HTML 5.0

❖ What is HTML5 ?

- HTML5 is the latest version of Hypertext Markup Language, the code that describes web pages. HTML5 is a standard for structuring and presenting content on the World Wide Web so it is used to build incredibly complicated applications that run in your browser.
- It's actually three kinds of code: HTML, which provides the structure; Cascading Style Sheets (CSS), which take care of presentation; and JavaScript, which makes things happen.
- 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.

Browser Support

- The latest versions of Apple Safari, Google Chrome, Mozilla Firefox, and Opera all support many HTML5 features and Internet Explorer 9.0 will also have support for some HTML5 functionality.
- The mobile web browsers that come pre-installed on iPhones, iPads, and Android phones all have excellent support for HTML5.
- Firefox generally supports the widest selection of HTML5 features, with Chrome and Safari following shortly afterwards, but as we said HTML5 is an evolving standard and the latest versions of each browser more than cover the basics.
- HTML5shiv is a javscript workaround to provide support for the new HTML5 elements in IE Browsers older than IE 9.

Syntax

```
<head>
<!--[if lt IE 9]>
  <script src="/js/html5shiv.js"></script>
<![endif]-->
</head>
```

Note

- HTML5shiv is found within the <head> tag.
- HTML5shiv is a javascript file that is referenced in a <script> tag.
- You should use HTML5shiv when you are using the new HTML5 elements such as:<article>, <section>, <aside>, <nav>, <footer>.

- Download the latest version of HTML5shiv from github or reference the Open Source Software CDN version at https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js
- You will require the HTML5shiv to provide compatibility for IE Browsers older than IE 9.

Features of HTML5

- HTML5 has New content related elements, like, <header>, <footer>, <article>, <section>, <nav>, etc.
- It offers new form controls, like date, calendar, time, URL, email, and search,etc.
- The <canvas> element for drawing 2D diagram
- Support for CSS3, the newer and version of CSS.
- The <video> and <audio> elements for media playback
- Figure element can be combined with elements to easily associate a caption with the other image elements.
- HTML is capable of handling incorrect syntax.

Difference between HTML and HTML5

HTML Doctype declaration is too long and complicated. <!DOCTYPE HTML PUBLIC “-//W3C//DTD HTML4.01//EN” “http://www.w3.org/TR/html4/strict.dtd”>	DOCTYPE declaration in HTML5 is simple and easy. <!DOCTYPE html>
HTML Character encoding is longer. <Meta http-equiv=”content-type” content=”text/html; charset=UTF-8”>	HTML5 Character encoding declaration is simple. <meta charset=”UTF-8”>
It didn’t support the audio and video tag without the support of flash player plugin.	It supports audio and video controls with the use of <audio> and <video> tags.
It is possible to draw a vector with the help of other technologies like Silverlight, Flash, VML, etc.	Vector graphics are a part of HTML5, e.g., canvas, SVG.
You can use HTML with all old browsers.	You can use HTML5 with all new browsers like Firefox, Mozilla, Chrome, Safari, etc.
Older version of HTML are less mobile-friendly.	HTML5 language is more mobile-friendly.
HTML does not allow drag and drop effects	HTML5 allows drag and drop effects.
It can not handle inaccurate syntax. There is no process to handlestructurally incorrect HTML codes.	It is capable of handling inaccurate syntax. HTML5 supports persistent error handling via the improvised error handling process.

It is almost impossible to get true GeoLocation of user with the help of browser.	One can track the GeoLocation of a user easily by using JS GeoLocation API.
Elements like nav, header were not present.	New element for web structure like nav, header, footer etc.
Not possible to draw shapes like circle, rectangle, triangle etc.	HTML5 allows to draw shapes like circle, rectangle, triangle etc.
It allows you to run JavaScript in a browser.	It enables you to run JavaScript code in the background.

Disadvantages of HTML5

- It requires modern browsers to access it.
- There are issues related to media licensing.
- Multiple device responsiveness can be a headache.
- The HTML5 language is still a work in progress.
- Gaming struggles with JavaScript under HTML5.
- There are no good IDEs that are available in HTML5.

Difference in Tags

Changes in existing HTML4 Tags

- **<!DOCTYPE html>**: In HTML 4.01, there are three different <!DOCTYPE> declarations but In HTML 5 there is only one <! DOCTYPE html>.
- **<a>**: In HTML 4.01, the <a> tag could be either a hyperlink or an anchor. In HTML5, the <a> tag is always a hyperlink, but if it has no href attribute, it is only a placeholder for a hyperlink.
- **<acronym>**: The <acronym> tag is not supported in HTML5. Use the <abbr> tag instead. The <acronym> tag was used to define acronyms in HTML 4.01.
- **<strong>:** In HTML 4.01, the <strong> tag defines strong emphasized text, but in HTML5 it defines important text.
- **<hr>**: In HTML 4.01, the <hr> tag represented a horizontal rule. In HTML5, the <hr> tag defines a thematic break. The <hr> element is used to separate content (or define a change) in an HTML page.
- **<meta>** : The scheme attribute is not supported in HTML5.
  HTML5 has a new attribute, charset, which makes it easier to define charset:
  HTML 4.01: <Meta http-equiv=”content-type” content=”text/html; charset=UTF-8”>
  HTML5 : <meta charset=”UTF-8”>
- **<script>** : The “type” attribute is required in HTML 4, but optional in HTML5.
- **<link>** : The “type” attribute is required in HTML 4, but optional in HTML5.

- **<style>** : The "type" attribute is required in HTML 4, but optional in HTML5.
- **<small>** : In HTML 4.01 the small element is displayed as smaller text. In HTML5 the small element defines small text and other side comments, and is displayed as smaller text.
- **<table>**: Only the "border" attribute is supported in HTML5, and it only allows the values " " or "1".

HTML5 offers New Semantic/Structural Elements, New Form Elements, New Input Types, New Input Attributes.

HTML5 offers New Elements

- **Semantic/Structural Elements**
- **Media Elements**
- **Form Elements**
- **Graphic Elements**

1. Semantic/Structural Elements:

- **What are Semantic Elements?**

A semantic element clearly describes its meaning to both the browser and the developer.

Examples of **non-semantic** elements: <div> and <span> - Tells nothing about its content.

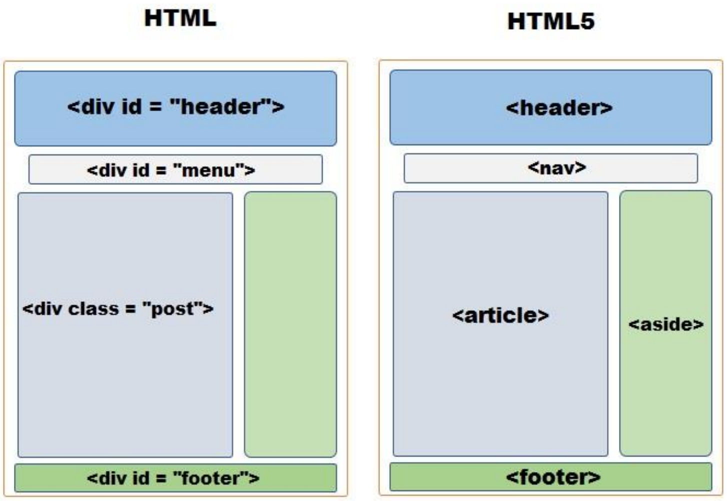
Examples of **semantic** elements: <form>, <table>, and <article> - Clearly defines its content.

- **HTML5 offers new elements for better structure**

Tag	Description
<main>	Defines the main content of a document. There must not be more than one <main> element in a document. The <main> element must NOT be a descendant of an <article>, <aside>, <footer>, <header>, or <nav> element.
<article>	Defines an article
<aside>	Defines content aside from the page content
<details>	Defines additional details that the user can view or hide
<summary>	Defines a visible heading for a <details> element
<figure>	Specifies self-contained content, like illustrations, diagrams,

	photos, code listings, etc.
<figcaption>	Defines a caption for a <figure> element
<footer>	Defines a footer for a document or section
<header>	Defines a header for a document or section
<hgroup>	Groups a set of <h1> to <h6> elements when a heading has multiple levels
<mark>	Defines marked/highlighted text
<meter>	Defines a scalar measurement within a known range (a gauge)
<nav>	Defines navigation links
<progress>	Represents the progress of a task
<section>	Defines a section in a document
<time>	Defines a date/time

- Difference between HTML4 and HTML5 Page Structure.



2. New Media Elements:

- **HTML5 offers new elements for media content:**

Tag	Description
<audio>	Defines sound content

<video>	Defines a video or movie
<source>	Defines multiple media resources for <video> and <audio>
<embed>	Defines a container for an external application or interactive content (a plug-in)

3. New Form Elements

- **HTML5 offers new form elements, for more functionality.**

Tag	Description
<datalist>	Specifies a list of pre-defined options for input controls
<output>	Defines the result of a calculation

4. Graphic elements

- **HTML5 offers new Graphic elements**

Tag	Description
<canvas>	Draw graphics, on the fly, via scripting (usually JavaScript)
<svg>	Draw scalable vector graphics

HTML5 offers New Input Types

New Input Types
Color date time datetime-local month week email number range search tel url image

- **HTML5 offers New Input Attributes**

New Input Attributes
• Autocomplete • autofocus • list • min and max • multiple • pattern • placeholder(regexp) • required

HTML5 - New Attribute Syntax

- HTML5 allows four different syntaxes for attributes.
- This example demonstrates the different syntaxes used in an <input> tag:

Type	Example
Empty	<input type="text" value="John" disabled>
Unquoted	<input type="text" value=John>
Double-quoted	<input type="text" value="John Doe">
Single-quoted	<input type="text" value='John Doe'>

- In HTML5, all four syntaxes may be used, depending on what is needed for the attribute.

Removed Elements

- **The following HTML elements are removed from HTML5:** <acronym>, <applet>, <basefont>, <big>, <center>, <dir>, <font>, <frame>, <frameset>, <noframes>, <strike>, <tt>

Conclusion

- HTML5 is being proved as one of the powerful tool for webpage design. With HTML5, streaming of audio and video without need of third party plug-in such as flash is possible. Player controls can be created that are fully programmable with JavaScript. In HTML5 we have new structural elements instead of traditional div tags to create page template, the final result will be a cleaner and more organized code. HTML5 introduces new elements and features that allow developers to improve interoperability, handling elements in a precise way saving time and costs.

❖ Examples with Output and Description

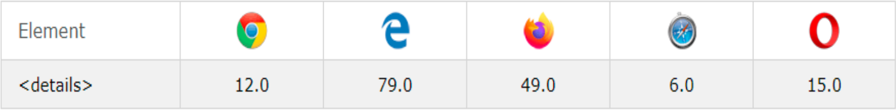
- Here is the list of all HTML5 Elements and Input Types and Attributes with output:

Structural/Semantic Tags

➢ Details and Summary Tag:

- HTML <details> tag is used to specify the additional details on the web page that the user can view or hide on demand.
- The <details> tag wraps all the content which you want to show or hide (like expand and collapse)
- Any sort of element can be placed inside the details element.
- **In short we can say that, <details> tag is often used to create an interactive widget that the user can open and close.** By default, the widget is closed. When open, it expands, and displays the content within.
- Details tag is used together with a relevant tag known as <summary>. Technically, there is no need of summary tag, but if you ignore this then the browser will use some default text.
- the <summary> tag contains the summary and the title of the section.
- Details and Summary Tag is a **Block** Element

Browser Support



Element					
<details>	12.0	79.0	49.0	6.0	15.0

Attributes

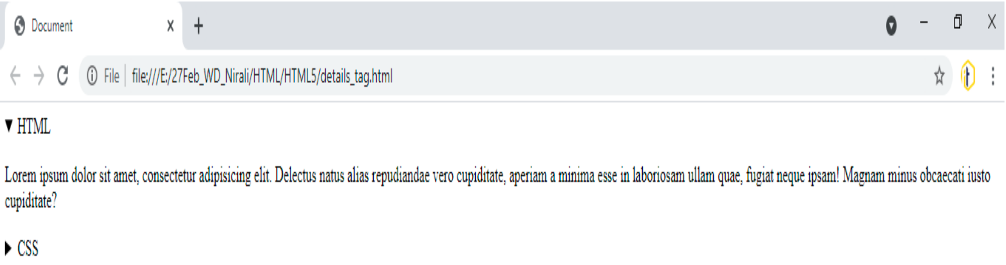
The HTML details and summary tag supports the Global attributes in HTML. Details tag provides one specific attribute open.

Attribute	Value	Description
Open	open	It specifies that the details will be displayed (open) to the user.

Example

```
<body>
        <details open>
                <summary>HTML</summary>
                <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit.
Delectus natus alias repudiandae </p>
        </details>
        <details>
               <summary>CSS</summary>
               <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit.
Delectus natus alias repudiandae vero cupiditate, </p>
        </details>
</body>
```

Output:



file:///E:/27Feb_WD_Nirali/HTML/HTML5/details_tag.html

▼ HTML

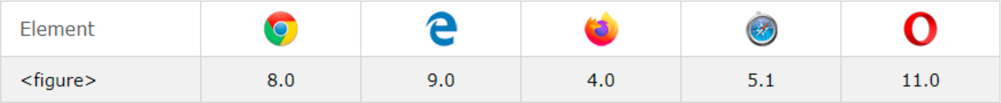
Lorem ipsum dolor sit amet, consectetur adipisicing elit. Delectus natus alias repudiandae vero cupiditate, aperiam a minima esse in laboriosam ullam quae, fugiat neque ipsam! Magnam minus obcaecati iusto cupiditate?

▶ CSS

➢ Figure and Figcaption Tag:

- **HTML <figure> tag** is used to mark up a photo in the document on a web page.
- As we know image tag is already available in HTML to display the pictures on web pages. But HTML 5 <figure> tag is used to handle the group of diagrams, photos, code listing etc. with some embedded content. You can also add a caption for the photo with the help of <figcaption> tag.
- Figure and Figcaption Tag is a **Block** Element.

Browser Support



Element					
<figure>	8.0	9.0	4.0	5.1	11.0

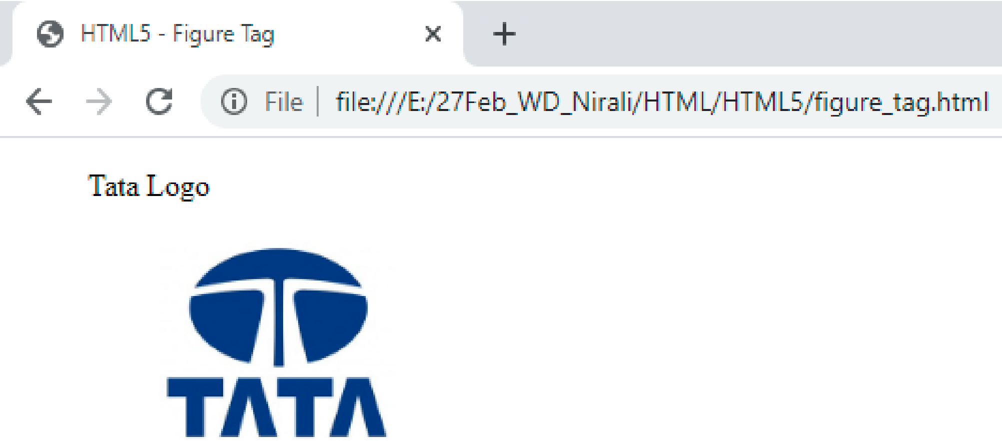
Attributes

The HTML figure tag and figcaption tag supports the Global attributes in HTML. No specific attributes for figure tag and figcaption tag.

Example

```
<body>
  <figure>
     <figcaption>Tata Logo</figcaption>
     <img src="../../images/tata-logo.jpg" width="200px">
  </figure>
</body>
```

Output:



HTML5 - Figure Tag

File | file:///E:/27Feb_WD_Nirali/HTML/HTML5/figure_tag.html

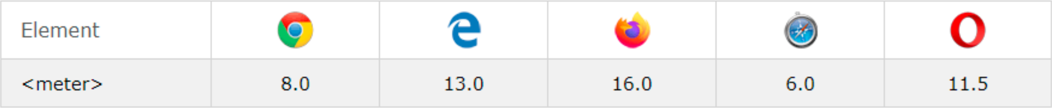
Tata Logo

➢ Meter Tag:

- **HTML <meter> tag** is used to measure data within a given range. It defines a scalar measurement with range.
- It should be used to display disk usage, voting population, profile completion etc.
- Meter tag is an inline element.

The meter tag should not be used to indicate progress (as in a progress bar). For progress bars, use the progress tag.

Browser Support



Element					
<meter>	8.0	13.0	16.0	6.0	11.5

Attributes

The HTML meter supports the Global attributes in HTML. Specific attributes for Meter tag is defined as below:

Attribute	Value	Description
max	number	Specifies the maximum value of the range
min	number	Specifies the minimum value of the range
value	number	Specifies current value of the gauge – *Required*.

Example

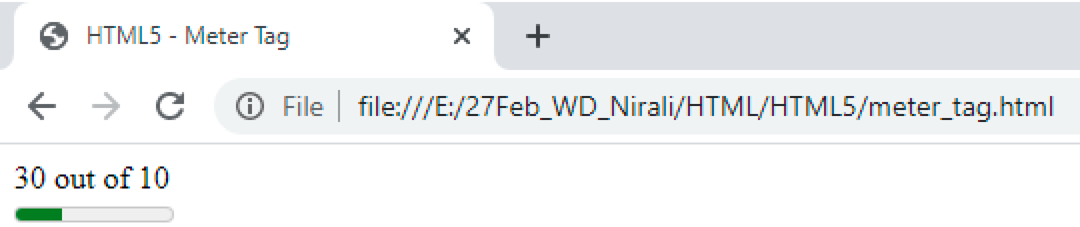
```
<body>

        <label>30 out of 10</label>

        <meter min="1" max="100" value="30"></meter>

</body>
```

Output



30 out of 10

- **Progress Tag:**
  - The **HTML <progress> tag** specifies a completion **progress** of a task. It is displayed as a **progress bar**. It provides an easy way for web developers to create progress bar on the website.
  - It is mostly used to show the progress of a file uploading on the web page. It is also define that how much work is done and how much is left to download a things.
  - The value of progressbar can be manipulated by JavaScript.
  - Progress tag is an inline element.

Progressbar is not used to display disk usage, voting population, profile completion, etc. For this purpose use meter tag.

Browser Support

Element					
<progress>	8.0	10.0	16.0	6.0	11.0

Attributes

The HTML Progress tag supports the Global attributes in HTML. Specific attributes for progress tag is defined as below.

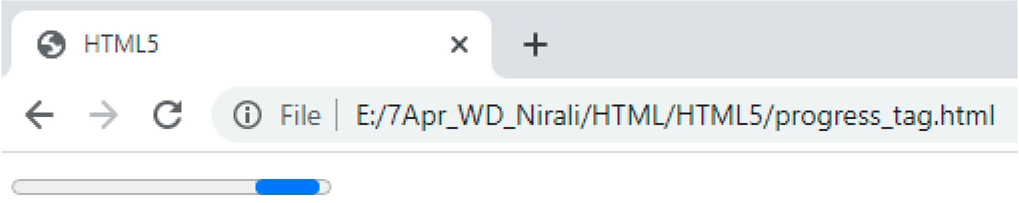
Attribute	Value	Description
max	number	It defines that how much work the task requires in total.
value	number	It defines that how much work the task has been completed.

Example

- HTML progress example without attribute.

```
<body>
    <progress></progress>
</body>
```

Output

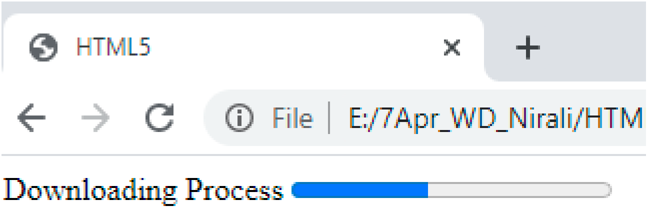


E:/7Apr_WD_Nirali/HTML/HTML5/progress_tag.html

- HTML progress example value and max attributes.

```
<body> <label>Downloading Process</label>
   <progress  value="43" max="100"></progress>  </body>
```

Output:



E:/7Apr_WD_Nirali/HTM

Downloading Process

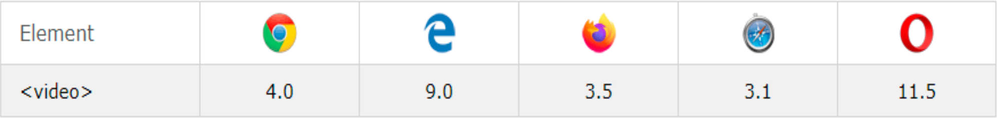
New Media Elements

Video Tag:

- HTML 5 supports <video> tag also. The HTML video tag is used for streaming video files such as a movie clip, song clip or other video streams on the web page.
- Before HTML 5 came into existence, videos could only be played in a browser using a plugin like flash. But after the release of HTML 5, adding a video to a webpage is as easy as adding an image. The HTML5 "video" element specifies a standard way to embed a video on a web page.
- Currently, there are three supported video formats in HTML: MP4, WebM, and OGG.
- Let's see the table that defines which web browser supports video file format.

Browser	mp4	webM	ogg
Internet Explorer	yes	yes	yes
Google Chrome	yes	yes	yes
Mozilla Firefox	yes	yes	yes
Opera	yes	yes	yes
Apple Safari	yes	yes	no

Browser Support



Element		e			O
<video>	4.0	9.0	3.5	3.1	11.5

Attributes

The HTML Video tag supports the Global attributes in HTML. Specific attributes for video tag is defined as below

Attribute	Value	Description
autoplay	autoplay	Specifies that the video will play automatically.

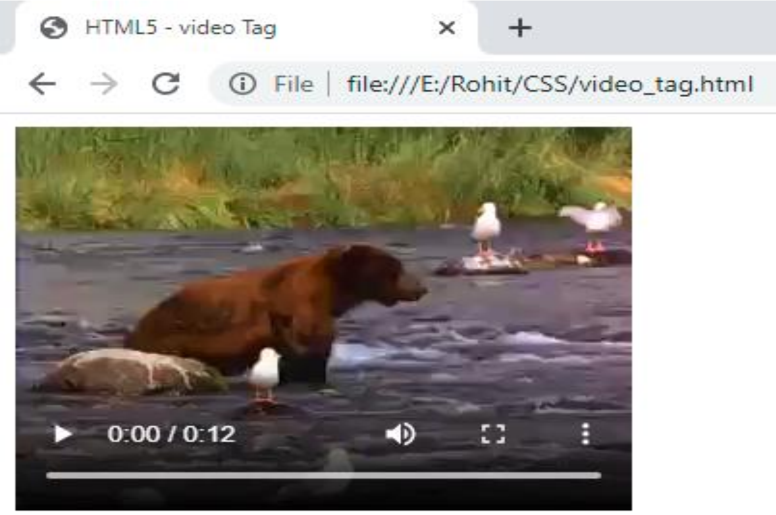
controls	controls	Specifies that the video controls gets displayed.
height	pixels	Specifies the height
loop	loop	Specifies that the video will start again every time after finish.
muted	muted	Specifies that the audio should be muted
poster	URL	Specifies the image to be shown while the video is downloading.
src	URL	Specifies the URL
width	pixels	Specifies the width

Example

- HTML Video tag example with width, height and controls attributes

```
<body>
<video controls height="300" width="500">
                    <source src="../../images/movie.mp4" type="video/mp4">
          </video>
</body>
```

Output



- HTML Video tag example with autoplay and muted attributes

```
<body>
<video controls height="300" width="500" autoplay muted>
                    <source src="../../images/movie.mp4" type="video/mp4">
          </video>
</body>
```

Chromium browsers do not allow autoplay in most cases. However, muted autoplay is always allowed.

- HTML Video tag example with poster attributes

```
<body>
   <video controls poster="../images/tata-motors.jpg">
      <source src="../images/movie.mp4" type="video/mp4">
   </video>
</body>
```

- HTML Video tag example with autoplay and muted attributes

```
<body>
<video controls height="300" width="500" loop>
                    <source src="../../images/movie.mp4" type="video/mp4">
          </video>
</body>
```

MIME Types for HTML Video Format

The available MIME type HTML video tag is given below.

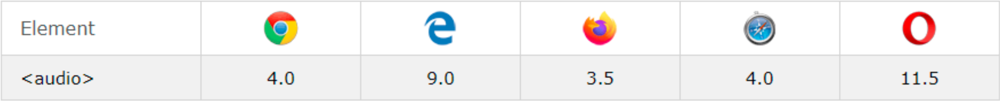
Video Format	MIME Type
mp4	video/mp4
ogg	video/ogg
webM	video/webM

➢ Audio Tag:

- **HTML audio tag** is used to define sounds such as music and other audio clips.
- HTML5 supports <video> and <audio> controls. The Flash, Silverlight and similar technologies are used to play the multimedia items. After the release of HTML 5 no need to install these plugins, already exists in browsers.
- But after the release of HTML 5, adding an audio to a webpage is as easy as adding an image. The HTML5 "audio" element specifies a standard way to embed an audio on a web page.
- Currently, there are three supported video formats in HTML: MP3, WAV, and OGG.
- This table defines that which web browser supports which audio file format.

Browser	mp3	wav	ogg
Internet Explorer	yes	no	no
Google Chrome	yes	yes	yes
Mozilla Firefox	yes*	yes	yes
Opera	no	yes	yes
Apple Safari	yes	yes	no

Browser Support



Element					
<audio>	4.0	9.0	3.5	4.0	11.5

Attributes

The HTML Audio tag supports the Global attributes in HTML. Specific attributes for Audio tag is defined as below:

Attribute	Value	Description
autoplay	autoplay	Specifies that the audio will play automatically.
controls	controls	Specifies that the audio controls gets displayed.
height	pixels	Specifies the height
loop	loop	Specifies that the audio will start again every

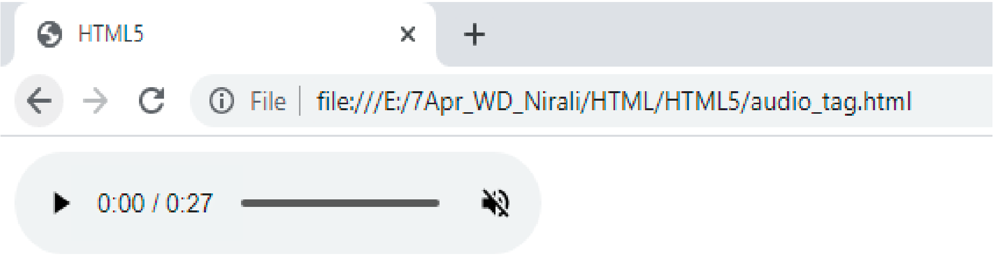
		time after finish.
muted	muted	Specifies that the audio should be muted
src	URL	Specifies the URL
width	pixels	Specifies the width

Example

- HTML Video tag example with controls and loop attributes

```
<body>
  <audio controls autoplay muted height="500" loop>
  <source src="../../images/file_example_MP3_700KB.mp3" type="audio/mp3">
  </audio>
</body>
```

Output



MIME Types for HTML Audio Format

The available MIME type HTML Audio tag is given below.

Audio Format	MIME Type
mp3	audio/mp3
ogg	audio/ogg
wav	audio/wav

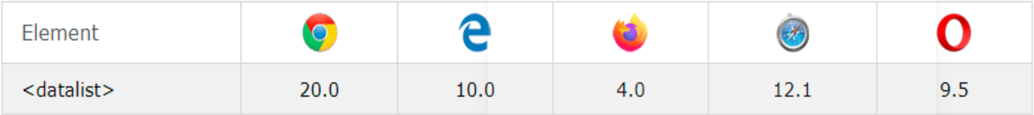
New Form Elements

Datalist Tag

- **The HTML <datalist> tag** is is used to provide an auto complete feature on form element. It provides a list of predefined options to the users to select data.

- The datalist tag is introduced in HTML5.
- The <datalist> tag should be used with an <input< element that contains a "list" attribute. The value of "list" attribute is linked with the datalist id.
- HTML5 Datalist tag is a **Block** element

Browser Support



Element					
<datalist>	20.0	10.0	4.0	12.1	9.5

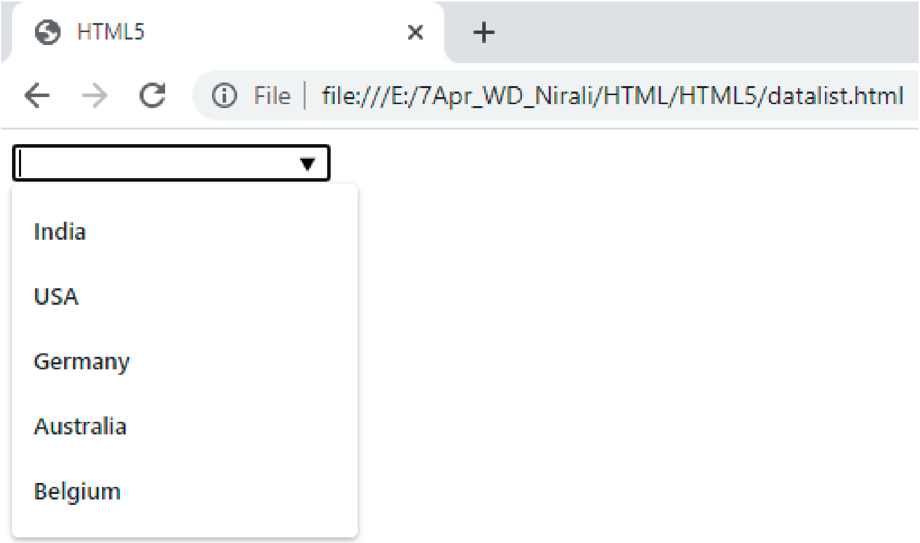
Attributes

The HTML datalist tag supports Global attributes.No specific Attributes for Datalist tag.

Example

```
<body>
    <input type="text" list="cntry">
    <datalist id="cntry">
        <option>India</option>
        <option>USA</option>
        <option>Germany</option>
        <option>Australia</option>
        <option>Belgium</option>
    </datalist>
</body>
```

Output



Output Tag

- **HTML <output> tag** is used to display the result of some calculation (performed by JavaScript) or the outcome of a user action (such as Input data into a form element).
- HTML5 Output tag is an **Inline** Element

Browser Support

Element					
<output>	10.0	13.0	4.0	5.1	11.0

Attributes

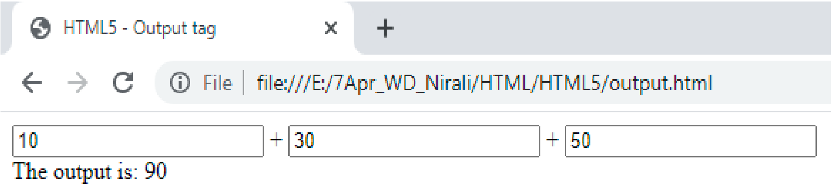
The HTML output tag supports the Global attributes in HTML. Specific attributes of output tag is defined as below:

Attribute	Value	Description
for	for	List of IDs of other elements, i.e it indicates the elements who have contributed input value to the calculation.
form	form	Enables to place output elements anywhere within a document.
name	name	It is the name of the element.

Example

```
<body>
    <form oninput = "sumresult.value = parseInt(z1.value)+parseInt(z2.value)+p
arseInt(z3.value)">
        <input type = "number" name = "z1" value = "0" /> +
        <input type = "number" name = "z2" value = "20" /> +
        <input type = "number" name = "z3" value = "40" /><br />
        The output is: <output name = "sumresult"></output>
     </form>
</body>
```

Output



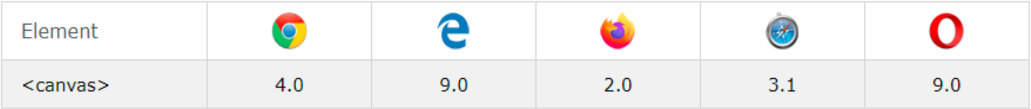
10 + 30 + 50
The output is: 90

New Graphic Elements

Canvas Tag

- The **HTML 5 <canvas> tag** is used to draw graphics using scripting language like JavaScript.
- The <canvas> element is only a container for graphics, you must need a scripting language to draw the graphics.
- Canvas has several methods for drawing paths, boxes, circles, text, and adding images.
- A canvas is a rectangle like area on an HTML page. It is specified with canvas element. By default, the <canvas> element has no border and no content, it is like a container.
- HTML5 Output tag is an **Inline** Element

Browser Suppoert



Element					
<canvas>	4.0	9.0	2.0	3.1	9.0

Attributes

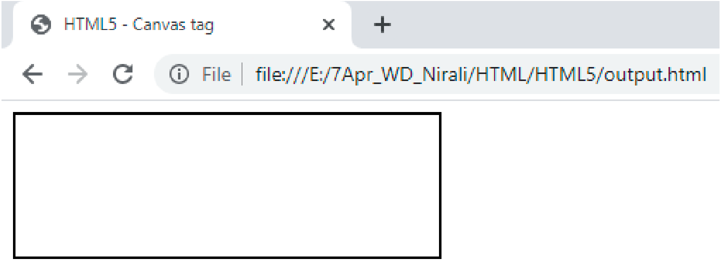
- The <canvas> element must have an id attribute so it can be referred to by JavaScript.
- The width and height attribute is necessary to define the size of the canvas.
- The <canvas> element also supports the Global Attributes in HTML.

Example

- HTML canvas tag example without javascript

```
<body>
   <canvas  width="300" height="100" style="border:2px solid;">
   </canvas>
</body>
```

Output

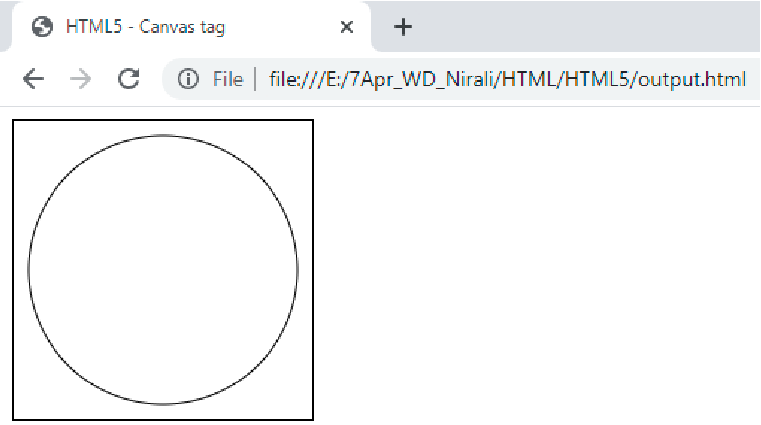


It is always necessary to specify the id attribute and the height & width attribute to define the size of the canvas. You can have multiple canvas elements on one HTML page.

- HTML canvas tag example with javascript

```
<body>
    <canvas id = "geeks" height = "200" width = "200" style = "border:1px solid
black">
    </canvas>
    <script>
        var c = document.getElementById("geeks");
        var cx = c.getContext("2d");
        cx.beginPath();
        cx.arc(100, 100, 90, 0, 2 * Math.PI);
        cx.stroke();
    </script>
</body>
```

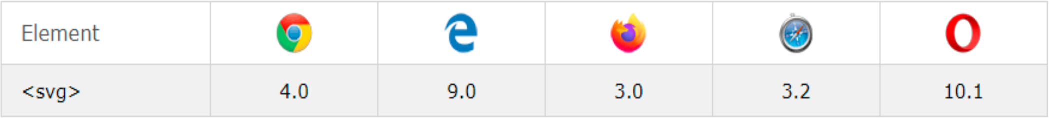
Output



➢ SVG Tag

- The `<svg>` element is used to embed SVG graphics in an HTML document.
- The HTML <svg> element is a container for SVG graphics.
- SVG has several methods for drawing paths, boxes, circles, text, and graphic images.
- The Scalable Vector Graphics (SVG) is an XML-based image format that is used to define two-dimensional vector based graphics for the web.
- HTML svg tag is an **inline** element.

Browser Support



Element					
<svg>	4.0	9.0	3.0	3.2	10.1

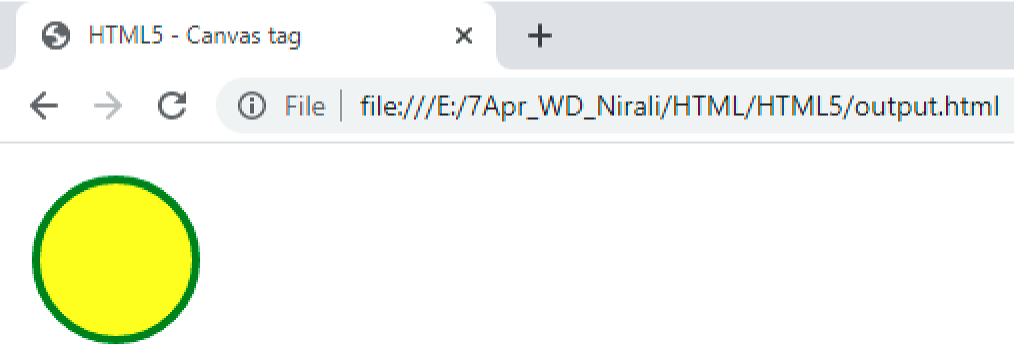
Attributes

The HTML datalist tag supports the Global attributes in HTML. No specific attributes of SVG Tag.

Example

```
<body>
        <svg width="100" height="100">
             <circle cx="50" cy="50" r="40" stroke="green" stroke-
width="4" fill="yellow" />
        </svg>
</body>
```

Output



- Above example is displaying how to create a circle using SVG method, more methods are available to create different-different shapes.

New Input Types and Input Attributes

Date input type and Max – Min Attributes

- The **date** is the value of the **type** attribute of an **<input>** element. It creates a calendar that allows a user to choose the date. The resulting value includes the **day, month**, and **year**.

Browser Support

Attribute					
type="date"	20.0	12.0	57.0	Not supported	10.1

Attributes

Along with the attributes common to all <input> elements, date inputs have the following specific attributes:

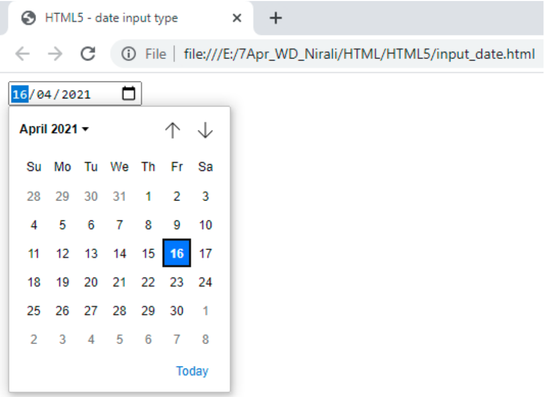
Attribute	Value	Description
max	dateformat (yyyy-mm-dd)	The latest acceptable date
min	dateformat (yyyy-mm-dd)	The earliest acceptable date.

Example

- HTML <input type=”date”> example without any attributes

```
<body>
    <input type="date">
</body>
```

Output



Su	Mo	Tu	We	Th	Fr	Sa
28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	1
2	3	4	5	6	7	8

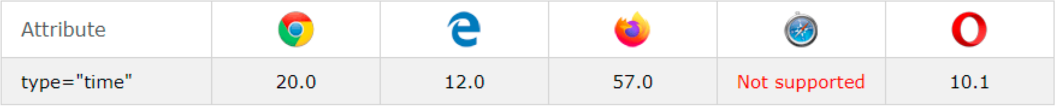
- HTML <input type=”date”> example with min, max and value attributes.

```
<body>
    <input type="date" min="1960-01-01" max="2021-12-31" value="2021-04-
16">
</body>
```

Time input type

- The **time** is the value of the **type** attribute of an **<input>** element
- The time input type can be used for selecting a time (hours and minutes).

Browser Support



Attribute					
type="time"	20.0	12.0	57.0	Not supported	10.1

Attributes

Along with the attributes common to all <input> elements, time input have the following specific attributes:

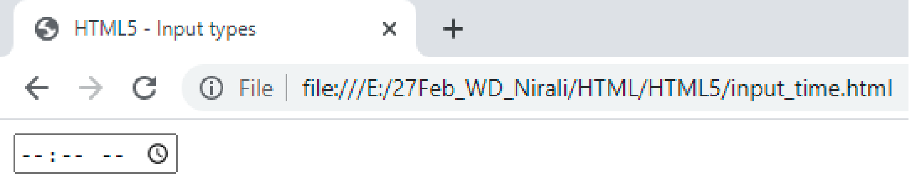
Attribute	Value	Description
max	hh:mm (24 – hour format)	The latest time to accept
min	hh:mm (24 – hour format)	The earliest time to accept as a valid input

Example

- HTML <input type=”time”> example without attributes.

```
<body>
    <input type="time">
</body>
```

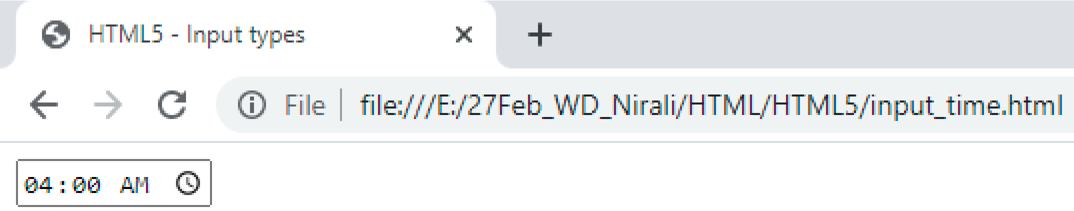
Output



- HTML <input type=”time”> example with value attribute.

```
<body>
    <input type="time" value="04:00">
</body>
```

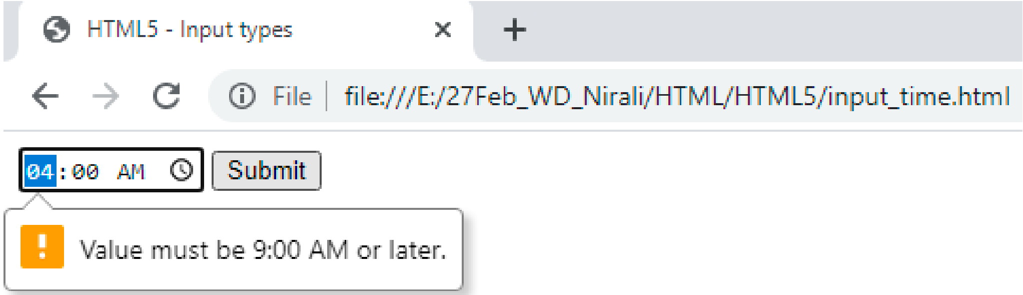
Output



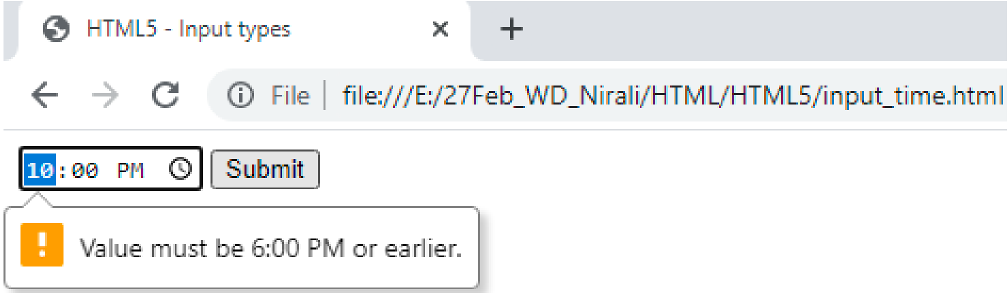
- HTML <input type=”time”> example with min, max attributes.

```
<body>
   <form action="#">
       <input type="time" name="appt" min="09:00" max="18:00" required
value="04:00">
       <input type="submit">
   </form>
</body>
```

Output



Value must be 9:00 AM or later.



Value must be 6:00 PM or earlier.

datetime-local input type

- The **datetime-local** is the value of the **type** attribute of an **<input>** element.
- <input> elements of type datetime-local create input controls that let the user easily enter both a date and a time, including the year, month, and day as well as the time in hours and minutes.

Browser Support

Attribute					
type="datetime-local"	20.0	13.0	Not supported	Not supported	10.1

Attributes

Along with the attributes common to all <input> elements, datetime-local input have the following specific attributes:

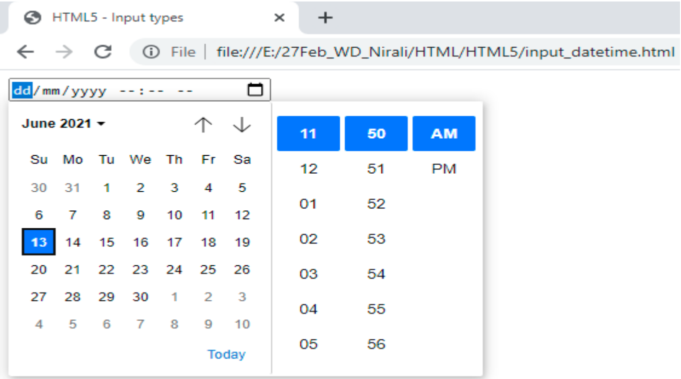
Attribute	Value	Description
max	yyyy-MM-ddThh:mm	The latest date and time to accept
min	yyyy-MM-ddThh:mm	The earliest date and time to accept

Example

- HTML <input type=”datetime-local”> example without attributes.

```
<body>
      <input type="datetime-local">
</body>
```

Output



HTML5 - Input types

File | file:///E:/27Feb_WD_Nirali/HTML/HTML5/input_datetime.html

dd/mm/yyyy --:-- --

June 2021

Su	Mo	Tu	We	Th	Fr	Sa
30	31	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	1	2	3
4	5	6	7	8	9	10

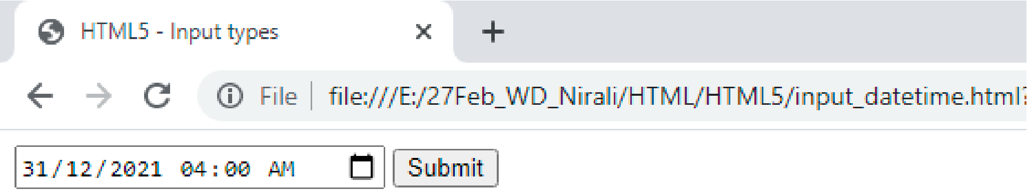
Today

11	50	AM
12	51	PM
01	52	
02	53	
03	54	
04	55	
05	56	

- HTML <input type=”datetime-local”> example with min, max and value attributes.

```
<body>
            <input type="datetime-local" min="2018-06-07T00:00" max="2021-
12-31T00:00" value="2021-12-31T04:00">
</body>
```

Output



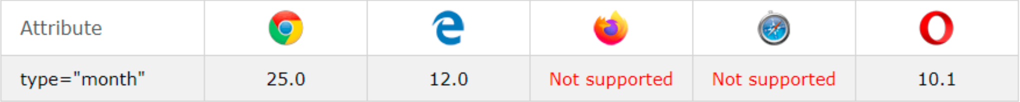
file:///E:/27Feb_WD_Nirali/HTML/HTML5/input_datetime.html

31/12/2021 04:00 AM

➢ month input type

- The **month** is the value of the **type** attribute of an **<input>** element.
- The <input> type "month" creates an input field which allows a user to easily enter month and year in the format of "YYYY-MM" where MM defines month value, and YYYY defines the year value.

Browser Support



Attribute					
type="month"	25.0	12.0	Not supported	Not supported	10.1

Attributes

Along with the attributes common to all <input> elements, month input have the following specific attributes:

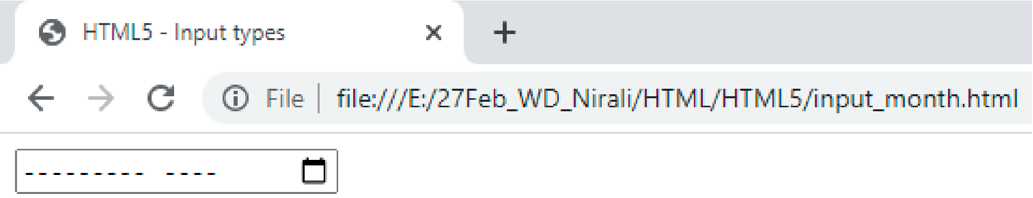
Attribute	Value	Description
max	YYYY-MM	The latest year and month to accept as a valid input
min	YYYY-MM	The earliest year and month to accept as a valid input

Example

- HTML <input type=”month”> example without attributes.

```
<body>
        <input type="month">
</body>
```

Output

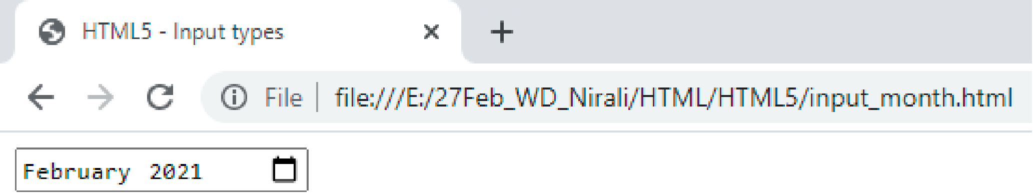


--------- ----

- HTML <input type="month"> example with min, max and value attributes.

```
<body>
  <input type="month" min="2018-01" max="2021-12" value="2021-02">
</body>
```

Output



February 2021

week input type

- The **week** is the value of the **type** attribute of an **<input>** element.
- The <input> type week creates an input field which allows a user to select a week and year form the drop-down calendar without time zone.

Browser Support

Attribute					
type="week"	25.0	12.0	Not supported	Not supported	10.1

Attributes

Along with the attributes common to all <input> elements, week input have the following specific attributes:

Attribute	Value	Description
max	yyyy-Www	The latest year and month to accept as a valid input

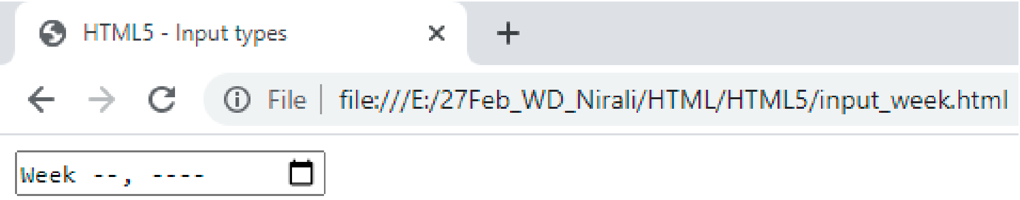
min	yyyy-Www	The earliest year and month to accept as a valid input

Example

- HTML <input type=”week”> example without attributes.

```
<body>
  <input type="week">
</body>
```

Output



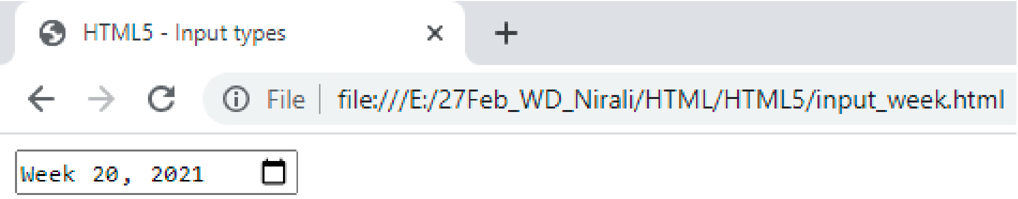
file:///E:/27Feb_WD_Nirali/HTML/HTML5/input_week.html

Week --, ----

- HTML <input type=”week”> example with min, max and value attributes.

```
<body>
  <input type="week" min="2018-W18" max="2018-W26" value="2021-
W20">
</body>
```

Output



file:///E:/27Feb_WD_Nirali/HTML/HTML5/input_week.html

Week 20, 2021

➢ email input type

- The **email** is the value of the **type** attribute of an **<input>** element.
- The <input type="email"> is used for input fields that should contain an e-mail address.
- The email input element has been specifically designed to handle and validate email addresses. Depending on the browser, it will alert the user that the entered information is not correct and the form will not submit until the user makes corrections.

- **Some smartphones recognize the email type, and add "@" and ".com" to the keyboard to match email input.**

Browser Support

Attribute					
type="email"	5.0	10.0	4.0	5.0	10.1

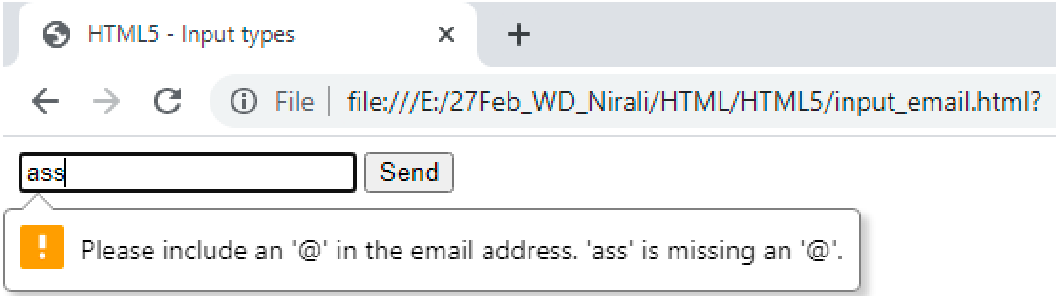
Attributes

HTML <input type=”email”> supports all common attributes of <input> element. No specific attributes for email input

Example

```
<form>
        <input type="email">
        <input type="submit" value="Send">
</form>
```

Output



Please include an '@' in the email address. 'ass' is missing an '@'.

URL input type

- The **url** is the value of the **type** attribute of an **<input>** element.
- The <input type="url"> defines a field for entering a URL.
- The input value is automatically validated before the form can be submitted.
- Some smartphones recognize the url type, and adds ".com" to the keyboard to match url input.

Browser Support

Attribute					
type="url"	Yes	10.0	Yes	Yes	10.1

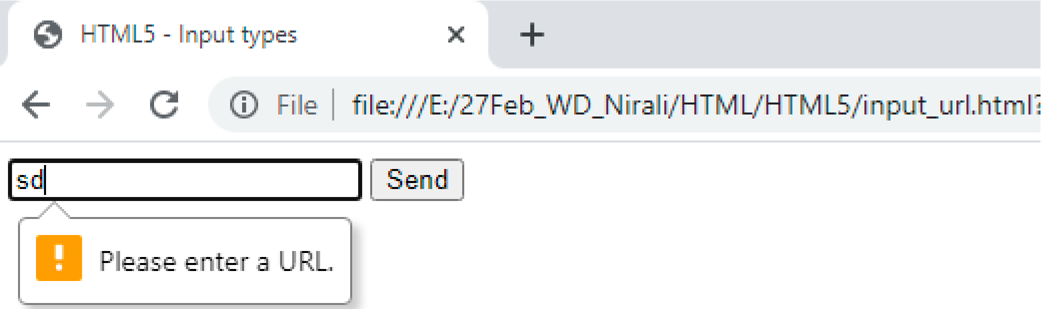
Attributes

HTML <input type=”url”> supports all common attributes of <input> element. No specific attributes for url input

Example

```
<form>
        <input type="url">
        <input type="submit" value="Submit">
</form>
```

Output

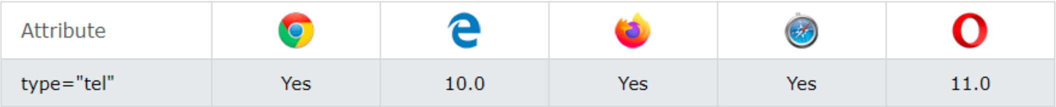


Please enter a URL.

➢ tel input type

- The **tel** is the value of the **type** attribute of an **<input>** element.
- The <input type="tel"> creates an input filed to enter the telephone number. The "tel" type does not have default validation such as email, because telephone number pattern can vary worldwide.
- We can use tel type, where we want numbers as an input from user (like pincode, otp, etc).
- Some smartphones recognize the tel type, and display numbers to the keyboard to match tel input.

Browser support



Attribute		e			O
type="tel"	Yes	10.0	Yes	Yes	11.0

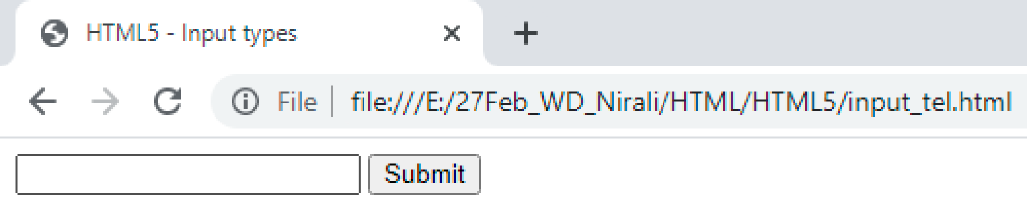
Attributes

HTML <input type=”url”> supports all common attributes of <input> element. No specific attributes for tel input.

Example

```
<form>
        <input type="tel">
        <input type="submit" value="Submit">
</form>
```

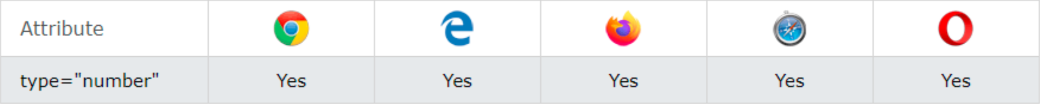
Output



➢ Number input type

- ○ The **number** is the value of the **type** attribute of an **<input>** element.
- ○ The <input type=”number> creates **numeric** input filed which allows a user to enter the numeric value. You can also restrict to enter a minimum and maximum value using min and max attribute.
- ○ In HTML5, you can have a number input field as a spinner which you have up and down arrow at the right of the textbox to increase or decrease the number value.

Browser Support



Attribute					
type="number"	Yes	Yes	Yes	Yes	Yes

Attributes

Along with the attributes common to all <input> elements, number input have the following specific attributes:

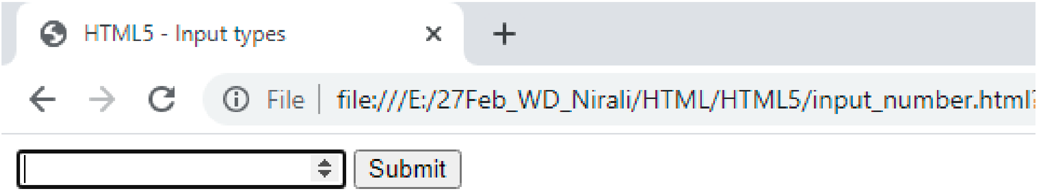
Attribute	Value	Description
max	number	The maximum value to accept for this input
min	number	The minimum value to accept for this input
step	number	A stepping interval to use when using up and down arrows to adjust the value, as well as for validation

Example

- HTML <input type=”number”> example without any attributes

```
<form>
      <input type="number">
      <input type="submit" value="Submit">
   </form>
```

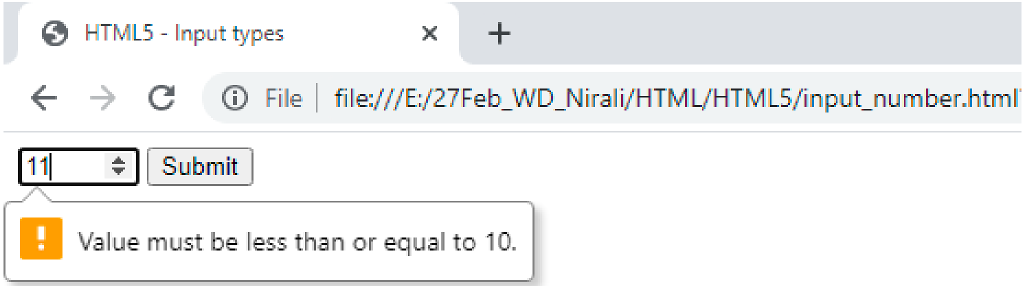
Output



- HTML <input type=”number”> example with min, max and value attributes

```
<form>
         <input type="number" min="1" max="10" value="1">
         <input type="submit" value="Submit">
 </form>
```

Output



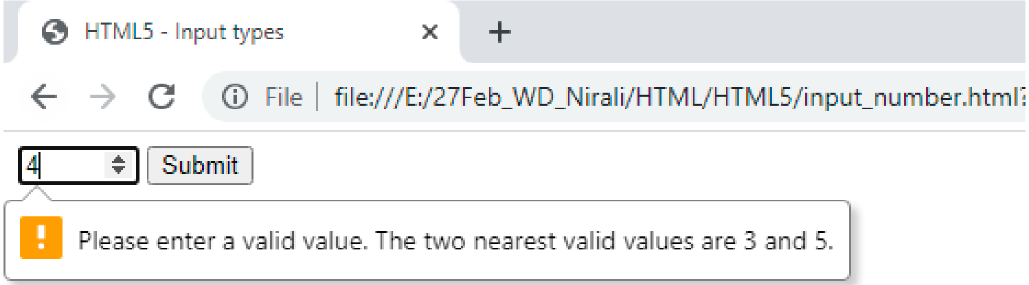
Value must be less than or equal to 10.

- HTML <input type=”number”> example with step attribute.

Example

```
<form>
         <input type="number" min="1" max="10" value="1" step="2">
         <input type="submit" value="Submit">
</form>
```

Output



4

Please enter a valid value. The two nearest valid values are 3 and 5.

Search input type

- The **number** is the value of the **type** attribute of an **<input>** element.
- The <input type="search"> defines a text field for entering a search string. These are functionally symmetrical to the text input type, but may be styled differently.
- When user enter search string in search box, it will display X sign.

Browser Support

Attribute					
type="search"	26.0	10.0	4.0	5.1	12.1

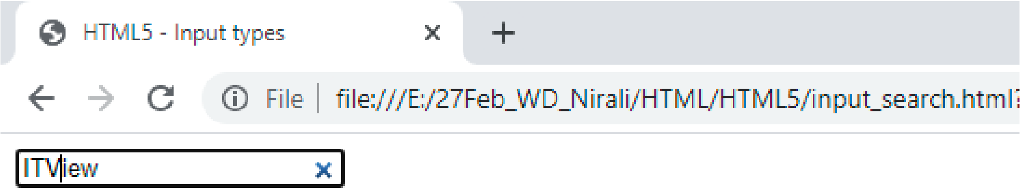
Attributes

HTML <input type=”search”> supports all common attributes of <input> element. No specific attributes for search input.

Example

```
<form>
        <input type="search">
</form>
```

Output

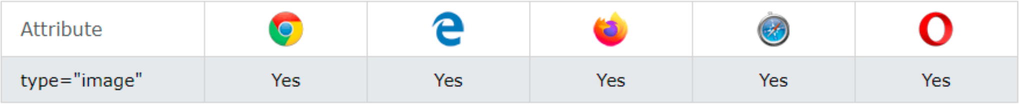


ITView

Image input type

- The **number** is the value of the **type** attribute of an **<input>** element.
- The <input type="image"> defines an image as a submit button.

Browser Support



Attribute					
type="image"	Yes	Yes	Yes	Yes	Yes

Attributes

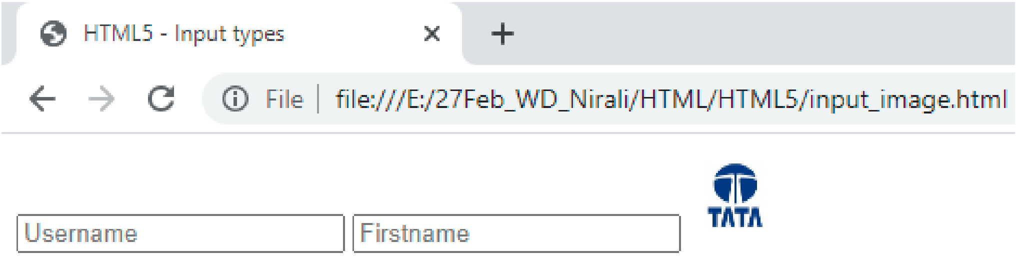
Along with the attributes common to all <input> elements, number input have the following specific attributes:

Attribute	Value	Description
height	pixels	The height, in CSS pixels, at which to draw the image
width	pixels	The minimum value to accept for this input
src	url	The URL from which to load the image
alt	string	The width, in CSS pixels, at which to draw the image

Example

```
<form>
    <input type="text" placeholder="Username">
    <input type="text" placeholder="Firstname">
    <input type="image" src="../../images/tata-logo.jpg" height="50px"
    width="50px">
</form>
```

Output



- **Color input type**
  - The **number** is the value of the **type** attribute of an **<input>** element.
  - The **HTML <input type=”color”>** is used to *define a color picker*. the value should be a seven character hexadecimal notation.

- It has a Default value which is **#000000**(black). The value must be in seven-character hexadecimal notation.

Browser Support

Attribute					
type="color"	20.0	38.0	29.0	10.1	11.01

Attributes

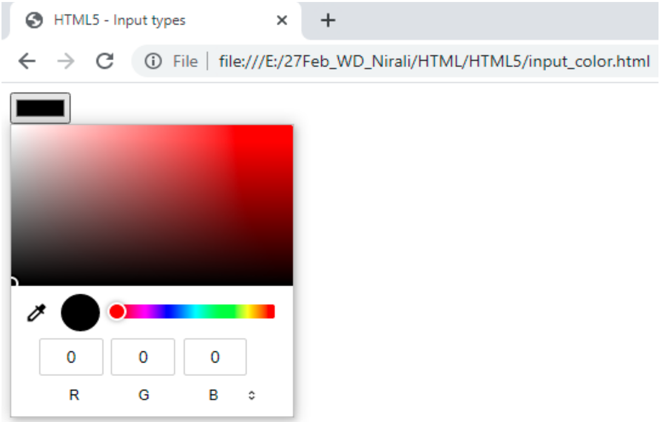
HTML <input type=”color”> supports all the common attributes of <input> element. No specific attributes for color input.

Example

- HTML <input type=”color”> example without any attributes.

```
<form method="post">
        <input type="color">
</form>
```

Output

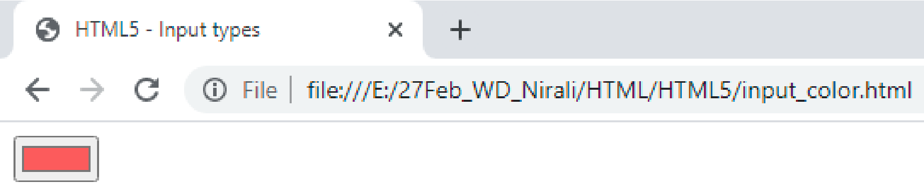

HTML5 - Input types
File | file:///E:/27Feb_WD_Nirali/HTML/HTML5/input_color.html

R	G	B
0	0	0

- HTML <input type=”color”> example with value attribute.

```
<form method="post">
        <input type="color" value="#e66465">
</form>
```

Output



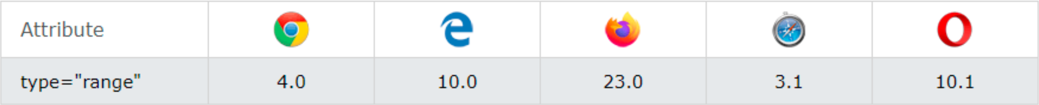
HTML5 - Input types

File | file:///E:/27Feb_WD_Nirali/HTML/HTML5/input_color.html

➢ Range input type

- The **number** is the value of the **type** attribute of an **<input>** element.
- The <input type="range"> defines a control for entering a number whose exact value is not important (like a slider control). Default range is 0 to 100. However, you can set restrictions on what numbers are accepted with the min, max, and step attributes:

Browser Support



Attribute					
type="range"	4.0	10.0	23.0	3.1	10.1

Attributes

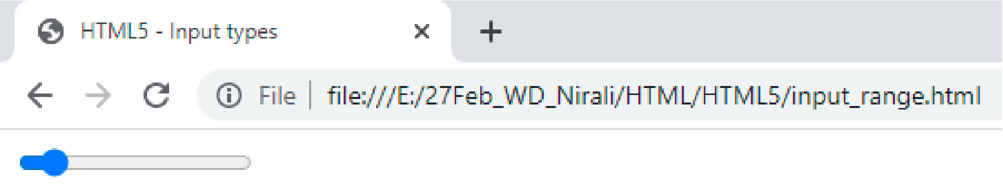
Along with the attributes common to all <input> elements, number input have the following specific attributes:

Attribute	Value	Description
max	number	specifies the maximum value allowed
min	number	specifies the minimum value allowed
step	number	specifies the legal number intervals

Example

```
<form>
  <input type="range" min="1000" max="20000" value="3000">
</form>
```

Output



HTML5 - Input types

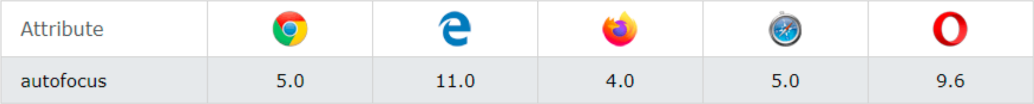
file:///E:/27Feb_WD_Nirali/HTML/HTML5/input_range.html

➢ Autofocus attribute

- The autofocus attribute is a boolean attribute.
- When present, it specifies that an <input> element should automatically get focus when the page loads.

> **Note**: It supports only the following elements: <button>, <input>, <select> and <textarea>

Browser Support

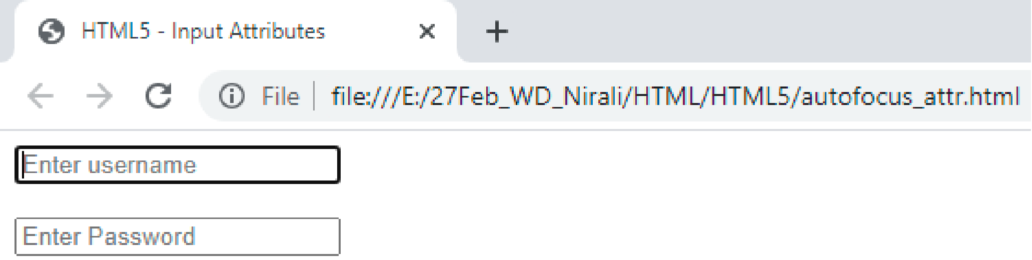


Attribute					
autofocus	5.0	11.0	4.0	5.0	9.6

Example

```
<form>
        <input type="text" placeholder="Enter username" autofocus>
        <input type="text" placeholder="Enter Password">
</form>
```

Output



HTML5 - Input Attributes

file:///E:/27Feb_WD_Nirali/HTML/HTML5/autofocus_attr.html

Enter username

Enter Password

➢ Autocomplete attribute

- The **autocomplete** attribute specifies whether a form or an input field should have autocomplete on or off.

- Autocomplete allows the browser to predict the value. When a user starts to type in a field, the browser should display options to fill in the field, based on earlier typed values.
- It is possible to have autocomplete "on" for the form, and "off" for specific input fields, or vice versa.
- The autocomplete attribute can be used with form and input elements.

> **Note**: The autocomplete attribute works with the following <input> types: text, search, url, tel, email, password, datepickers, range, and color.

Attribute Values:

- **on:** It specifies that autocomplete is enabled.
- **off:** It specifies that the autocomplete is disabled.

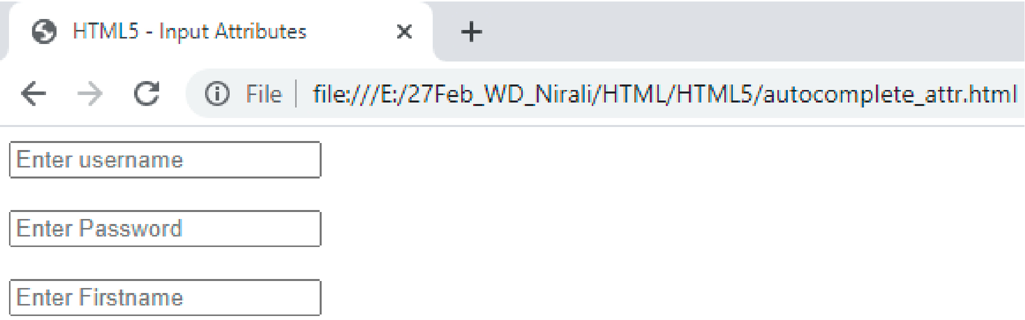
Browser Support

Element					
form	Yes	Yes	4.0	5.2	15.0
input	17.0	5.0	4.0	5.2	9.6

Example

```
<form autocomplete="off">
        <input type="text" placeholder="Enter username" name="uname">
        <input type="text" name="pass" placeholder="Enter Password">
        <input type="text" name="fname" placeholder="Enter Firstname" autoco
mplete="on">
    </form>
```

Output


HTML5 - Input Attributes
File
file:///E:/27Feb_WD_Nirali/HTML/HTML5/autocomplete_attr.html

Enter username

Enter Password

Enter Firstname

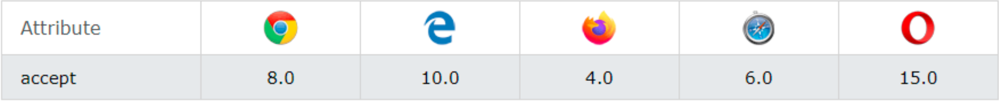
Accept attribute

- The accept Attribute *is used to* specifies the type of file that the server accepts.
- This attribute can be used with **<input type=”file”>** element only.
- This attribute is not used for validation tool because file uploads should be validated on the Server.

Attribute Value

Type of value of HTML accept attribute is **a valid MIME Type(for example image/gif).**

Browser Support



Attribute					
accept	8.0	10.0	4.0	6.0	15.0

Example

- HTML accept attribute example with single value

```
<form>
        <input type="file" accept="image/*">
</form>
```

- HTML accept attribute example with multiple values

```
<form>
        <input type="file" accept="image/jpg,image/jpeg">
</form>
```

Multiple Attribute

- The multiple attribute in HTML allows user to enter more than one value. It is a Boolean attribute and can be used on <input> as well as <select> element,
- To allow multiple file uploads in HTML forms, use the multiple attribute. The multiple attribute works with email and file input types.

For <input type="file">: To select multiple files, hold down the CTRL or SHIFT key while selecting.

For <input type="email">: Separate each email with a comma, like: mail@example.com, mail2@example.com, mail3@example.com in the email field.

For <select>: To select multiple options, hold down the CTRL key while selecting.

Browser Support

Attribute					
multiple	6.0	10.0	3.6	5.0	11.0

Example

```
<form>
        <input type="file" multiple>
              <input type="email" multiple>
              <select multiple>
                        <option>Select Courses</option>
                        <option>HTML</option>
                        <option>CSS</option>
                        <option>Java Script</option>
                        <option>jQuery</option>
                        <option>Bootstrap</option>
              </select>
</form>
```

➢ Required attribute

- The required attribute of the <input> element is used to set a field which is required to be filled before the form is submitted. If the field set with required attribute is not filled, then on clicking the SUBMIT button, the form won’t submit.
- This attribute used with the **<input>, <select>, <textarea>** elements

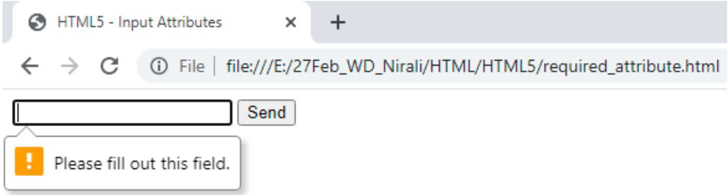
Browser Support

Attribute					
required	5.0	10.0	4.0	10.1	9.6

Example

```
<form>
        <input type="text" required>
        <input type="submit" value="Send">
</form>
```

Output



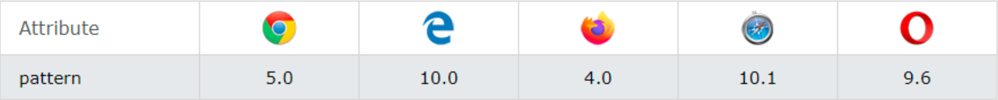
HTML5 - Input Attributes
×
+
File
file:///E:/27Feb_WD_Nirali/HTML/HTML5/required_attribute.html
Send

Please fill out this field.

➢ Pattern attribute

- The pattern attribute specifies a regular expression that the <input> element's value is checked against on form submission.
- The pattern attribute works with the following **input types(text, date, search, url, tel, email, and password) and textarea.**
- Use the global **title** attribute to describe the pattern or display validation message to help the user.
- Use regular expression as a pattern attribute value

Browser support



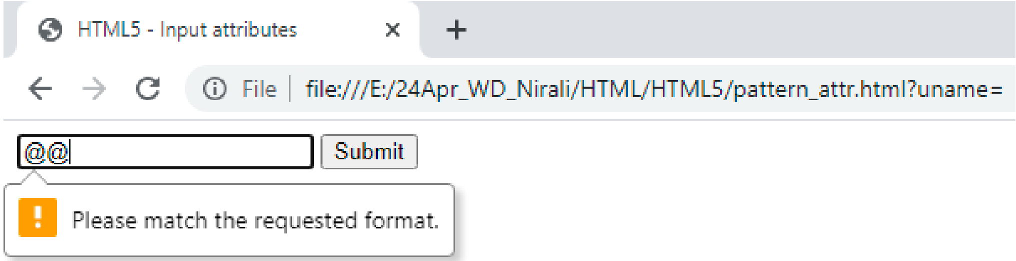
Attribute					
pattern	5.0	10.0	4.0	10.1	9.6

Example

- HTML pattern attribute example without title attribute.

```
<form action="">
        <input type="text" name="uname" id="" pattern="[a-zA-Z0-9]{4,8}">
        <input type="submit">
</form>
```

Output



File | file:///E:/24Apr_WD_Nirali/HTML/HTML5/pattern_attr.html?uname=

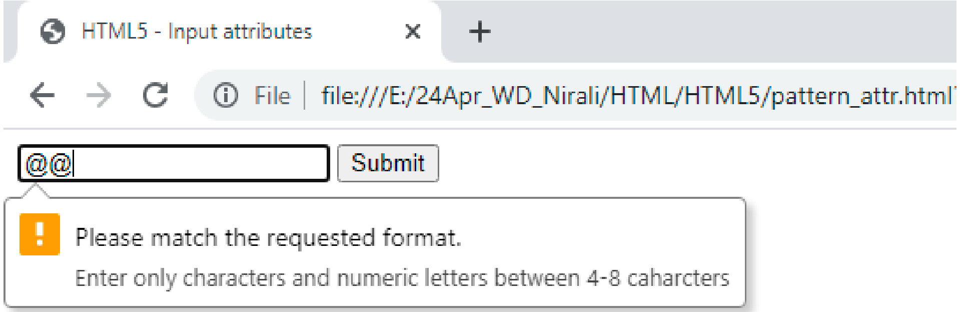
@@

! Please match the requested format.

- HTML pattern attribute example with title attribute.

```
<form action="">
        <input type="text" name="uname" id="" pattern="[a-zA-Z0-9]{4,8}">
        <input type="submit">
</form>
```

Output



@@

Please match the requested format.
Enter only characters and numeric letters between 4-8 caharcters

Placeholder attribute

- If you want to set a hint for text area or input field, then use the HTML placeholder attribute. The hint is the expected value, which gets displayed before the user enters a value, for example, name, details, etc.
- The placeholder attribute works with textarea and following input types: text, search, url, tel, email, number and password.

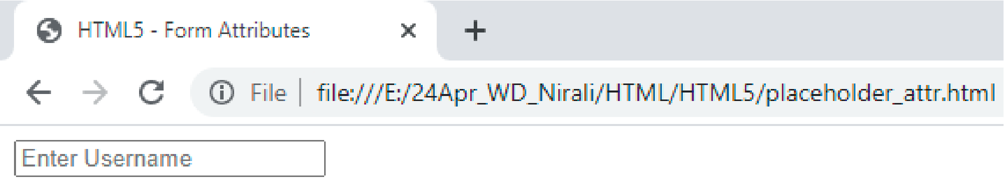
Browser Support

Attribute					
placeholder	10.0	10.0	4.0	5.0	11.0

Example

```
<form>
        <input type="text" name="uname" placeholder="Enter Username">
</form>
```

Output



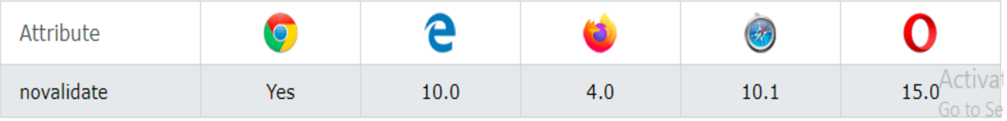
Enter Username

Novalidate attribute

- The **HTML novalidate Attribute** is used to *specify* that the form-data should not be validated when submitting the form. It is a Boolean attribute. It can be used with the **<form>** element.

- In the following example, if you will not add email id in proper format in the <input type=”email” > field, then it won’t show an error. Same for <input type=”number”>, <input type=”url”>, <input type=”number”>, etc.

Browser Support



Attribute					
novalidate	Yes	10.0	4.0	10.1	15.0

Example

```
<form novalidate>
        <input type="number " name="number" id="">
        <input type="email" name="fname" id="">
        <input type="submit" value="submit">
</form>
```

❖ Global Attributes in HTML

- Here is the list of some common global attributes:

Attribute	Description
class	Specifies one or more classnames for an element (refers to a class in a style sheet)
contenteditable	Specifies whether the content of an element is editable or not
dir	Specifies the text direction for the content in an element
hidden	Specifies that an element is not yet, or is no longer, relevant
id	Specifies a unique id for an element
spellcheck	Specifies whether the element is to have its spelling and grammar checked or not
style	Specifies an inline CSS style for an element
tabindex	Specifies the tabbing order of an element
title	Specifies extra information about an element

❖ Input Restrictions

- Here is the list of some common Input Restrictions:

Attribute	Description
checked	Specifies that an input field should be pre-selected when the page loads (for type="checkbox" or type="radio")
disabled	Specifies that an input field should be disabled
max	Specifies the maximum value for an input field
maxlength	Specifies the maximum number of character for an input field
min	Specifies the minimum value for an input field
pattern	Specifies a regular expression to check the input value against
readonly	Specifies that an input field is read only (cannot be changed)
required	Specifies that an input field is required (must be filled out)
size	Specifies the width (in characters) of an input field
step	Specifies the legal number intervals for an input field
value	Specifies the default value for an input field