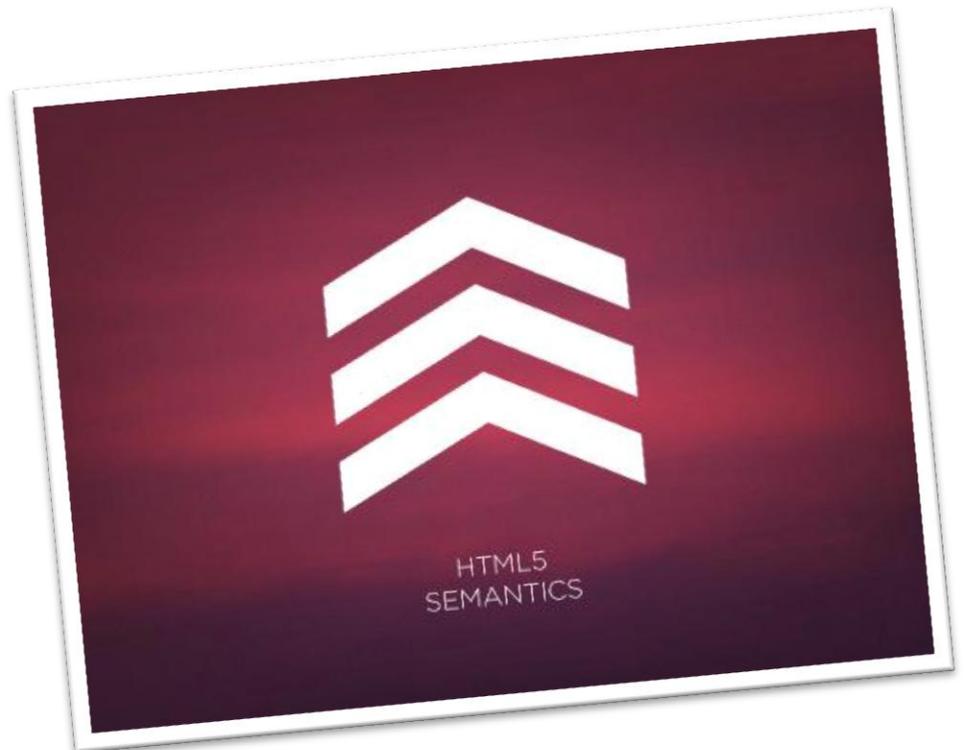




A QUICK START GUIDE FOR: HTML 5

- Get Started
- Learn New Elements
- Tools we use
- Learn HTML 5
- Start Building
- Building your first web-page
- Become a Pro



A quick start guide to HTML 5

Lavish Thakkar

[The Geeks Club](#)

Authors Note



Contact:

E-mail: lavishthakkar@ymail.com

Facebook: [lavish.thakkar](https://www.facebook.com/lavish.thakkar)

Twitter: [@lavishthakkar](https://twitter.com/lavishthakkar)

Skype: [lavish.thakkar1](https://www.skype.com/people/lavish.thakkar1)

Hi everyone. First of all, a great thanks to all of you for reading this book. I would like to tell you something about this book; this is actually something I recommend you for start learning HTML 5. HTML 5 is now a day's popular language for designing beautiful, awesome and a highlighted webpage. This book is not a complete and official guide to HTML 5, but this book would definitely guide you to you first steps to start developing HTML 5 pages. When you are developing or you have learned how to develop with HTML 5, you can discover more and create more.

Before starting reading this book I wanted to tell you that, you should have a little bit of knowledge of previous versions of HTML 5 and a very little knowledge of Java Script.

Once again, robust thanks for reading this book, hope you would like it.

-Llavish Thakkar

Contents

1. Introduction.....	5
2. The New Elements.....	7
3. Getting Started.....	9
4. Working with media components.....	12
5. Drag and drop functionality.....	16
6. HTML 5 Canvas.....	20
7. Scalable Vector Graphics.....	23
8. HTML 5 Geo Locations.....	25
9. Local Web Storage.....	27
10. Web Workers.....	33
11. HTML 5 Forms.....	35
12. Designing Forms.....	39
13. Building your first web page.....	42
14. Getting Certified.....	44
15. Answers.....	45

1. Introduction

When we say 'HTML', we mean HYPER TEXT MARKUP LANGUAGE. HTML is basically a language used for web designing and constructing the Web Sites that are presented all over the World Wide Web. HTML was originally projected and developed by Opera Software. HTML 5 is the fifth amendment of HTML language. The basic HTML standard was created in 1990 and HTML 4, the last version before HTML 5 was released and created in 1997.

HTML 5 is still under development, it is developed keeping in mind the all new media necessities and the high-end graphics need of WWW. According to me, HTML 5 will incorporate the previous versions of HTML. HTML 5 has come up with lots of new tags, elements and new features that we would discuss in this book later.



Like the previous versions of HTML, HTML5 is also written in text format and then saved in **".html"** or **".htm"** format.

Anyone can learn HTML as it is an easy language, while you go through this book, you will see many tutorials, from them you would learn and finally you would be ready to go with your HTML skills. The future of the web is HTML 5 so I think we need to have knowledge about HTML 5 as the future web would be based on HTML 5.

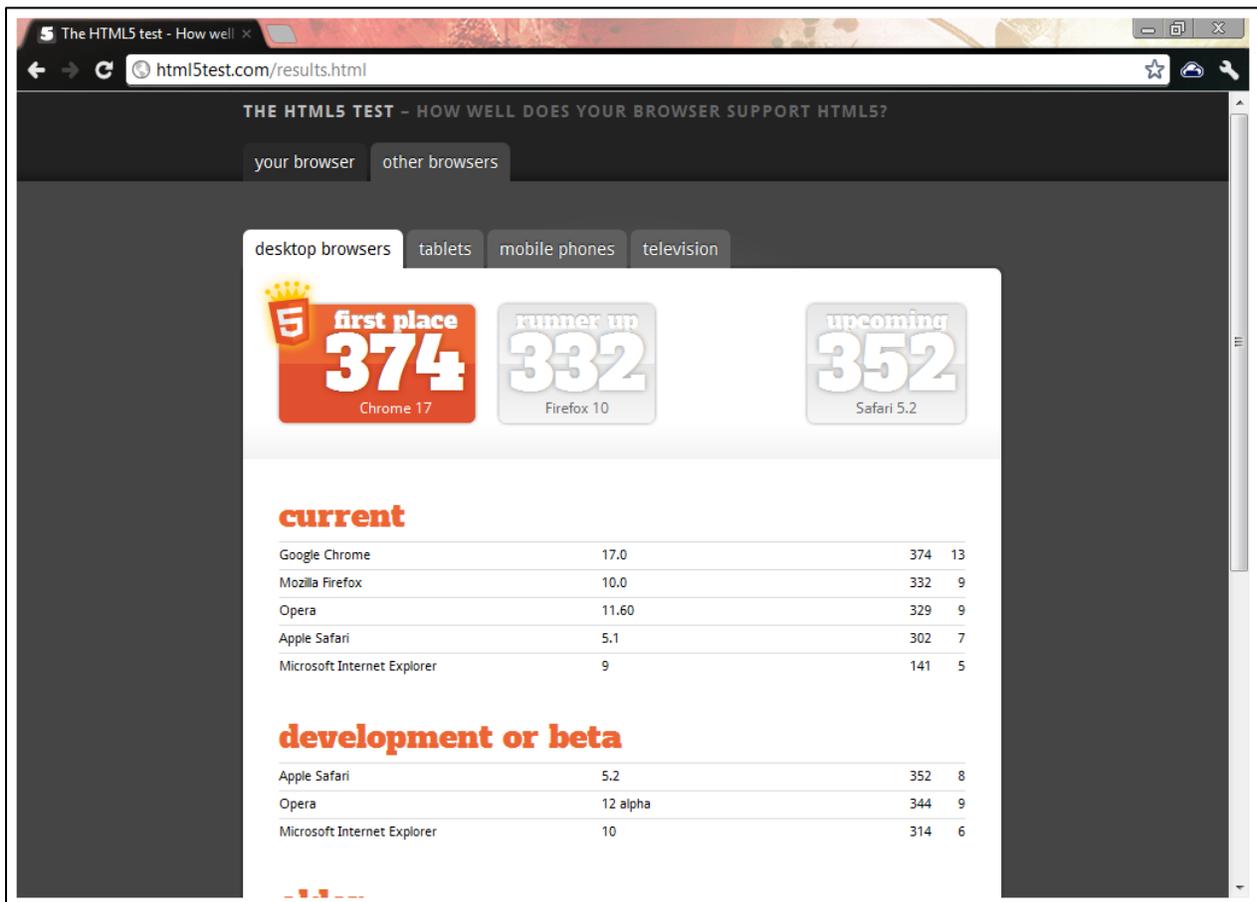
According to reports and some surveys conducted, the top websites are using HTML 5 to provide better web experience to their users.

In this book, I will teach you step by step, how to program with HTML 5, while you read the book always keep a notepad window opened so that you can direct away check the code and test it if you want. I would recommend you to use "[Notepad ++](#)", it is free and open source text editor.

1.1 HTML 5 Supports

As the HTML 5 is new comer in the web designing languages, some browsers does not support some components of HTML 5, but if you want a real full support, then you must try Opera, as already told HTML 5 was created by Opera Software, they try to put the best out of HTML 5 in their browser.

In the HTML 5 support tests done by www.html5test.com, they've awarded Chrome as the winner in HTML 5 support and Firefox after it, you can even check whether your browser supports HTML 5 or not by just clicking on the link above.



2. The New Elements

The web has changed a lot since 1997, when the last version of HTML was launched. Now the users have different requirements than earlier, now we feel like having more media on page rather than information, our first sight on a website is its looks. We are not likely to visit ugly or bad-looking websites. So to meet the new requirements of users, there are lots of new things introduced in HTML 5.

2.1 New Rules

Before we go through the new tags and elements, we should know about the new rules introduced in HTML 5 so that we can avoid mistakes and can follow the rules:

- New applications to be based on HTML, CSS, DOM, and JavaScript
- Applications should be device independent
- The development to be visible to public

So, now we know the rules, let's know about the new elements.

2.2 New Elements

Now coming to the new elements of HTML 5, there are some new structural, media, form and canvas elements introduced and even some of the elements were removed from HTML, so let's see what is new and what is removed.

2.2.1 New Structural Elements

Tag	Short Description
<code><article></code>	Used for defining an Article
<code><aside></code>	Defines content that is aside from the page content
<code><command></code>	For creating a new command button
<code><details></code>	Add some additional details that a user can view or hide
<code><summary></code>	Visible heading for a <code><details></code> element
<code><figure></code>	Shows self-contained content
<code><figcaption></code>	Caption for <code><figure></code> element
<code><footer></code>	Used to insert a footer in the document or page
<code><header></code>	Used to insert a header in the document or page
<code><hgroup></code>	Heading levels starting from H1 to H6
<code><mark></code>	Used to mark or highlight the text
<code><meter></code>	Scalar measurement
<code><nav></code>	Used to defines navigation links
<code><progress></code>	To show the progress of a task
<code><ruby></code>	Ruby Annotation

<code><rt></code>	Explanation/pronunciation of characters
<code><rp></code>	What to show in browsers that do not support ruby annotations
<code><section></code>	Section in a document
<code><time></code>	Used to insert date/time
<code><wbr></code>	Line-break

These were all the new structural elements introduced in HTML 5; now let's talk about some New Media Elements in HTML 5.

2.2.2 New Media Elements

Tag	Short Description
<code><audio></code>	Inserts sound content
<code><video></code>	Inserts video or movie
<code><source></code>	Inserts multiple media resources for <code><video></code> and <code><audio></code>
<code><embed></code>	Used to embed some thing

2.2.3 New Form Elements

Tag	Description
<code><datalist></code>	List of pre-defined options for input controls
<code><keygen></code>	Key-pair generator field
<code><output></code>	The result of a calculation

2.2.4 The “`<canvas>`” Element

You would love this element; this tag is basically used to draw graphics by codes, the graphics are displayed via Java Script.

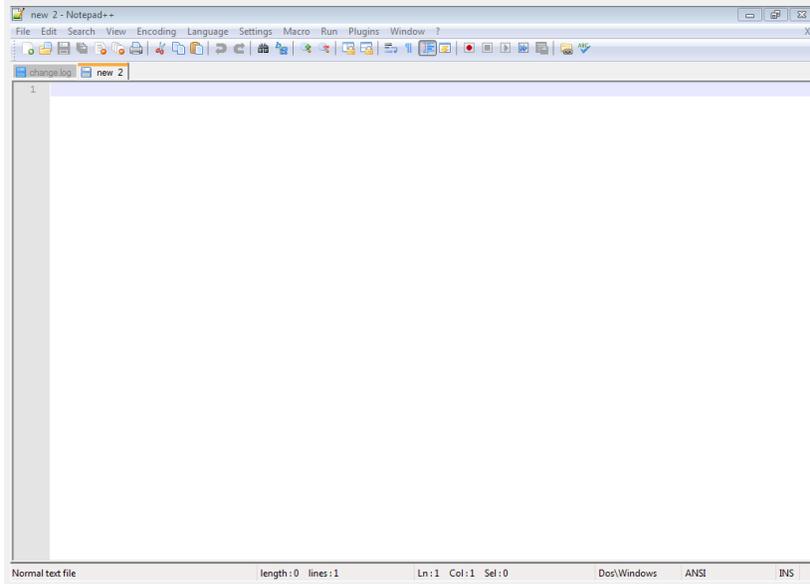
2.2.5 Removed Elements

Now we have read about the new elements now just have a look on the removed elements:

- `<acronym>`
- `<applet>`
- `<basefont>`
- `<big>`
- `<center>`
- `<dir>`
- ``
- `<frame>`
- `<frameset>`
- `<noframes>`
- `<strike>`
- `<tt>`
- `<u`

3. Getting Started

So, now we have discussed a lot about HTML 5, now we are on our way to create our first webpage, hope you have chosen a text editor for you that can save files in HTML format, I recommend you Notepad ++, download it, it's free and open source.



To get started, we will look to a simple code to understand how to start with HTML 5:

```
<!DOCTYPE html>
<html>
<head>
<title>Title for your page</title>
</head>
<body>
The content goes here...
</body>
</html>
```

As we earlier started the code with <html> tag, but in HTML 5 <! DOCTYPE html> tag is necessary, we can't start our code without that, this was the first change in coding we noticed till now. So let's develop our first Web Page with HTML 5.

1. Copy the code from above and paste it on your Notepad or text editor.
2. Change the title if you want and insert something in the body if you want. For e.g. we will create a hello world page, so have look on the code.

```
<!DOCTYPE html>

<html>

<head>

<title>Hello World</title>

</head>

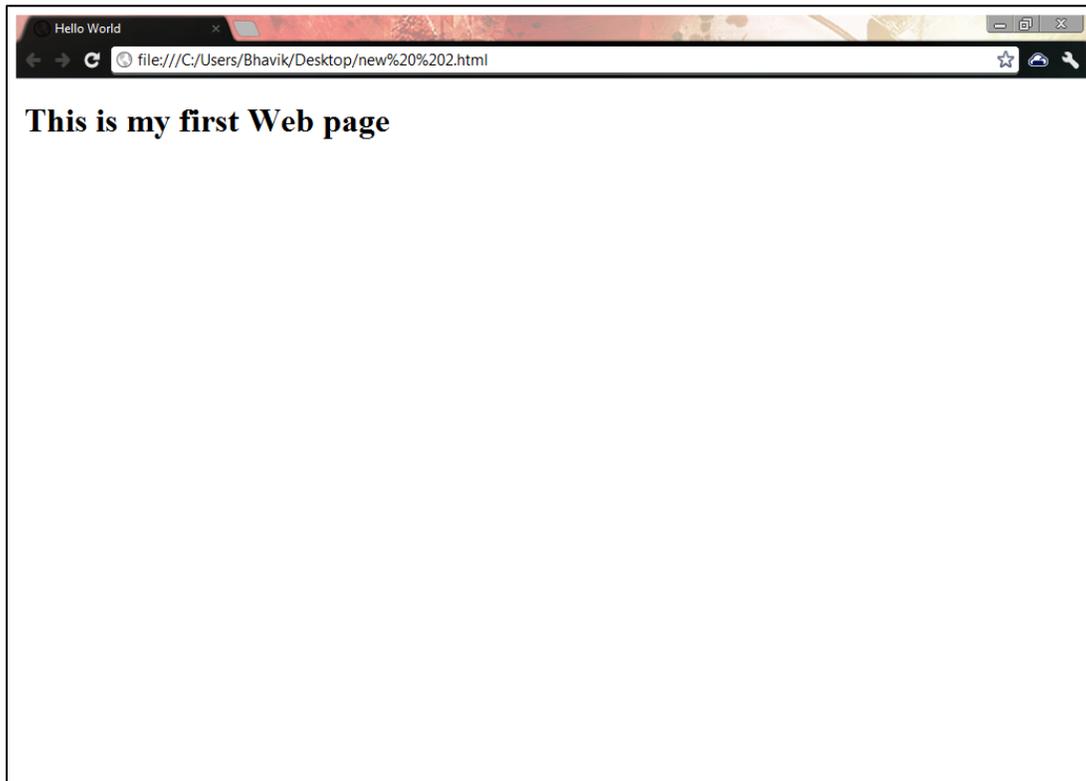
<body>

<h1> This is my first Web page </h1>

</body>

</html>
```

3. Now save the file into “.html” extension and preview the saved file in your browser. It should look like this:



So, you have created your first webpage and learnt how to get started with HTML 5 so now it's time for a short exercise.

Exercise 3.1

1. Create a HTML page with the title as your name and in the body insert your name, age and your street address and try to view it in your browser.
2. Try some tags of previous version HTML tags and explore more in HTML 5.

If you face any problems creating any HTML document, you can feel free to contact me for help at: lavish.thakkar@hotmail.com

4. Working with media components



Earlier versions of HTML 5 didn't supported videos or audios directly, we took help of YouTube and some other services, but now we don't need to worry as HTML 5 natively supports Video and Audio with a pre default player in your browser.

Before starting with media components, let's check which browser supports which format of the movie or audio clip.

Video Formats:

There are 3 video formats that can be visualized with HTML 5; those are MP4, WebM and Ogg.

Browser	MP4	WebM	Ogg
Internet Explorer	YES	NO	NO
Firefox	NO	YES	YES
Google Chrome	YES	YES	YES
Apple Safari	YES	NO	NO
Opera	NO	YES	YES

Audio Formats:

There are 3 audio formats that can be embedded with HTML 5; those are MP3, Wav and Ogg.

Browser	MP3	Wav	Ogg
Internet Explorer	YES	NO	NO
Firefox	NO	YES	YES
Google Chrome	YES	YES	YES
Apple Safari	YES	YES	NO
Opera	NO	YES	YES

4.1 Video components

For inserting a video into a Web Page `<video>`, `<source>` and `<track>` tags are used, for e.g.

```
<!DOCTYPE html>
<html>
<head>
<title>Title Here</title>
</head>
<body>
<video width="320" height="240" controls="controls">
  <source src="movie.mp4" type="video/mp4" />
  Video Not Supported
</video>
</body>
</html>
```

In the above example, you can replace "movie.mp4" with you own movie link but remember video or movie should be in the category of supported media formats in HTML 5. You can change your video width and height if you want and is you don't want controls on your video you can delete the part where it says `controls="controls"`.

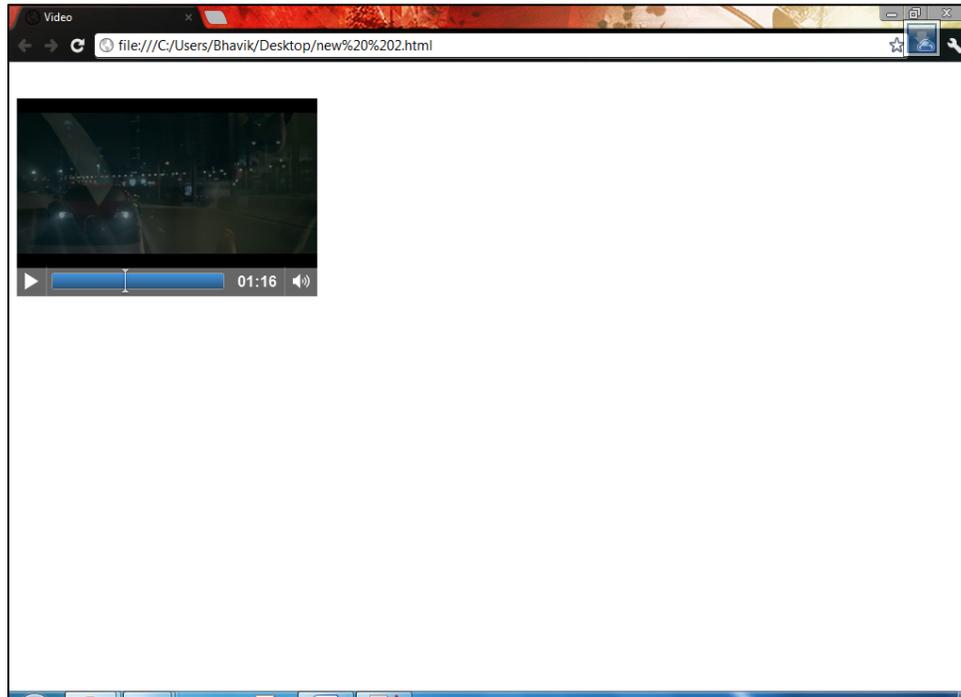
If you want to give multiple sources or multiple formats to a video, so that your video runs on different browsers, you can insert more source tags for e.g.

```
<video width="320" height="240" controls="controls">
  <source src="movie.mp4" type="video/mp4" />
  <source src="movie.ogg" type="video/ogg" />
  Video Not Supported
</video>
```

In the above example we have inserted one more media source that is in OGG format, now the video would run on the browsers that support MP4 or OGG videos. Like same method you can insert more sources, but remember the video should be the same, formats may vary. You might see in the code that we've written "Video Not Supported";

basically this is a message to convey to the user if video is not supported on the browser of the user, you can edit the message if you want.

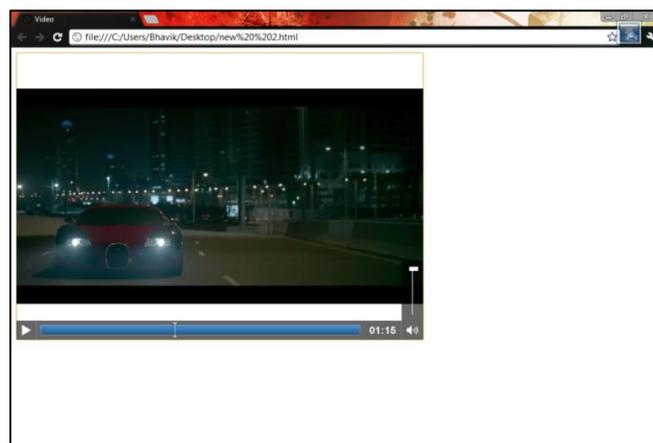
Your video should look like this:



Now you have seen how to insert videos on the page, now time for an exercise.

Exercise 4.1

1. Upload a video to any free hosting site (e.g. 4Shared.com) and create a webpage with that video inserted in it and the height of the video should 480 and width to be 640, give an error message of your choice.
2. Repeat question one and now insert 3 different video sources or video formats of your video and try to view it.

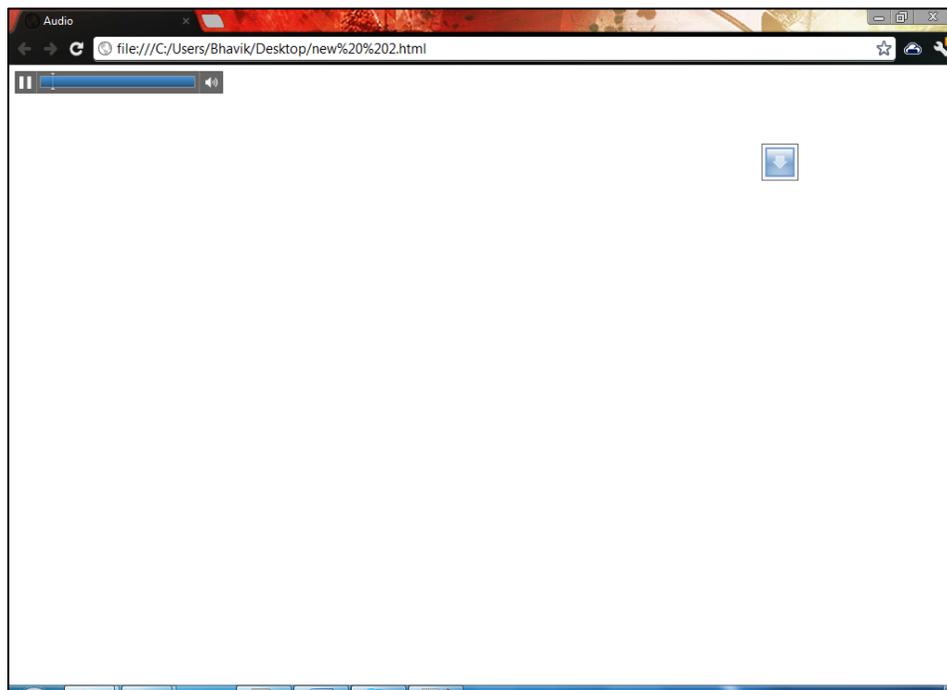


4.2 Audio Components

Like we inserted videos into our webpage we can even insert only audio files, what we need to do is instead of `<video>` tag we need to replace it with `<audio>` tag, for e.g.

```
<audio controls="controls">  
  <source src="audio.mp3" type="audio/mp3" />  
  <source src="audio.ogg" type="audio/ogg" />  
  Audio Not Supported  
</audio>
```

We don't need to add height and width in the tag as we are displaying nothing on the screen so no need to add width and height. We'll need to change the formats, as you have seen in the example above. Let's try whether it works or not. It should look like this:



Try working with audio controls by changing the sources URL to your files and build your webpage with Audio Controls in a Webpage.

5. Drag and Drop functionality

One of the most promising new features of HTML 5 is its drag and drop functionality, drag and drop is what we call in real life pick a thing and leave it where you want or leave it at a particular place. In this chapter we will learn how to add some drag and drop functionality to your webpage. In HTML 5 we can drag any element on page.

5.1 How to make an element drag able

To create an element or object drag able we need to change drag able property to True, for e.g. if we want to set property of image drag able as true then we will write:

```
<img draggable="true" />
```

5.1 What to drag?

To specify what to do when an object is dragged, a suitable code is written, so that our webpage can know what to do when an object is dragged. This code even specifies what to drag from an object for e.g.

```
function drag(dr)
{
dr.dataTransfer.setData("Text",dr.target.id);
}
```

“Function drag (dr)” is used to state that the object to be dragged and “dataTransfer.setData” is used to recognize the data type and set the value of data that is dragged over.

5.2 What to drop

Above was whole we discussed about dragging; now we have learnt how to drag, but how to drop and object? So let’s see what to drop and how to drop an object.

```
function allowDrop(dr)
{
dr.preventDefault();
}
```

The above code is used in the script so that our webpage supports dropping of an object.

```
function drop(dr)
{
var data=dr.dataTransfer.getData("Text");
dr.target.appendChild(document.getElementById(data));
dr.preventDefault();
}
```

The above code is used to append or describe what to do when an object is dropped on any other object. All the above codes are inserted in <head> under <script> tag, for better clarification let's check a full code of a page to understand drag and drop better.

```
<!DOCTYPE HTML>
<html>
<head>
<style type="text/css">
#rect {width:350px;height:70px;padding:10px;border:1px solid #aaaaaa;}
</style>
<script type="text/javascript">
function allowDrop(ve)
{
ve.preventDefault();
}

function drag(ve)
{
ve.dataTransfer.setData("Text",ve.target.id);
}

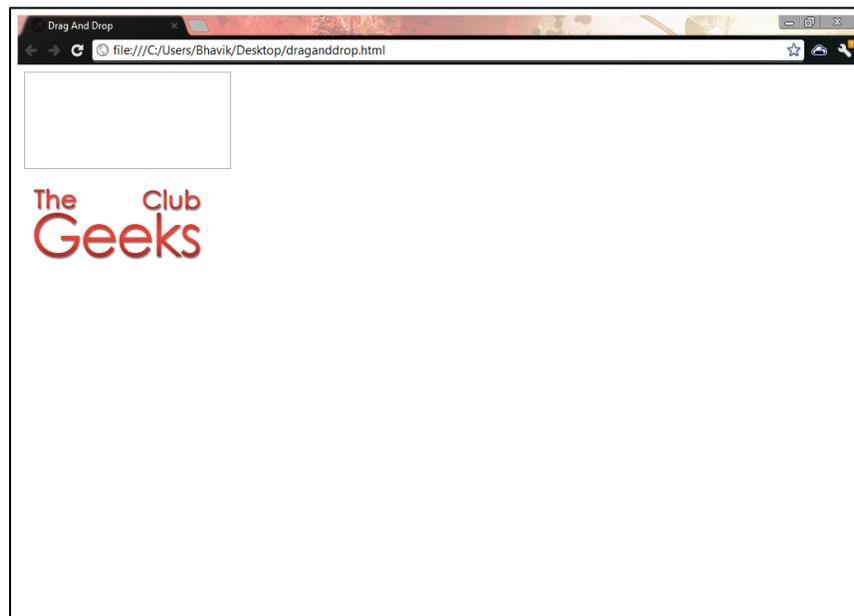
function drop(ve)
{
var data=ve.dataTransfer.getData("Text");
ve.target.appendChild(document.getElementById(data));
ve.preventDefault();
}
</script>
</head>
<body>

<div id="rect" ondrop="drop(event)" ondragover="allowDrop(event)"></div>
<br />

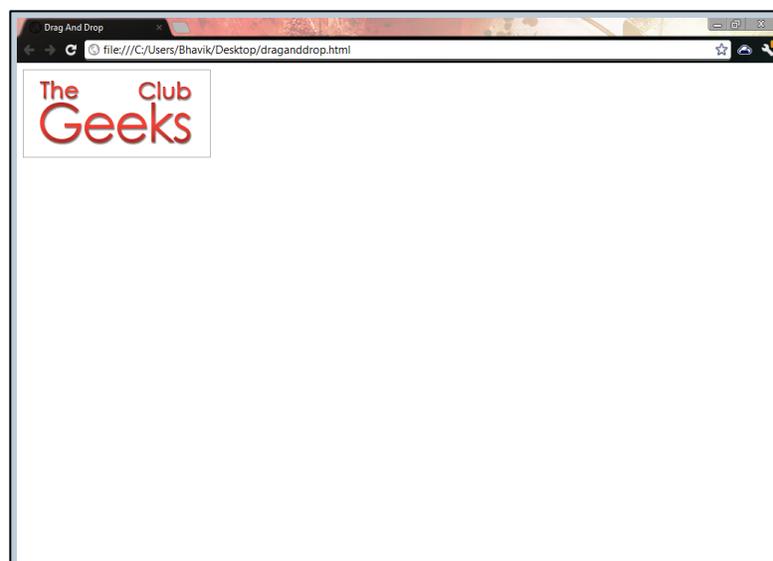

</body>
</html>
```

The above code was of a page in which you can drag an image to a rectangle, study the code thoroughly and try to create your one such page. You can edit the code and add what to do if the image is dragged over a rectangle. This was a quite difficult task, but don't panic, just go through the code twice thrice and more, so let's conclude what we did in this chapter:

- To get the dragged data `dataTransfer.getData("text")` is used.
- Dragged data can be said as the id of dragged element ("pic1").
- `preventDefault()` is used to prevent the browser default handling of the data.
- Check out a screenshot to see how would your page look alike:



After dragging and dropping:



6. HTML 5 Canvas

Canvas is that part of HTML where we'll learn about graphics on a Web Page. Canvas is that element of HTML 5 that lets you draw graphics on screen by JavaScript. Canvas element is denoted by the `<canvas>` tag. Canvas is that part of HTML that cannot be taught or learnt, the more you explore, the more you find, so in this chapter you will learn canvas element by different examples. So let's go.

Example 1

Drawing the canvas element:

```
<canvas id="Canvasname" width="300" height="150"></canvas>
```

Remember, that canvas does not have drawing capabilities by its own; we do it by JavaScript, so see some more examples, to see how to draw different things in a canvas.

Example 2

```
<script type="text/javascript">
var c=document.getElementById("canvasname");
var ctx=c.getContext("2d");
ctx.fillStyle="#FF0000";
ctx.fillRect(0,0,200,100);
</script>
```

The above as you can see is the JavaScript, in the above code we used `getElementById` command to get the canvas object and to specify in which canvas to draw. `getContext` command is used to tell whether the shape should be 2D or 3D. `fillstyle` is used to tell the color and `fillRect` was used to tell the shape and the size of shape, you can change `Rect` with your own desired shape, no need to change the `Rect` command with square command or shape, as you can draw squares with the `Rect` command only.

You may have noticed that in `fillRect` command we have used some numbers in the brackets, those numbers are the coordinates or we can say the location where the rectangle would be drawn, you can replace coordinates with your desired numbers.

Example 3

Drawing a right angle with lines:

```
<script type="text/javascript">
var c=document.getElementById("Canvasname");
var ctx=c.getContext("2d");
ctx.moveTo(10,10);
ctx.lineTo(10,80);
ctx.lineTo(80,80)
ctx.stroke();
</script>
```

MoveTo command is used to move the coordinates but not to draw the lines on those points. Stroke command is used to finish the line.

Example 4

```
<script type="text/javascript">
var c=document.getElementById("Canvasname");
var ctx=c.getContext("2d");
ctx.fillStyle="#FF0000";
ctx.beginPath();
ctx.arc(200,70,70,0,Math.PI*2,true);
ctx.closePath();
ctx.fill();
</script>
```

In the above code, `ctx.arc` is used to draw the circle and after that `ctx.fill` is used to fill the circle. In the command `ctx.arc` we specified three numbers in the bracket, first one shows the `x` coordinate, second one the `y` coordinate of location and third one the radius of the circle and after that `math.pi*2` formula is used to draw circle from radius and `x`, `y` coordinates.

Whoa! We've given you a lot of examples now it's your turn to discover and create, so now what you are going to do is little bit of practice, you would now just replace the codes with your own and create different shapes and different objects in the canvas. But remember to create the canvas first and then add script in which the id of the canvas is identified.

7. Scalable Vector Graphics (SVG)

One of the most promising and the best feature of HTML 5 is that it supports SVG (Scalable Vector Graphics). Wikipedia says:



“Scalable Vector Graphics (SVG) is a family of specifications of an XML-based file format for two-dimensional vector graphics, both static and dynamic (i.e. interactive or animated). The SVG specification is an open standard that has been under development by the World Wide Web Consortium (W3C) since 1999.”

Source: Wikipedia

As you read that SVG is inserted by scripting XML files, HTML 5 shows support for SVG, what you need to do is just code SVG in xml.

What are the advantages of using SVG?

- SVG is written in XML and can be scripted easily.
- SVG graphics are scalable, so they don't lose quality and pixelate if zoomed or resized.
- SVG can be animated.
- W3C recommends SVG
- SVG images can be printed with high quality at any resolution.
- SVG graphics are supported by all browsers.

Embedding SVG into HTML 5

```
<svg xmlns="http://www.w3.org/2000/svg" version="1.1">  
  <circle cx="100" cy="50" r="40" stroke="black"  
    stroke-width="2" fill="green" />  
</svg>
```

You can just simple embed SVG in your HTML 5 DOC by starting with `<svg>` tag, but before inserting SVG into HTML you need to learn SVG you can simply learn it online, there are lots of tutorials available.

8. HTML 5 Geo Locations

HTML 5 supports Geo Locations, Geo Locations are used track a user's location with the permission of user, if the device is packed with GPS antenna then, you can get the accurate results, if not then the results would be based on IP address.

We will use `getCurrentPosition(showPosition)` command to get the position of the user and then we would add commands what to do! For example:

```
<!DOCTYPE html>
<html>
<body>
<p id="plocation">Your location would be displayed here</p>
<button onclick="Location()">Try It</button>
<script>
var x=document.getElementById("plocation");
function Location()
{
if (navigator.geolocation)
{
navigator.geolocation.getCurrentPosition(showPosition);
}
else{x.innerHTML="Geolocation Not supported";}
}
function showPosition(position)
{
x.innerHTML="Latitude: " + position.coords.latitude +
"<br />Longitude: " + position.coords.longitude;
}
</script>
</body>
</html>
```

In the above example we used a button to get location of a user in a paragraph; you can replace controls and can get the users location by using `getCurrentPosition` method.

9. Local Web Storage

One of the best and latest features of HTML 5 is the local web storage, in local web storage, instead of cookies the data is stored in the user's browser, web storage is far away better than cookies and they are fast, secure and easy to manage. The data is stored when it is asked for and we can even store larger amounts of data with web storage without affecting the performance of web site. Web storage is supported by all major browsers so there might not be much problem accessing the data. The one more feature of web storage is that, the data stored by a web site can only be accessed by the same web site.

There are two types of Web Storage:

- Local Storage: In local storage the data is stored permanently in browser and can be accessed anytime from the same browser.
- Session Storage: In this type of storage data is stored only for one session and can be expired after a particular time.

Local Storage

How to save data in web storage

To save a data `localStorage.fieldhere="data"` is used, for e.g. we want to create a new field in browser, we will write:

```
<script>
if (typeof(Storage) !== "undefined")
{
    localStorage.name="Name Here";
}
</script>
```

Now we've entered some data into name field, now let's see how we will read that data.

How to read data from web storage

To read the data we simply need to write a script that would read the data from storage and would display it into a paragraph or heading.

```
document.getElementById("name").innerHTML="Name: "  
+ localStorage.name;
```

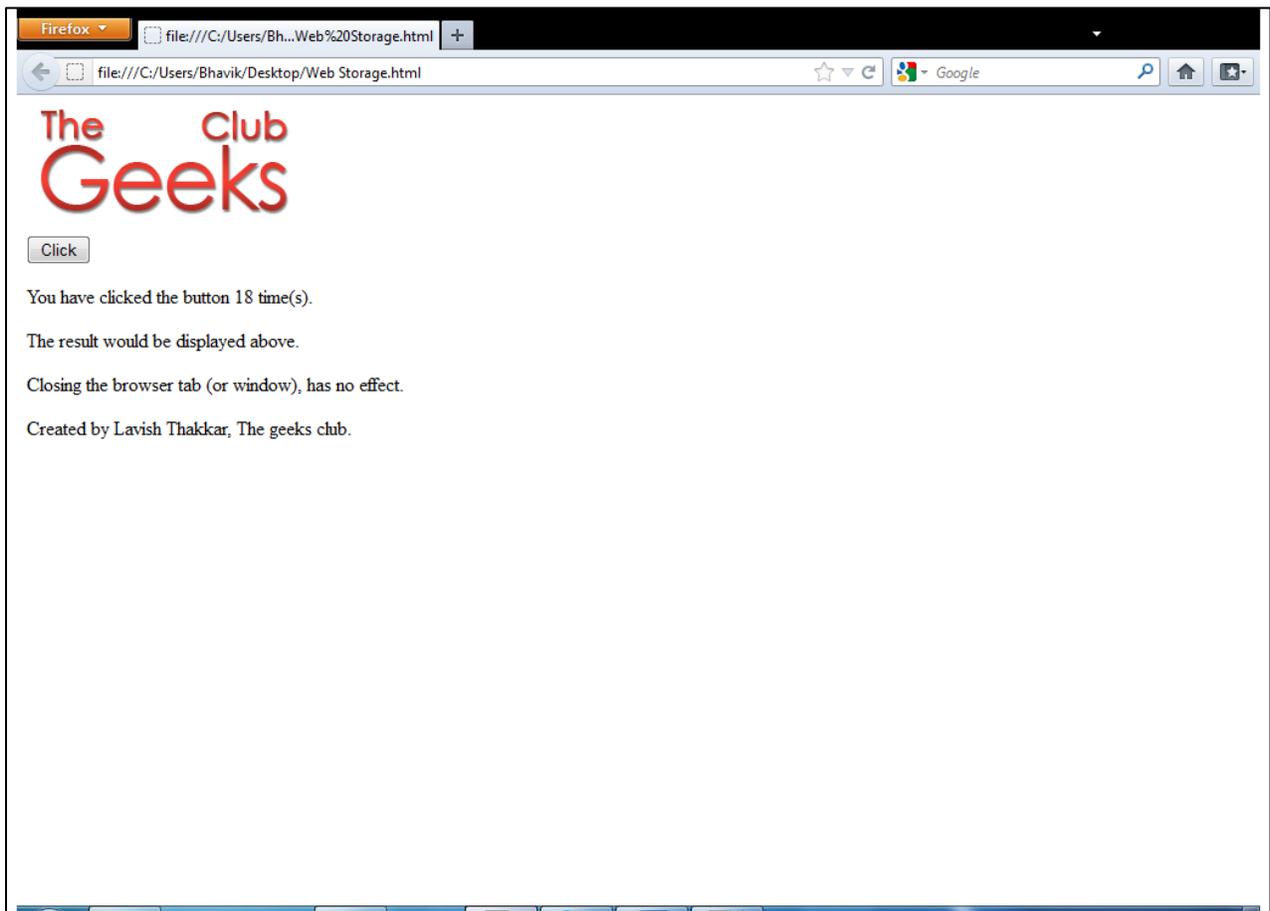
In the above example we've seen that, the local storage was displayed on the paragraph named as "name". See a full webpage example to know more about working of Web Storage.

```

<!DOCTYPE html>
<html>
<head>
<script>
function clicktime()
{
if (typeof(Storage) !== "undefined")
{
if (localStorage.num)
{
localStorage.num=Number(localStorage.num)+1;
}
else
{
localStorage.num=1;
}
document.getElementById("result").innerHTML="You have clicked the button " +
localStorage.num + " time(s).";
}
else
{
document.getElementById("result").innerHTML=" Web Storage not supported by your
browser.";
}
}
</script>
</head>
<body>
<img src = "http://www.thegeeksclub.com/wp-content/uploads/2011/03/TGC-Logo6.png" align
= "middle">
</br>
<p><button onclick="clicktime()" type="button">Click</button></p>
<div id="result"></div>
<p>The result would be displayed above.</p>
<p>Closing the browser tab (or window), has no effect.</p>
<p>Created by Lavish Thakkar, The geeks club.</p>
</body>

```

It would look like:



Closing, refreshing the window has no effect on the storage, it would continue from where you left it.

Session Storage

Now we've discussed about the local storage, now see the same example with session storage. What we will do is simply is just replace `localStorage` with `sessionStorage`.

Now try the example given below, when you will close your window or refresh the page, the storage would reset and the counting would start again.

```
<!DOCTYPE html>
<html>
<head>
<script>
function clicktime()
{
if(typeof(Storage)!=="undefined")
{
if (sessionStorage.num)
{
sessionStorage.num=Number(sessionStorage.num)+1;
}
else
{
sessionStorage.num=1;
}
document.getElementById("result").innerHTML="You have clicked the button " +
sessionStorage.num + " time(s).";
}
else
{
document.getElementById("result").innerHTML="Web Storage not supported by your
browser.";
}
}
</script>
</head>
<body>
<img src = "http://www.thegeeksclub.com/wp-content/uploads/2011/03/TGC-Logo6.png"
align = "middle">
</br>
<p><button onclick="clicktime()" type="button">Click</button></p>
<div id="result"></div>
<p>The result would be displayed above.</p>
<p>Closing the browser tab (or window), the storage would reset.</p>
<p>Created by Lavish Thakkar, The geeks club.</p>
</body>
</html>
```

10. Web Workers

Web Workers are individual scripts that run independently and perform their tasks on the page without affecting the performance of page.

You can do anything you want on the page, clicking, hovering, downloading but the web workers would enjoy doing their task without disturbing you or the user. When executing web workers in an HTML 5 page, the page becomes unresponsive until the script is finished. Web workers are supported in all major browsers except Internet Explorer.

HTML 5 has showed the support for external Web Workers, Web workers are usually written in JAVA Script and are saved in JS file.

How to read a JS file:

Before we get started with Web Workers, I want to tell you that you need to have a little bit knowledge of Java Script here, so before starting, grab some lessons of Java Script online.

To read Java Script web worker in HTML 5, we will write:

```
function startWorker()
{
  if(typeof(Worker) !=="undefined")
  {
    w=new Worker("java script file here");
    w.onmessage = function (event) {
      document.getElementById("output").innerHTML=event.data;
    };
  }
}
```

In the above code we added a worker that was in Java Script format, but you can add as many workers you want. To stop a running worker we'll use `w.terminate`

`W.terminate` is used to terminate the running worker. So, that was all in this chapter.

11. HTML 5 Forms

What are forms? All of you might have filled up some forms online to create an account or for any other task. All those forms can be created in HTML 5, there are lots of new form input types discussed and introduced in HTML 5 and in this chapter we will talk about the new input types introduced in HTML 5, following is the list of the new input types introduced in HTML 5, have a look:

- color
- date
- datetime
- datetime-local
- email
- month
- number
- range
- search
- tel
- time
- url
- week

Some of the new input types are not supported by some browsers, so when you code them they might appear as normal textboxes in the browser window.

Input Type: Color

This input type is used to select a color from the color picker:

```
<input type="color" name="color" />
```

Browser Support: Opera

Input Type: Date

This input type allows user to select a date:

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

Browser Support: Opera, Safari, Chrome

Input Type: E-mail

This input type allows user to enter an e-mail in textbox:

```
<input type="email" name="email" />
```

Browser Support: Opera, Firefox, Chrome

Input Type: Number

This input type defines a numeric value in a textbox with some restrictions:

```
<input type="number" name="number" min="1" max="10" />
```

Browser Support: Opera, Safari, Chrome

Input Type: Range

This input type is used to select a number from a range with a slider:

```
<input type="range" name="range" min="1" max="10" />
```

Browser Support: Opera, Safari, Chrome

There are lots of other input types which can be used by `<input>` tag, try them out and see what does your browser supports.

You might have seen that we've used "max" and "min" in our codes "max" and "min" are used to specify a value or the range within that of value. Those were the input types; now let's see what new form elements were introduced in HTML.

The new form elements in HTML 5 are

- `<datalist>`
- `<keygen>`
- `<output>`

Datalist element

This element is used to create a data list or a dropdown so that the user can select a specific option from the dropdown menu and this kind of element is used to provide pre-default options for an input type.

```
<input list="options" />

<datalist id="selectanoption">
  <option value="option1">
  <option value="option2">
  <option value="option3">
  <option value="option4">
  <option value="option5">
</datalist>
```

Keygen Element

Keygen element is used to create a safe and encrypted connection between the user and webpage. Keygen generates two types of codes, public and private code, private code is stored in the local server and public code is sent to the web server.

```
<form action="server url" method="get">
  Username: <input type="text" name="user_name" />
  Encryption: <keygen name="security" />
  <input type="submit" />
</form>
```

Output element

Output element is used to generate a result of a calculation, done in the webpage.

```
<form oninput="x.value=parseInt(a.value)+parseInt(b.value)">
<input type="number" name="a" value="50"/>
+<input type="number" name="b" value="50" />
=<output name="x" for="a b"></output>
</form>
```

Now, you have learnt a lot about HTML 5, now it time for some practice, go through the lessons for some exercise of HTML 5.

12. Designing forms

Designing a web form is quite a difficult thing, so there are some things that are to be kept in mind before designing a web form. A web form should be simple, easy and lite and it should be user friendly, a form without advertisements works as a form with advertisements looks like scam or a cheat.

The things to be kept in mind while designing a form are:

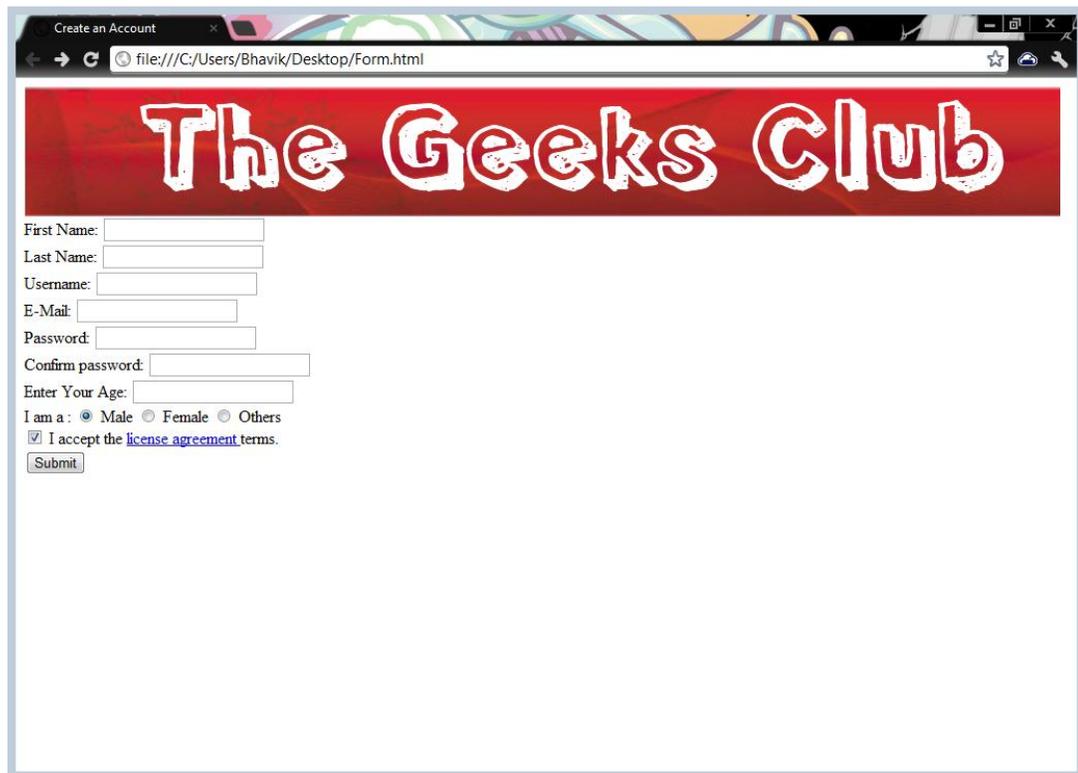
- Easy
- Should be in 2-3 steps
- It should be well ordered
- The fields should not relate to the users private information
- There should be an agreement at the end.
- Except of creating textboxes for a user to answer, you can create data lists or any other auto fill field.
- Form should be creative, with images
- No advertisements to be there.
- Proper alignment
- Good tab index
- Good header and footer
- Simple scripting or simple code to avoid mistakes

Here are some exercises for you so that you can create a better web form, answers for these exercises are provided at the very end of this book.

Exercise

1. Create a web form with your company banner at the top, the web form should include these fields:
 - a. First Name
 - b. Last Name
 - c. Username
 - d. Email
 - e. Password
 - f. Confirm Password
 - g. Age
 - h. Sex radio buttons
 - i. License agreement terms checkbox
 - j. Submit button

Screen Shot:



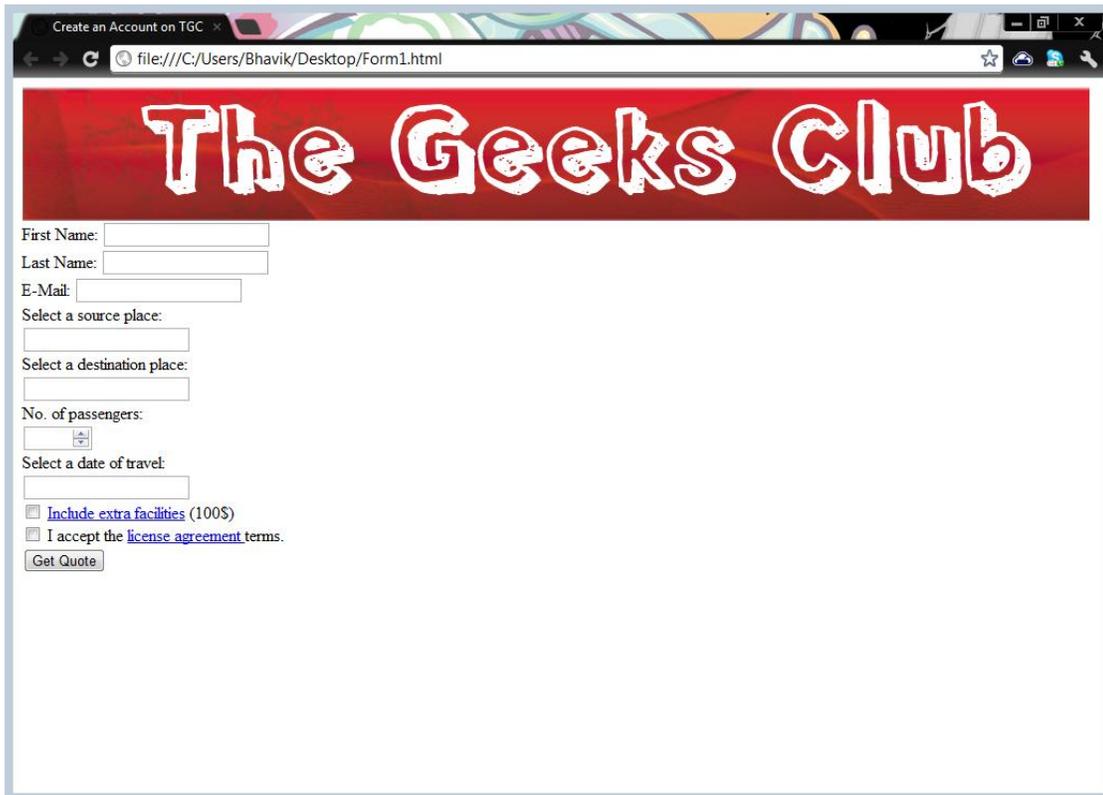
The screenshot shows a web browser window with the address bar displaying 'file:///C:/Users/Bhavik/Desktop/Form.html'. The page features a red banner at the top with the text 'The Geeks Club' in a white, stylized font. Below the banner is a registration form with the following fields and elements:

- First Name:
- Last Name:
- Username:
- E-Mail:
- Password:
- Confirm password:
- Enter Your Age:
- I am a : Male Female Others
- I accept the [license agreement](#) terms.
-

Click [here](#) for answer to this question.

2. Create a travel agency form for asking user about the traveling details, the form should include:
 - a. Company banner
 - b. Name of the ticket booker
 - c. Source Place (datalist)
 - d. Destination Place (datalist)
 - e. No. of passengers
 - f. Travel date
 - g. Include extra facilities (checkbox)
 - h. Get Quote Button

Screen Shot:



file:///C:/Users/Bhavik/Desktop/Form1.html

The Geeks Club

First Name:

Last Name:

E-Mail:

Select a source place:

Select a destination place:

No. of passengers:

Select a date of travel:

Include extra facilities (100\$)

I accept the [license agreement](#) terms.

Click [here](#) for answer to this question.

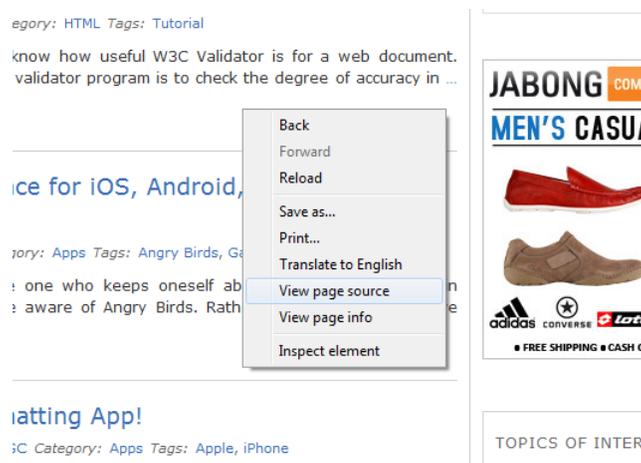
13. Building your first Web Page.

Huh! We've practiced and learned a lot, now it is time to explore and create something, how about building our first website or our first webpage with the new features of HTML 5?

So to start creating a webpage, first of all go through the codes again, you found interesting. And keep in mind that a webpage should have:

- A proper name and a company logo
- What is the webpage for?
- Some necessary and general details about the company or the person.
- Extra links
- Sharing buttons
- Lots of content
- Proper Alignment
- Fixed Size
- If personal website, advertisements may not work
- Catchy graphics
- Site Map
- Header
- Footer
- Favicon

To get references and some live examples, you can visit some websites and see how they have created there webpage, and you can even view the website code by just right clicking and selecting, view page source.



Do not copy the code, it would be illegal, but study the code and see how, they've managed and written the code and created a beautiful website with HTML.

14. Getting Certified

You've learned HTML 5, you've practiced it and now it is time for some quiz and a getting certified for HTML 5, there are lots of services out there which can certify you after a short quiz or any other way of testing your Knowledge about HTML 5. One such service is provided by W3 Schools, they would take a quiz and give you a certificate, and the certificate has a lot of worth.



Click [here](#) to get certified with W3 Schools.

You can find many such services out there, which would charge you a little fee but would certify you after a test.

15. Answers to designing forms questions

Code for question 1:

```
<!DOCTYPE html>
<head>
<title> Create an Account on TGC</title>
</head>
<body>
<img src = "image url here" align ="left">
<form action="server url here" autocomplete="on">
First Name: <input type="text"> </br>
Last Name: <input type="text"> </br>
Username: <input type="text"> </br>
E-Mail: <input type="email"> </br>
Password: <input type="password"> </br>
Confirm password: <input type="password"> </br>
Enter Your Age: <input type="date"> </br>
I am a : <input type="radio" name="sex" value="male" /> Male
<input type="radio" name="sex" value="female" /> Female
<input type="radio" name="sex" value="other" /> Others </br>
<input type="checkbox"> I accept the <a href="license link
here"> license agreement </a> terms.</br>
<input type="Submit" value="Submit">
</form>
</body>
</html>
```

Code for question 2:

```
<!DOCTYPE html>
<head>
<title> Create an Account on TGC</title>
</head>
<body>
<img src = "file:\\\C:\Users\Bhavik\Pictures\TGC1.png" align
="left">
<form action="server url here" autocomplete="on">
First Name: <input type="text"> </br>
Last Name: <input type="text"> </br>
E-Mail: <input type="email"> </br>
Select a source place: </br>
<input list="place" name="places" />
<datalist id="source">
  <option value="Place 1">
  <option value="Place 2">
  <option value="Place 3">
  <option value="Place 4">
  <option value="Place 5">
</datalist><br>
Select a destination place: </br>
<input list="dplace" name="dplaces" />
<datalist id="destination">
  <option value="Place 1">
  <option value="Place 2">
  <option value="Place 3">
  <option value="Place 4">
```

```
<option value="Place 5">
</datalist></br>
No. of passengers:</br>
<input type="number" max="50" min="1"></br>
Select a date of travel:</br>
<input type="date"></br>
<input type="checkbox"> <a href="extra facilities link
here">Include extra facilities</a> (100$) </br>
<input type="checkbox"> I accept the <a href="license link
here">license agreement </a> terms.</br>
<input type="Submit" value="Get Quote">
</form>
</body>
</html>
```

Some extra facts:

Once upon a time, there was a lovely language called HTML, which was so simple that writing websites with it was very easy. So, everyone did, and the Web transformed from a linked collection of physics papers to what we know and love today.

Most pages didn't conform to the simple rules of the language (because their authors were rightly concerned more with the message than the medium), so every browser had to be forgiving with bad code and do its best to work out what its author wanted to display.

In 1999, the W3C decided to discontinue work on HTML and move the world toward XHTML. This was all good, until a few people noticed that the work to upgrade the language to XHTML2 had very little to do with the real Web. Being XML, the spec required a browser to stop rendering if it encountered an error. And because the W3C was writing a new language that was better than simple old HTML, it deprecated elements such as `` and `<a>`.

A group of developers at Opera and Mozilla disagreed with this approach and presented a paper to the W3C in 2004 arguing that, "We consider Web Applications to be an important area that has not been adequately served by existing technologies... There is a rising threat of single-vendor solutions addressing this problem before jointly-developed specifications."

The paper suggested seven design principles:

1. Backwards compatibility and a clear migration path.
2. Well-defined error handling, like CSS (i.e. ignore unknown stuff and move on), compared to XML's "draconian" error handling.
3. Users should not be exposed to authoring errors.
4. Practical use: every feature that goes into the Web-applications specifications must be justified by a practical use case. The reverse is not necessarily true: every use case does not necessarily warrant a new feature.
5. Scripting is here to stay (but should be avoided where more convenient declarative mark-up can be used).
6. Avoid device-specific profiling.
7. Make the process open. (The Web has benefited from being developed in the open. Mailing lists, archives and draft specifications should continuously be visible to the public.)

The paper was rejected by the W3C, and so Opera and Mozilla, later joined by Apple, continued a mailing list called Web Hypertext Application Technology Working Group (WHATWG), working on their proof-of-concept specification. The spec extended HTML4 forms, until it grew into a spec called Web Applications 1.0, under the continued editorship of Ian Hickson, who left Opera for Google.

In 2006, the W3C realized its mistake and decided to resurrect HTML, asking WHATWG for its spec to use as the basis of what is now called HTML5.

Source: [Smashing Magazine](#).

A QUICK START GUIDE FOR: HTML 5



This is an ultimate start guide to HTML 5 written by Lavish Thakkar. After reading this book, you would have somewhat knowledge of HTML 5 and you would be able to create your first Web Page with HTML 5.

This book discusses the new elements and the new features of HTML 5 and you can learn about those new features after reading this book.

Before reading the book:

- You should have knowledge of previous versions of HTML 5
- You need to have a little knowledge of Java Script
- And a little knowledge of CSS

Published By:

The Club
Geeks

Disclaimer

The information contained in this eBook represents the views on HTML 5 and on its Applications. The publisher and author cannot guarantee the accuracy of any information presented or added after the date of publication in the operating system.

This eBook is for informational purposes only. TheGeeksClub MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS SUMMARY.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form, by any means (electronic, mechanical, photocopying, recording or otherwise), or for any purpose, without the express written permission of Publisher and Author or mentioning the eBook as source properly with the correct page number in reference.

Registered Trademarks, Logos and Icons used in this eBook are the property of their respective owners.

Unless otherwise noted, the example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, e-mail address, logo, person, place, or event is intended or should be inferred.

© 2011-12 TheGeeksClub. All rights reserved.

Thanks for reading the eBook. If you enjoy this release, then don't forget to subscribe to our feeds and email subscription for further notifications.

For any further queries or suggestions mail us at thegeeksclub@hotmail.com or visit:

<http://www.thegeeksclub.com>