edureka!

Module-2 Creating web pages with HTML5

www.edureka.co/complete-web-developer

Course Topics

edureka!

→ Module 1
  » Deploying the first Website to Amazon S3

→ Module 2
  » **Creating web pages with HTML5**

→ Module 3
  » Styling web pages using CSS

→ Module 4
  » CSS3 effects and animations

→ Module 5
  » Handling events with JavaScript

→ Module 6
  » Twitter Bootstrap 3

→ Module 7
  » Twitter Bootstrap 3 Project

→ Module 8
  » Bootstrap ScrollSpy, jQuery and jQuery UI

→ Module 9
  » Ajax, Google APIs, Social Plugins

→ Module 10
  » Project - Building Website Tour

Objectives

edureka!

At the end of this module, you will be able to:

→ Understand to encode the URL

→ Learn the use of XHTML, its rules and the advantages of XHTML

→ Learn the use of various Input Types and the Form Attributes in the webpage

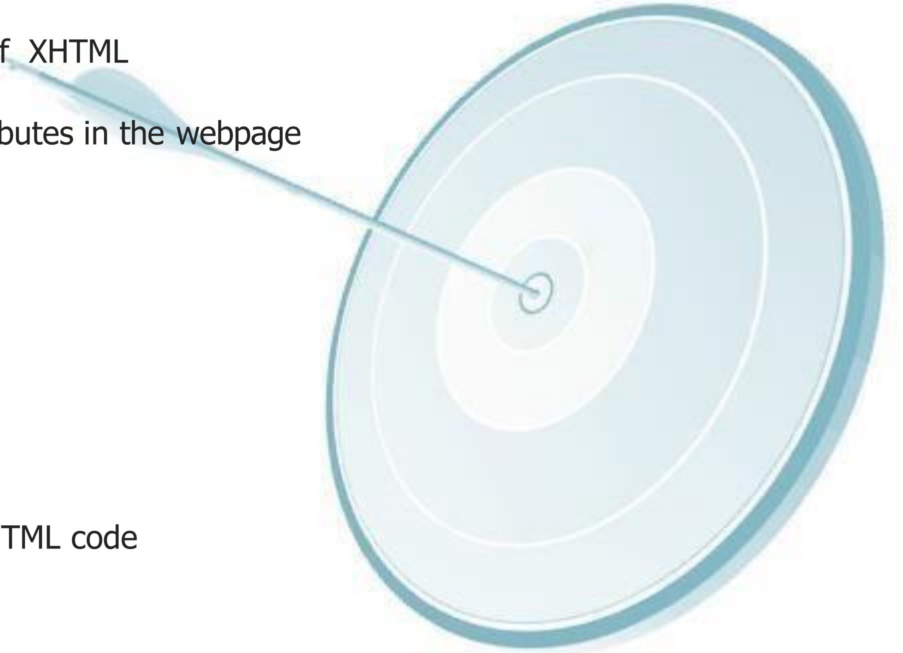
→ Set video background for website

→ Understand Canvas and SVG to design graphics

→ Embed audio, videos, geo-location in the webpage

→ Differentiate between localStorage and sessionStorage

→ Understand the importance of using JavaScript in the HTML code



HTML URL

edureka!

→ URL stands for Uniform Resource Locator. It is an address to the resource available on the Internet

→ The request from the Web browser to the Web server is given through the URL

→ User can either give the website address or IP (Internet Protocol) address

→ From the browser, user can type http://www.google.com or can type IP address of the domain google.com

HTML URL (Contd.)

→ URL must be in the following format:

  » Scheme: //IP-A ddress_Or_Domain-Name:port/path/filename

→ Here Scheme is http or https or ftp

  » http stands for hypertext transfer protocol

  » https stands for hypertext transfer protocol with security - https is used when the information between and browser and server needs to be secured

  For example, credit card payment in shopping sites, internet banking etc.

  » ftp – FTP stands for File transfer protocol. FTP is used to transfer file from one system to another system

HTML URL (Contd.)

The description of the components of the URL is as follows:

→ Host: www (world wide web)

→ Domain server name: It is the domain name of the server. Domain names are the hierarchical distributed naming system for the resources associated with the internet. E.g., for google it is google.com

→ Port: Defines the port at which the browser can communicate with the server

→ Path: Specifies the program to be executed on the server. If the path is not specified, root of the server is assumed

→ File name: Name of the file that is given in the path to be executed. Files can be html file, jsp file, asp file etc.

Encode URL

→ When the user enters a string in the browser's URL pane with spaces, URL encoding is done

→ http://www.myapp.com/new pricing.htm URL returns an error because spaces in the URL is not accepted by the server

→ %20 is the space character. Thus the URL can be modified as http://www.myapp.com/new%20pricing.htm

→ At the server side, it understands "new pricing.htm"

XHTML

edureka!

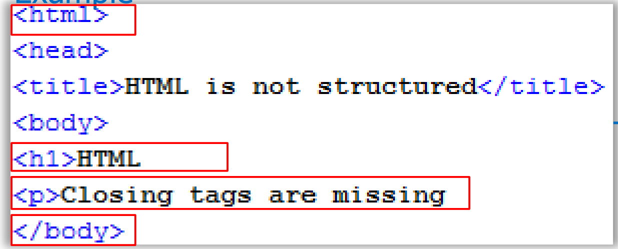
→ Some of the HTML code run perfectly though the HTML rules for writing the HTML script is not strictly followed

→ Such codes run on the computer systems but an error is encountered when opened on a mobile phone. To solve such issues XHTML was created

→ XHTML(Extensible Hypertext Mark-up Language) is a combination of XML and HTML

→ XML is a markup language where scripts must be well-formed. It is very rigid

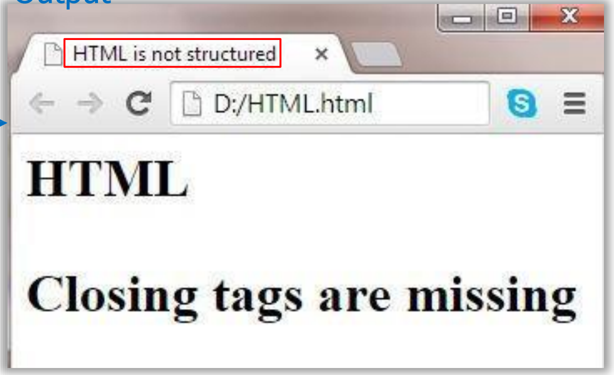
Example



```
<html>
<head>
<title>HTML is not structured</title>
<body>
<h1>HTML
<p>Closing tags are missing
</body>
```

As you can see, there is no closing tag for <h1>, <p>, <head> and <html>. But still the output is displayed

Output



HTML is not structured

D:/HTML.html

HTML

Closing tags are missing

Rules for XHTML

edureka!

- → DOCTYPE, HTML, HEADER, TITLE and BODY tags are mandatory in XHTML
- → XHTML elements should be properly nested
- → A ll the tags/elements should be lowercase and must be properly closed
- → XHTML should have one root element
- → A ll the attribute data must be in lowercase and should be placed in double quotes

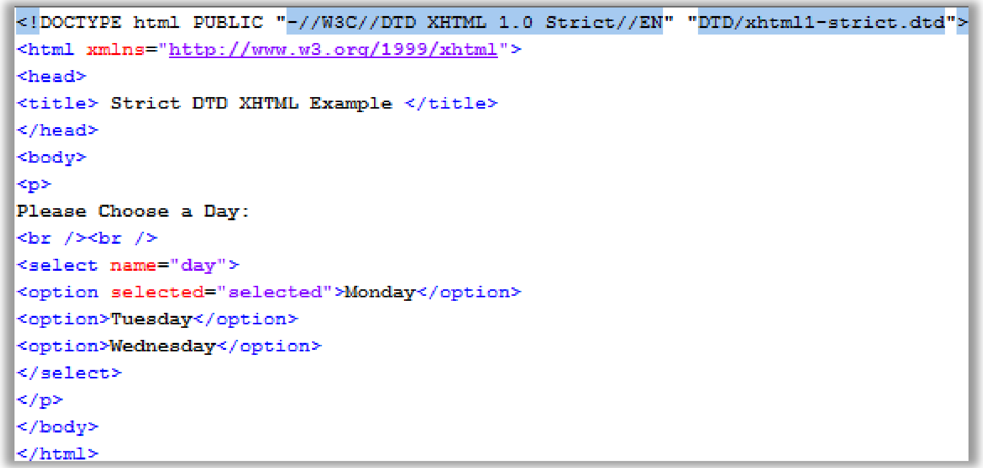
Advantages of XHTML

edureka!

- → Sustainability: A s the industry is moving towards XML, and XHTML has XML, it is widely accepted
- → Wide range of application: With XHTML, complex websites can be designed as it has data and UI
- → Compatibility: Since XHTML has the format of XML, developers can read/parse XHTML documents and can convert it into PDF, RTF formats
- → Efficiency of processing: Since XHTML is strict version of HTML, less number of errors will be there in the document and speed of processing the XHTML document is faster
- → A clean code: Since XHTML has mandatory closing tags unlike HTML, it is always a clean code
  - » For a beginner it is easy to read and understand and for programmers it is a good and clean code

XHTML

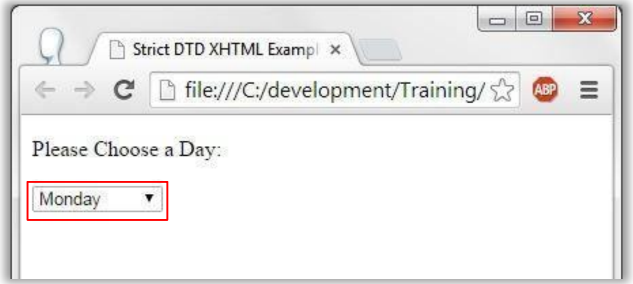
Example



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title> Strict DTD XHTML Example </title>
</head>
<body>
<p>
Please Choose a Day:
<br /><br />
<select name="day">
<option selected="selected">Monday</option>
<option>Tuesday</option>
<option>Wednesday</option>
</select>
</p>
</body>
</html>
```

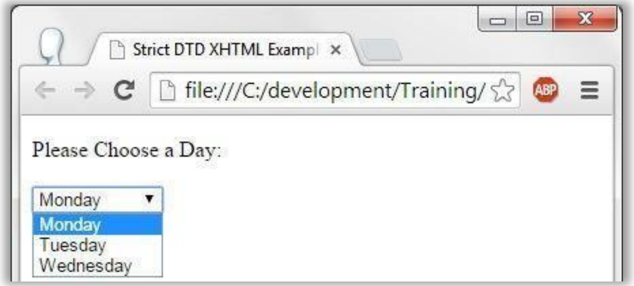
xmlns sta nds for XM L Namespace. Specified URL will give the information about the current HTML docume nt. Specifying URL is not mandatory

Output



Please Choose a Day:

Monday



Please Choose a Day:

Monday
Monday
Tuesday
Wednesday

New Elements in HTML – 5

edureka!

→ Better document structure: Article, header, figcaption, header, footer, section, summary etc.

→ Form elements: Datalist, keygen, output etc.

→ Form Input types: Date, datetime, color, email, telephone, time, url etc.

→ Graphics: SVG, Canvas

→ Media elements: A udio, video, embed, source, track etc.

Semantic Elements

edureka!

→ Semantic elements are the tags with meaning. The user can understand what has to be done just by looking at the tag

- » Example of Semantic elements: <table>, <img>, <form>
- » Examples of Non-semantic elements: <div>, <span> etc.

→ The below given HTML-5 semantic elements does nothing except for better documentation

- » Article: Defines an article
- » Aside: Like a side bar
- » Figure: To display set of figures
- » Header: Section for header
- » Footer: Section for footer
- » Summary: Section to display summary

→ Semantic elements which does operation are:

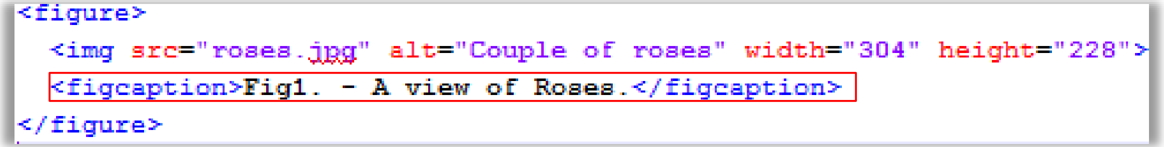
- » Figcaption: This displays the caption for the figure
- » Mark: It is the highlighter used to highlight the text

Semantic Element – <figcaption> Tag

→ In the below code, figcaption is used to display the caption for the image. It gives an idea of what the image is

→ Each figure will have a figure caption. Only one <figcaption> tag can be embedded in a <figure> tag

Example



```
<figure>
  <img src="roses.jpg" alt="Couple of roses" width="304" height="228">
  <figcaption>Fig1. - A view of Roses.</figcaption>
</figure>
```

Output

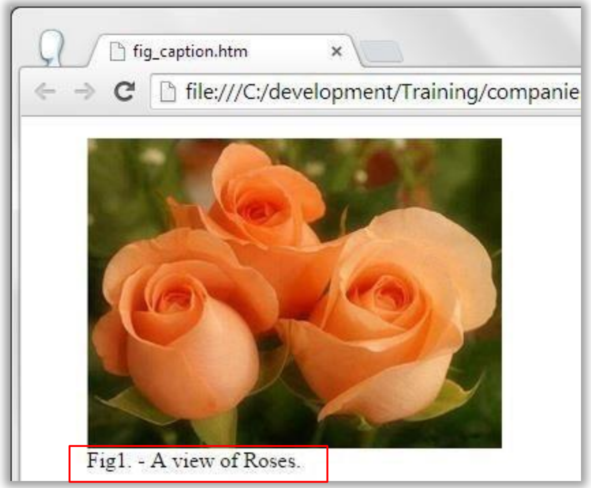


Fig1. - A view of Roses.

Annie's Question

What is the significance of alt attribute in <img> tag?

Annie’s Answer

edureka!

Ans. When the image is unable to display an alternative text will be displayed. This is made possible using the “alt” attribute

Semantic Element – <mark> Tag

edureka!

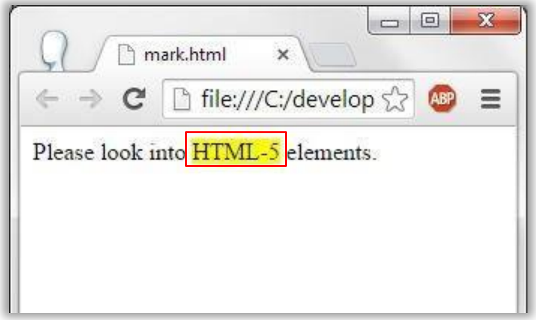
→ It is same as the <mark> attribute in html but in html-5 it is made as a tag for semantics

→ This is done to bring attention of the user to that part of the text

Example

```
<p>Please look into <mark>HTML-5</mark> elements.</p>
```

Output



Please look into HTML-5 elements.

Form Element – <datalist> Tag

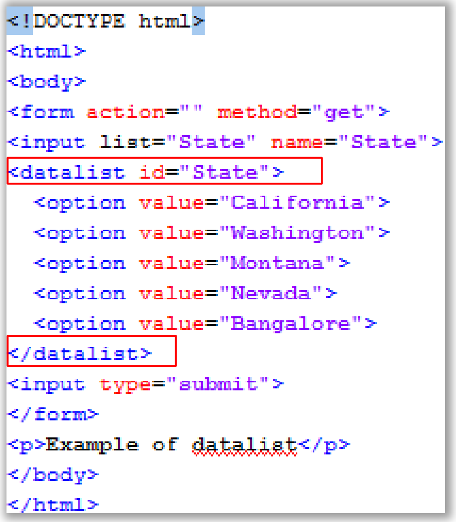
edureka!

→ New Form elements in HTML5 are <datalist> and <output>

→ <datalist> tag defines a list of data in drop-down menu for the user to select when clicked on the textbox

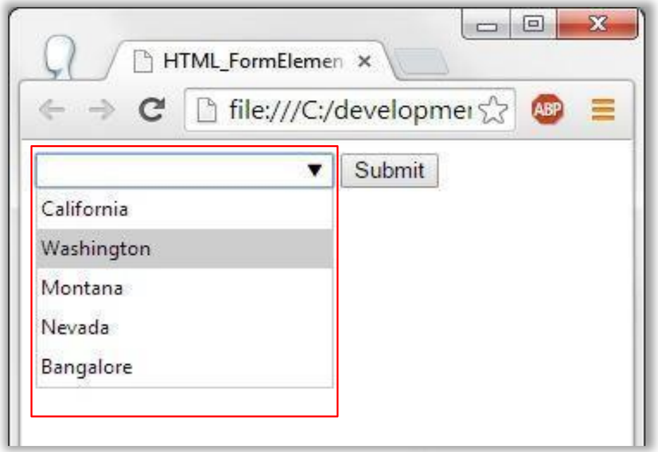
→ In this example datalist tag provides the data for the input list called "State"

Example



```
<!DOCTYPE html>
<html>
<body>
<form action="" method="get">
<input list="State" name="State">
<datalist id="State">
  <option value="California">
  <option value="Washington">
  <option value="Montana">
  <option value="Nevada">
  <option value="Bangalore">
</datalist>
<input type="submit">
</form>
<p>Example of datalist</p>
</body>
</html>
```

Output



Annie's Question

Why is it required to display list of items in the datalist?

Annie's Answer

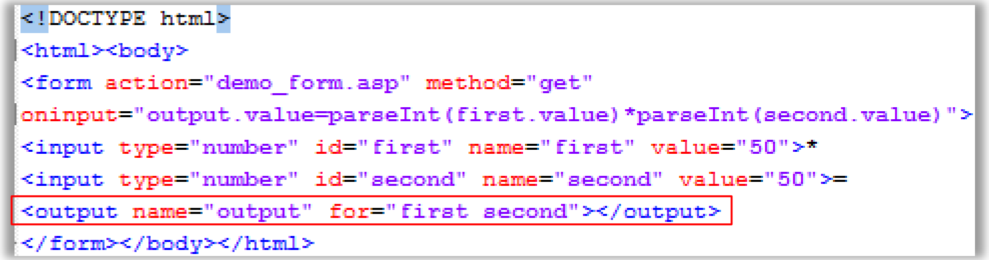
Ans. When there are many options and user has to select one amongst them, then list of items are displayed in data list

Form Element – <output> Tag

edureka!

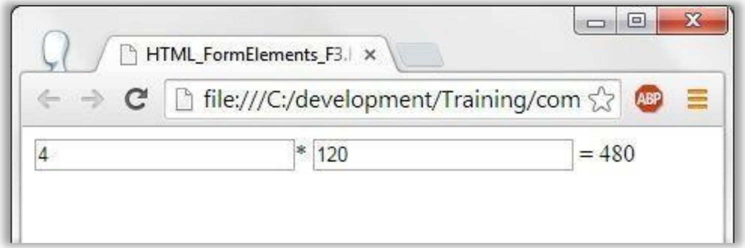
- → When you want to perform any calculations, <output> is used
- → <output> tag performs the calculation based on the user's input and displays the result
- → Here two input numbers are taken using input tag
- → On entering data, they are converted to integer data and multiplied and displayed in output tag

Example



```
<!DOCTYPE html>
<html><body>
<form action="demo_form.asp" method="get"
oninput="output.value=parseInt(first.value)*parseInt(second.value)">
<input type="number" id="first" name="first" value="50">*
<input type="number" id="second" name="second" value="50">=
<output name="output" for="first second"></output>
</form></body></html>
```

Output



```
4 * 120 = 480
```

HTML5 – Input Type

edureka!

→ The Input Types supported by HTML are: text, password, radio buttons, check boxes and submit

→ In HTML5, various Input Types are introduced. They are:

- » Color
- » Date
- » Datetime
- » Email
- » Url
- » Week
- » Number

Each one will be explained in detail in the upcoming slides

Input Type – Number

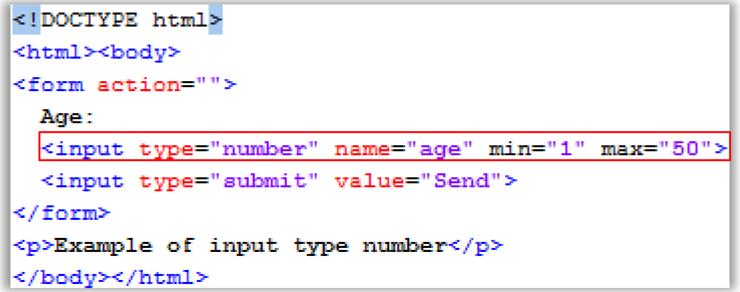
edureka!

→ When an input field must contain a number, the "number" input type is used

→ Here the input type is number. User enters a number. Minimum and maximum limit will be specified

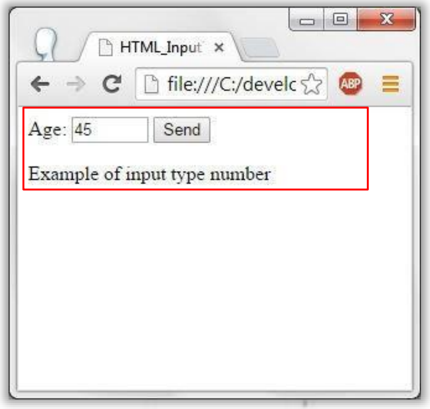
→ If the user does not enter the number in this bound then an appropriate message will be displayed to the user, as shown

Example

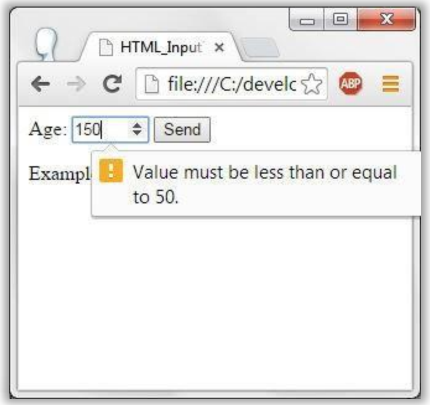


```
<!DOCTYPE html>
<html><body>
<form action="">
  Age:
  <input type="number" name="age" min="1" max="50">
  <input type="submit" value="Send">
</form>
<p>Example of input type number</p>
</body></html>
```

Output



Example of input type number



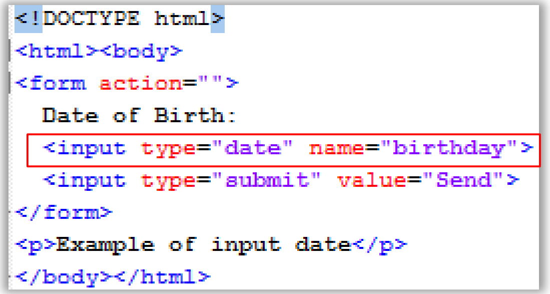
Value must be less than or equal to 50.

Input Type – Date

edureka!

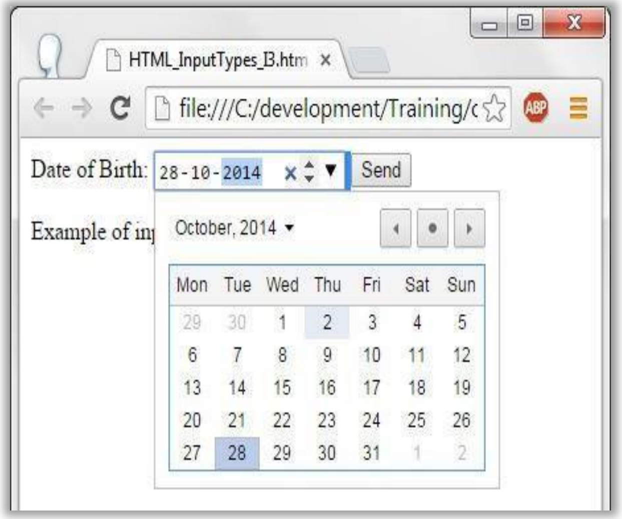
→ When you want to specify a date in the input field, "date" input type is used

Example



```
<!DOCTYPE html>
<html><body>
<form action="">
  Date of Birth:
  <input type="date" name="birthday">
  <input type="submit" value="Send">
</form>
<p>Example of input date</p>
</body></html>
```

Output



HTML_InputTypes_B.htm
file:///C:/development/Training/c
Date of Birth: 28-10-2014
Send
Example of in
October, 2014
Mon Tue Wed Thu Fri Sat Sun

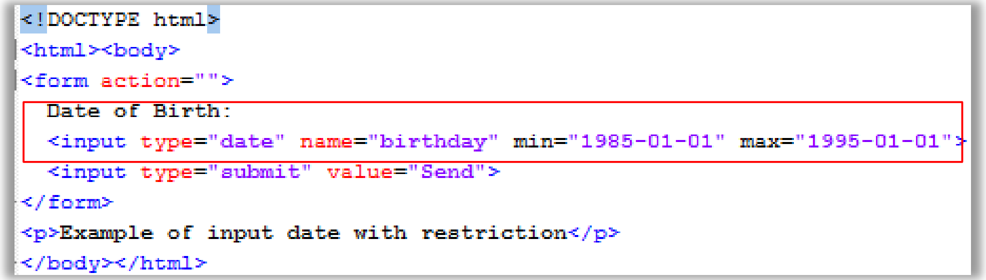
Mon	Tue	Wed	Thu	Fri	Sat	Sun
29	30	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2

Input Type – Date-range

→ When you have to restrict the date, use min and max keywords to specify the range

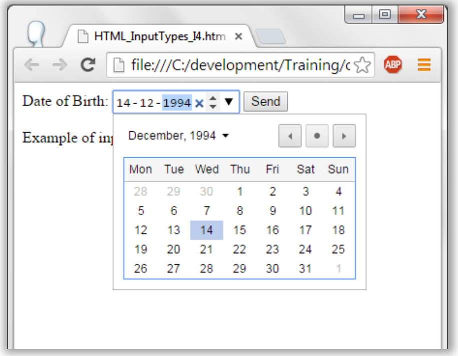
→ In this code, user can select date range between min and max attributes

Example

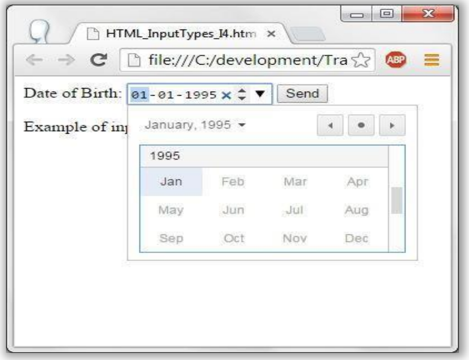


```
<!DOCTYPE html>
<html><body>
<form action="">
  Date of Birth:
  <input type="date" name="birthday" min="1985-01-01" max="1995-01-01">
  <input type="submit" value="Send">
</form>
<p>Example of input date with restriction</p>
</body></html>
```

Output



HTML_InputTypes_I4.htm
file:///C:/development/Training/c
Date of Birth: 14-12-1994
Send
Example of in
December, 1994
Mon Tue Wed Thu Fri Sat Sun



Annie's Question

Give a practice example of date range.

Annie's Answer

Ans. When the bank statement are to be displayed between the date range, then we use date-range

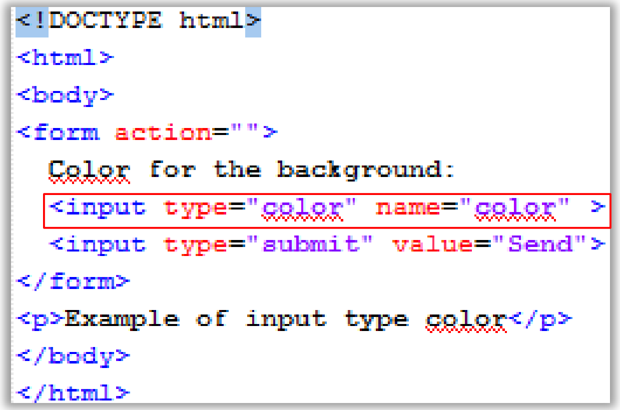
Input Type – Color

→ When the user wants to apply color to any element, input type "color" is used

→ When the user clicks on the color button, color window is displayed

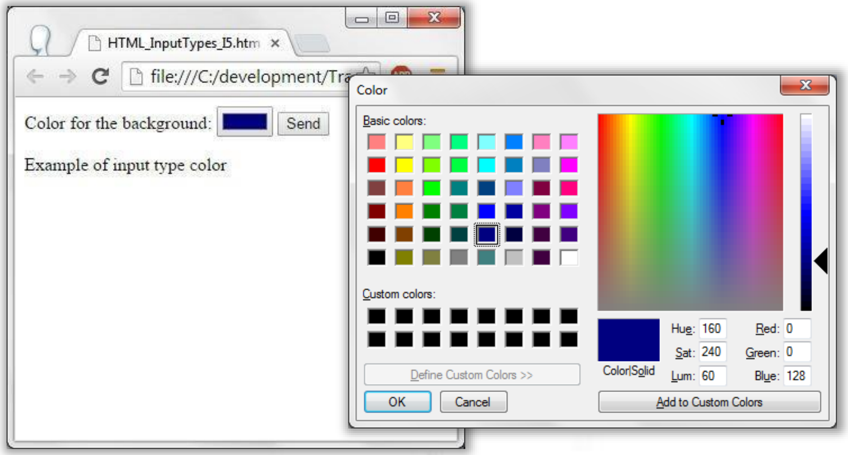
→ User can choose the colors in the color window and can click on ok button to select the color

Example



```
<!DOCTYPE html>
<html>
<body>
<form action="">
  Color for the background:
  <input type="color" name="color" >
  <input type="submit" value="Send">
</form>
<p>Example of input type color</p>
</body>
</html>
```

Output



Color for the background: Send

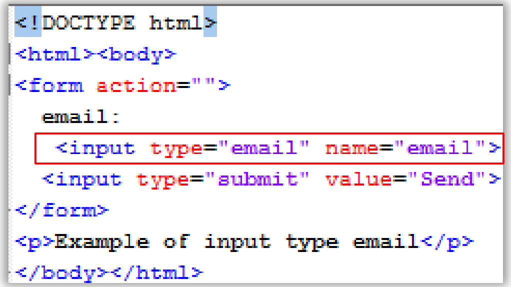
Example of input type color

Input Type – Email

→ Here the input type email is used. The email is validated when sent to the webserver

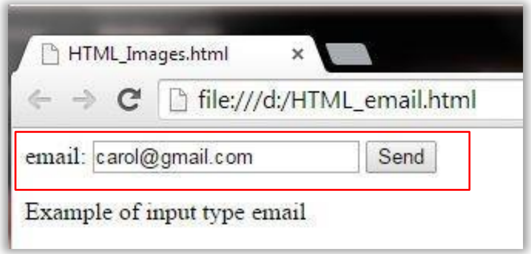
→ If the user types invalid email id then HTML5 will display an error message

Example



```
<!DOCTYPE html>
<html><body>
<form action="">
  email:
   <input type="email" name="email">
  <input type="submit" value="Send">
</form>
<p>Example of input type email</p>
</body></html>
```

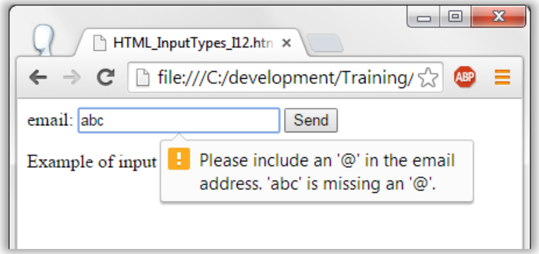
Output



file:///d:/HTML_email.html

email: carol@gmail.com Send

Example of input type email



file:///C:/development/Training/

email: abc Send

Example of input

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

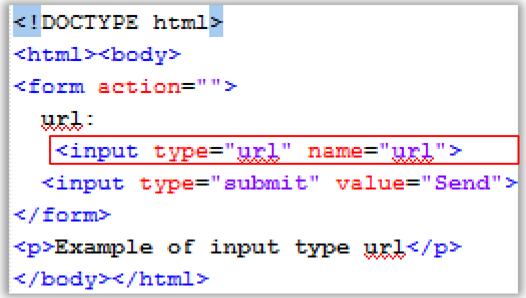
Input Type – URL

edureka!

→ Here the input type URL is used. The URL is validated when sent to the webserver

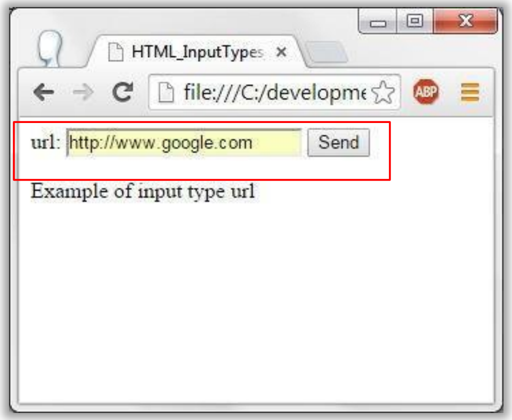
→ If the user does not enter a valid URL then error message is displayed by browser

Example



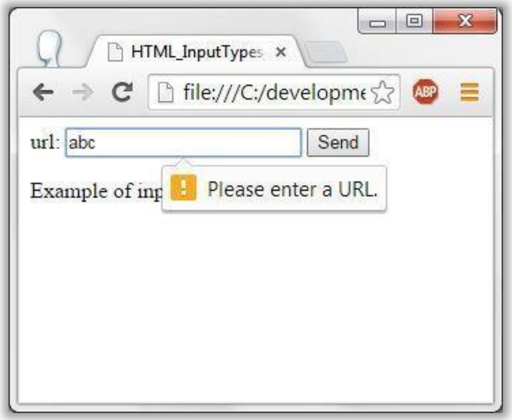
```
<!DOCTYPE html>
<html><body>
<form action="">
  url:
  <input type="url" name="url">
  <input type="submit" value="Send">
</form>
<p>Example of input type url</p>
</body></html>
```

Output



url: http://www.google.com

Example of input type url



url: abc

Example of inp

Please enter a URL.

Form Attribute – Autocomplete

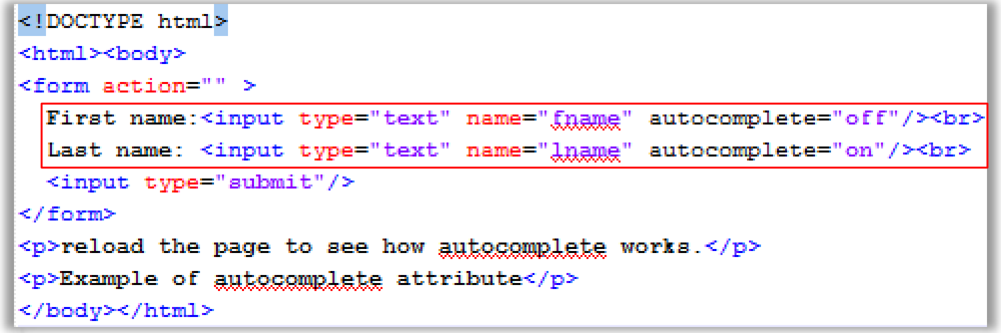
edureka!

→ When you type text in the text box, the suggestions appears to the user to make their choice easier. This is done with the help of autocomplete attribute in HTML

→ With the autocomplete on, user will be suggested with the option which were entered previously

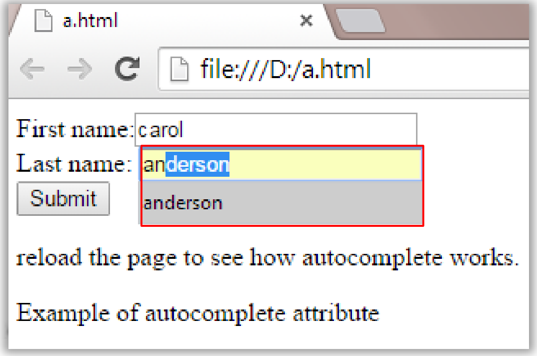
→ First input field autocomplete is off and for the second input field – Autocomplete is on

Example



```
<!DOCTYPE html>
<html><body>
<form action="" >
  First name:<input type="text" name="fname" autocomplete="off"/><br>
  Last name: <input type="text" name="lname" autocomplete="on"/><br>
  <input type="submit"/>
</form>
<p>reload the page to see how autocomplete works.</p>
<p>Example of autocomplete attribute</p>
</body></html>
```

Output



First name: carol

Last name: anderson

reload the page to see how autocomplete works.

Example of autocomplete attribute

Form Attribute –Autofocus

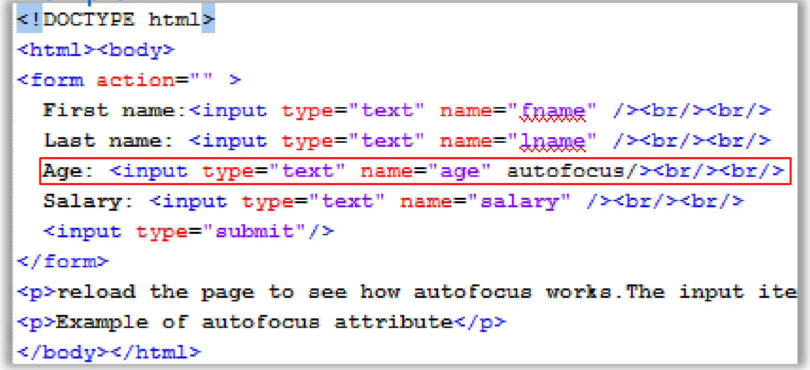
edureka!

→ When you want the user to focus on an important field or a mandatory field in the form, autofocus attribute is used

→ Autofocus lets the user to automatically focus on a particular input field in the webpage

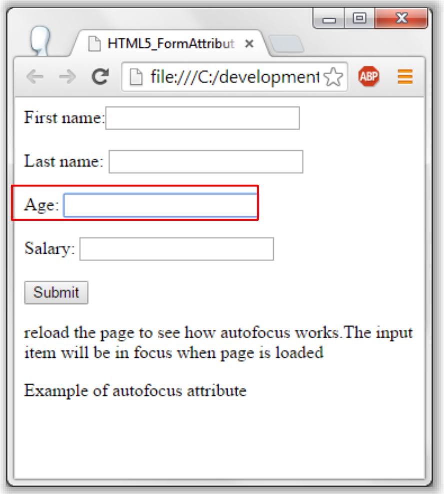
→ Using the attribute autofocus, when the form is displayed, control will be in auto focus field first which is age field in this example

Example



```
<!DOCTYPE html>
<html><body>
<form action="" >
  First name:<input type="text" name="fname" /><br/><br/>
  Last name: <input type="text" name="lname" /><br/><br/>
  Age: <input type="text" name="age" autofocus/><br/><br/>
  Salary: <input type="text" name="salary" /><br/><br/>
  <input type="submit"/>
</form>
<p>reload the page to see how autofocus works.The input ite
<p>Example of autofocus attribute</p>
</body></html>
```

Output



reload the page to see how autofocus works.The input item will be in focus when page is loaded

Example of autofocus attribute

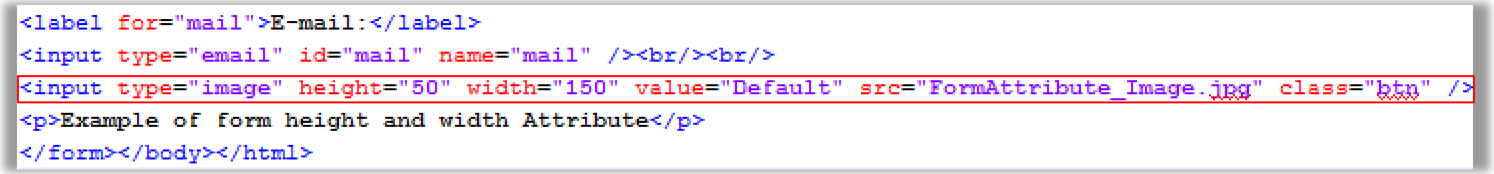
Form Attribute – Height and Width

→ When an element has to be resized, height and width attributes are used

→ Here height and width are specified in the input element

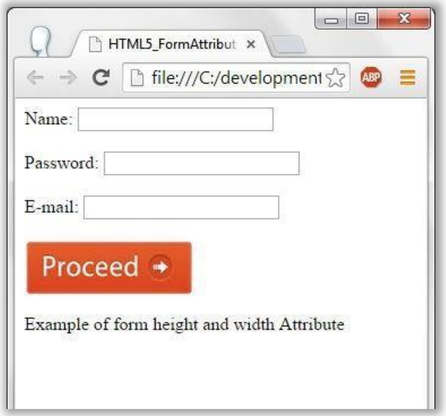
→ In this example, the image "Proceed" is using the height and width attributes

Example



```
<label for="mail">E-mail:</label>
<input type="email" id="mail" name="mail" /><br/><br/>
<input type="image" height="50" width="150" value="Default" src="FormAttribute_Image.jpg" class="btn" />
<p>Example of form height and width Attribute</p>
</form></body></html>
```

Output



Example of form height and width Attribute

Form Attribute –Multiple

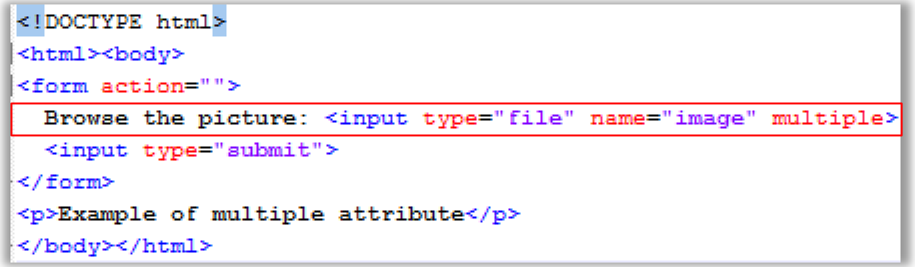
edureka!

→ When the user wants to select a particular file from a collection of files, multiple attribute in the form is used

→ With the "multiple" option in the input type, user can select multiple files from the file selection box

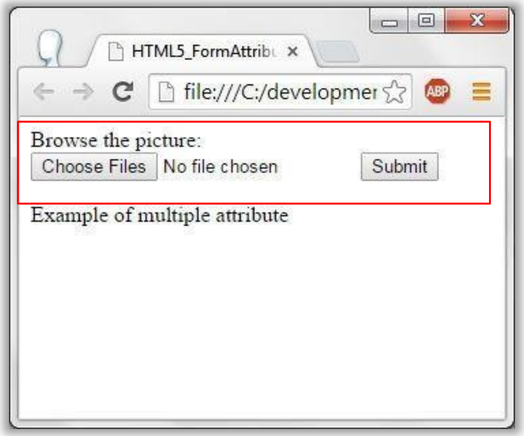
→ Click on choose files, the window where the user makes choice is displayed. User can select the file from this window

Example



```
<!DOCTYPE html>
<html><body>
<form action="">
  Browse the picture: <input type="file" name="image" multiple>
  <input type="submit">
</form>
<p>Example of multiple attribute</p>
</body></html>
```

Output



Browse the picture:
No file chosen
Example of multiple attribute

Form Attribute –Novalidate

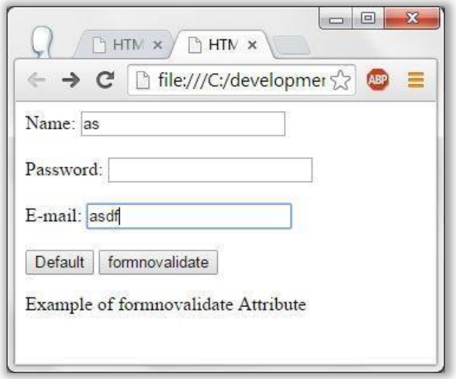
→ When the submitted form needs no validation, novalidation attribute in the form is used

→ In this example, there is a form with two text fields and two buttons. When the first button is clicked form is validated and when the second button is clicked form is not validated

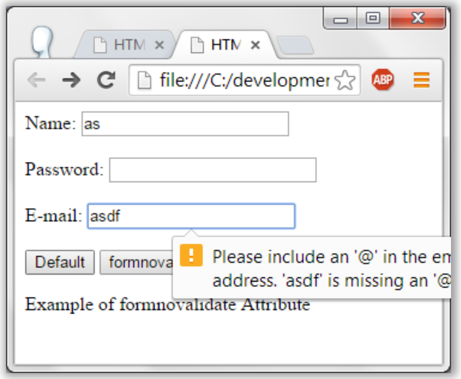
Example

```
<input type="email" id="mail" name="mail" class="textbox"/><br/><br/>
<input type="submit" value="Default" class="btn" />
<input type="submit" value="formnovalidate" class="btn"   formnovalidate  />
<p>Example of formnovalidate Attribute</p>
</form></body></html>
```

Output



```
Name: as
Password:
E-mail: asdf
Default  formnovalidate
Example of formnovalidate Attribute
```



```
Name: as
Password:
E-mail: asdf
Default  formnova
Please include an '@' in the em
address. 'asdf' is missing an '@
Example of formnovalidate Attribute
```

Canvas

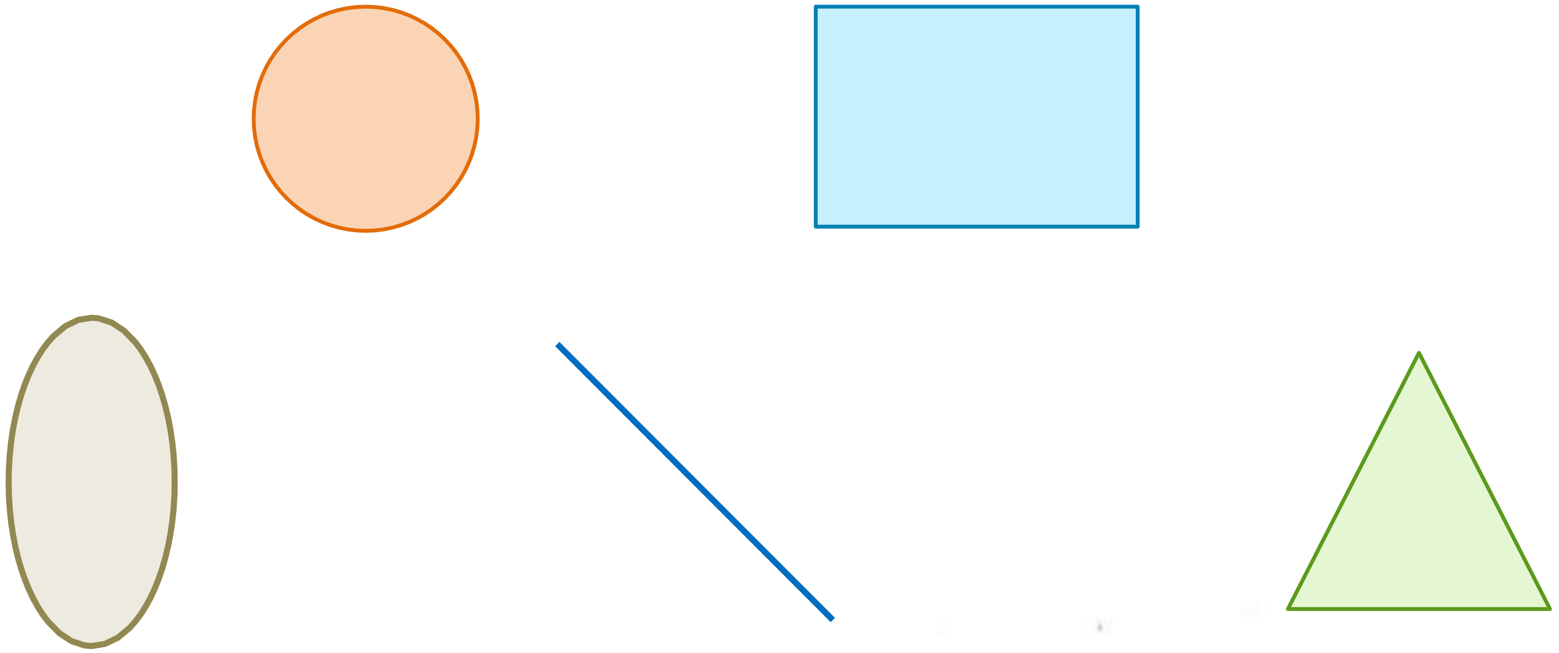
edureka!

Jenny, I want to draw different shapes in my webpage, how can that be done?
Hey! that is easy in HTML. You can draw shapes using the <canvas> tag

Canvas

edureka!

→ When you have to draw different shapes in your webpage, you can use the <canvas> tag



Canvas

edureka!

- → When the user wants to draw a circle, rectangle, paths of different sizes, <canvas> is used
- → <canvas> element is used to draw graphics and are defined in HTML
- → Canvas uses script (JavaScript) to draw the graphics on web page
- → Canvas tag/element is a container to store and draw graphics
- → Script will be written to write graphics in this container
- → Boxes, text, images, lines can be written in Canvas container

Canvas is a Container

→ Canvas is a rectangular area in the webpage where the graphics are drawn

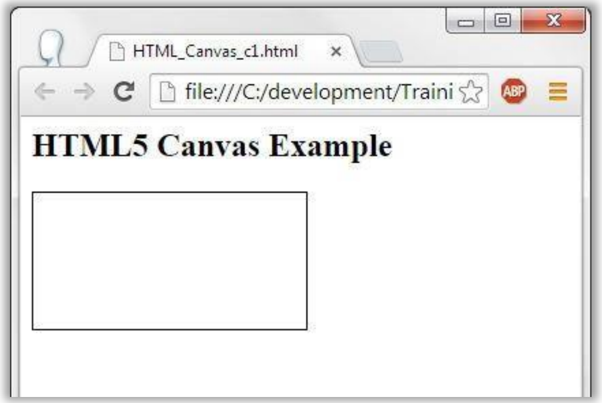
→ In this code, canvas is of the width 200 pixels and height of 100 pixels and border size is of 1px

→ Now let us see how to draw different graphics and shapes using the <canvas> tag in HTML

Example

```
<body>
  <h2> HTML5 Canvas Example</h2>
  <canvas width="200" id="can" height="100" style="border:1px solid #000000; ">
  </canvas>
 </body>
```

Output



HTML5 Canvas Example

Canvas – Rectangle

→ In the script, object of Canvas is obtained by getElementById()

→ var context = Canvas1.getContext("2d") - obtains the context of the object

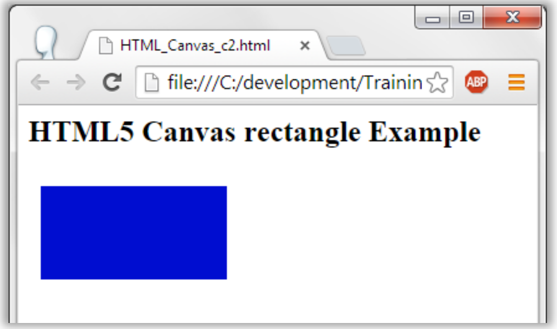
→ To draw a rectangle we use the 2D context functions like – fillRect()

→ fillRect() is used to draw a filled rectangle. fillStyle specifies the color to be filled in the Rectangle and the rectangle co-ordinates are specified using the fillRect() which has the (x1, y1, width, height) as parameters

Example

```
<script>
  var canvas1= document.getElementById("can");
  var context = canvas1.getContext("2d");
  context.fillStyle = "#000dd0";
  context.fillRect(0,0,200,100);
  </script>
```

Output



HTML5 Canvas rectangle Example

Canvas – Line

→ In the canvas container,

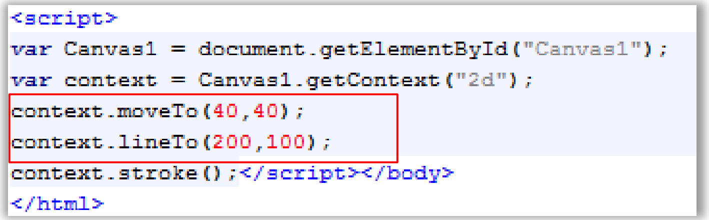
» moveTo(x,y) is used to position the cursor at the specified coordinates where the line must begin i.e., (40,40)

» lineTo(x,y) is used to draw a line from the start position to the new position i.e., (200,100)

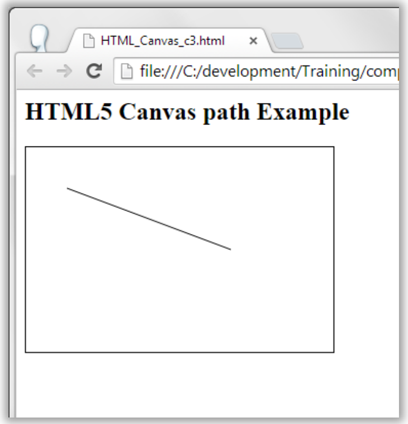
→ The line is made visible using the stroke()

Example

Output



```
<script>
var Canvas1 = document.getElementById("Canvas1");
var context = Canvas1.getContext("2d");
context.moveTo(40,40);
context.lineTo(200,100);
context.stroke();</script></body>
</html>
```



HTML5 Canvas path Example

Canvas – Arc

edureka!

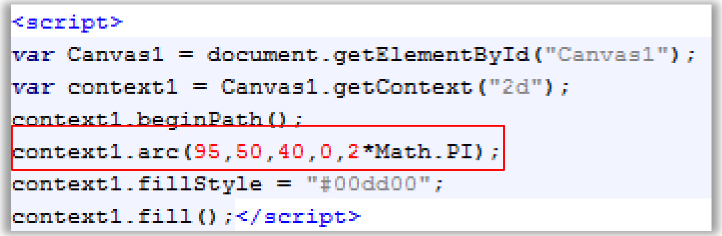
→ To draw an arc in HTML5, we use the arc()

→ The Syntax of arc is: arc (x, y, radius, starting angle, ending angle)

→ X and Y are the centre point of the arc

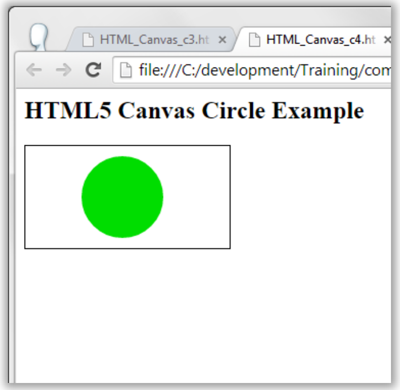
→ Starting and ending angles are in radians

Example



```
<script>
var Canvas1 = document.getElementById("Canvas1");
var context1 = Canvas1.getContext("2d");
context1.beginPath();
context1.arc(95,50,40,0,2*Math.PI);
context1.fillStyle = "#00dd00";
context1.fill();</script>
```

Output



HTML5 Canvas Circle Example

www.edureka.co/complete-web-developer

Creating an HTML5 Canvas Clock

There are lots of cool things that can be done with HTML5 Canvas. Please check the document on Creating an HTML5 Canvas Clock uploaded to LMS.

Creating an HTML5 Canvas Clock

Canvas – Text

→ We have learnt to draw figures in the canvas, now let us learn how to write text into the Canvas

→ fillText() is used to write text in Canvas container. Font is used to specify the font style of the text

→ The syntax for fillText is (text,x,y);

→ The x, y coordinate specify the point from where start painting the text

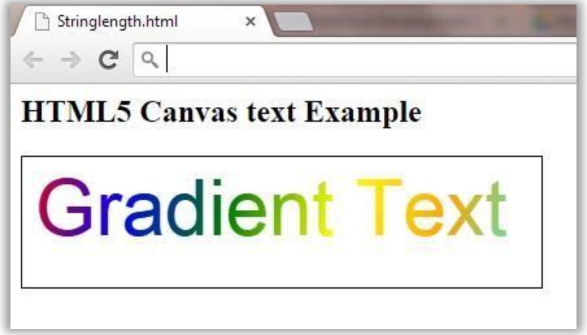
Canvas – Text (Contd.)

edureka!

Example

```
<script>
var Canvas2 = document.getElementById("Canvas2");
var context1 = Canvas2.getContext("2d");
context1.font="60px Arial";
// Create gradient
var gra =context1.createLinearGradient(0,0,Canvas2.width,Canvas2.height);
gra.addColorStop("0","Red");
gra.addColorStop("0.2","Blue");
gra.addColorStop("0.4","Green");
gra.addColorStop("0.6","Yellow");
gra.addColorStop("0.8","Orange");
gra.addColorStop("1.0","Cyan");
// Fill with gradient
context1.fillStyle=gra;
context1.fillText("Gradient Text",10,60);
</script>
```

Output



Canvas – Text (Contd.)

→ In the script, Canvas is defined. The object of Canvas is obtained by getElementById()

→ Context of canvas is obtained and assigned to the variable context1

→ Linear Gradient object is created using createLinearGradient() function. The parameters specify that gradient is from x1,y1 till x2 to y2

→ Specify the colors for the gradient using addColorStop() function. First parameter specifies the value between 0.0 to 1.0

→ fillText() function is used to display the text at the given location

SVG

edureka!

- → SVG (Scalar Vector Graphics) is used to write two-dimensional graphics and animations on web page
- → SVG does not lose the quality if they are zoomed
- → SVG can be printed in very high quality at any resolution
- → SVG is used world wide and is recommended by W3C
- → SVG images can be created and edited by any text editor
- → SVG defines the graphics in XML format

SVG– Circle

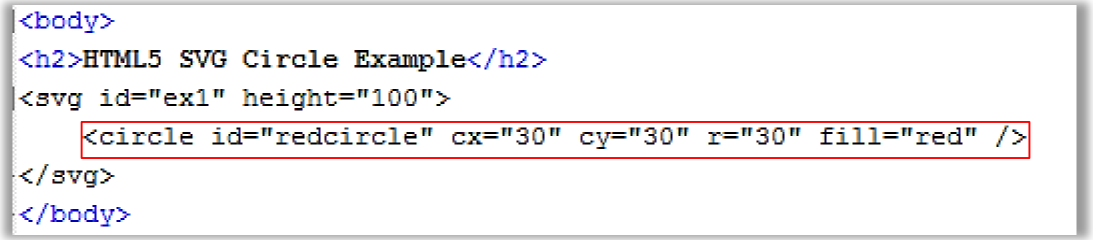
edureka!

→ SVG can draw circle using the <circle> tag by giving x, y co-ordinates and radius as attributes

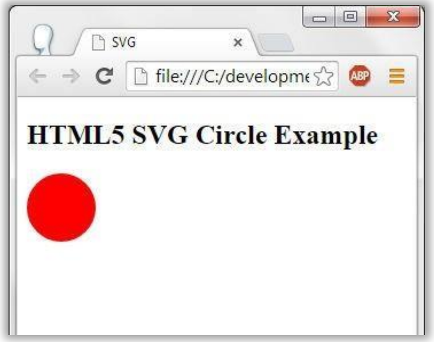
→ Circle is drawn and the color red is filled into the circle

→ The cx and cy attributes define the x and y coordinates of the center of the circle and the r attribute indicates the radius of the circle

Example


```
<body>
<h2>HTML5 SVG Circle Example</h2>
<svg id="ex1" height="100">
    <circle id="redcircle" cx="30" cy="30" r="30" fill="red" />
</svg>
</body>
```

Output



HTML5 SVG Circle Example

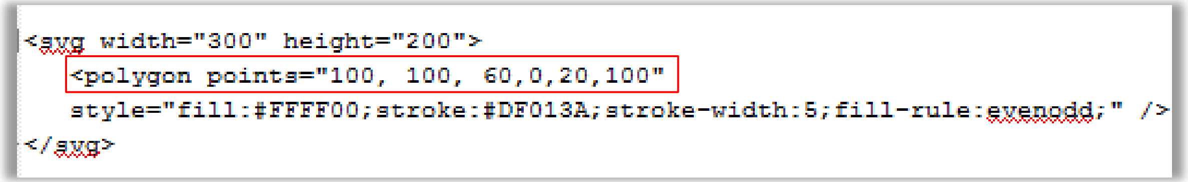
SVG– Rectangle

→ Polygons in HTML are drawn using the <polygon> tag. You will specify the points for your polygon depending on the shape you need

→ You can also style your polygon by fillingin colors and giving it a border

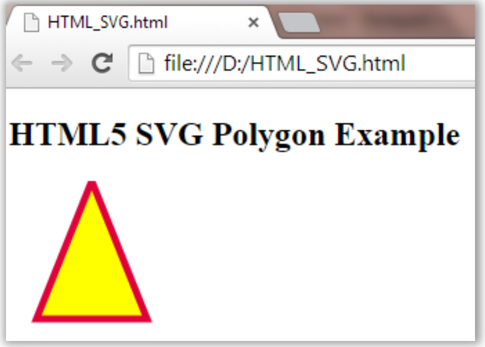
→ Here in our example, we are drawing a triangle with red border of width 5px and we fill yellow color into it

Example



```
<svg width="300" height="200">
   <polygon points="100, 100, 60,0,20,100"
   style="fill:#FFFF00;stroke:#DF013A;stroke-width:5;fill-rule:evenodd;" />
</svg>
```

Output



HTML_SVG.html

file:///D:/HTML_SVG.html

HTML5 SVG Polygon Example

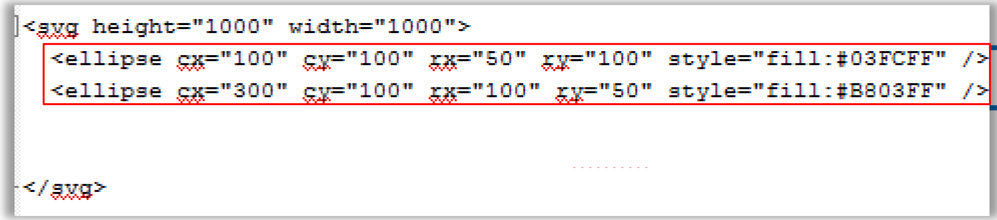
SVG– Ellipse

edureka!

→ Now let's draw an ellipse. It is similar to circle but it has a different radius for the coordinates, x and y

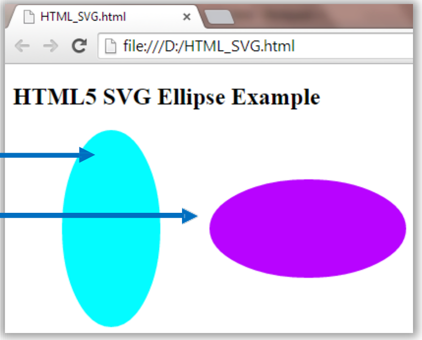
→ Ellipse is drawn using the <ellipse> tag

Example



```
<svg height="1000" width="1000">
  <ellipse cx="100" cy="100" rx="50" ry="100" style="fill:#03FCFF" />
  <ellipse cx="300" cy="100" rx="100" ry="50" style="fill:#B803FF" />


</svg>
```

Output



HTML5 SVG Ellipse Example

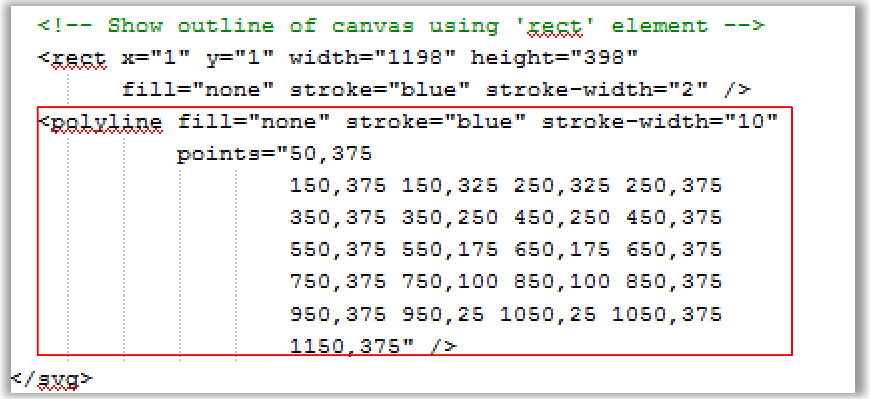
SVG–Polyline

edureka!

→ Polyline is a set of straight lines connecting several points

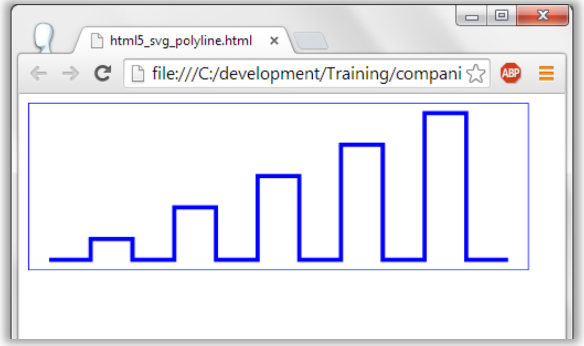
→ Polyline syntax is – First two coordinates is the position control moves to. From there on a line is drawn to the next two (x,y) coordinates and so on

Example



```
<!-- Show outline of canvas using 'rect' element -->
<rect x="1" y="1" width="1198" height="398"
      fill="none" stroke="blue" stroke-width="2" />
<polyline fill="none" stroke="blue" stroke-width="10"
        points="50,375
                150,375 150,325 250,325 250,375
                350,375 350,250 450,250 450,375
                550,375 550,175 650,175 650,375
                750,375 750,100 850,100 850,375
                950,375 950,25 1050,25 1050,375
                1150,375" />
</svg>
```

Output



www.edureka.co/complete-web-developer

Canvas v/s SVG

edureka!

Canvas	SVG
Canvas is drawn using the JavaScript code	SVG is drawn using the XML script code
Canvas is resolution dependent	SVG is not resolution dependent
Canvas doesn't support event handlers	SVG in XML can be represented by SVG DOM and hence can support event handlers
Canvas can be used for gaming applications	SVG has DOM which makes it slow and hence can not be used for gaming applications

HTML5 – Video

edureka!

→ Video tag is used to display the given video or a movie clip using the <video> tag by specifying the height and width of the video

→ The "src" attribute is used to locate the position of the video file that has to be played

→ Controls attribute is used to display the play/pause buttons and other controls on the video

→ HTML5 supports uploading video of 3 formats. They are:

  » MP4

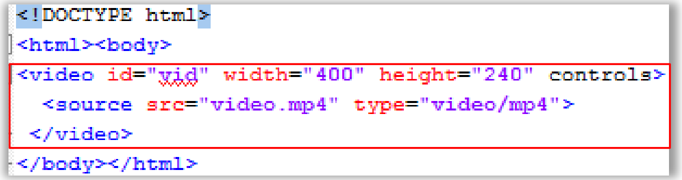
  » WebM

  » Ogg

In our example we are playing the mp4 video

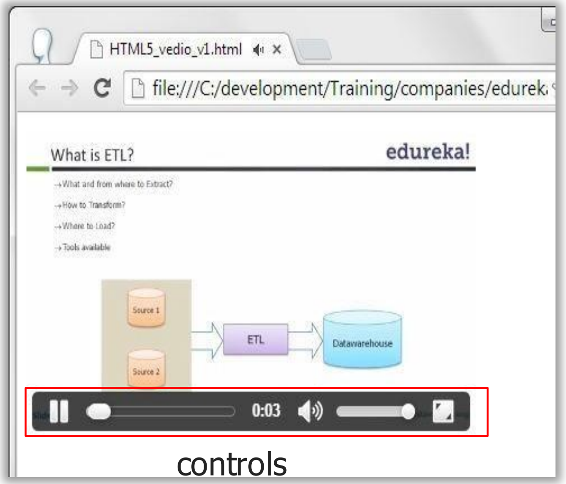
HTML5 – Video (Contd.)

Example



```
<!DOCTYPE html>
<html><body>
<video id="vid" width="400" height="240" controls>
  <source src="video.mp4" type="video/mp4">
 </video>
</body></html>
```

Output



controls

HTML5 – Audio

edureka!

→ HTML5 embeds audio in the webpage. <audio> tag is used to play the audio file. Here we are playing mp3 file

→ The "src" attribute is used to locate the position of the audio file that has to be played

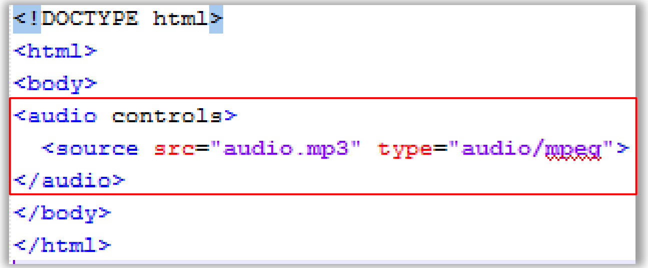
→ HTML5 supports uploading audio of 3 formats. They are:

» MP3

» Wav

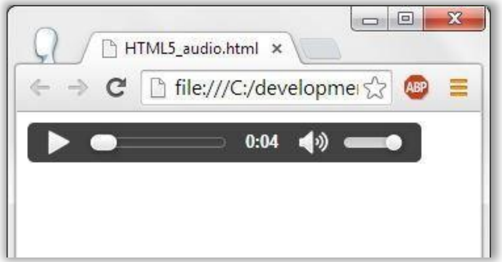
» Ogg

Example



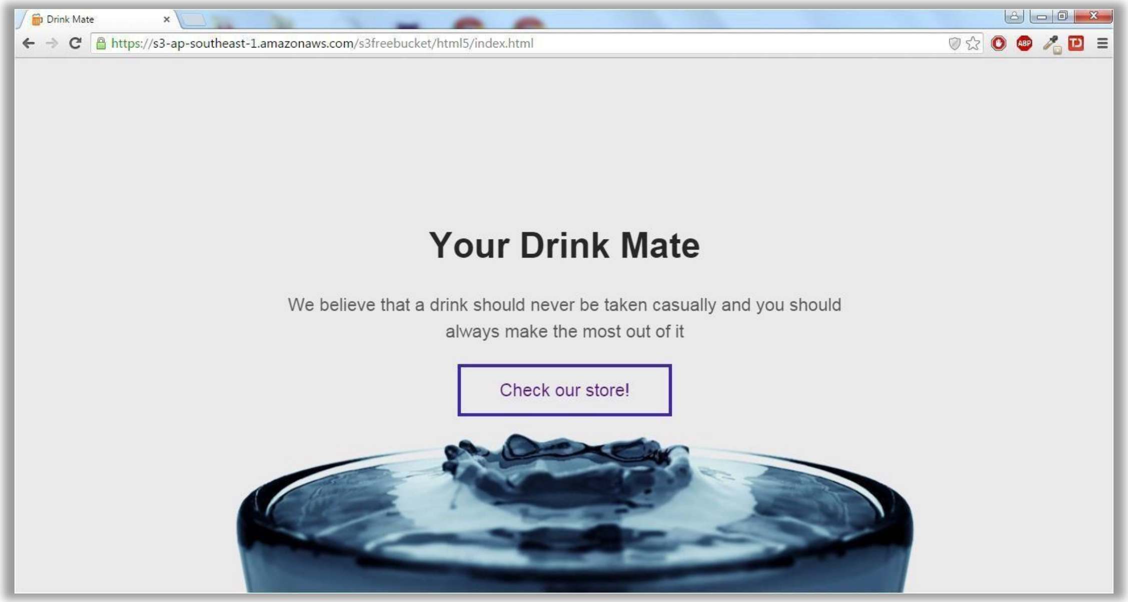
```
<!DOCTYPE html>
<html>
<body>
<audio controls>
  <source src="audio.mp3" type="audio/mpeg">
</audio>
</body>
</html>
```

Output



HTML5 – Setting Video Backgrounds

→ With HTML5 we can set a video as a background for the website.



Drink Mate
https://s3-ap-southeast-1.amazonaws.com/s3freebucket/html5/index.html

Your Drink Mate

We believe that a drink should never be taken casually and you should always make the most out of it

HTML5 – Setting Video Backgrounds

edureka!

→ We can use the HTML5 video tag to set a video as background. The code is shown below

```
<video id="my-video" class="video" autoplay loop  muted>
    <source src="video/demo.mp4" type="video/mp4">
    <source src="video/demo.webm" type="video/webm">
</video>
```

```
<script>
(function() {
  /**
   * Video element
   * @type {HTMLElement}
   */
   var video = document.getElementById("my-video");

      /**
     * Check if video can play, and play it
     */
   video.addEventListener( "canplay", function() {
    video.play();
  });
})();
</script>
```

Conditional Video Play with JavaScript

→ The most important attributes for video backgrounds are:

→ autoplay – if specified, the video will automatically begin to play

→ muted – indicates the default setting of the audio contained in the video

→ loop – if specified, it will loop again through video on reaching the end

All browsers support differe nt se t of video formats so its be tter to specify the video in different formats. If a browser doesn' t support one forma t it will try with the ne xt one and so on. To convert a video in different format you can use http://video.online-convert.com/

HTML5 – Plug-ins

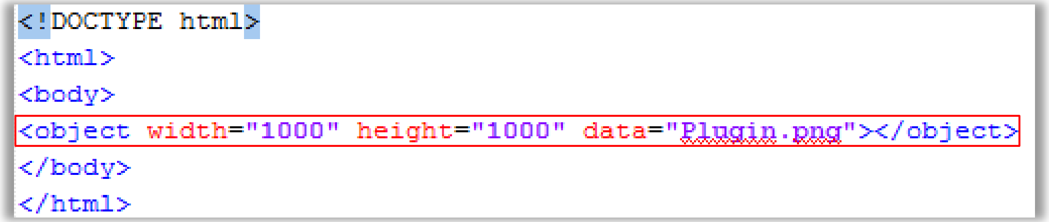
edureka!

→ Plug-ins are software code that adds a special feature to the existing application

→ Plug-ins are nothing but embedding objects into the webpage

→ <object> or <embed> tags are used to embed an object into the current web page

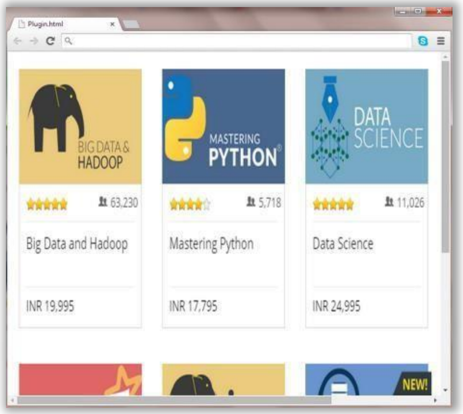
→ Code snippet is given below:



```
<!DOCTYPE html>
<html>
<body>
<object width="1000" height="1000" data="Plugin.png"></object>
</body>
</html>
```

Object element is used to embed multimedia (like audio, video, Java applets, ActiveX, PDF, and Flash) in web pages.
You can also use <object> tag to embed another webpage into your HTML document

Output



HTML5 – GEO Location

edureka!

→ GEO Location is used in finding the location of the user in the world using various techniques like specifying the IP address, GPS hardware

→ Now let us try to find the location of the user in the Google Maps using the HTML5 tags

→ Here we use GEO Location API to allow the user share his location only to the trusted websites

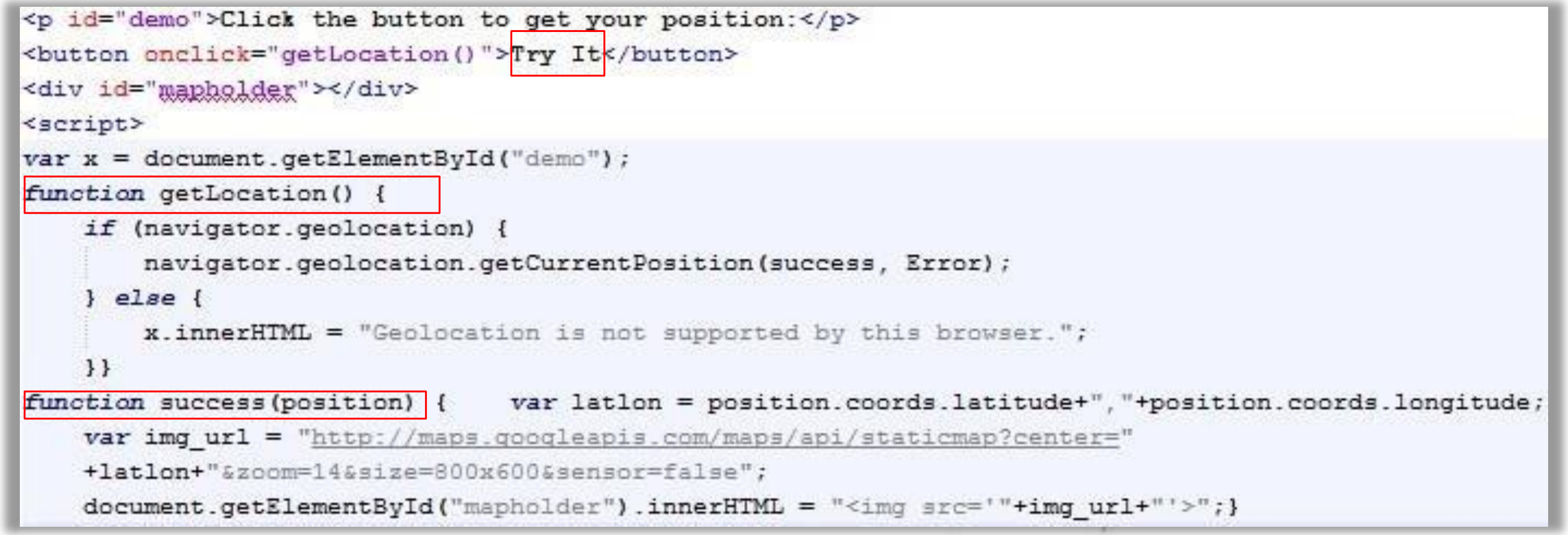
→ GEO Location API is supported by the following browsers :
- » IE 9.0
- » Firefox 3.5
- » Safari 5.0
- » Chrome 5.0
- » Opera 16.0

Geolocation is much more accurate for devices with GPS, like android mobiles and iPhone

HTML5 – GEOLocation (Contd.)

→ The code snippet for finding the location of the user is shown below:

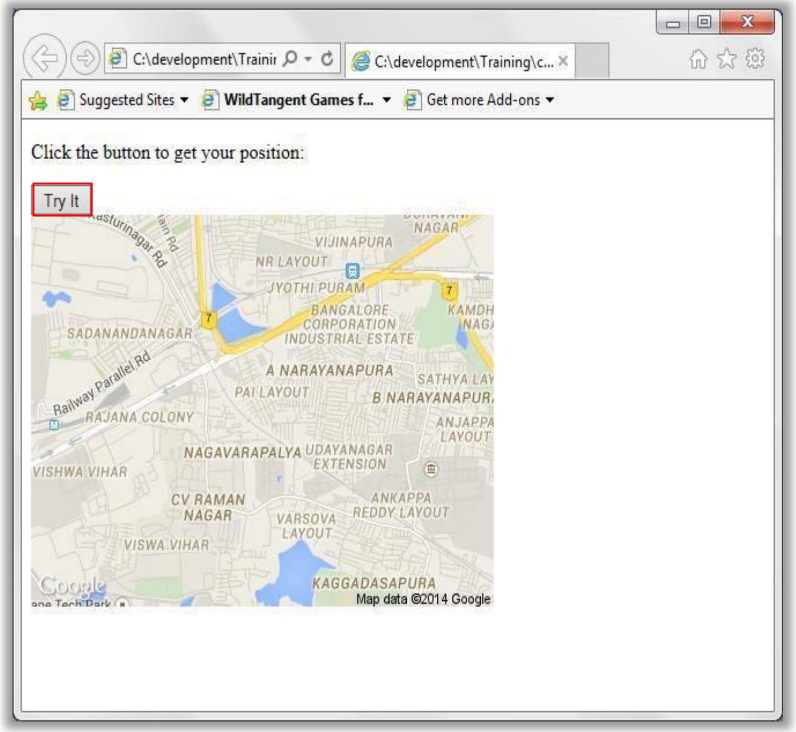
Example



```
<p id="demo">Click the button to get your position:</p>
<button onclick="getLocation()">Try It</button>
<div id="mapholder"></div>
<script>
var x = document.getElementById("demo");
function getLocation() {
    if (navigator.geolocation) {
        navigator.geolocation.getCurrentPosition(success, Error);
    } else {
        x.innerHTML = "Geolocation is not supported by this browser.";
    }}
function success(position) {     var latlon = position.coords.latitude+","+position.coords.longitude;
    var img_url = "http://maps.googleapis.com/maps/api/staticmap?center="
    +latlon+"&zoom=14&size=800x600&sensor=false";
    document.getElementById("mapholder").innerHTML = "<img src='"+img_url+"'>";}
```

HTML5 – GEOLocation (Contd.)

Output



Click the button to get your position:

HTML5 – GEOLocation (Contd.)

→ On clicking the button "Try it", getLocation() method is called

→ In this method, if the browser supports GeoLocation then current location is obtained by getCurrentPosition()

» On success, it calls success() method. In this method, an image is created with the longitude and latitude co-ordinates and this image is displayed. The latitude and longitude are specified using the Javascript

» On any error, Error() method is called to display the current issue in displaying Geo Location

We will cover JavaScript in later modules

Annie's Question

What is the practical application of geo location?

Annie's Answer

Ans. To display the office location, to display the route between two places, geo-location can be used

HTML5 – Drag and Drop

edureka!

→ When the user wants to make his task easy to copy, delete, or reorder the elements in the webpage, Drag and drop is used

→ The user will have to click and hold on the item he wants to move, and drag it to the location and release the mouse button

→ Drag and Drop is done in HTML 5 using the APIs supported whereas in HTML, the user had to write JavaScript to perform such action

HTML5 – Drag and Drop (Contd.)

edureka!

→ Two images, recycle bin and folder are displayed

→ Folder can be dragged and dropped into recycle bin

→ For the folder, when draggable=true is given, the image can be dragged

  » When the folder image is started to drag, it calls a method drag(event)

  » In this method, we are setting data type and id of the dragged image

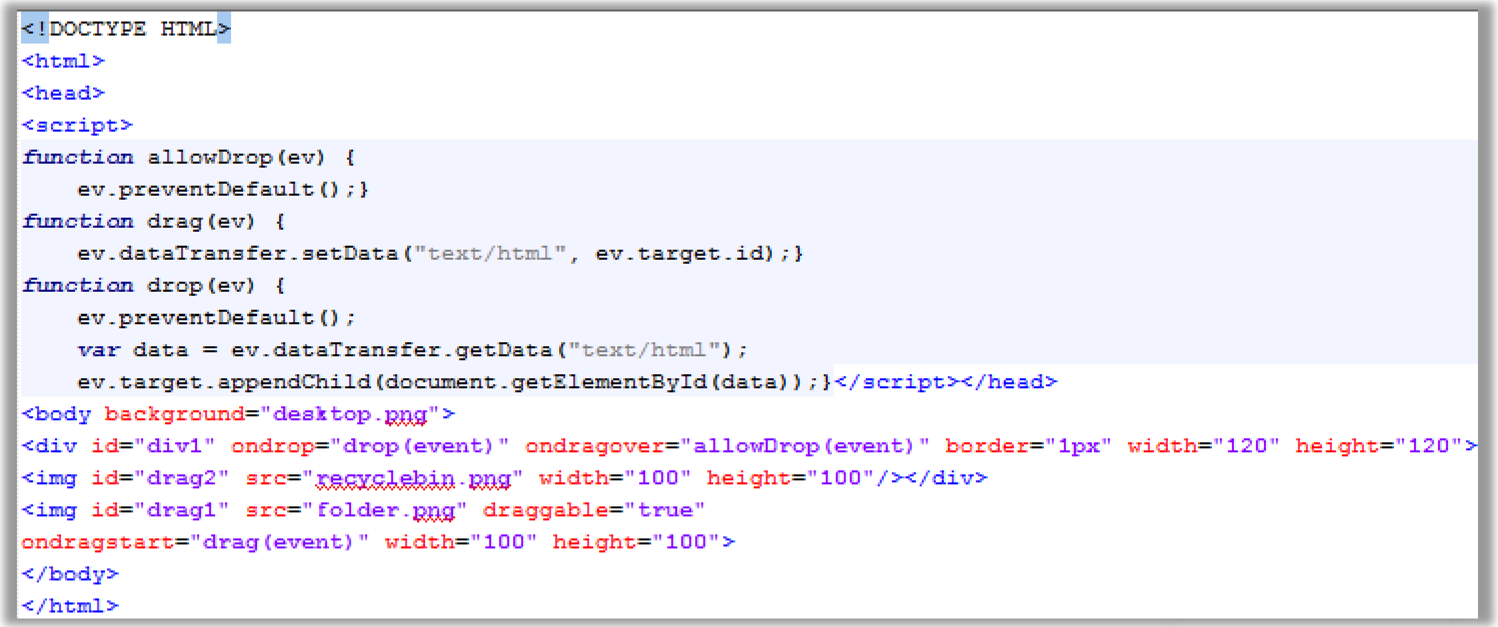
→ In our example, we are moving the folder into the recycle bin and the recycle bin is using onDragOver and OnDrop() methods

  » OnDragOver() method removes the default way of handling the elements by using the preventDefault() method

  » When folder is dropped on Recycle Bin, it calls OnDrop() which gets the data of the folder image and gets appended to recycle bin element

HTML5 – Drag and Drop (Contd.)

Example

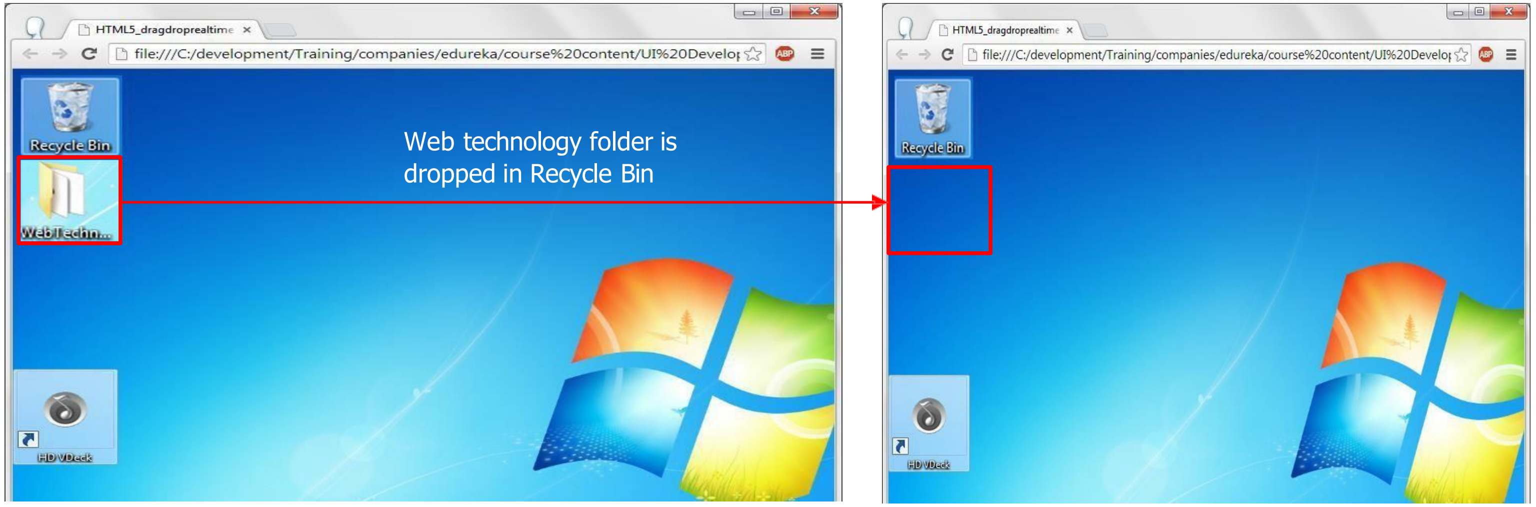


```
<!DOCTYPE HTML>
<html>
<head>
<script>
function allowDrop(ev) {
    ev.preventDefault();}
function drag(ev) {
    ev.dataTransfer.setData("text/html", ev.target.id);}
function drop(ev) {
    ev.preventDefault();
    var data = ev.dataTransfer.getData("text/html");
    ev.target.appendChild(document.getElementById(data));}</script></head>
<body background="desktop.png">
<div id="div1" ondrop="drop(event)" ondragover="allowDrop(event)" border="1px" width="120" height="120">
<img id="drag2" src="recyclebin.png" width="100" height="100"/></div>
<img id="drag1" src="folder.png" draggable="true"
ondragstart="drag(event)" width="100" height="100">
</body>
</html>
```

HTML5 – Drag and Drop (Contd.)

edureka!

Output


HTML5_dragdroprealtime
file:///C:/development/Training/companies/edureka/course%20content/UI%20Develo
Recycle Bin
WebTechn...
HD VDeck
HTML5_dragdroprealtime
file:///C:/development/Training/companies/edureka/course%20content/UI%20Develo
Recycle Bin
HD VDeck

Web technology folder is dropped in Recycle Bin

HTML5 – Local Storage

→ Before HTML5 Cookies were used to store local information. But the problem with cookies is the size limit and for each request made to the server all the cookies stored at client will be transferred to the server

→ Therefore HTML5 came up with a solution to use the localstorage

→ Here, Local Storage is used to store the data locally for the user's convenience

- » Local Storage is secure and can store large amount of information on the web browser without affecting the browser's performance
- » Local Storage is done using the 2 objects: localStorage and sessionStorage
- » Local Storage is better than cookies in terms of storage space and the data is never transferred to the server

localStorage - stores data with no expiration date
sessionStorage - stores data for one session (data is lost when the browser tab is closed)

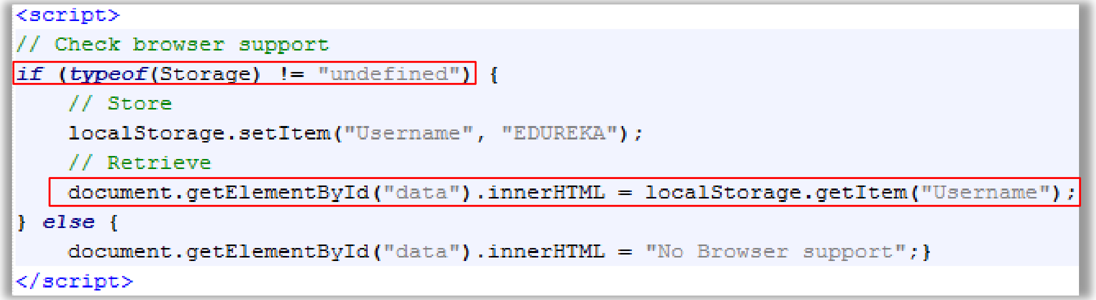
Local Storage - localStorage

edureka!

→ First check whether the browser supports for local storage by using typeof(Storage) != "undefined". If the browser supports then EDUREKA is set as Username in localStorage

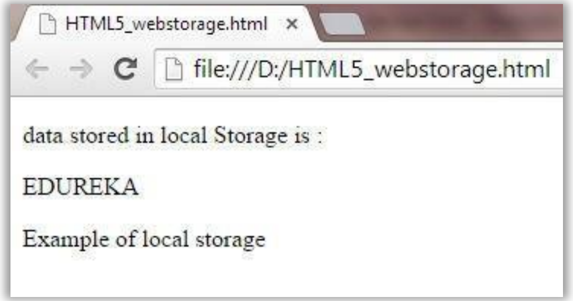
→ Once data is stored in local storage it can easily be retrieved from localStorage and used as required. You can retrieve the elements from localStorage as localStorage.getItem("element_name");

Example



```
<script>
// Check browser support
if (typeof(Storage) != "undefined") {
    // Store
    localStorage.setItem("Username", "EDUREKA");
    // Retrieve
    document.getElementById("data").innerHTML = localStorage.getItem("Username");
} else {
    document.getElementById("data").innerHTML = "No Browser support";}
</script>
```

Output



```
data stored in local Storage is :

EDUREKA

Example of local storage
```

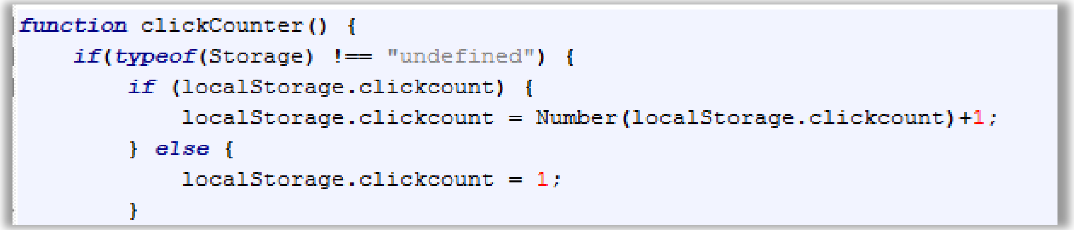
Local Storage – localStorage (Contd.)

edureka!

→ If you want to show the user how many times he has visited a particular site, localStorage should be used

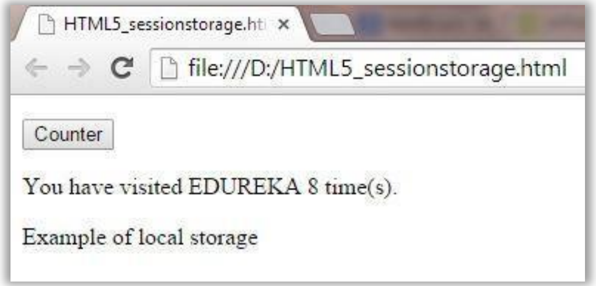
→ localStorage stores the data without the expiration date so data will not be deleted after the browser is closed

Example



```
function clickCounter() {
    if(typeof(Storage) !== "undefined") {
        if (localStorage.clickcount) {
            localStorage.clickcount = Number(localStorage.clickcount)+1;
        } else {
            localStorage.clickcount = 1;
        }
```

Output



HTML5_sessionstorage.ht ×

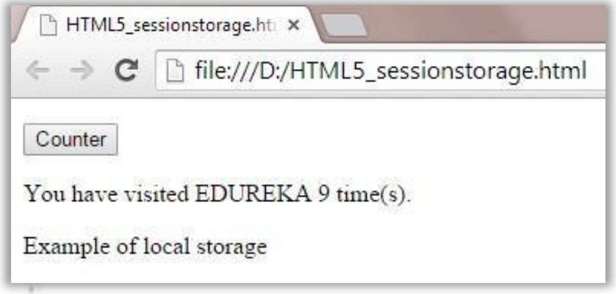
file:///D:/HTML5_sessionstorage.html

Counter

You have visited EDUREKA 8 time(s).

Example of local storage

In local storage the data will be stored without e xpira tion , so eve n after the browser is closed, the count is not affected



HTML5_sessionstorage.ht ×

file:///D:/HTML5_sessionstorage.html

Counter

You have visited EDUREKA 9 time(s).

Example of local storage

Local Storage – sessionStorage

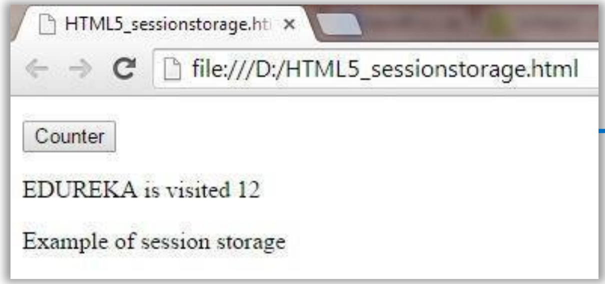
→ If you want to show the user how many times he has visited a particular site in the current session, sessionStorage should be used

→ SessionStorage stores the data until a particular session ends and the data gets deleted immediately after the browser is closed

Example

```
function Counter() {
    if(typeof(Storage) !== "undefined") {
        if (sessionStorage.Counter) {
            sessionStorage.Counter = Number(sessionStorage.Counter)+1;
        } else {
            sessionStorage.Counter = 1;         }
```

Output



HTML5_sessionstorage.ht

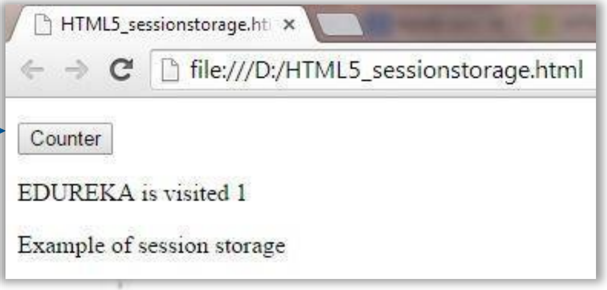
file:///D:/HTML5_sessionstorage.html

Counter

EDUREKA is visited 12

Example of session storage

After the browser is closed, the count starts from the start



HTML5_sessionstorage.ht

file:///D:/HTML5_sessionstorage.html

Counter

EDUREKA is visited 1

Example of session storage

HTML5 – Application Cache

→ HTML5 introduces application cache, which means that a web application can be cached, and accessible without an internet connection

→ It has 3 advantages:

- » A web page can be accessed even if user is offline
- » Cached pages are loaded faster
- » It reduces the load on the server as it does not request the server to display the web page

→ To use the cached web page, manifest attribute has to be specified in <html> tag as given below:

- » <html manifest="mypage.appcache">
- » Cache manifest file mypage tell the browser what to cache and what not to cache

For more information, refer: http://html5doctor.com/go-offline-with-application-cache/

Only the HTML file with manifest attribute can be cached, otherwise it cannot be cached

Application Cache – Manifest File

edureka!

→ The manifest file has three sections:

→ CACHE MANIFEST - Files listed under this section will be cached after they are downloaded for the first time

→ NETWORK - Files listed under this section require a connection to the server, and will never be cached

→ FALLBACK - Files listed under this section specifies fallback pages if a page is inaccessible

Below is an example of Manifest file

```
CACHE MANIFEST
/main.css
/background.jpg
/scripts.js

NETWORK:
login.jsp

FALLBACK:
/html/ /offline.html
```

HTML5 – Web Worker

→ JavaScript runs in a single threaded environment. While JavaScript code is executed, the user will not be able to perform any operation(clicking or selecting an element) in the web page

→ To avoid this problem, the Javascript is embedded in the HTML script and the Javascript code runs in background so that foreground web page can be accessed as usual

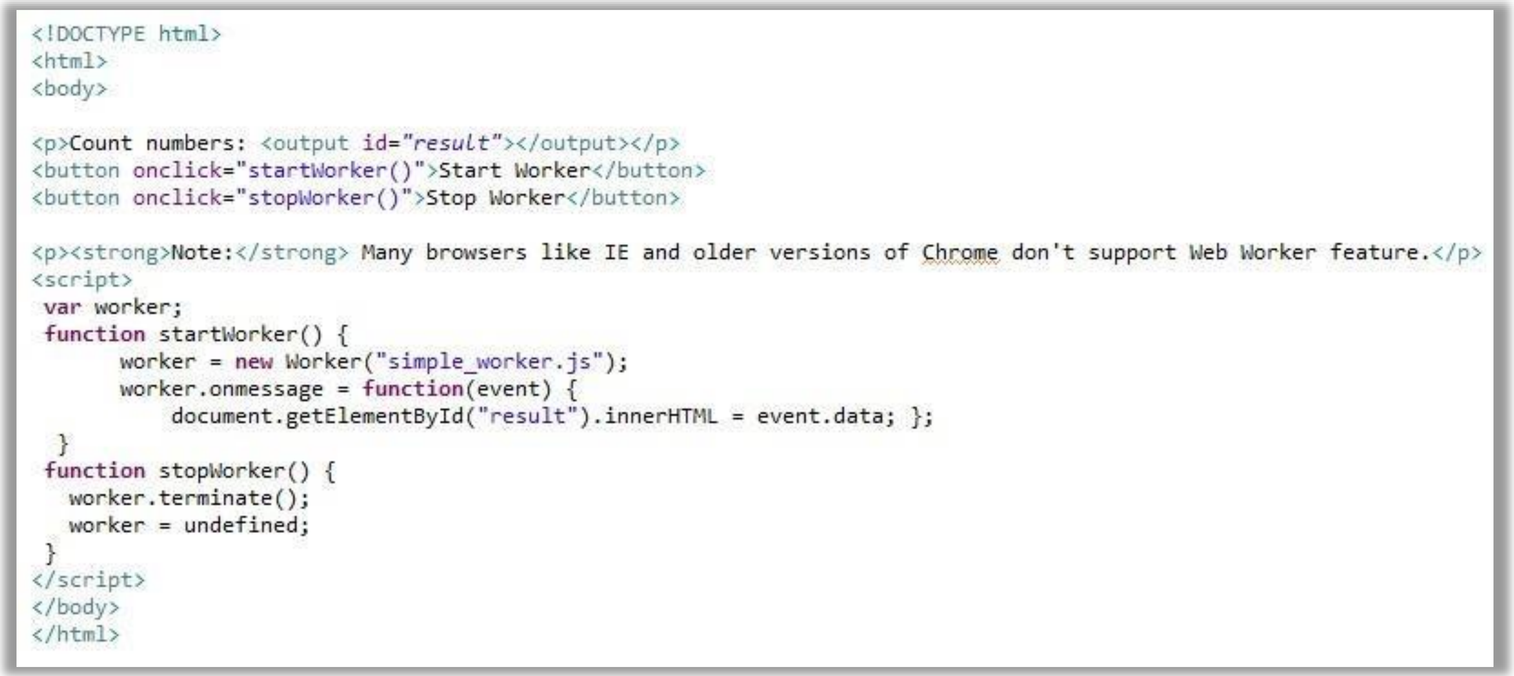
→ Web Workers are responsible to run Javascript code in the background

→ A web worker is a task running in the background which is written in JavaScript without affecting the performance of the page being displayed

→ Users can continue to do whatever they want while web worker runs in background

Since web workers are in external files, they do not have access to JavaScript objects window, document and parent

HTML5 – Web Worker



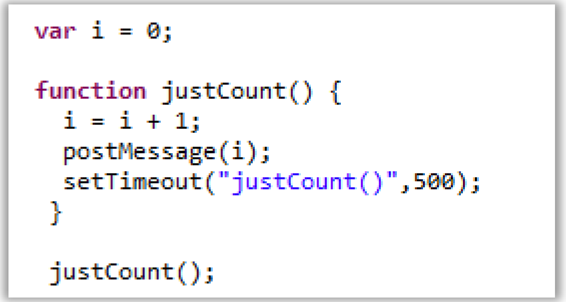
```
<!DOCTYPE html>
<html>
<body>

<p>Count numbers: <output id="result"></output></p>
<button onclick="startWorker()">Start Worker</button>
<button onclick="stopWorker()">Stop Worker</button>

<p><strong>Note:</strong> Many browsers like IE and older versions of Chrome don't support Web Worker feature.</p>
<script>
 var worker;
 function startWorker() {
        worker = new Worker("simple_worker.js");
        worker.onmessage = function(event) {
            document.getElementById("result").innerHTML = event.data; };
  }
 function stopWorker() {
   worker.terminate();
   worker = undefined;
 }
</script>
</body>
</html>
```

web-worker.html

HTML5 – Web Worker

edureka!

→ In our external JavaScript file (simple_worker.js) we have defined a function justCount() which will be called after 500 milliseconds



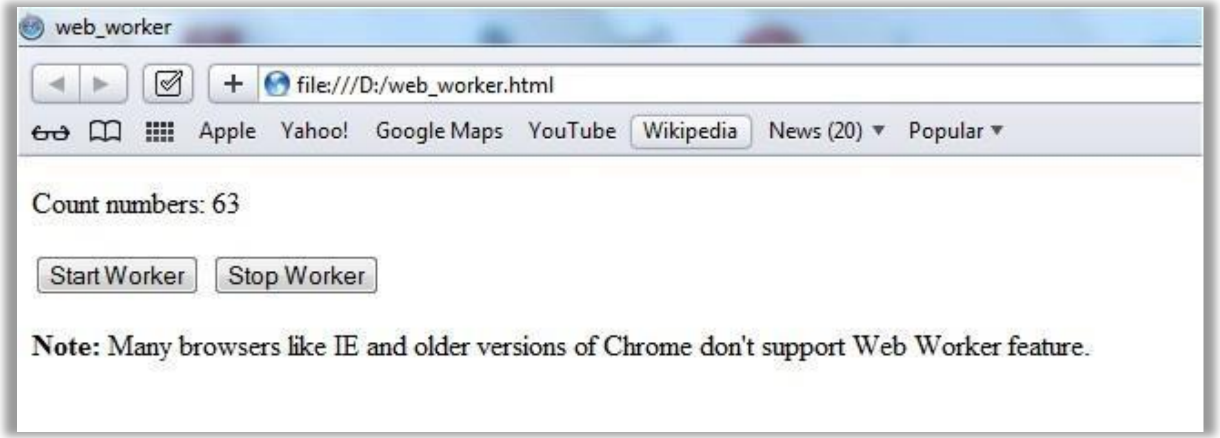
```
var i = 0;

function justCount() {
  i = i + 1;
  postMessage(i);
  setTimeout("justCount()",500);
 }

 justCount();
```

simple_worker.js

postMessage() method is used to post the message back to the HTML page

HTML5 – Web Worker (Contd.)

edureka!



Count numbers: 63

Note: Many browsers like IE and older versions of Chrome don't support Web Worker feature.

Output in Safari

Web workers should not be used in large numbers as they could hog system resources. Usualy web workers are used to do computational tasks like finding large prime numbers, showing timers, file compression etc.

Annie's Question

What is the advantage of Web Worker?

Annie's Answer

Ans. Time to load the webpage is faster and UI is not affected by background work of Web Worker

HTML5 – Server Sent Events

edureka!

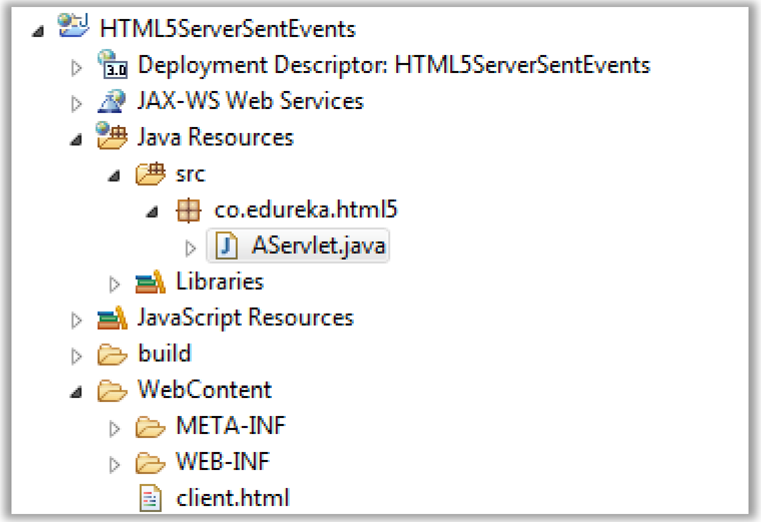
- → SSE (Server Sent Events) are the updates sent by server to the browser or the client
- → Web page should be open to receive the events or messages from the server
- → Some of the examples of SSE are Facebook updates, Sports results, stock price updates etc.
- → EventSource object is used to receive server-sent event notifications
- → Each time a message is received from the server, onmessage event occurs
- → JavaScript handles the onmessage() event and performs the required operation

EventSource object can handle following events

onopen	When a connection to the server is opened
Onmessage	When a message is received
onerror	When an error occurs

HTML5 – Server Sent Events Example

→ We are going to build an application where html page will receive events from the server side component Servlet



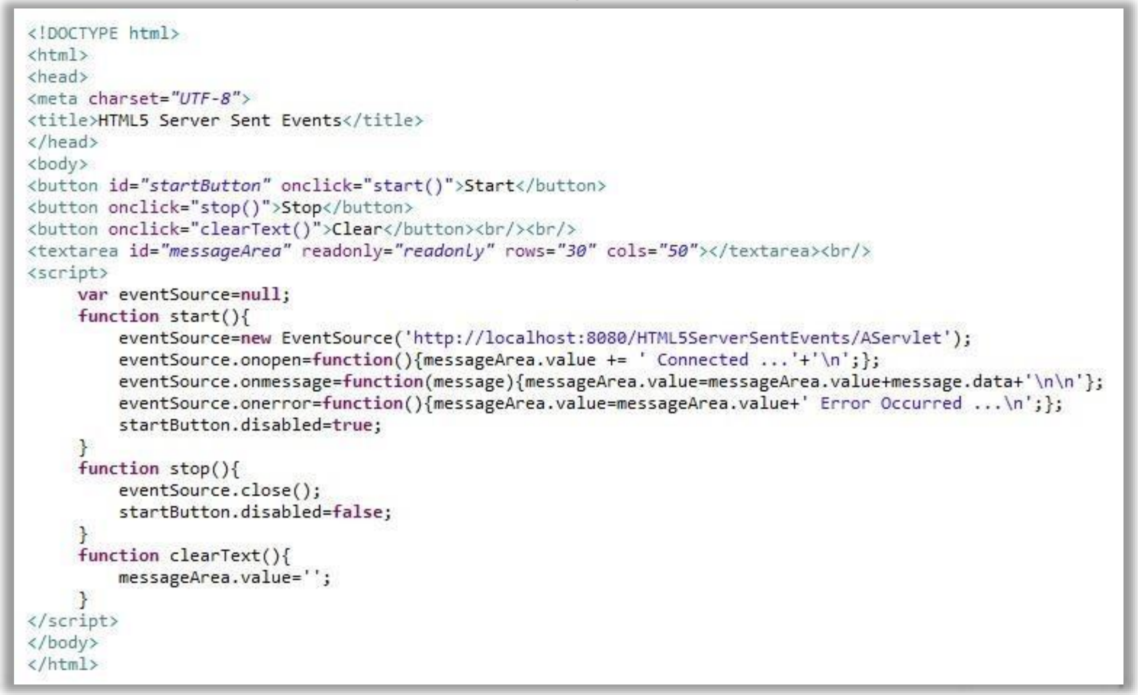
```
◢ HTML5ServerSentEvents
    ▷ Deployment Descriptor: HTML5ServerSentEvents
    ▷ JAX-WS Web Services
    ◢ Java Resources
        ◢ src
            ◢ co.edureka.html5
                ▷ AServlet.java
        ▷ Libraries
    ▷ JavaScript Resources
    ▷ build
    ◢ WebContent
        ▷ META-INF
        ▷ WEB-INF
          client.html
```

Project Structure

Server side events can be sent from any server side technology like PHP, ASP, JSP, Servlet etc.

HTML5 – Server Sent Events Example (Contd.)

→ Note that in client.html we have configured the servlet as the event source

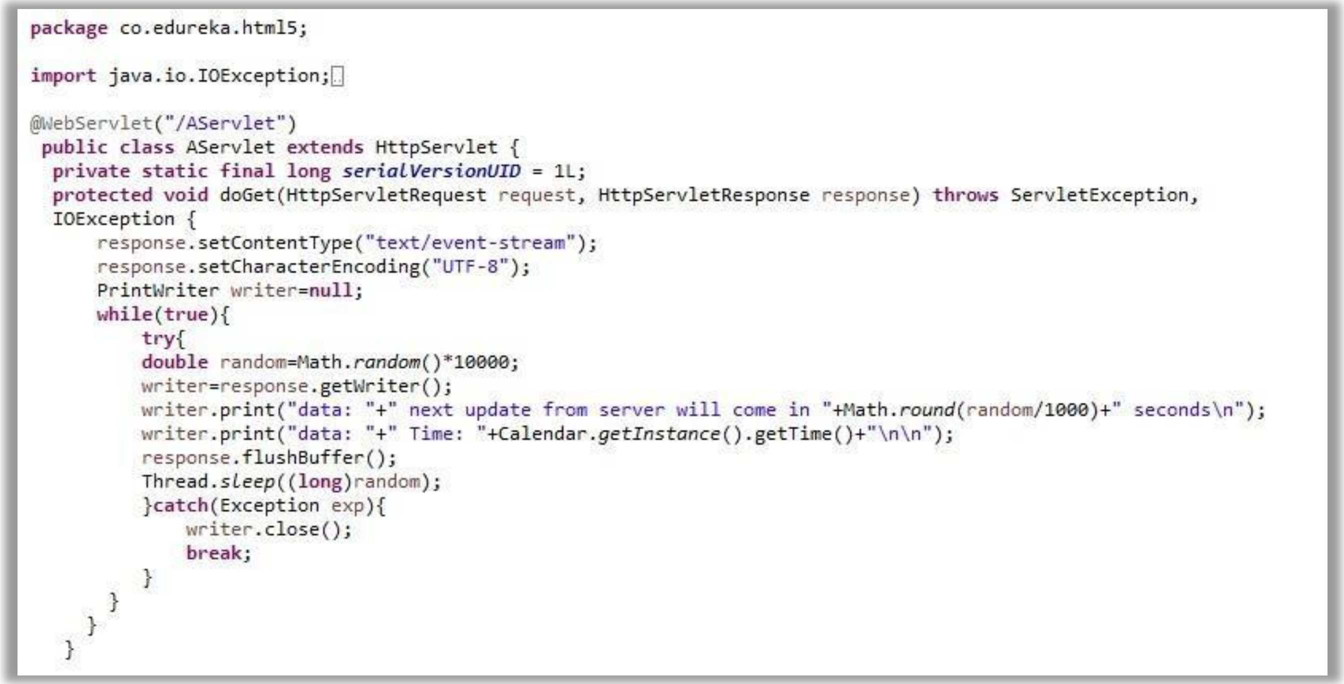


```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>HTML5 Server Sent Events</title>
</head>
<body>
<button id="startButton" onclick="start()">Start</button>
<button onclick="stop()">Stop</button>
<button onclick="clearText()">Clear</button><br/><br/>
<textarea id="messageArea" readonly="readonly" rows="30" cols="50"></textarea><br/>
<script>
     var eventSource=null;
     function start(){
         eventSource=new EventSource('http://localhost:8080/HTML5ServerSentEvents/AServlet');
         eventSource.onopen=function(){messageArea.value += ' Connected ...'+'\n';};
         eventSource.onmessage=function(message){messageArea.value=messageArea.value+message.data+'\n\n'};
         eventSource.onerror=function(){messageArea.value=messageArea.value+' Error Occurred ...\n';};
         startButton.disabled=true;
     }
     function stop(){
         eventSource.close();
         startButton.disabled=false;
     }
     function clearText(){
         messageArea.value='';
     }
</script>
</body>
</html>
```

client.html

HTML5 – Server Sent Events Example (Contd.)

→ We have used an infinite while loop in the doGet() which prints two lines of text at random interval of time



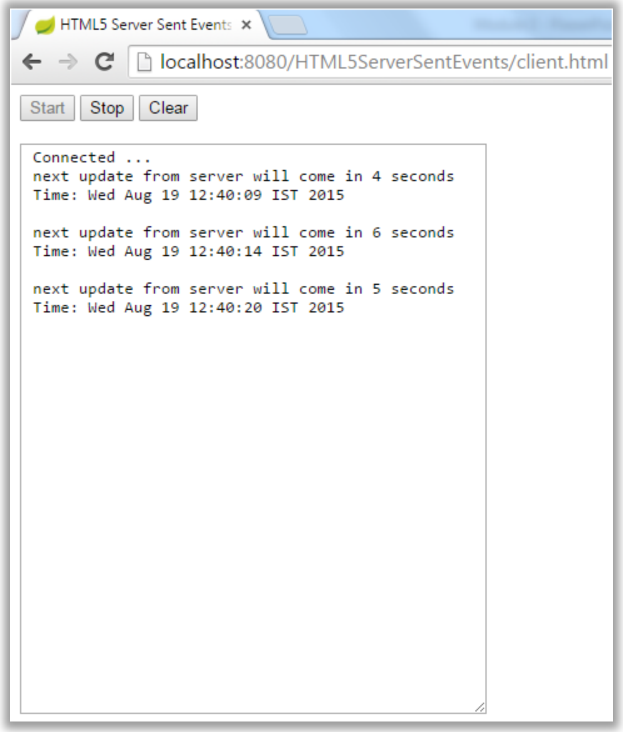
```
package co.edureka.html5;

import java.io.IOException;

@WebServlet("/AServlet")
 public class AServlet extends HttpServlet {
  private static final long serialVersionUID = 1L;
  protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException,
  IOException {
      response.setContentType("text/event-stream");
      response.setCharacterEncoding("UTF-8");
      PrintWriter writer=null;
      while(true){
          try{
          double random=Math.random()*10000;
          writer=response.getWriter();
          writer.print("data: "+" next update from server will come in "+Math.round(random/1000)+" seconds\n");
          writer.print("data: "+" Time: "+Calendar.getInstance().getTime()+"\n\n");
          response.flushBuffer();
          Thread.sleep((long)random);
          }catch(Exception exp){
              writer.close();
              break;
          }
      }
   }
 }
```

AServlet.java

We have set the response content type to text/event-stream and prepended the response content with data: which is mandatory for, the clients to retrieve the server response

HTML5 – Server Sent Events Example (Contd.)

→ Access the client.html and click on start button to start getting events from the Servlet



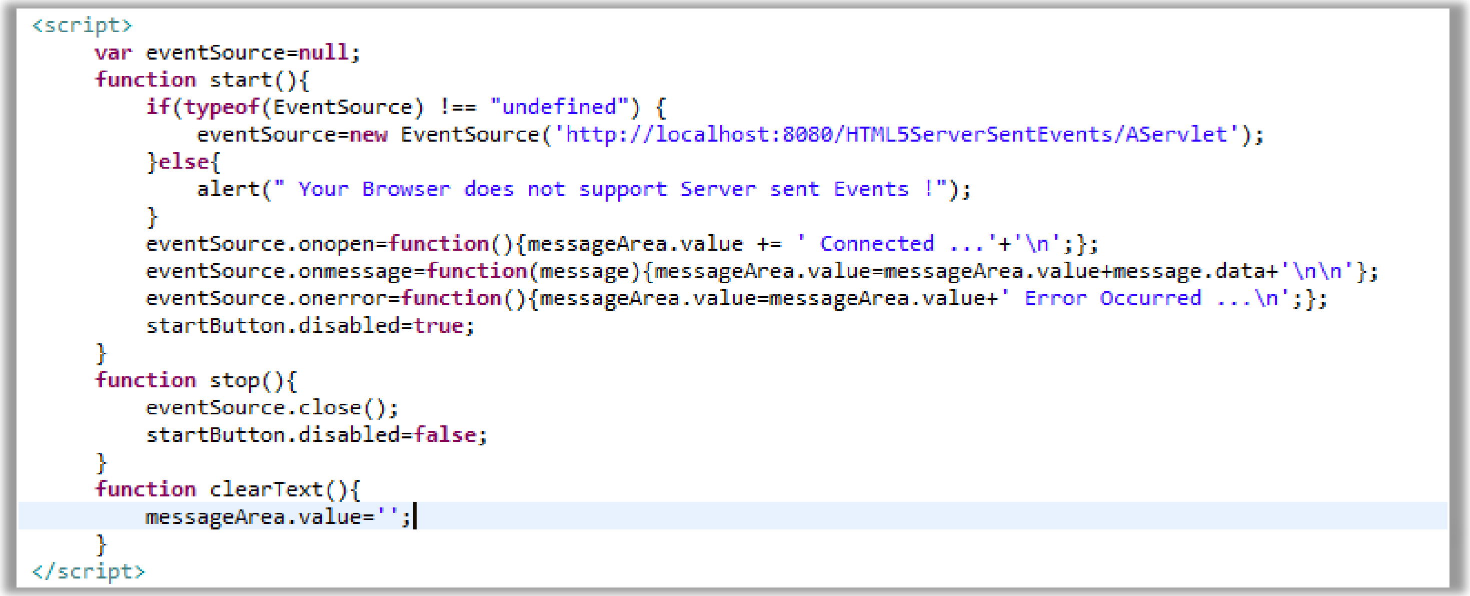
```
Connected ...
next update from server will come in 4 seconds
Time: Wed Aug 19 12:40:09 IST 2015

next update from server will come in 6 seconds
Time: Wed Aug 19 12:40:14 IST 2015

next update from server will come in 5 seconds
Time: Wed Aug 19 12:40:20 IST 2015
```

Output

Client.html gets events from the servlet at random times as shown in the output

HTML5 – Server Sent Events Example (Contd.)

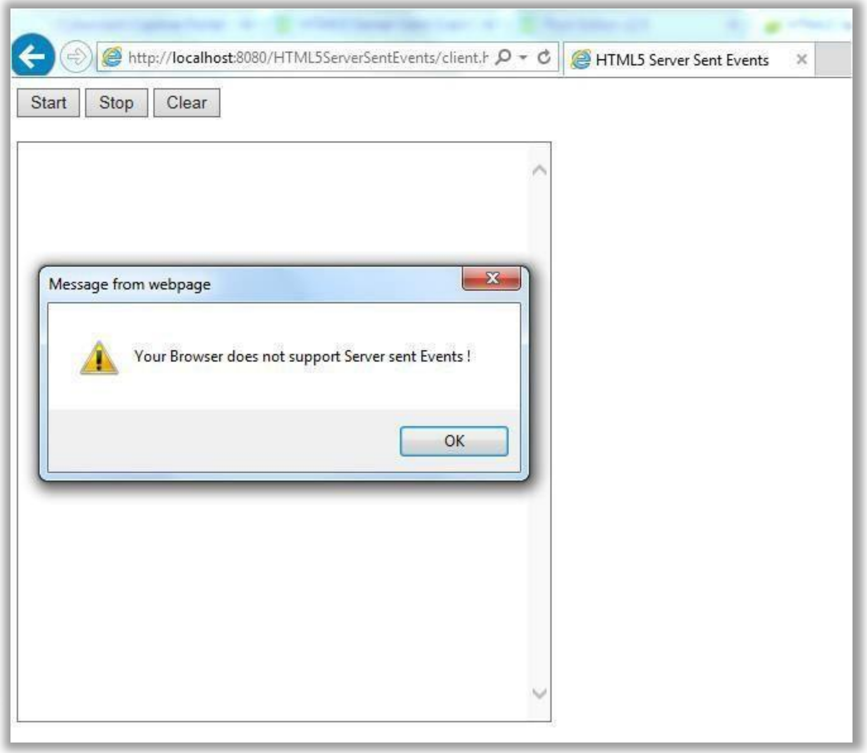
→ Its always good to first check whether browser supports the server sent events feature or not. Internet Explorer does not support Server Sent Event feature. If browser supports the feature then only EventSource object can be created otherwise error message should be displayed



```
<script>
     var eventSource=null;
     function start(){
         if(typeof(EventSource) !== "undefined") {
             eventSource=new EventSource('http://localhost:8080/HTML5ServerSentEvents/AServlet');
         }else{
             alert(" Your Browser does not support Server sent Events !");
         }
         eventSource.onopen=function(){messageArea.value += ' Connected ...'+'\n';};
         eventSource.onmessage=function(message){messageArea.value=messageArea.value+message.data+'\n\n'};
         eventSource.onerror=function(){messageArea.value=messageArea.value+' Error Occurred ...\n';};
         startButton.disabled=true;
     }
     function stop(){
         eventSource.close();
         startButton.disabled=false;
     }
     function clearText(){
         messageArea.value='';
     }
</script>
```

client.html

HTML5 – Server Sent Events in IE

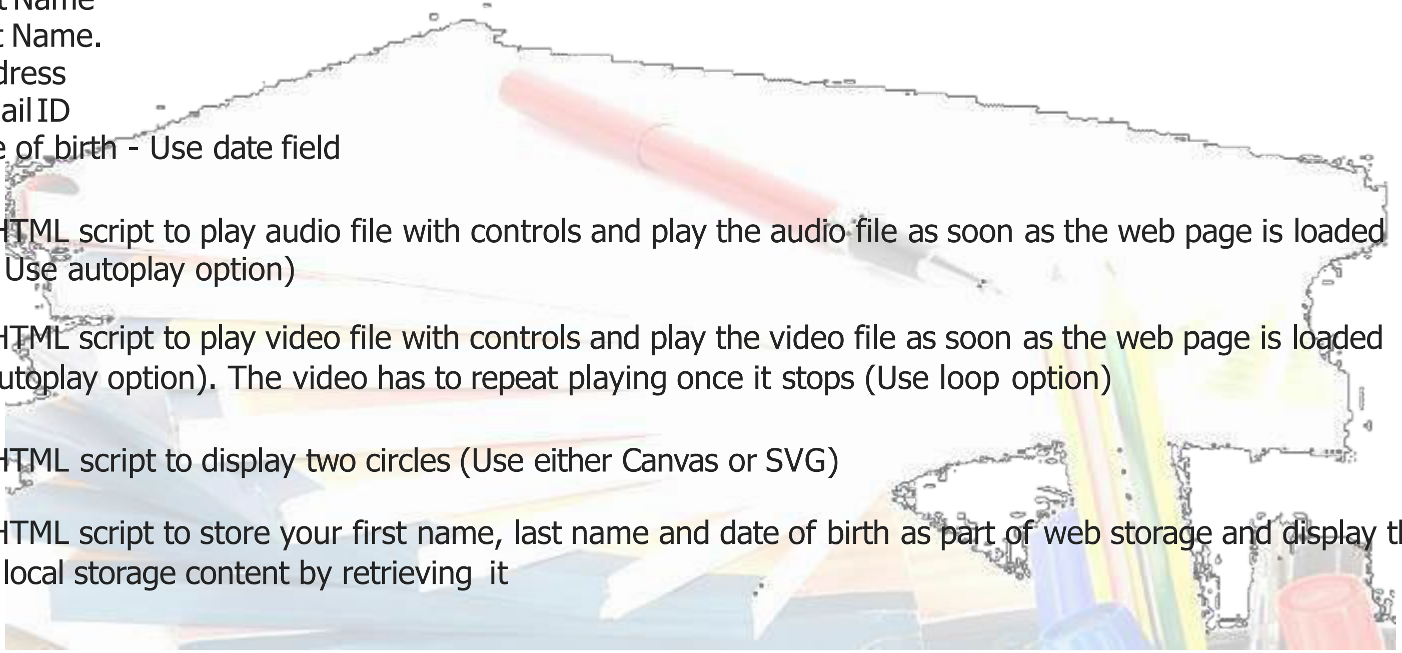


Your Browser does not support Server sent Events !

Internet Explorer does not support Server sent events so an appropriate error message is shown on clicking the start button

Questions

Assignment



→ Display the web page with “USER INFORMATION” as heading using the <h1> tag and include the following fields as the form fields:

» First Name
» Last Name.
» A ddress
» Email ID
» date of birth - Use date field

→ Write HTML script to play audio file with controls and play the audio file as soon as the web page is loaded (Note: Use autoplay option)

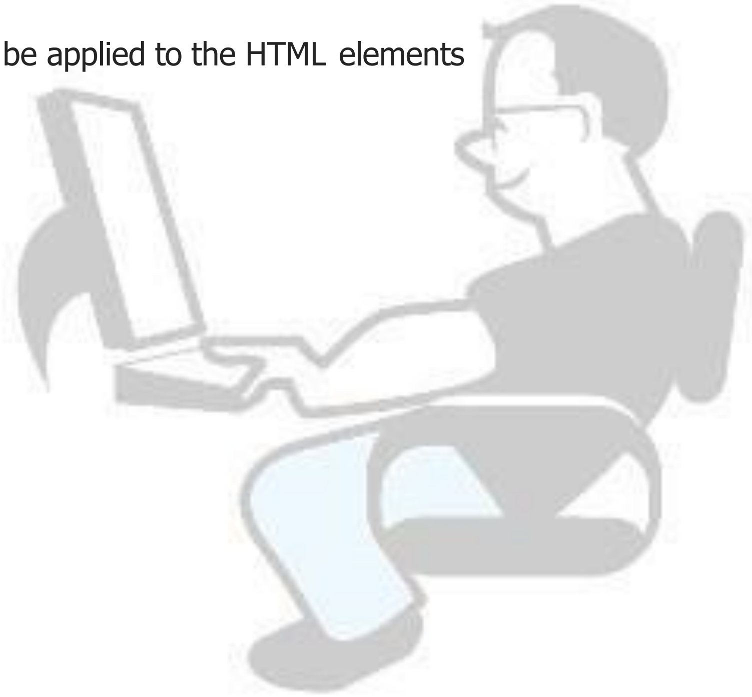
→ Write HTML script to play video file with controls and play the video file as soon as the web page is loaded (Use autoplay option). The video has to repeat playing once it stops (Use loop option)

→ Write HTML script to display two circles (Use either Canvas or SVG)

→ Write HTML script to store your first name, last name and date of birth as part of web storage and display the stored local storage content by retrieving it

Pre-work

- Go through the various attributes supported in CSS
- The different styles that can be applied to the HTML elements



Further Reading

→ http://ionicframework.com/html5-input-types/

→ http://diveintohtml5.info/canvas.html

→ http://www.arungudelli.com/html5/html5-local-storage/

→ https://github.com/Atmosphere/atmosphere/wiki/Getting-started-with-HTML5-Server-Sent-Events-(SSE)

Agenda for the next class

→ Understand the importance of selectors in the webpage

→ How to style lists and tables in CSS

→ Understand the box model in CSS

Survey

Your feedback is important to us, be it a compliment, a suggestion or a complaint. It helps us to make the course better!

Please spare few minutesto take the survey after the webinar.

Thank you!