

A decorative graphic consisting of three blue circles of varying sizes and two thin blue lines. One large circle is at the top center, a smaller one is below it, and another large one is at the bottom right. Two lines intersect to form a V-shape, with one line passing through the top circle and the other passing through the middle circle.

# HTML 5

Module Training about HTML 5

**SF 09-2**  
**3/8/2011**

## Table of Contents

Introduction .....	3
What is html5? .....	3
Why use html5?.....	3
What's new in HTML5? .....	3
How to create drag and drop effect .....	5
Make your element draggable .....	5
Understand the drag and drop event .....	5
Simple drag and drop.....	7
How to create a drawing canvas .....	9
How to create bouncing ball using canvas and worker HTML 5 .....	11
HTML5.....	11
Web Worker .....	11
Canvas.....	12
Canvas and Worker.....	12
How to create a video player .....	15
Video tag .....	15
How to create resizable and draggable video using HTML5 and JQuery .....	16
How to design web page HTML tag.....	20
Design a web page.....	21
Reference.....	24

## Introduction

### What is html5?

1. From en.wikipedia.org,  
HTML5 is a language for structuring and presenting content for the World Wide Web, a core technology of the Internet. It is the latest revision of the HTML standard (originally created in 1990) and currently remains under development.
2. From www.w3schools.com,  
HTML5 will be the new standard for HTML, XHTML, and the HTML DOM.  
The previous version of HTML came in 1999. The web has changed a lot since then.  
HTML5 is still a work in progress. However, most modern browsers have some HTML5 support.  
So, HTML5 is new and under development, but it is powerful and easy to learn.

### Why use html5?

1. Reduce the need of external plugin like flash. Normally, if we want to play video with need external plugin, we need to add this or that or whatever it is to make our video playable. But in HTML5, we can play video and audio with one tag without external plugin.
2. A lot of new features. For example, drawing canvas that enable user to draw, animate and play games using canvas and offline Storage to store your information.
3. Better error handling
4. More markup to replace scripting

### What's new in HTML5?

1. HTML5 new Tags

Article	Aside	Audio	Canvas
Command	Datalist	details	embed
figcaption	figure	footer	header
hgroup	keygen	mark	meter
nav	output	progress	rp
rt	ruby	section	source
summary	time	video	wbr

2. HTML5 new global Attributes

contenteditable	contextmenu	dir	draggable
dropzone	hidden	spellcheck	

3. HTML5 new Events

- a. Window Events

Onafterprint	Onbeforeprint	Onbeforeunload	Onerror
Onhaschange	Onmessage	Onoffline	Ononline
Onpagehide	Onpageshow	Onpopstate	Onredo
Onresize	Onstorage	Onundo	Onunload

**b. Form Events**

Oncontextmenu	Onformchange	Onforminput	Oninput
Oninvalid			

**c. Mouse Events**

Ondrag	Ondragend	Ondragenter	Ondragleave
Ondragover	Ondragstart	Ondrop	Onmousewheel
Onscroll			

**d. Media Events**

Oncanplay	Oncanplaythrough	Ondurationchange	Onemptied
Onended	Onerror	Onloadeddate	Onloadedmetadata
Onloadstart	Onpause	Onplay	Onplaying
Onprogress	Onratechange	Onreadystatechange	Onseeked
Onseeking	Onstalled	Onsuspend	Onupdate
Onvolumechange	onwaiting		

## How to create drag and drop effect

### Make your element draggable

To begin drop the item you just need to add `draggable="true"` to your tag. For example,

```
<div draggable="true">Hallo</div>
```

It will allowed your element to be dragged.

### Understand the drag and drop event

Ondrag	Script to be run when an element is being dragged
Ondragend	Script to be run when the element finish to be dragged
Ondragenter	Script to be run when an dragged element enter
Ondragleave	Script to be run when an dragged element leave
Ondragover	Script to be run when an dragged element over
Ondragstart	Script to be run when the element begin to be dragged
Ondrop	Script to be run when an dragged element drop

Let's begin to practice. In the example bellow. I create 3 div, trash, area and message. In here, I use a function to show the message in div. First, we need to add `draggable="true"` at trash. Add `ondrag`, `ondragstart` and `ondragend` event to trash. Add `ondragenter`, `ondragover`, and `ondragleave` to area. Now let's see the result. When I dragged the trash, the message about the event will show in message div. Show you can see when the script run.

Here is the screenshot when the trash is dragged:



## Full Code :

```
<html>
  <head>
    <style>
      #message
      {
        position:absolute;
        right:20px;
        top:0px;
      }
      #trash{
        width:100px;
        height:30px;
        background-color:pink;
      }
      #area
      {
        width:300px;
        height:300px;
        background-color:Black;
      }
    </style>
  </head>
  <body>
    <script>
      function showMessage (msg)
      {
        document.getElementById("message").innerHTML += msg + "<br/>";
      }
    </script>
    <div id="trash" draggable="true"
      Ondrag="showMessage('Garbage is being dragged')"
      Ondragend="showMessage('Garbage drag end')"
      Ondragstart="showMessage('Garbage drag start')">I am trash</div>
    <br />
    <div id="area"
      Ondragenter="showMessage('Garbage enter the area')"
      Ondragover="showMessage('Garbage over the area')"
      Ondragleave="showMessage('Garbage leave the area')"
    >Drop area</div>
    <div id="message"></div>
  </body>
```

## Simple drag and drop

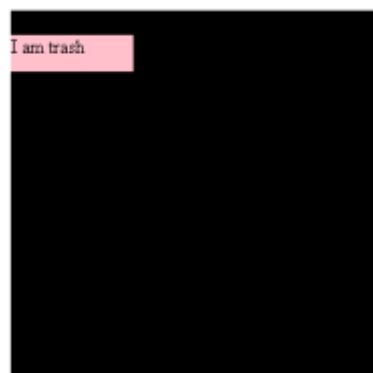
First, we need to create two div, one as trash and one as drop area. At the trash, we add need to add the ondragstart event and at the drop area, we add ondragover and ondrop event. When the drag start, we add `e.dataTransfer.setData('Text',target.id);` to set the data. In HTML5, it is allowed dataTransfer while we drag and drop. We add this to setData to the element id. After it, ondragover we just need to write `return false`, in some browser it doesn't run the ondrop script if we don't add this. Now, at ondrop we need to get the element that we want to add to the black area. After that we simple use appendChild to append the element to black area. I also use preventDefault to prevent browser run the default script. Let's see the result.

Here is the screenshot:

Before drop the trash



After drop the trash at the black area, the trash will be append to the black area.



## Full Code

:

```
<html>
  <head>
    <style>
      #trash{
        width:100px;
        height:30px;
        background-color:pink;
      }
      #area
      {
        width:300px;
        height:300px;
        background-color:Black;
      }
    </style>
  </head>
  <body>
    <script>
      function begin(target,e)
      {
        e.dataTransfer.setData('Text',target.id);
      }
      function drop(target,e)
      {
        var element = document.getElementById(e.dataTransfer.getData("Text"));
        document.getElementById("area").appendChild(element);
        e.preventDefault();
      }
    </script>
    <div id="trash" draggable="true" OnDragstart="begin(this,event)">I am trash</div>
    <br />
    <div id="area" OnDrop="drop(this,event)" OnDragover="return false">Drop area</div>
  </body>
</html>
```



## How to create a drawing canvas

I am using HTML5 and JQuery to create a simple drawing demo. First, `$(function() {});` is the function that called when page is loaded or ready. In here, i am getting the canvas offset to know the position of myCanvas and setting the lineWidth.

After that, i add mousemove event for myCanvas. I use isDraw as a flag whether it is drawing or not. If it is true, i am getting the endX and endY. endX and endY is the position where our line end. They are getting from mouse position minus canvas position. And then i am calling function draw which will draw the line. At last, i am setting begin position to end position.

I also add mouseup event to myCanvas. I call the function draw to drawing the line and ended at mouse position (minus 8 to get the exact position). Don't forget to setting isDraw to false.

Here is an mousedown event for myCanvas. In here, i am setting isDraw to true and get the current mousePosition.

Here is the result.



A simple drawing canvas, you can draw whatever you want.

## Full Code :

```
<html>
  <head>
    <title>Drawing Canvas</title>
    <style type="text/css">
      .classDown
      {
        background:gray;
      }
      #x img
      {
        border: 1px solid black;
        width:30px;
        height:30px;
      }
    </style>
    <script type="text/javascript" src="jquery-1.5.min.js"></script>
    <script type="text/javascript">

      $(function() {
        canvas = $("#myCanvas").offset();
        lineWidth = 5;

        $("#myCanvas").mousemove(function(e) {
          if (isDraw) {
            endX = e.pageX - canvas.left;
            endY = e.pageY - canvas.top;
            draw(beginX, beginY, endX, endY);
            beginX = endX;
            beginY = endY;
          }
        });
        $("#myCanvas").mouseup(function(e) {
          draw(beginX, beginY, e.pageX-8 , e.pageY -8);
          isDraw = false;
        });
        $("#myCanvas").mousedown(function(e) {
          isDraw = true;
          beginX = e.pageX;
          beginY = e.pageY;
        });
      });

      function draw(x1, y1, x2, y2) {
        myCanvas = document.getElementById("myCanvas");
        myContext = myCanvas.getContext("2d");
        myContext.lineWidth = lineWidth;
        myContext.lineCap = "round";
        myContext.beginPath();
        myContext.moveTo(x1, y1);
        myContext.lineTo(x2 + 1, y2 + 1);
        myContext.stroke();
      }
    </script>
  </head>
  <body>
    <canvas id="myCanvas" style="border:1px black solid;" width="600px"
height="300px">

    </canvas>
    <br />
    <br />
  </body>
</html>
```

## How to create bouncing ball using canvas and worker HTML 5

### HTML5

What is HTML5? HTML5 is a language for structuring and presenting content for the World Wide Web, a core technology of the Internet. It is the latest revision of the HTML standard. There is a lot of new features in HTML5. Why we use HTML5? Because there's many new features and which is very powerful and easy to learn. In here, we are going to learn about worker and canvas in HTML5.

### Web Worker

The Web Workers specification defines an API for spawning background scripts in your web application. Web Workers allow you to do things like fire up long-running scripts to handle computationally intensive tasks, but without blocking the UI or other scripts to handle user interactions.

Normally, we write `worker = new Worker("myscript.js")` and use `postMessage` to transfer information between `myscript.js` and our html page. In "myscript.js" we have an event handler to define what we are going to do if we receive a message and we can use `postMessage` to send a message to our HTML page. In our html page, we have an event handler just like in "myscript.js" to define what we are going to do if we receive a message.

There's a simple example about it. This is the html page. First, we need two buttons and one div to show the message. When the button is clicked, the worker will post a message and it will be handled at "worker.js". At the script in the html page, don't forget to declare the worker and `addEventListener` to handle the event if the worker posts a message.

```
<html>
  <body>
    <script>
      var worker = new Worker("worker.js");
      worker.addEventListener("message", function(e) {
        document.getElementById("msg").innerHTML = e.data;
      }, false);
    </script>
    <button onclick="worker.postMessage('msg')">Send message to
worker</button>
    <button onclick="worker.postMessage('close')">Close worker</button>
    <div id="msg"></div>
  </body>
</html>
```

This is the "worker.js". In here, we handle the message event. If the data we receive is `msg` then we will `postMessage` to the html page that the message has been received. If the data is `close`, the worker will be closed.

```
self.addEventListener("message", function(e) {
  if(e.data == "msg")self.postMessage("I have been process");
  else if(e.data == "close"){
    self.close();
    self.postMessage("I have been closed");
  }
}, false);
```

## Canvas

The canvas is the html5 element uses javascript to draw graphics in web page. With html5, we are free to draw anything we want and control the animation in the canvas area. How to begin? Just add `<canvas> </canvas>` in your body and begin draw with javascript.

This is a simple example to draw circle using canvas in HTML5. We need to get the canvas element and `getContext` from canvas. After that we begin to draw using context. `fillStyle` to arrange the fill style, `beginPath` to begin. After that, use `arc`, with `arc` we can draw circle, half circle and ext. `context.arc(x,y,radius,startAngle,endAngle, clockwise)`; we can adjust what we want to draw by change the parameter in `arc`. Last we need to call `closePath` and `fill`.

```
<html>
  <head>
    <script>
      function begin()
      {
        var canvas = document.getElementById("myCanvas");
        var context = canvas.getContext("2d");
        context.fillStyle = "#000000";
        context.beginPath();
        context.arc(20,20,10,0,Math.PI*2,false);
        context.closePath();
        context.fill();
      }
    </script>
  </head>
  <body onload="begin()">
    <canvas id="myCanvas" style="border:1px solid black">
    </canvas>
  </body>
</html>
```

## Canvas and Worker

In this demo, i am using canvas and worker in HTML 5 to make multiple bouncing ball. Each of the worker will handle one ball. First of all, don't forget to add the canvas area in your web. The canvas's id is "myCanvas". Let's begin to the javascript part.

I have a begin function that called when the page is ready. In this function i am declare worker using looping. Each of the worker call `postMessage` and add event listener to it. In `postMessage`, we declare what we want to send. In `addEventListener`, we use `draw` as the function that call when a message is sent from worker.

In the `draw` function, we handle the message that is sent. If the 'cmd' is `draw`, we will draw circle according the information that sent and if the 'cmd' is `clear`, we are goint to clear pixel according the information that sent.

```

var worker = new Array();
var speed = 5;
var flag = new Array();
var i,j;
var canvas;
var context;
var total = 5;

function begin()
{
    canvas = document.getElementById("myCanvas");
    context = canvas.getContext("2d");
    for(i=0;i<total;i++)
    {
        worker[i] = new Worker("ball.js");
        worker[i].postMessage({'cmd': 'draw', 'msg': ''});
        worker[i].addEventListener('message',draw,false);
    }
}

function draw(e) {
    switch(e.data.cmd)
    {
        case 'draw' :
            context.fillStyle = e.data.msg[2];
            context.beginPath();
            context.arc(e.data.msg[0],e.data.msg[1],10,0,Math.PI*2,true);
            context.closePath();
            context.fill();
            break;
        case 'clear' :
            context.clearRect(e.data.msg[0],e.data.msg[1],20,20);
            break;
    }
}

```

In ball.js, we have a function to handle the postMessage. In the function, i arrange the position of my ball and let it loop using setInterval. The position is according to the current position and how the ball will bounce. I am using random to manage the color of the ball. After arrange the position and color, we use postMessage to send message to our html page to draw the ball. Remember, each time before draw the ball, we need to erase the ball first.

```

self.addEventListener('message', function(e) {
  var data = e.data;
  var x,y,ax,ay;
  var ballSize = 10;
  var canvasW = 500;
  var canvasH = 300;
  var speed = 5;
  var bX,bY;

  switch (data.cmd) {
    case 'draw':
      x = rand(canvasW);
      y = rand(canvasH);
      ax = ay = speed;
      setInterval(function()
        {
          bX = x - ballSize/2;
          bY = y - ballSize/2;
          self.postMessage({'cmd':'clear', 'msg': new Array(bX,bY)});
          var color = "rgb("+rand(255)+" "+rand(255)+" "+rand(255)+" ");
          if(x + ballSize > canvasW || x < 0)ax *= -1;
          if(y + ballSize > canvasH || y < 0)ay *= -1;
          x += ax;
          y += ay;
          self.postMessage({'cmd':'draw', 'msg': new
Array(x+ballSize/2,y+ballSize/2,color)});
        },50);
      break;
    case 'close':self.close();
      break;
  };
}, false);

function rand(num)
{
  return Math.round(Math.random()*num);
}

```

Result:



## How to create a video player

### Video tag

Normally, if we want to play video and audio on our browser we need to use other plugin. But in html5, we just need to insert the <video> tag and <audio> tag. Currently the video format supported is ogg, mpeg4 and webm (Not all browser support this). You just need to add "`<video src="file_name"></video>`"

Here is some the attributes for video tag :

1. `height="500"` or `height="500px"` to adjust the height of the video
2. `width="500"` or `width="500px"` to adjust the width of the video
3. `controls="controls"` to adjust the controls of the video. By default the controls is not visible, but if you add `controls="controls"` you will able to use the play/pause button, the volume adjustment and the video slider
4. `autoplay="autoplay"` to adjust if you want to autoplay or not. By default the video is need to play by the user
5. `audio="muted"` to mute the video
6. `loop="loop"` to loop the video after it finish
7. `src="video_name"` video\_name is the name of your video file. For example : "`http://luhuiya.byethost7.com/my_video.ogg`" or "`my_video.ogg`"
8. `poster="file_name"` to show image when you haven't play the video. It should be picture that can explain your video or one of your video frame. video\_name is the name of your video file. For example : "`http://luhuiya.byethost7.com/my_poster.png`" or "`my_poster.png`"
9. `preload="load"` to preload the video. The value can also be metadata, auto or none.

Sample Code :

```
<html>
  <body>
    <video src="funny-cartoon.ogg" autoplay="autoplay" height="200px"
width="300px">
    </video>
  </body>
</html>
```

## How to create resizable and draggable video using HTML5 and JQuery

Do you know that canvas in HTML5 is allowed to draw an image from the video element? With drawImage function in HTML5, you can use to draw an image from an image or video element. In this post, I am going to explain how to make resizable and droggable video in canvas.

In HTML5, you can play video in your browser without any other plugin. That's simple and amazing. That also allowed you to play audio too.

The canvas is the html5 element uses javascript to draw graphics in web page. With html5, we are free to draw anything we want and control the animation in the canvas area. How to begin? Just add < canvas> </canvas> in your body and begin draw with javascript.

First, let's look at the html. I just need the canvas and video in my body. In canvas, I am setting the width, border, and height. In video, I am setting the controls, autoplay, loop and source. The video is setting to invisible, so we just need to see the video in canvas.

```
<canvas id="c" style="border:1px solid black" width="600"
height="400"></canvas>
<video id="v" src="funny-cartoon.ogg" style="display:none"
controls="controls" autoplay="autoplay" loop="loop"></video>
```

Like I said above, we just need to call drawImage. For example,

```
var c = document.getElementById("c");
var v = document.getElementById("v");
var con = c.getContext("2d");
con.drawImage(v,videoX,videoY,videoW,videoH);
```

We can use it another way too. Like the sample bellow. We can crop the image to how we want to. That's allow us to separate our video to many side. We can do a lot of thing with it

```
var c = document.getElementById("c");
var v = document.getElementById("v");
var con = c.getContext("2d");
con.drawImage(v,videoX,videoY,videoW,videoH,destX,destY,destW,destH);
```

To make the video resizable and draggable, we just simple manipulate the coordinat when mouseup and mousedown event. To handle the mouseevent, I am using JQuery because it's more simple and easy for me. It's fine if you want to another way. When the mouse down, we need to validate whether it is in the video or not. If it is true, we are going to validate if is the resize place or not. I use two lines to mark the resize place. I am using isResize and isMove as flag to notice which one is for current event. At the mouseup, we just need to know the coordinat.

I create a draw function to draw the video. The function is called and going to run every 100 milliseconds. That make our image seem animate like video. In here, I also draw the line as resize icon.

It is my logic to create this way, you can also create a lot of demo using HTML5. For example, you can create a puzzle that the image is video and moving. Or you can create a canvas that allowed you to draw on video.

You also need to understand that HTML5 is still under development and not all the browser can run it. But there's plenty way to handle it.



Full Code

:

```

<html>
  <head>
    <script type="text/javascript" src="jquery-1.5.min.js"></script>
    <script type="text/javascript">
      var posX = 0 , posY = 0;
      var mouseX = 0, mouseY = 0;
      var videoX = 10, videoY = 10;
      var videoW = 100, videoH = 80;
      var beginX = 0, beginY = 0;
      var isMove = false;
      var isResize = false;

      $(function(e)
      {
        $("body").mouseup(function(e){
          if(isMove)
          {
            mouseX = e.pageX;
            mouseY = e.pageY;
            isMove = false;
          }
          if(isResize)
          {
            videoH = videoH + (e.pageY - beginY);
            videoW = videoW + (e.pageX - beginX);
            isResize = false;
          }
        });
        $("body").mousedown(function(e){
          if( e.pageX >= videoX + videoW - 10 &&
            e.pageX <= videoX + videoW + 10 &&
            e.pageY >= videoY + videoH - 10 &&
            e.pageY <= videoY + videoH + 10)
          {
            isResize = true;
            beginX = e.pageX;
            beginY = e.pageY;
          }
          else if(e.pageX >= videoX &&
            e.pageX <= videoX + videoW &&
            e.pageY >= videoY &&
            e.pageY <= videoY + videoH)
          {
            isMove = true;
            posX = e.pageX - videoX;
            posY = e.pageY - videoY;
          }
        });
        draw();
      });
    }
  }

```

```

function draw()
{
    var c = document.getElementById("c");
    var v = document.getElementById("v");
    var con = c.getContext("2d");
    c.width = c.width;
    videoX = mouseX - posX;
    videoY = mouseY - posY;
    con.drawImage(v,videoX,videoY,videoW,videoH);

    con.moveTo(videoX+videoW,videoY+videoH);
    con.lineTo(videoX+videoW,videoY+videoH-10);
    con.moveTo(videoX+videoW,videoY+videoH);
    con.lineTo(videoX+videoW-10,videoY+videoH);

    con.stroke();
    setTimeout("draw()",100);
}

</script>
</head>
<body>
    <canvas id="c" style="border:1px solid black" width="600"
height="400"></canvas>
    <video id="v" src="funny-cartoon.ogg" style="display:none" controls="controls"
autoplay="autoplay" loop="loop"></video>
</body>
</html>

```

Result :



## How to design web page

### HTML tag

There's a series of new tags in HTML5 and the tag in bellow is specified for design a better web page. Normally, if we want to create a web header most of us will use `div id="header"` to mark its as a header but now in HTML5, we can use `<header>` to specify that this is the header of our web.

Here is the tag use for help us in structure of our web,

1. Article , an independent content.  
Contents of the article should be able to stand alone without being influenced by the other content in the web.
2. Aside, content aside from the content its in.  
Contents of the aside should connected to the content where aside is placed in.
3. Details, an additional information.  
Contents of the details is an additional information, user can specified whether want to see it or not.
4. Figcaption, a caption in figure.  
Caption of the figure where its in, it should be place in the first or the last.
5. Figure, self-contained flow content.  
Contents of the figure must relevant to where its in, but it is ok if you want to remove it. For example : photos, diagram etc
6. Footer, the footer for a document or web page.  
Contens of footer is footer of something, for example, copyright, date, the author or contact information
7. Header, the header of web page.  
Contents of header normally is the introduction of what this page is doing or what the purpose is. It also can contains a group of navigation
8. Hgroup, a group of headings.  
Represent a group of headings in a section or document.
9. Nav, a section of navigations.  
Represent a group of navigations in a section or web page.
10. Section, defines a section in document.  
In a web page, we can contains many sections, like section of buying, section of product etc.
11. Summary, a header of details element.  
Contents of summary must describe what contains in the details its in.

## Design a web page

Here is a simple example to i make, instead of using div , I will use header , footer etc to design the web.  
So instead of using

```
<html>
<body>
  <div id="divHeader">Header</div>
  <div id="divNav">
    Navigation
    <ul>
      <li><a href="#">First</a></li>
      <li><a href="#">Second</a></li>
    </ul>
  </div>
  <div id="divSection">
    Section
    <div id="divArticle">Article</div>
    <div id="divArticle">Article</div>
    <div id="divArticle">Article</div>
  </div>
  <br style="clear:both">
  <div id="divFooter">Footer</div>
</body>
</html>
```

We are going to use this

```
<html>
<body>
  <header>Header</header>
  <nav>
    Navigation
    <ul>
      <li><a href="#">First</a></li>
      <li><a href="#">Second</a></li>
    </ul>
  </nav>
  <section>
    Section
    <article>Article</article>
    <article>Article</article>
    <article>Article</article>
  </section>
  <br style="clear:both">
  <footer>Footer</footer>
</body>
</html>
```

But there's a problem because not all browser support HTML5 element so we can fix it using createElement in javascript

```
<script>
  document.createElement ("header" );
  document.createElement ("nav" );
  document.createElement ("section" );
  document.createElement ("article" );
  document.createElement ("footer" );
</script>
```

Full Code

:

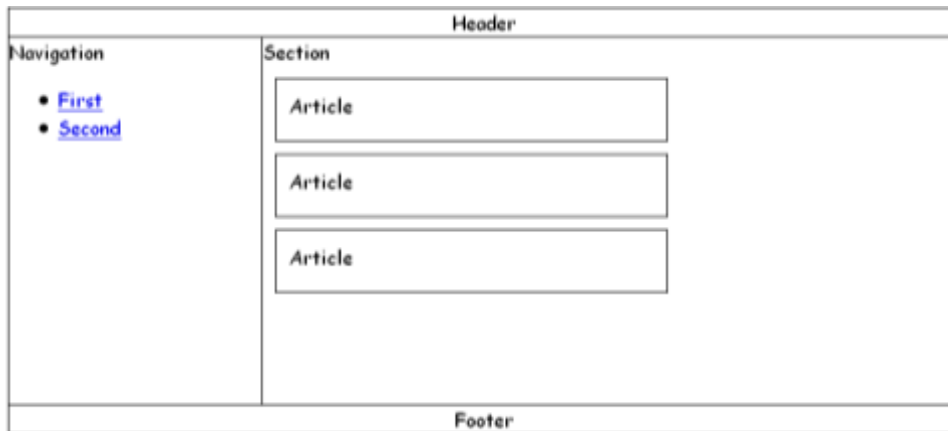
```
<!DOCTYPE html>
<html>
<head>
  <title>Design Web with HTML5</title>
  <style>
    nav
    {
      float:left;
      width:207px;
      border-left: 1px solid black;
      height:300px;
      display:block;
    }
    section
    {
      float:left;
      width:570px;
      border-left: 1px solid black;
      border-right: 1px solid black;
      height:300px;
      display:block;
    }
    footer,header
    {
      text-align : center;
      border: 1px solid black;
      width:778px;
      display:block;
    }
    article
    {
      border: 1px solid black;
      width:300px;
      margin:10px;
      height:30px;
      padding:10px;
      display:block;
    }
    body
    {
      font-family:Comic Sans MS;
      width:780px;
    }
  </style>
</head>
```

```

<body>
  <script>
    document.createElement("header");
    document.createElement("nav");
    document.createElement("section");
    document.createElement("article");
    document.createElement("footer");
  </script>
  <header>Header</header>
  <nav>
    Navigation
    <ul>
      <li><a href="#">First</a></li>
      <li><a href="#">Second</a></li>
    </ul>
  </nav>
  <section>
    Section
    <article>Article</article>
    <article>Article</article>
    <article>Article</article>
  </section>
  <br style="clear:both">
  <footer>Footer</footer>
</body>
</html>

```

Result :



## Reference

<http://dev.w3.org/html5/html4-differences/>

<http://www.w3.org>

<http://www.w3schools.com/html5>

<http://diveintohtml5.org/>