, td {

```
3. border: 1px solid pink;
4. border-collapse: collapse;
5. }
6. th, td {
7. padding: 10px;
8. }
9. </style>
```

Output:

Name	Last Name	Marks
Sonoo	Jaiswal	60
James	William	80
Swati	Sironi	82
Chetna	Singh	72

HTML Table width:

We can specify the HTML table width using the **CSS width** property. It can be specify in pixels or percentage.

We can adjust our table width as per our requirement. Following is the example to display table with width.

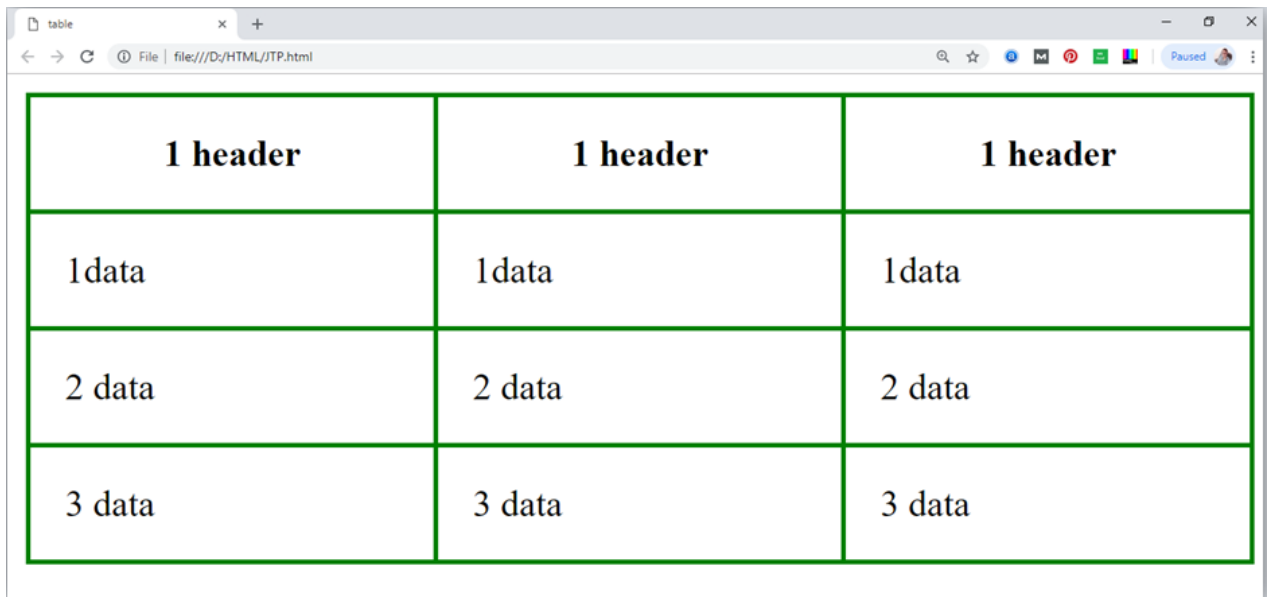
```
1. table{
2. width: 100%;
3. }
```

Example:

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4. <title>table</title>
5. <style>
6. table{
7. border-collapse: collapse;
8. width: 100%;
9. }
```

```
10.  th,td{
11.      border: 2px solid green;
12.      padding: 15px;
13.  }
14.
15.  </style>
16. </head>
17. <body>
18.  <table>
19.    <tr>
20.      <th>1 header</th>
21.      <th>1 header</th>
22.      <th>1 header</th>
23.    </tr>
24.    <tr>
25.      <td>1 data</td>
26.      <td>1 data</td>
27.      <td>1 data</td>
28.    </tr>
29.    <tr>
30.      <td>2 data</td>
31.      <td>2 data</td>
32.      <td>2 data</td>
33.    </tr>
34.    <tr>
35.      <td>3 data</td>
36.      <td>3 data</td>
37.      <td>3 data</td>
38.    </tr>
39. </table>
40. </body>
41. </html>
```

Output:



1 header	1 header	1 header
1 data	1 data	1 data
2 data	2 data	2 data
3 data	3 data	3 data

HTML Table with colspan

If you want to make a cell span more than one column, you can use the colspan attribute.

It will divide one cell/row into multiple columns, and the number of columns depend on the value of colspan attribute.

Let's see the example that span two columns.

CSS code:

1. `<style>`
2. `table, th, td {`
3. `border: 1px solid black;`
4. `border-collapse: collapse;`
5. `}`
6. `th, td {`
7. `padding: 5px;`
8. `}`
9. `</style>`

HTML code:

1. `<table style="width:100%">`
2. `<tr>`
3. `<th>Name</th>`
4. `<th colspan="2">Mobile No.</th>`
5. `</tr>`
6. `<tr>`
7. `<td>Ajeet Maurya</td>`

8. `<td>7503520801</td>`
9. `<td>9555879135</td>`
10. `</tr>`
11. `</table>`

Output:

Name	Mobile No.	
Ajeet Maurya	7503520801	9555879135

HTML Table with rowspan

If you want to make a cell span more than one row, you can use the rowspan attribute.

It will divide a cell into multiple rows. The number of divided rows will depend on rowspan values.

Let's see the example that span two rows.

CSS code:

1. `<style>`
2. `table, th, td {`
3. `border: 1px solid black;`
4. `border-collapse: collapse;`
5. `}`
6. `th, td {`
7. `padding: 10px;`
8. `}`
9. `</style>`

HTML code:

1. `<table>`
2. `<tr><th>Name</th><td>Ajeet Maurya</td></tr>`
3. `<tr><th rowspan="2">Mobile No.</th><td>7503520801</td></tr>`
4. `<tr><td>9555879135</td></tr>`
5. `</table>`

Output:

Name	Ajeet Maurya
------	--------------

Mobile No.	7503520801
	9555879135

HTML table with caption

HTML caption is displayed above the table. It must be used after table tag only.

1. `<table>`
2. `<caption>Student Records</caption>`
3. `<tr><th>First_Name</th><th>Last_Name</th><th>Marks</th></tr>`
4. `<tr><td>Vimal</td><td>Jaiswal</td><td>70</td></tr>`
5. `<tr><td>Mike</td><td>Warn</td><td>60</td></tr>`
6. `<tr><td>Shane</td><td>Warn</td><td>42</td></tr>`
7. `<tr><td>Jai</td><td>Malhotra</td><td>62</td></tr>`
8. `</table>`

Styling HTML table even and odd cells

CSS code:

1. `<style>`
2. `table, th, td {`
3. `border: 1px solid black;`
4. `border-collapse: collapse;`
5. `}`
6. `th, td {`
7. `padding: 10px;`
8. `}`
9. `table#alter tr:nth-child(even) {`
10. `background-color: #eee;`
11. `}`
12. `table#alter tr:nth-child(odd) {`
13. `background-color: #fff;`
14. `}`
15. `table#alter th {`
16. `color: white;`
17. `background-color: gray;`
18. `}`
19. `</style>`

Output:

First_Name	Last_Name	Marks
Sonoo	Jaiswal	60
James	William	80
Swati	Sironi	82
Chetna	Singh	72

HTML Lists

HTML Lists are used to specify lists of information. All lists may contain one or more list elements. There are three different types of HTML lists:

1. Ordered List or Numbered List (ol)
2. Unordered List or Bulleted List (ul)
3. Description List or Definition List (dl)

Note: We can create a list inside another list, which will be termed as nested List.

HTML Ordered List or Numbered List

In the ordered HTML lists, all the list items are marked with numbers by default. It is known as numbered list also. The ordered list starts with `` tag and the list items start with `` tag.

1. ``
2. `Aries`
3. `Bingo`
4. `Leo`
5. `Oracle`
6. ``

Output:

1. Aries
2. Bingo
3. Leo
4. Oracle

Click here for full details of HTML ordered list. [HTML Ordered List](#)

HTML Unordered List or Bulleted List

In HTML Unordered list, all the list items are marked with bullets. It is also known as bulleted list also. The Unordered list starts with `` tag and list items start with the `` tag.

1. ``
2. `Aries`
3. `Bingo`
4. `Leo`
5. `Oracle`
6. ``

Output:

- Aries
- Bingo
- Leo
- Oracle

Click here for full details of HTML unordered list. [HTML Unordered List](#)

HTML Description List or Definition List

HTML Description list is also a list style which is supported by HTML and XHTML. It is also known as definition list where entries are listed like a dictionary or encyclopedia.

The definition list is very appropriate when you want to present glossary, list of terms or other name-value list.

The HTML definition list contains following three tags:

1. `<dl>` tag defines the start of the list.
2. `<dt>` tag defines a term.
3. `<dd>` tag defines the term definition (description).

1. `<dl>`
2. `<dt>Aries</dt>`
3. `<dd>-One of the 12 horoscope sign.</dd>`
4. `<dt>Bingo</dt>`
5. `<dd>-One of my evening snacks</dd>`
6. `<dt>Leo</dt>`
7. `<dd>-It is also an one of the 12 horoscope sign.</dd>`
8. `<dt>Oracle</dt>`
9. `<dd>-It is a multinational technology corporation.</dd>`
10. `</dl>`

Output:

Aries

-One of the 12 horoscope sign.

Bingo

-One of my evening snacks

Leo

-It is also an one of the 12 horoscope sign.

Oracle

-It is a multinational technology corporation.

Click here for full details of HTML description list. [HTML Description List](#)

HTML Nested List

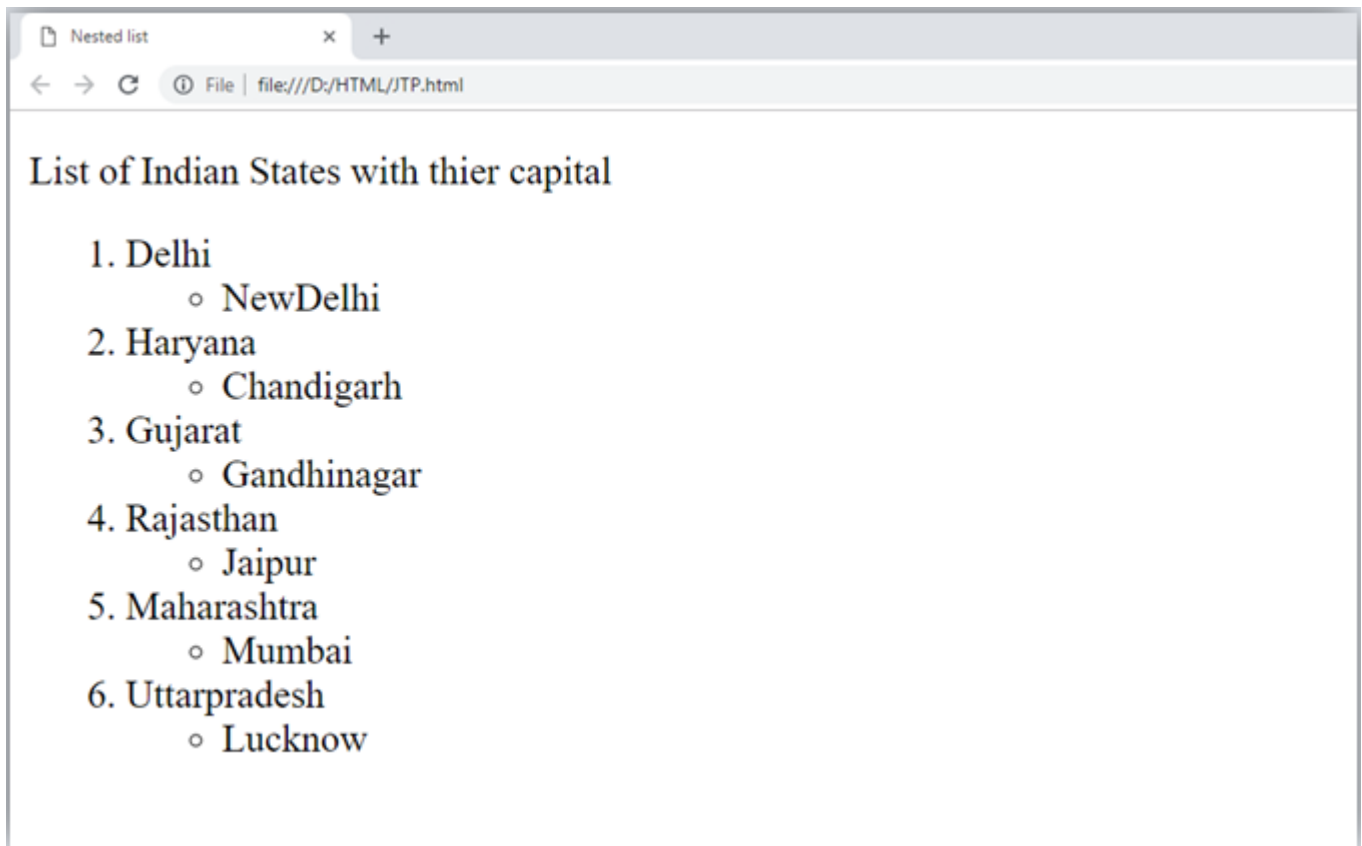
A list within another list is termed as nested list. If you want a bullet list inside a numbered list then such type of list will called as nested list.

Code:

1. `<!DOCTYPE html>`
2. `<html>`
3. `<head>`
4. `<title>Nested list</title>`
5. `</head>`
6. `<body>`
7. `<p>List of Indian States with thier capital</p>`

```
8. <ol>
9.   <li>Delhi
10.    <ul>
11.      <li>NewDelhi</li>
12.    </ul>
13.  </li>
14.  <li>Haryana
15.    <ul>
16.      <li>Chandigarh</li>
17.    </ul>
18.  </li>
19.  <li>Gujarat
20.    <ul>
21.      <li>Gandhinagar</li>
22.    </ul>
23.  </li>
24.  <li>Rajasthan
25.    <ul>
26.      <li>Jaipur</li>
27.    </ul>
28.  </li>
29.  <li>Maharashtra
30.    <ul>
31.      <li>Mumbai</li>
32.    </ul>
33.  </li>
34.  <li>Uttarpradesh
35.    <ul>
36.      <li>Lucknow</li></ul>
37.  </li>
38. </ol>
39. </body>
40. </html>
```

Output:



HTML Form

An **HTML form** is *a section of a document* which contains controls such as text fields, password fields, checkboxes, radio buttons, submit button, menus etc.

An HTML form facilitates the user to enter data that is to be sent to the server for processing such as name, email address, password, phone number, etc. .

Why use HTML Form

HTML forms are required if you want to collect some data from of the site visitor.

For example: If a user want to purchase some items on internet, he/she must fill the form such as shipping address and credit/debit card details so that item can be sent to the given address.

HTML Form Syntax

1. `<form action="server url" method="get|post">`
 2. `//input controls e.g. textfield, textarea, radiobutton, button`
 3. `</form>`
-

HTML Form Tags

Let's see the list of HTML 5 form tags.

Tag	Description
<form>	It defines an HTML form to enter inputs by the used side.
<input>	It defines an input control.
<textarea>	It defines a multi-line input control.
<label>	It defines a label for an input element.
<fieldset>	It groups the related element in a form.
<legend>	It defines a caption for a <fieldset> element.
<select>	It defines a drop-down list.
<optgroup>	It defines a group of related options in a drop-down list.
<option>	It defines an option in a drop-down list.
<button>	It defines a clickable button.

HTML 5 Form Tags

Let's see the list of HTML 5 form tags.

Tag	Description
<datalist>	It specifies a list of pre-defined options for input control.
<keygen>	It defines a key-pair generator field for forms.
<output>	It defines the result of a calculation.

HTML <form> element

The HTML <form> element provide a document section to take input from user. It provides various interactive controls for submitting information to web server such as text field, text area, password field, etc.

Note: The `<form>` element does not itself create a form but it is container to contain all required form elements, such as `<input>`, `<label>`, etc.

Syntax:

1. `<form>`
 2. `//Form elements`
 3. `</form>`
-

HTML `<input>` element

The HTML `<input>` element is fundamental form element. It is used to create form fields, to take input from user. We can apply different input field to gather different information from user. Following is the example to show the simple text input.

Example:

1. `<body>`
2. `<form>`
3. Enter your name `
`
4. `<input type="text" name="username">`
5. `</form>`
6. `</body>`

Output:



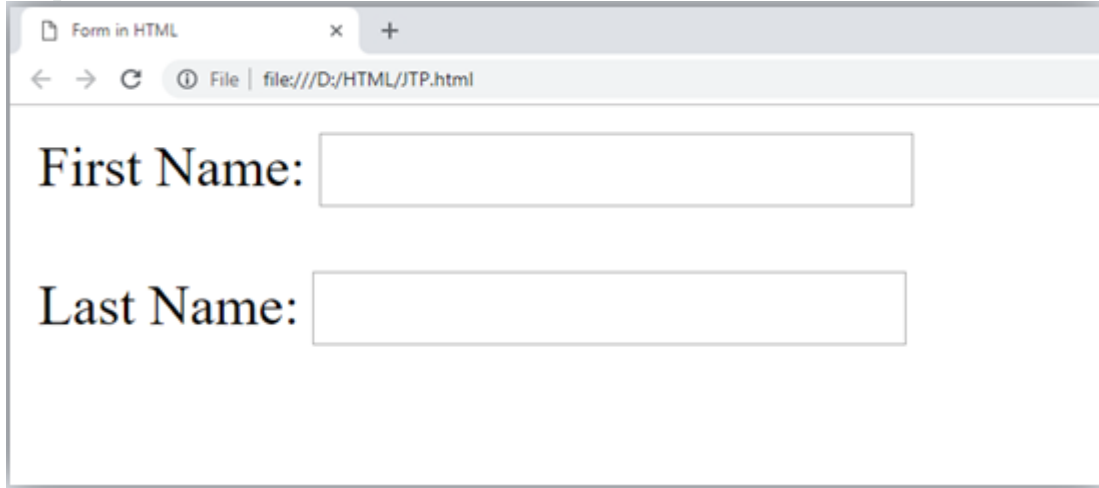
HTML TextField Control

The `type="text"` attribute of input tag creates textfield control also known as single line textfield control. The `name` attribute is optional, but it is required for the server side component such as JSP, ASP, PHP etc.

1. `<form>`

2. First Name: `<input type="text" name="firstname"/>
`
3. Last Name: `<input type="text" name="lastname"/>
`
4. `</form>`

Output:

A screenshot of a web browser window titled "Form in HTML". The address bar shows the file path "file:///D:/HTML/JTP.html". The browser content displays a simple form with two text input fields. The first field is preceded by the label "First Name:" and the second by "Last Name:". Both fields are empty and have a light gray border.

Note: If you will omit 'name' attribute then the text filed input will not be submitted to server.

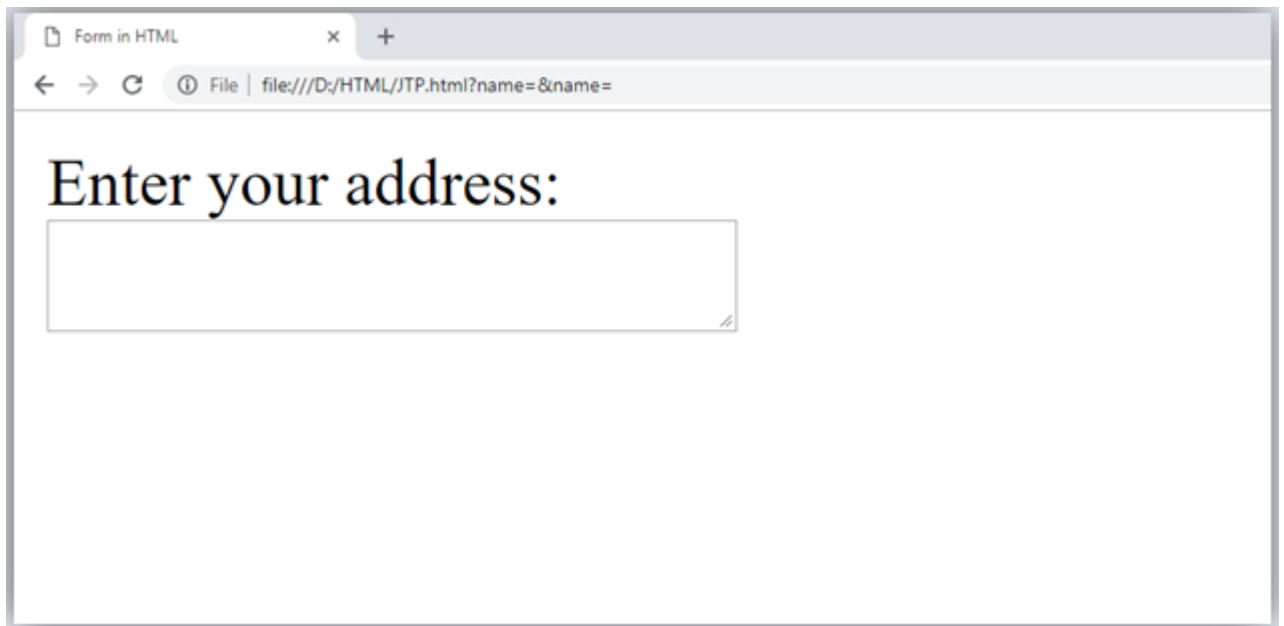
HTML `<textarea>` tag in form

The `<textarea>` tag in HTML is used to insert multiple-line text in a form. The size of `<textarea>` can be specify either using "rows" or "cols" attribute or by CSS.

Example:

1. `<!DOCTYPE html>`
2. `<html>`
3. `<head>`
4. `<title>Form in HTML</title>`
5. `</head>`
6. `<body>`
7. `<form>`
8. `Enter your address:
`
9. `<textarea rows="2" cols="20"></textarea>`
10. `</form>`
11. `</body>`
12. `</html>`

Output:



Label Tag in Form

It is considered better to have label in form. As it makes the code parser/browser/user friendly.

If you click on the label tag, it will focus on the text control. To do so, you need to have for attribute in label tag that must be same as id attribute of input tag.

NOTE: It is good to use <label> tag with form, although it is optional but if you will use it, then it will provide a focus when you tap or click on label tag. It is more worthy with touchscreens.

1. `<form>`
2. `<label for="firstname">First Name: </label>
`
3. `<input type="text" id="firstname" name="firstname"/>
`
4. `<label for="lastname">Last Name: </label>`
5. `<input type="text" id="lastname" name="lastname"/>
`
6. `</form>`

Output:

First Name:

Last Name:

HTML Password Field Control

The password is not visible to the user in password field control.

1. `<form>`
2. `<label for="password">Password: </label>`
3. `<input type="password" id="password" name="password"/>
`
4. `</form>`

Output:

Password:

HTML 5 Email Field Control

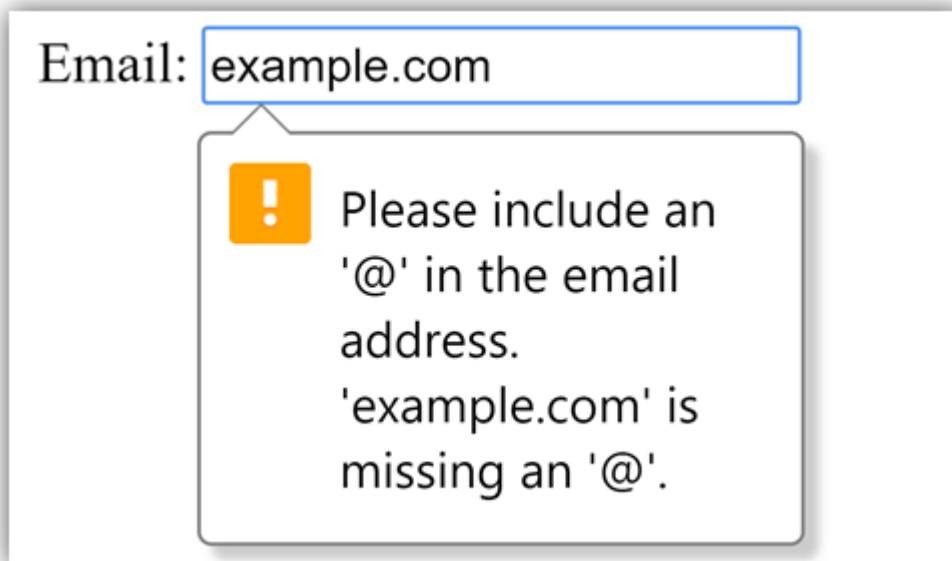
The email field is new in HTML 5. It validates the text for correct email address. You must use @ and . in this field.

1. `<form>`
2. `<label for="email">Email: </label>`
3. `<input type="email" id="email" name="email"/>
`
4. `</form>`

It will display in browser like below:

Email:

Note: If we will not enter the correct email, it will display error like:



Radio Button Control

The radio button is used to select one option from multiple options. It is used for selection of gender, quiz questions etc.

If you use one name for all the radio buttons, only one radio button can be selected at a time.

Using radio buttons for multiple options, you can only choose a single option at a time.

1. `<form>`
2. `<label for="gender">Gender: </label>`
3. `<input type="radio" id="gender" name="gender" value="male"/>Male`
4. `<input type="radio" id="gender" name="gender" value="female"/>Female
`
5. `</form>`

Gender: ☐ Male ☒ Female

Checkbox Control

The checkbox control is used to check multiple options from given checkboxes.

1. `<form>`
2. Hobby:`
`
3. `<input type="checkbox" id="cricket" name="cricket" value="cricket"/>`
4. `<label for="cricket">Cricket</label>
`
5. `<input type="checkbox" id="football" name="football" value="football"/>`
6. `<label for="football">Football</label>
`
7. `<input type="checkbox" id="hockey" name="hockey" value="hockey"/>`
8. `<label for="hockey">Hockey</label>`
9. `</form>`

Note: These are similar to radio button except it can choose multiple options at a time and radio button can select one button at a time, and its display.

Output:

Hobby:

☒ Cricket

☒ Football

☐ Hockey

Submit button control

HTML `<input type="submit">` are used to add a submit button on web page. When user clicks on submit button, then form get submit to the server.

Syntax:

1. `<input type="submit" value="submit">`

The type = submit , specifying that it is a submit button

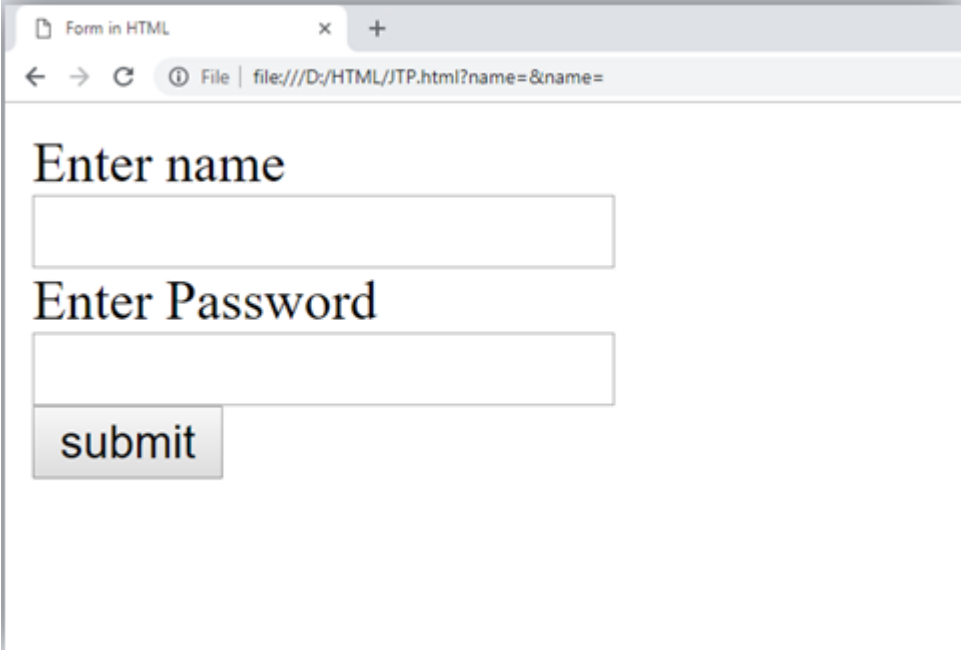
The value attribute can be anything which we write on button on web page.

The name attribute can be omit here.

Example:

1. `<form>`
2. `<label for="name">Enter name</label>
`
3. `<input type="text" id="name" name="name">
`
4. `<label for="pass">Enter Password</label>
`
5. `<input type="Password" id="pass" name="pass">
`
6. `<input type="submit" value="submit">`
7. `</form>`

Output:



HTML <fieldset> element:

The <fieldset> element in HTML is used to group the related information of a form. This element is used with <legend> element which provide caption for the grouped elements.

Example:

1. <form>
2. <fieldset>
3. <legend>User Information:</legend>
4. <label for="name">Enter name</label>

5. <input type="text" id="name" name="name">

6. <label for="pass">Enter Password</label>

7. <input type="Password" id="pass" name="pass">

8. <input type="submit" value="submit">
9. </fieldset>
10. </form>

Output:



HTML Form Example

Following is the example for a simple form of registration.

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <title>Form in HTML</title>
5. </head>
6. <body>
7. <h2>Registration form</h2>
8. <form>
9. <fieldset>
10. <legend>User personal information</legend>
11. <label>Enter your full name</label>

12. <input type="text" name="name">

13. <label>Enter your email</label>

14. <input type="email" name="email">


```

15.    <label>Enter your password</label><br>
16.    <input type="password" name="pass"><br>
17.    <label>confirm your password</label><br>
18.    <input type="password" name="pass"><br>
19.    <br><label>Enter your gender</label><br>
20.    <input type="radio" id="gender" name="gender" value="male"/>Male <br>
21.    <input type="radio" id="gender" name="gender" value="female"/>Female <br>
22.    <input type="radio" id="gender" name="gender" value="others"/>others <br>
23.    <br>Enter your Address:<br>
24.    <textarea></textarea><br>
25.    <input type="submit" value="sign-up">
26. </fieldset>
27. </form>
28. </body>
29. </html>

```

Output:

HTML Form Example

Let's see a simple example of creating HTML form.

```

1. <form action="#">
2. <table>
3. <tr>
4.   <td class="tdLabel"><label for="register_name" class="label">Enter name:</label></td>
5.   <td><input type="text" name="name" value="" id="register_name" style="width:160px"/></td>
6. </tr>
7. <tr>
8.   <td class="tdLabel"><label for="register_password" class="label">Enter password:</label></td>

```

```

9.    <td><input type="password" name="password" id="register_password" style="width:160px"/></t
    d>
10. </tr>
11. <tr>
12.    <td class="tdLabel"><label for="register_email" class="label">Enter Email:</label></td>
13.    <td
14. ><input type="email" name="email" value="" id="register_email" style="width:160px"/></td>
15. </tr>
16. <tr>
17.    <td class="tdLabel"><label for="register_gender" class="label">Enter Gender:</label></td>
18.    <td>
19. <input type="radio" name="gender" id="register_gendermale" value="male"/>
20. <label for="register_gendermale">male</label>
21. <input type="radio" name="gender" id="register_genderfemale" value="female"/>
22. <label for="register_genderfemale">female</label>
23. </td>
24. </tr>
25. <tr>
26.    <td class="tdLabel"><label for="register_country" class="label">Select Country:</label></td>
27.    <td><select name="country" id="register_country" style="width:160px">
28.    <option value="india">india</option>
29.    <option value="pakistan">pakistan</option>
30.    <option value="africa">africa</option>
31.    <option value="china">china</option>
32.    <option value="other">other</option>
33. </select>
34. </td>
35. </tr>
36. <tr>
37.    <td colspan="2"><div align="right"><input type="submit" id="register_0" value="register"/>
38. </div></td>
39. </tr>
40. </table>
41. </form>

```

HTML Classes

Class Attribute in HTML

The HTML class attribute is used to specify a single or multiple class names for an HTML element. The class name can be used by CSS and JavaScript to do some tasks for HTML

elements. You can use this class in CSS with a specific class, write a period (.) character, followed by the name of the class for selecting elements.

A class attribute can be defined within <style> tag or in separate file using the (.) character.

In an HTML document, we can use the same class attribute name with different elements.

Defining an HTML class

To create an HTML class, firstly define style for HTML class using <style> tag within <head> section as following example:

Example:

```
1. <head>
2.   <style>
3.     .headings{
4.       color: lightgreen;
5.       font-family: cursive;
6.       background-color: black; }
7.   </style>
8. </head>
```

We have define style for a class name "headings", and we can use this class name with any of HTML element in which we want to provide such styling. We just need to follow the following syntax to use it.

```
1. <tag class="ghf"> content </tag>
```

Example 1:

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4.   <style>
5.     .headings{
6.       color: lightgreen;
7.       font-family: cursive;
8.       background-color: black; }
9.   </style>
10. </head>
11. <body>
12. <h1 class="headings">This is first heading</h1>
13. <h2 class="headings">This is Second heading</h2>
14. <h3 class="headings">This is third heading</h3>
15. <h4 class="headings">This is fourth heading</h4>
16. </body>
17. </html>
```

Another Example with different class name

Example:

Let's use a class name "Fruit" with CSS to style all elements.

```
1. <style>
2. .fruit {
3.     background-color: orange;
4.     color: white;
5.     padding: 10px;
6. }
7. </style>
8.
9. <h2 class="fruit">Mango</h2>
10. <p>Mango is king of all fruits.</p>
11.
12. <h2 class="fruit">Orange</h2>
13. <p>Oranges are full of Vitamin C.</p>
14.
15. <h2 class="fruit">Apple</h2>
16. <p>An apple a day, keeps the Doctor away.</p>
```

Here you can see that we have used the class name "fruit" with (.) to use all its elements.

Note: You can use class attribute on any HTML element. The class name is case-sensitive.

Class Attribute in JavaScript

You can use JavaScript access elements with a specified class name by using the `getElementsByClassName()` method.

Example:

Let's hide all the elements with class name "fruit" when the user click on the button.

```
1. <!DOCTYPE html>
2. <html>
3. <body>
4.
```

```
5. <h2>Class Attribute with JavaScript</h2>
6. <p>Click the button, to hide all elements with the class name "fruit", with JavaScript:</p>
7.
8. <button onclick="myFunction()">Hide elements</button>
9.
10.
11. <h2 class="fruit">Mango</h2>
12. <p>Mango is king of all fruits.</p>
13.
14. <h2 class="fruit">Orange</h2>
15. <p>Oranges are full of Vitamin C.</p>
16.
17. <h2 class="fruit">Apple</h2>
18. <p>An apple a day, keeps the Doctor away.</p>
19.
20. <script>
21. function myFunction() {
22.   var x = document.getElementsByClassName("fruit");
23.   for (var i = 0; i < x.length; i++) {
24.     x[i].style.display = "none";
25.   }
26. }
27. </script>
28.
29. </body>
30. </html>
```

Note: You will learn more about JavaScript in our JavaScript tutorial.

Multiple Classes

You can use multiple class names (more than one) with HTML elements. These class names must be separated by a space.

Example:

Let's style elements with class name "fruit" and also with a class name "center".

```
1. <!DOCTYPE html>
2. <html>
3. <style>
4. .fruit {
```

```

5.    background-color: orange;
6.    color: white;
7.    padding: 10px;
8.  }
9.
10. .center {
11.   text-align: center;
12. }
13. </style>
14. <body>
15.
16. <h2>Multiple Classes</h2>
17. <p>All three elements have the class name "fruit". In addition, Mango also have the class name
    "center", which center-aligns the text.</p>
18.
19. <h2 class="fruit center">Mango</h2>
20. <h2 class="fruit">Orange</h2>
21. <h2 class="fruit">Apple</h2>
22.
23. </body>
24. </html>

```

You can see that the first element `<h2>` belongs to both the "fruit" class and the "center" class.

Same class with Different Tag

You can use the same class name with different tags like `<h2>` and `<p>` etc. to share the same style.

Example:

```

1. <!DOCTYPE html>
2. <html>
3. <style>
4. .fruit {
5.   background-color: orange;
6.   color: white;
7.   padding: 10px;
8. }
9. </style>
10. <body>
11. <h2>Same Class with Different Tag</h2>
12. <h2 class="fruit">Mango</h2>

```

13. `<p class="fruit">Mango is the king of all fruits.</p>`
14. `</body>`
15. `</html>`

HTML Id Attribute

The id attribute refers to a unique value for an HTML element. This HTML id value can be used with CSS and JavaScript to perform certain task.

HTML id with CSS

In CSS, if you want to select an element with a specific id, write a hash (#) character, followed by the id of the element.

Example:

Use the HTML id "myid" with CSS:

1. `<style>`
2. `#myid {`
3. `background-color: lightpink;`
4. `color: black;`
5. `padding: 40px;`
6. `text-align: center;`
7. `}`
8. `</style>`
9. `<h1 id="myid">Example of HTML id</h1>`

Note: Here the id attribute is "myid" which can be used on any HTML element. The HTML id value is case-sensitive and it must contain at least one character, and must not contain whitespace (spaces, tabs, etc.).

Difference between HTML Class and ID

An HTML class name can be used by multiple elements while An HTML element can only have one unique id that belongs to that single element.

Example:

1. `<!DOCTYPE html>`
2. `<html>`
3. `<head>`
4. `<style>`

```

5. /* Style the element with the id "myid" */
6. #myid {
7.     background-color: pink;
8.     color: black;
9.     padding: 40px;
10.    text-align: center;
11. }
12.
13. /* Style all elements with the class name "fruit" */
14. .fruit {
15.     background-color: orange;
16.     color: white;
17.     padding: 10px;
18. }
19. </style>
20. </head>
21. <body>
22.
23. <h2>Difference Between Class and ID</h2>
24.
25. <h1 id="myid">My Favorite Fruits</h1>
26.
27. <h2 class="fruit">Mango</h2>
28. <p>The King of all fruits.</p>
29.
30. <h2 class="fruit">Orange</h2>
31. <p>Full of Vitamin C</p>
32.
33. <h2 class="fruit">Apple</h2>
34. <p>An apple a day, keeps the doctor away.</p>
35.
36. </body>
37. </html>

```

HTML id with JavaScript

You can use HTML id with JavaScript `getElementById()` method to access an element with a specified id.

Example:

```

1. <!DOCTYPE html>
2. <html>

```

3. `<body>`
4. `<h2>HTML id with JavaScript</h2>`
5. `<h1 id="myid">Hello JavaTpoint!</h1>`
6. `<button onclick="displayResult()">Change text</button>`
7. `<script>`
8. `function displayResult() {`
9. `document.getElementById("myid").innerHTML = "All the best for future!";`
10. `}`
11. `</script>`
12. `</body>`
13. `</html>`

HTML Id Attribute

The id attribute refers to a unique value for an HTML element. This HTML id value can be used with CSS and JavaScript to perform certain task.

HTML id with CSS

In CSS, if you want to select an element with a specific id, write a hash (#) character, followed by the id of the element.

Example:

Use the HTML id "myid" with CSS:

1. `<style>`
2. `#myid {`
3. `background-color: lightpink;`
4. `color: black;`
5. `padding: 40px;`
6. `text-align: center;`
7. `}`
8. `</style>`
9. `<h1 id="myid">Example of HTML id</h1>`

Note: Here the id attribute is "myid" which can be used on any HTML element. The HTML id value is case-sensitive and it must contain at least one character, and must not contain whitespace (spaces, tabs, etc.).

Difference between HTML Class and ID

An HTML class name can be used by multiple elements while An HTML element can only have one unique id that belongs to that single element.

Example:

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4. <style>
5. /* Style the element with the id "myid" */
6. #myid {
7.     background-color: pink;
8.     color: black;
9.     padding: 40px;
10.    text-align: center;
11. }
12.
13. /* Style all elements with the class name "fruit" */
14. .fruit {
15.     background-color: orange;
16.     color: white;
17.     padding: 10px;
18. }
19. </style>
20. </head>
21. <body>
22.
23. <h2>Difference Between Class and ID</h2>
24.
25. <h1 id="myid">My Favorite Fruits</h1>
26.
27. <h2 class="fruit">Mango</h2>
28. <p>The King of all fruits.</p>
29.
30. <h2 class="fruit">Orange</h2>
31. <p>Full of Vitamin C</p>
32.
33. <h2 class="fruit">Apple</h2>
34. <p>An apple a day, keeps the doctor away.</p>
35.
36. </body>
37. </html>
```

HTML id with JavaScript

You can use HTML id with JavaScript `getElementById()` method to access an element with a specified id.

Example:

1. `<!DOCTYPE html>`
2. `<html>`
3. `<body>`
4. `<h2>HTML id with JavaScript</h2>`
5. `<h1 id="myid">Hello JavaTpoint!</h1>`
6. `<button onclick="displayResult()">Change text</button>`
7. `<script>`
8. `function displayResult() {`
9. `document.getElementById("myid").innerHTML = "All the best for future!";`
10. `}`
11. `</script>`
12. `</body>`
13. `</html>`

HTML `` tag

HTML `` tag is used as a generic container of inline elements. It is used for styling purpose to the grouped inline elements (using class and id attribute or inline style).

The `` tag does not have any default meaning or rendering.

The `` tag can be useful for the following task:

- To change the language of a part of the text.
- To change the color, font, background of a part of text using CSS
- To apply the scripts to the particular part of the text.

Note: HTML `` is much similar as `<div>` tag, but `<div>` is used for block-level elements and `` tag is used for inline elements.

Syntax

1. ``Write your content here.....``

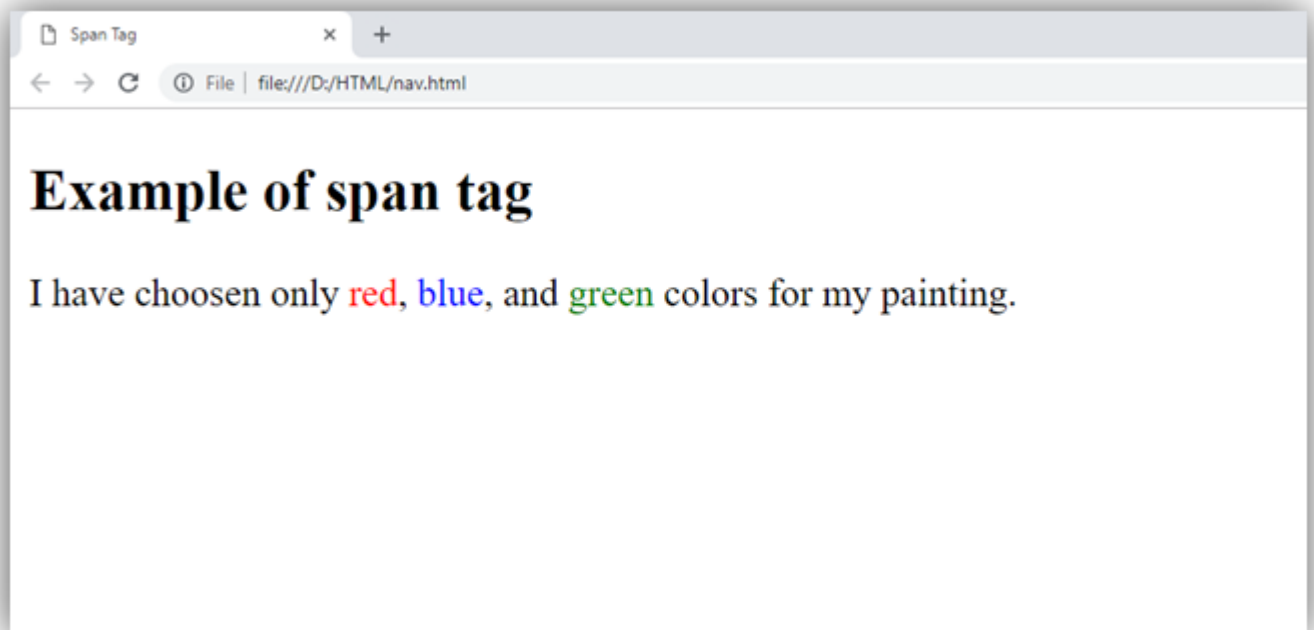
Following are some specifications about the HTML `` tag

Display	Inline
Start tag/End tag	Both Start and End tag
Usage	Styles and semantics

Example

1. `<!DOCTYPE html>`
2. `<html>`
3. `<head>`
4. `<title>Span Tag</title>`
5. `</head>`
6. `<body>`
7. `<h2>Example of span tag</h2>`
8. `<p>I have choosen only`
9. `red,`
10. `blue, and`
11. `green colors for my painting.`
12. `</p>`
13. `</body>`
14. `</html>`

Output:



Attribute:

Tag-specific attributes:

The `` tag does not contain any specific attribute in HTML.

Global attribute:

The `` tag supports the Global attributes in HTML.

Event attribute:

The `` tag supports the Event attributes in HTML.

HTML Div Tag

The **HTML `<div>` tag** is used *to group the large section of HTML elements together.*

We know that every tag has a specific purpose e.g. `p` tag is used to specify paragraph, `<h1>` to `<h6>` tag are used to specify headings but the `<div>` tag is just like a container unit which is used to encapsulate other page elements and divides the HTML documents into sections.

The `div` tag is generally used by web developers to group HTML elements together and apply CSS styles to many elements at once. For example: If you wrap a set of paragraph elements into a `div` element so you can take the advantage of CSS styles and apply font style to all paragraphs at once instead of coding the same style for each paragraph element.

1. `<div style="border:1px solid pink;padding:20px;font-size:20px">`
2. `<p>Welcome to Javatpoint.com, Here you get tutorials on latest technologies.</p>`
3. `<p>This is second paragraph</p>`
4. `</div>`

Output:

Welcome to Javatpoint.com, Here you get tutorials on latest technologies.

This is second paragraph.

Difference between HTML div tag and span tag

div tag	span tag
HTML div is a block element.	HTML span is an inline element
HTML div element is used to wrap large sections of elements .	HTML span element is used to wrap small portion of texts, image etc.

HTML div example: Login Form

In this example, we are creating box using div tag. There is a login form inside the box. Let's see the CSS and HTML code.

CSS Code:

```
1. .loginform{
2.   padding:10px;
3.   border:1px solid pink;
4.   border-radius:10px;
5.   float:right;
6.   margin-top:10px;
7. }
8. .formheading{
9.   background-color:red;
10.  color:white;
11.  padding:4px;
12.  text-align:center;
13. }
14. .sub{
15. background-color:blue;
16. padding: 7px 40px 7px 40px;
17. color:white;
18. font-weight:bold;
19. margin-left:70px;
20. border-radius:5px;
21. }
```

HTML Code:

```
1. <div class="loginform">
2. <h3 class="formheading">Please Login</h3>
3. <form action="LoginServlet" method="post">
4. <table>
```

5. `<tr><td>Email:</td><td><input type="email" name="email"/></td></tr>`
6. `<tr><td>Password:</td><td><input type="password" name="password"/></td></tr>`
7. `<tr><td colspan="2" style="text-align:center"><input class="sub" type="submit" value="login"/></td></tr>`
8. `</table>`
9. `</form>`
10. `</div>`

HTML iframes

HTML Iframe is used to display a nested webpage (a webpage within a webpage). The HTML `<iframe>` tag defines an inline frame, hence it is also called as an Inline frame.

An HTML iframe embeds another document within the current HTML document in the rectangular region.

The webpage content and iframe contents can interact with each other using JavaScript.

Iframe Syntax

An HTML iframe is defined with the `<iframe>` tag:

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

Here, "src" attribute specifies the web address (URL) of the inline frame page.

Set Width and Height of iframe

You can set the width and height of iframe by using "width" and "height" attributes. By default, the attributes values are specified in pixels but you can also set them in percent. i.e. 50%, 60% etc.

Example: (Pixels)

1. `<!DOCTYPE html>`
2. `<html>`
3. `<body>`
4. `<h2>HTML Iframes example</h2>`
5. `<p>Use the height and width attributes to specify the size of the iframe:</p>`
6. `<iframe src="https://www.javatpoint.com/" height="300" width="400"></iframe>`
7. `</body>`
8. `</html>`

Example: (Percentage)

1. `<!DOCTYPE html>`

2. `<html>`
3. `<body>`
4. `<h2>HTML Iframes</h2>`
5. `<p>You can use the height and width attributes to specify the size of the iframe:</p>`
6. `<iframe src="https://www.javatpoint.com/" height="50%" width="70%"></iframe>`
7. `</body>`
8. `</html>`

You can also use CSS to set the height and width of the iframe.

Example:

1. `<!DOCTYPE html>`
2. `<html>`
3. `<body>`
4. `<h2>HTML Iframes</h2>`
5. `<p>Use the CSS height and width properties to specify the size of the iframe:</p>`
6. `<iframe src="https://www.javatpoint.com/" style="height:300px;width:400px"></iframe>`
7. `</body>`
8. `</html>`

Remove the border of iframe

By default, an iframe contains a border around it. You can remove the border by using `<style>` attribute and CSS border property.

Example:

1. `<!DOCTYPE html>`
2. `<html>`
3. `<body>`
4. `<h2>Remove the Iframe Border</h2>`
5. `<p>This iframe example doesn't have any border</p>`
6. `<iframe src="https://www.javatpoint.com/" style="border:none;"></iframe>`
7. `</body>`
8. `</html>`

You can also change the size, color, style of the iframe's border.

Example:

1. `<!DOCTYPE html>`
2. `<html>`

3. <body>
4. <h2>Custom Iframe Border</h2>
5. <iframe src="https://www.javatpoint.com/" style="border:2px solid tomato;"></iframe>
6. </body>
7. </html>

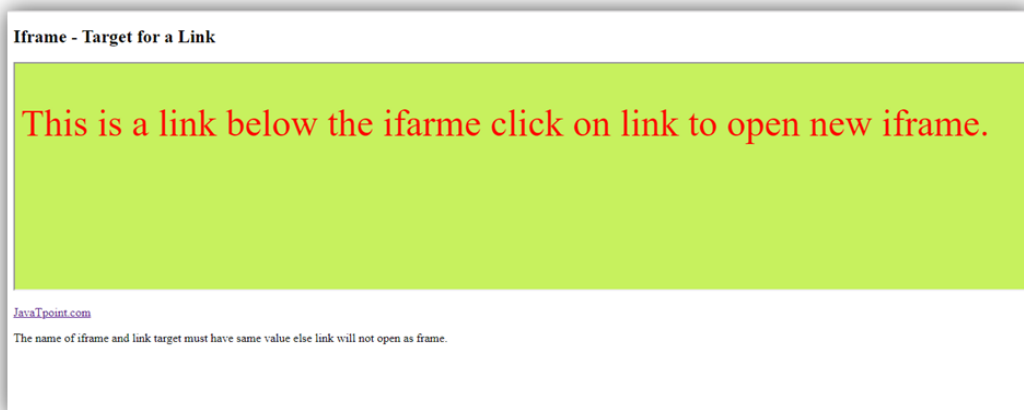
Iframe Target for a link

You can set a target frame for a link by using iframe. Your specified target attribute of the link must refer to the name attribute of the iframe.

Example:

1. <!DOCTYPE html>
2. <html>
3. <body>
- 4.
5. <h2>Iframe - Target **for** a Link</h2>
6. <iframe height="300px" width="100%" src="new.html" name="iframe_a"></iframe>
7. <p>JavaTpoint.com</p>
8. <p>The name of iframe and link target must have same value **else** link will not open as a frame.
</p>
- 9.
10. </body>
11. </html>

Output



new.html output code:

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

```
<head>
  <style>

    p{ font-size: 50px;
      color: red;}
</style>
</head>
<body style="background-color: #c7f15e;">
  <p>This is a link below the ifarme click on link to open new iframe. </p>
</body>
</html>
```

Embed YouTube video using iframe

You can also add a YouTube video on your webpage using the `<iframe>` tag. The attached video will be played at your webpage and you can also set height, width, autoplay, and many more properties for the video.

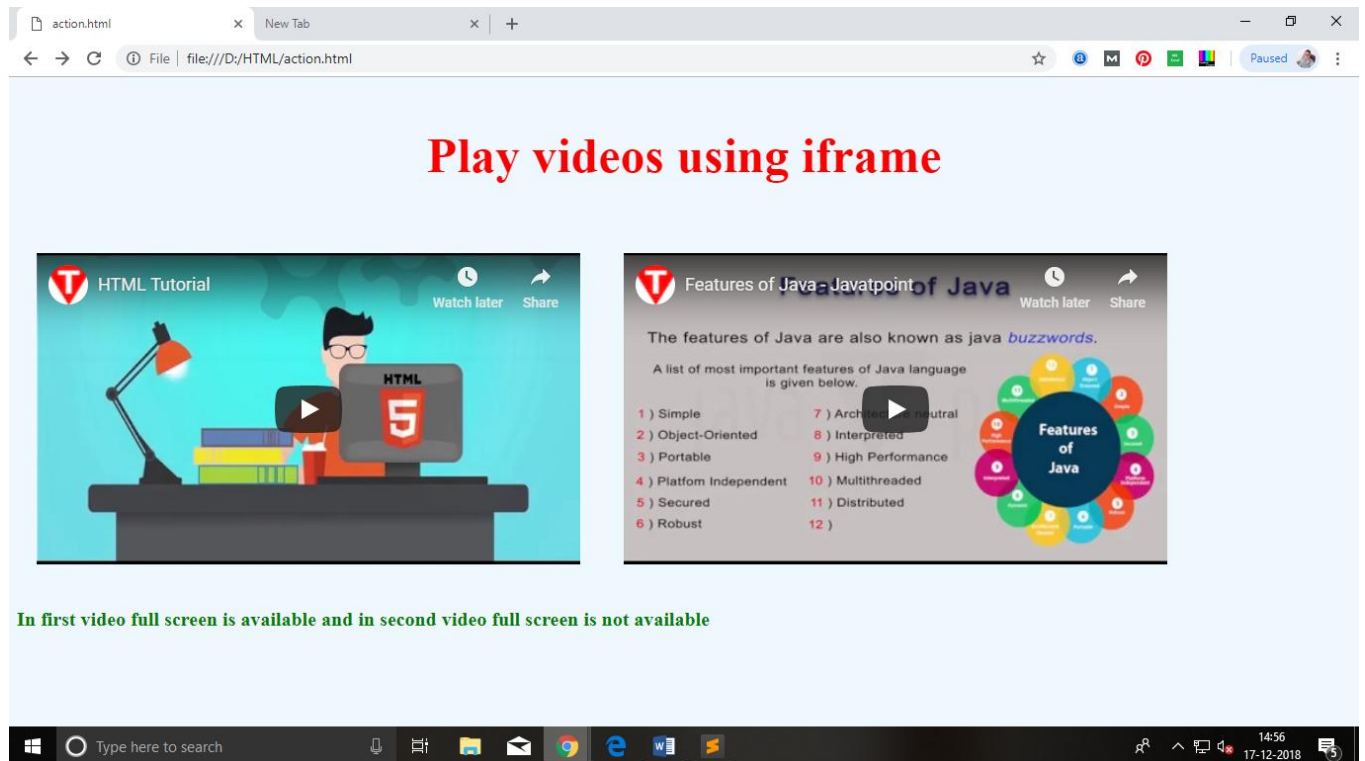
Following are some steps to add YouTube video on your webpage:

- Goto YouTube video which you want to embed.
- Click on **SHARE** ➡ under the video.
- Click on Embed <> option.
- Copy HTML code.
- Paste the code in your HTML file
- Change height, width, and other properties (as per requirement).

Example:

1. `<iframe width="550" height="315" src="https://www.youtube.com/embed/JHq3pL4cdy4" frameborder="0" allow="accelerometer; autoplay; encrypted-media; gyroscope; picture-in-picture" allowfullscreen style="padding:20px;"></iframe>`
2. `<iframe width="550" height="315" src="https://www.youtube.com/embed/O5hShUO6wxs" frameborder="0" allow="accelerometer; autoplay; encrypted-media; gyroscope; picture-in-picture" style="padding:20px;"></iframe>`

Output:



Attributes of <iframe>

Attribute name	Value	Description
allowfullscreen		If true then that frame can be opened in full screen.
height	Pixels	It defines the height of the embedded iframe, and the default height is 150 px.
name	text	It gives the name to the iframe. The name attribute is important if you want to create a link in one frame.
frameborder	1 or 0	It defines whether iframe should have a border or not. (Not supported in HTML5).
Width	Pixels	It defines the width of embedded frame, and default width is 300 px.
src	URL	The src attribute is used to give the path name or file name which content to be loaded into iframe.
sandbox		This attribute is used to apply extra restrictions for the

		content of the frame
	allow-forms	It allows submission of the form if this keyword is not used then form submission is blocked.
	allow-popups	It will enable popups, and if not applied then no popup will open.
	allow-scripts	It will enable the script to run.
	allow-same-origin	If this keyword is used then the embedded resource will be treated as downloaded from the same source.
srcdoc		The srcdoc attribute is used to show the HTML content in the inline iframe. It overrides the src attribute (if a browser supports).
scrolling		It indicates that browser should provide a scroll bar for the iframe or not. (Not supported in HTML5)
	auto	Scrollbar only shows if the content of iframe is larger than its dimensions.
	yes	Always shows scroll bar for the iframe.
	no	Never shows scrollbar for the iframe.