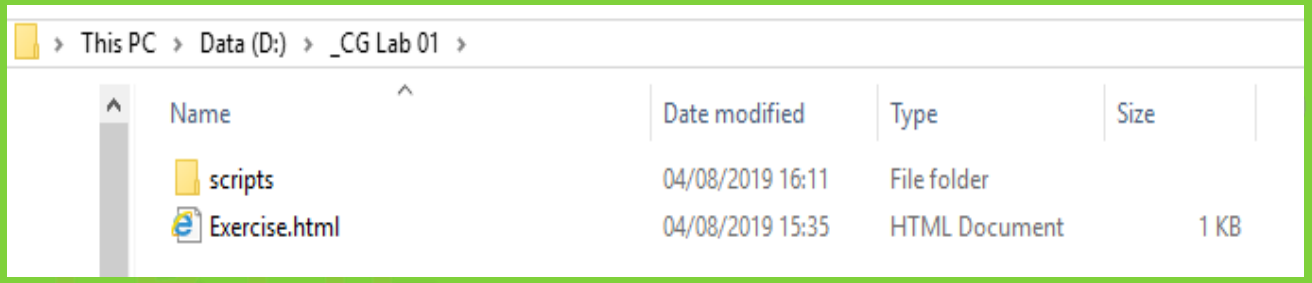
Lab 01- Drawing and Animation in a Browser

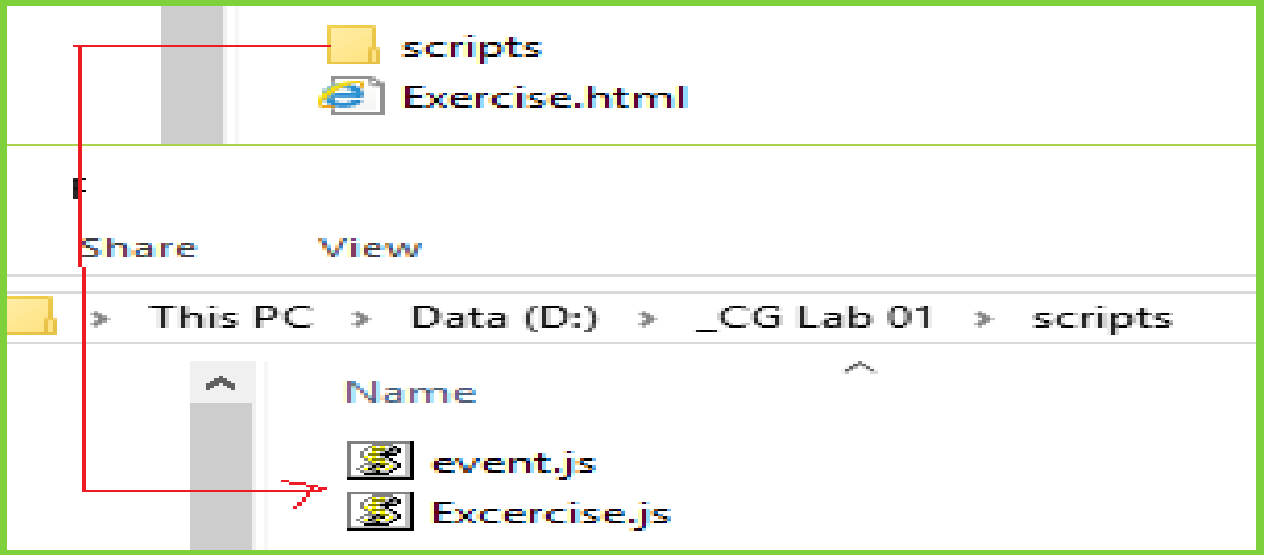
Exercises are taken from F.S. Hill Chapter 02,03

Lab 01 Objectives /Tasks

1. HTML Code Organization
2. The <canvas> tag
3. Adding Javascript into HTML
4. Study Concepts:
   - output primitives, canvas, viewport, pixel depth,
   - color depth, callbacks, moveTo(), lineTo()
5. Loading and Animating RGB images
   - click handler, keyboard handler, timer
7. Try: Google Colab, C++ .....
8. Lab Practice and Major Assignment (chap 02,chap 03)

Step I : Folder Creation and File Organization



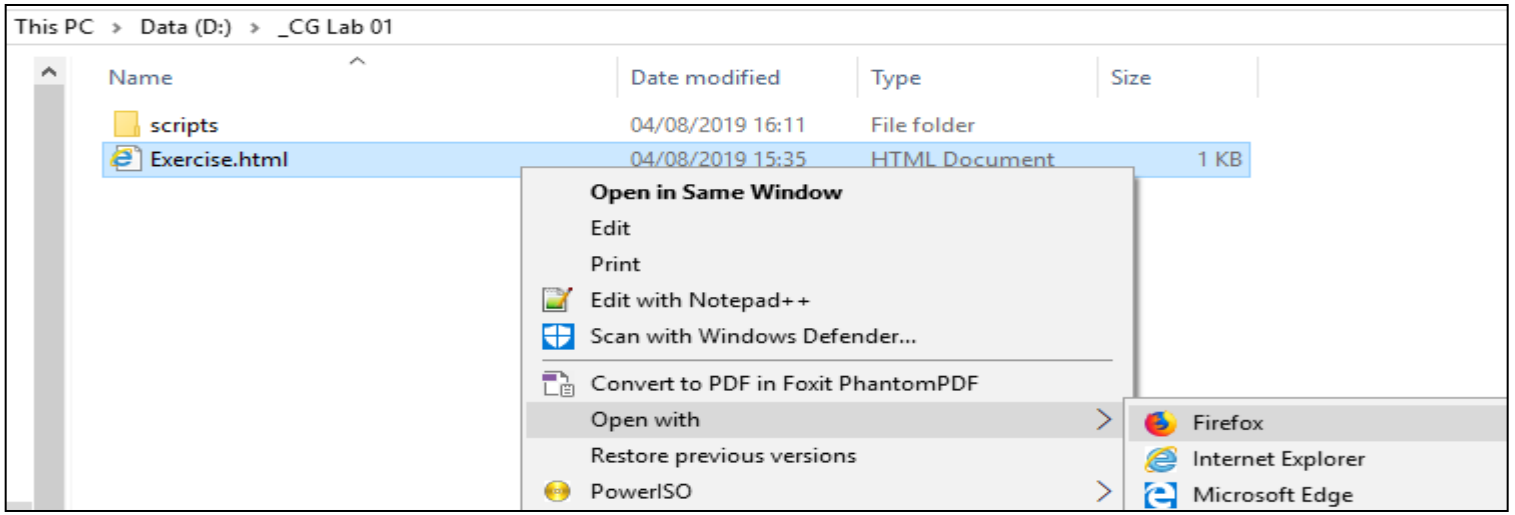


Step 2. Using <canvas> tag

Exercise.html | Excercise.js | event.js

```
1
2
3   <head>
4
5   <title> HT Javascript Lab 01 </title>
6   <script  src="scripts/Excercise.js"> </script>
7
8   </head>
9
10  <body>
11
12  <h1> Hill Chapter 02,03 Exercise 01 </h1>
13  <canvas  id="mycanvas" width="640" height="480" style="border:1px solid"
14
15  <p>  Enter default content here </p>
16
17  </canvas>
18
19
20  <p> Above this is canvas area </p>
21
22  </body>
```

Canvas renders its's content with Java script



Hill Chapter 02,03 Exercise 01

Above this is canvas area

Step 3. Adding Javascript into HTML

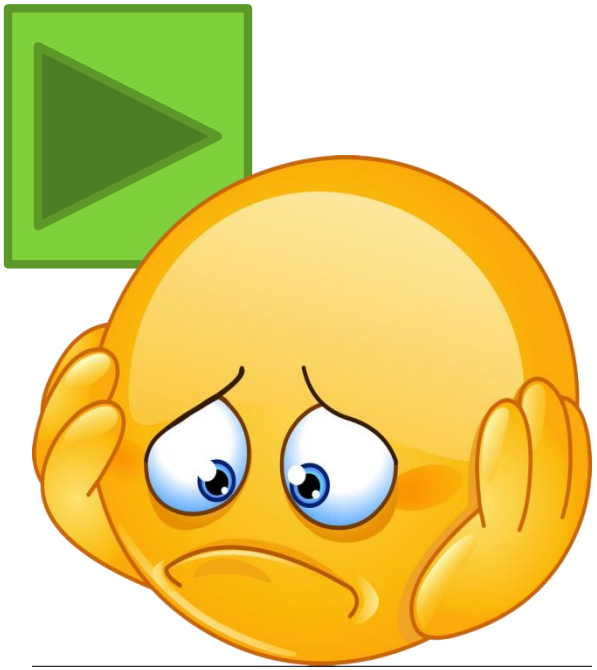
Exercise.html | Excercise.js | event.js

```
1
2   window.onLoad = myDisplay(); //executes call back function as the page finished loading
3
4   function myDisplay(){
5
6               var cvs = document.getElementById('mycanvas');  // Get canvas by ID
7                                                               // Canvas is by default invisible
8               if(cvs.getContext){
9
10                  var ctx = cvs.getContext('2d');  // Get Rendering context
11
12                  ctx.fillStyle="blue";
13                  ctx.fillRect(0,0,640,480);
14
15                  ctx.moveTo(0,0);      // set start position
16                  ctx.lineTo(320,240); //accept finish point and draw line
17
18              }
19  }
```

- Read callback from chap 02
- Read moveTo, lineTo from Chap 03

Recall <script> tag in header part of html file

```
<head>

<title> HT Javascript Lab 01 </title>
<script  src="scripts/Excercise.js"> </script>

</head>
```



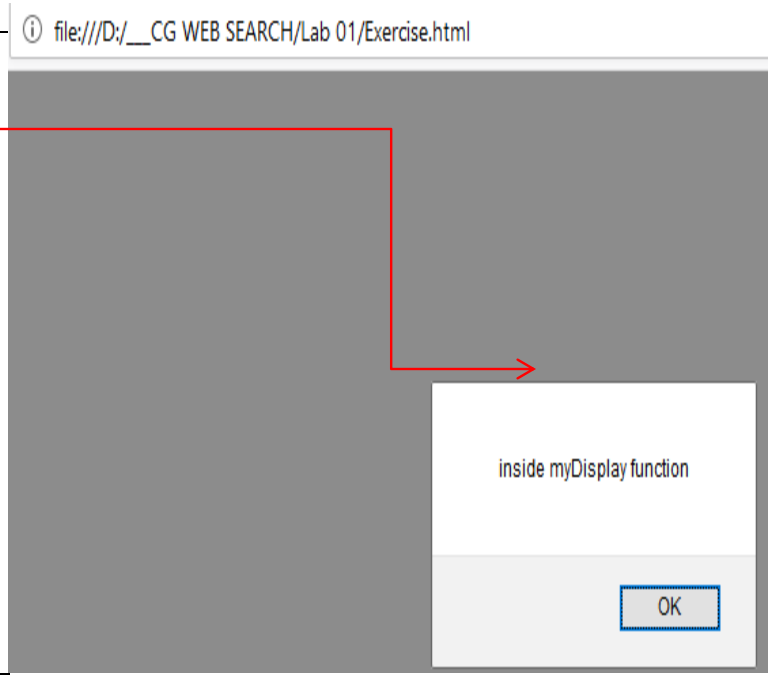
Hill Chapter 02,03 Exercise 01

Above this is canvas area

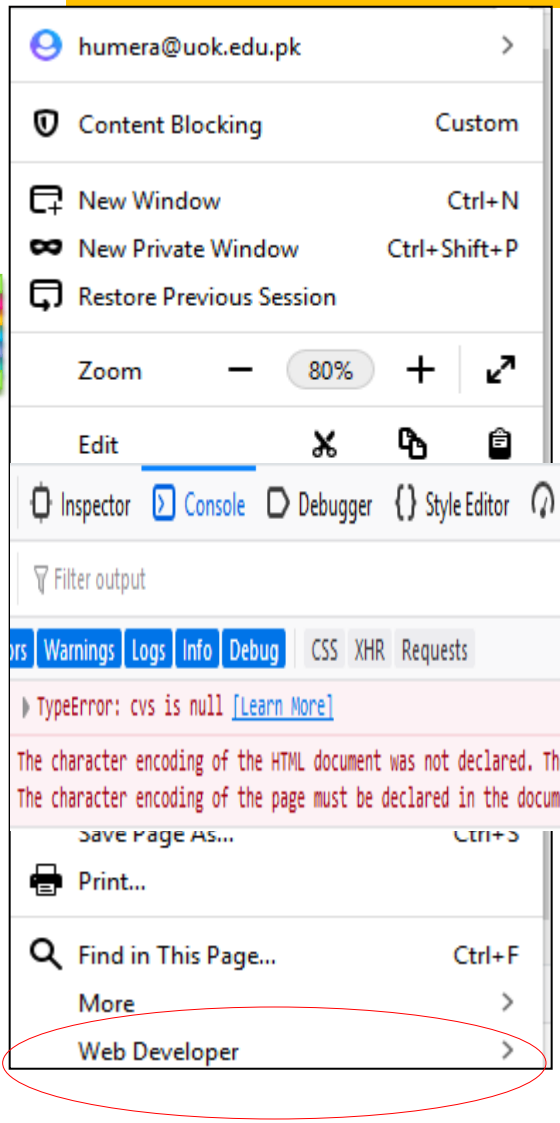
How to check that browser is loading script file

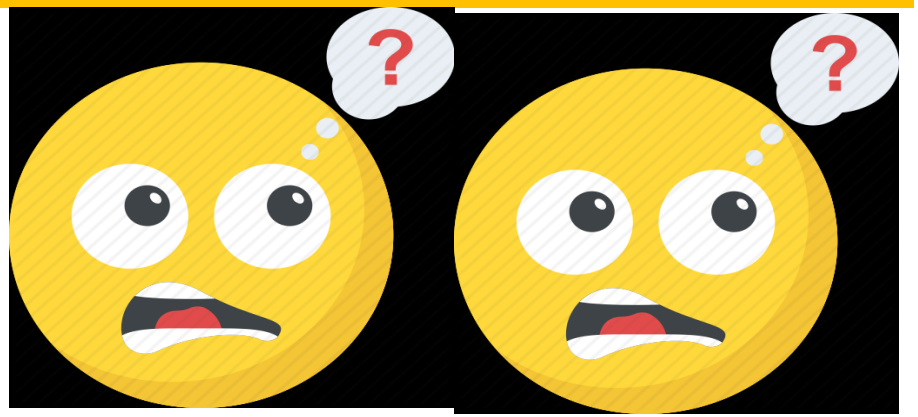
```
<head>

<title> HT Javascript Lab 01 </title>
<script  src="scripts/Excercise.js"> </script>

</head>
```

```
function myDisplay(){

    alert("inside myDisplay function");

        var cvs = document.getElementById('mycanvas');  // Get canvas by ID
                                                        // Canvas is by default invisible
            if(cvs.getContext){

              var ctx = cvs.getContext('2d');  // Get Rendering context

              ctx.fillStyle="blue";
              ctx.fillRect(0,0,640,480);

              ctx.moveTo(0,0);     // set start position
              ctx.lineTo(320,240); //accept finish point and draw line

            }
}
```



How to debug ??



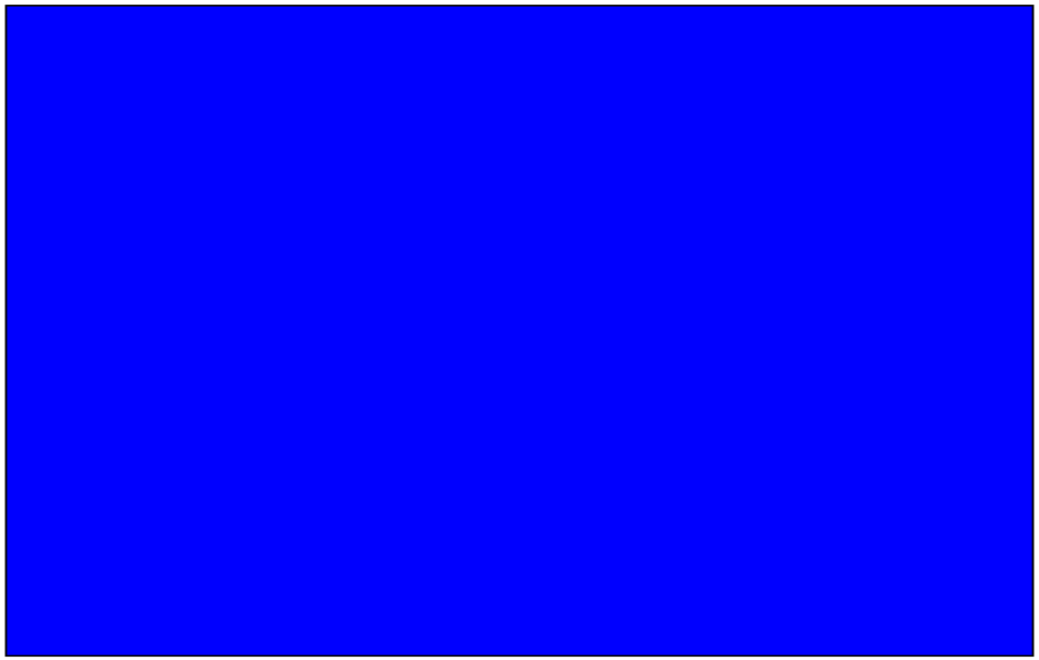


- cvs is null
- Language and encoding Warnings for html

<script> tag should placed after canvas in html as shown

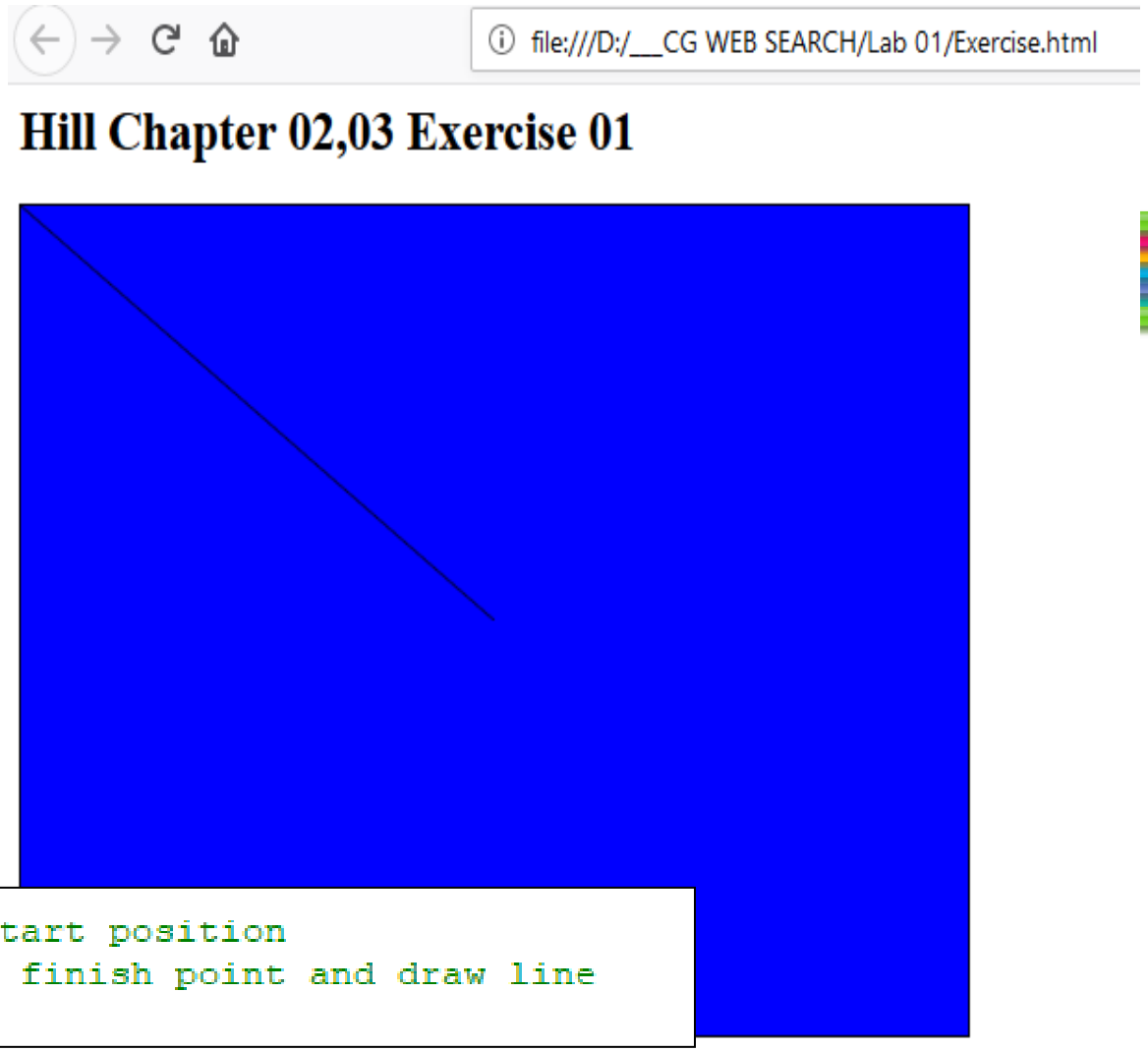
```
<body>

<h1> Hill Chapter 02,03 Exercise 01 </h1>
<canvas  id="mycanvas" width="640" height="480" style="border:1px solid"

<p>  Enter default content here </p>

</canvas>

<p> Above this is canvas area </p>
<script  src="scripts/Excercise.js"> </script>
</body>
```

Hill Chapter 02,03 Exercise 01



Above this is canvas area

<script> tag should placed after canvas in html as shown



www.colorpicker.com

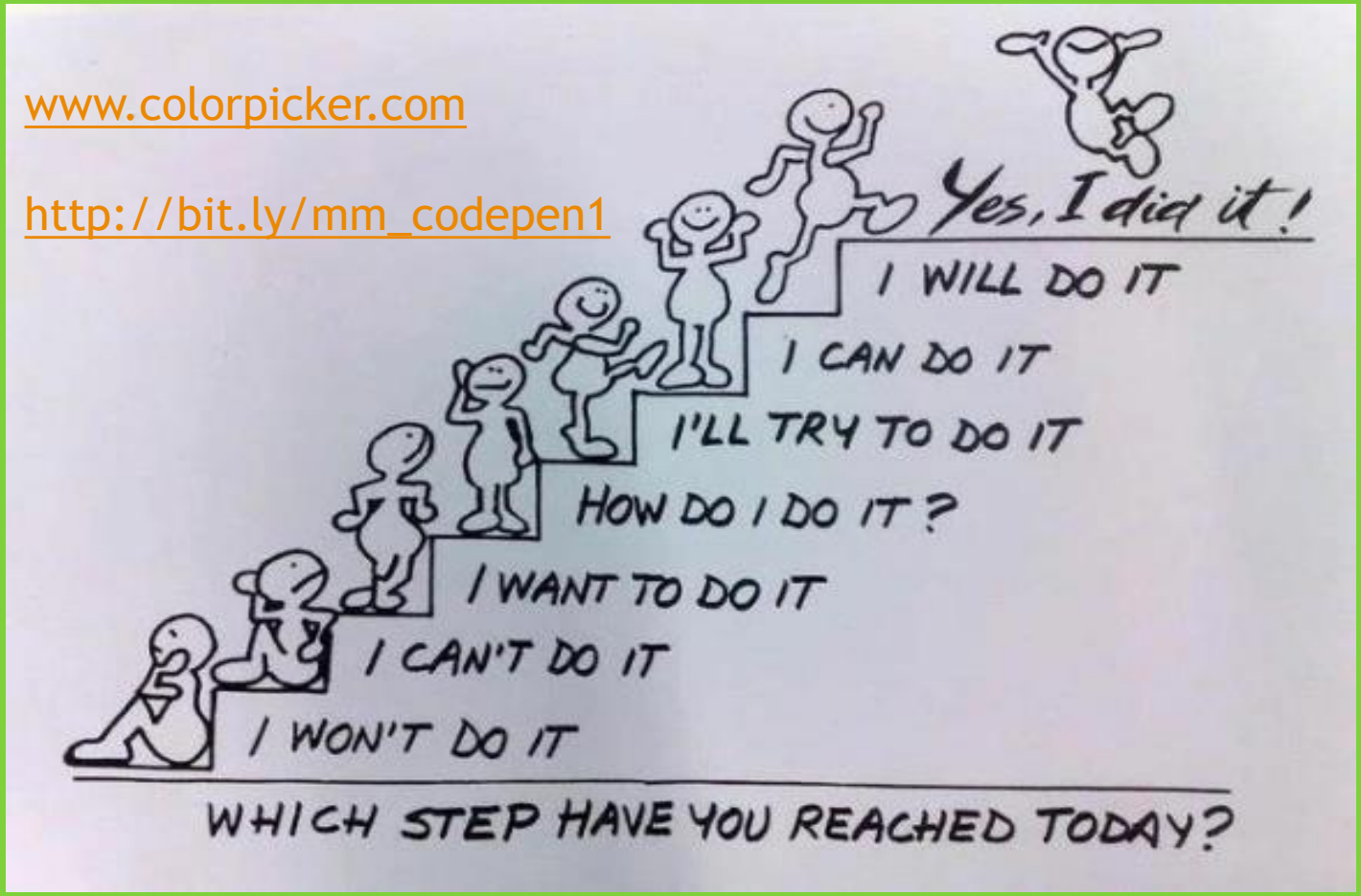
```
ctx.moveTo(0,0);      // set start position
ctx.lineTo(320,240); //accept finish point and draw line
ctx.stroke();
```

Practice makes a man better if not perfect

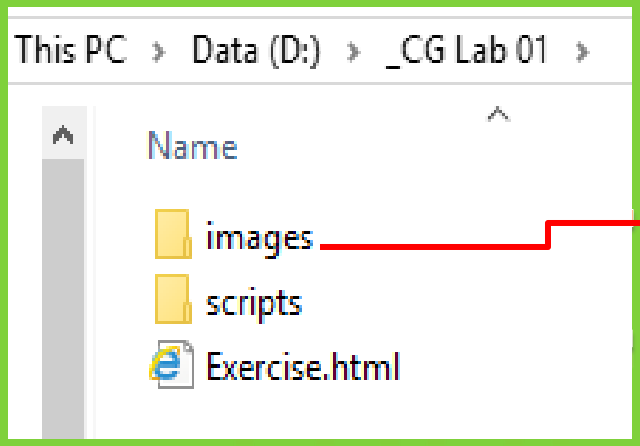
Check following links and do some practice

www.colorpicker.com

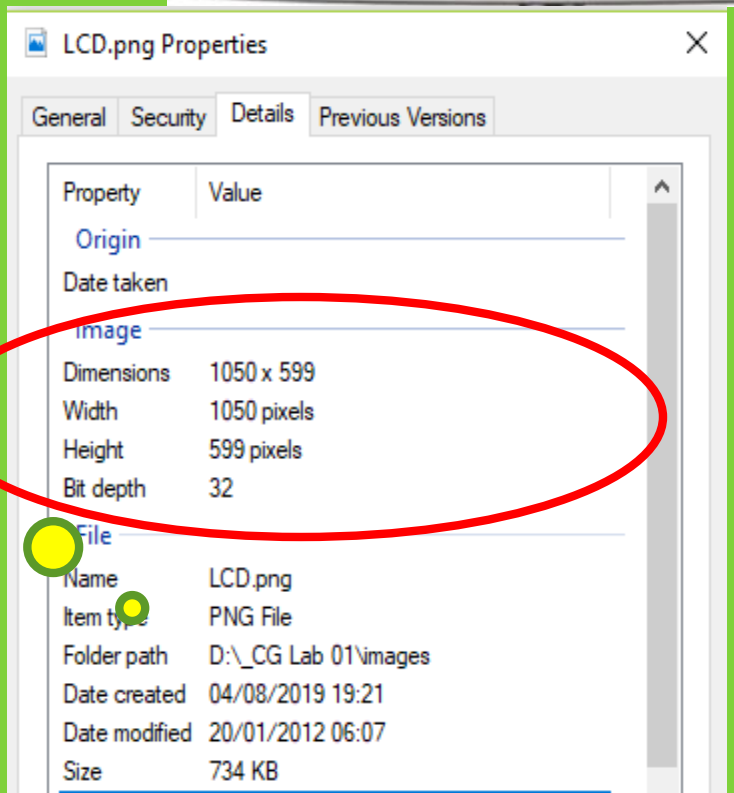
http://bit.ly/mm_codepen1



Step 4 : Loading and Displaying RGB images



- Read pixel depth, color depth, type of images, from chap 02 of book



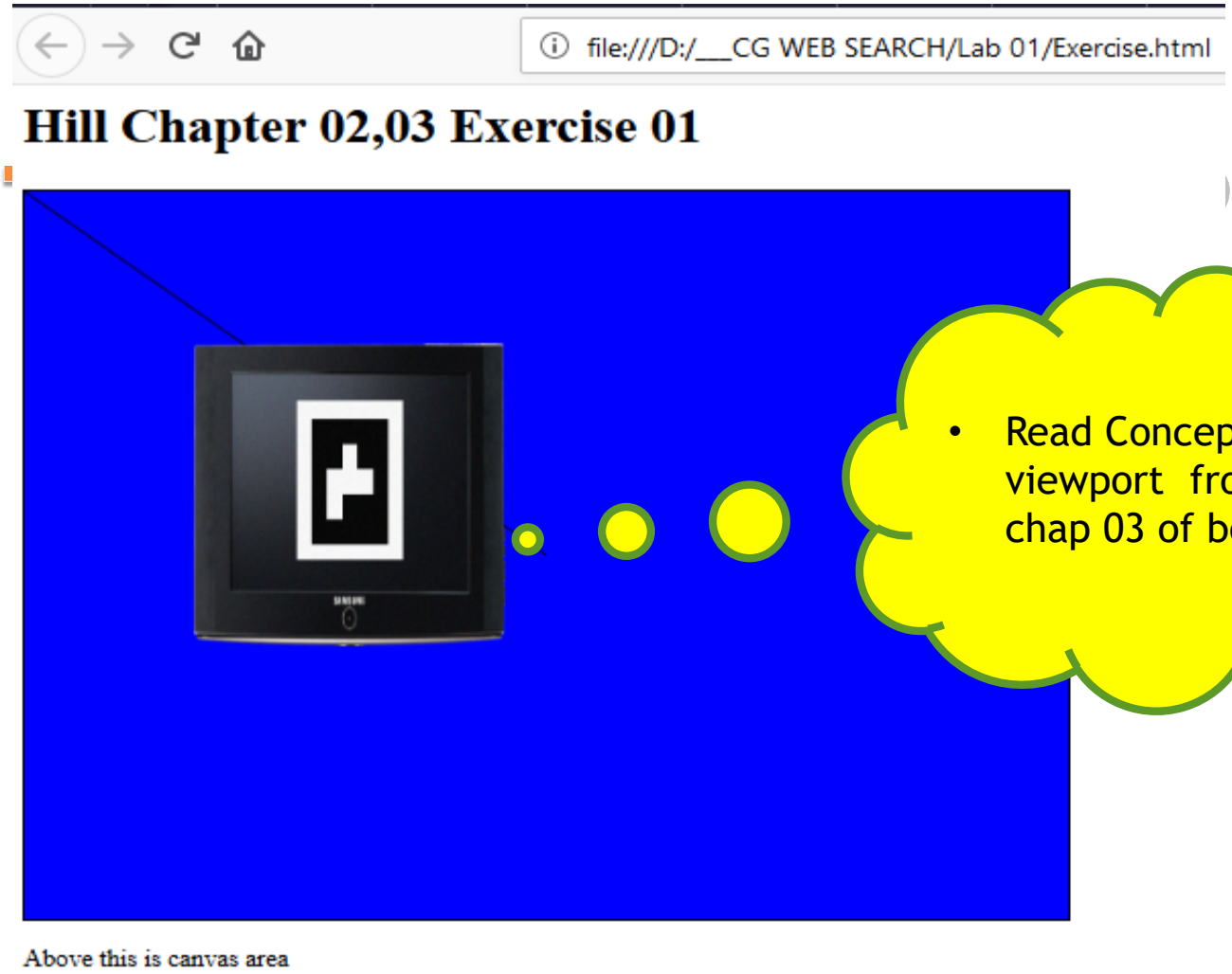
Modify .js file and simply refresh html in browser

.html | Excercise.js | event.js

```
window.onLoad = myDisplay(); //executes call back function as the page finished loading

function myDisplay(){

      alert("inside myDisplay function");

            var cvs = document.getElementById('mycanvas');  // Get canvas by ID
                                                            // Canvas is by default invisible
                if(cvs.getContext){

                  var ctx = cvs.getContext('2d');  // Get Rendering context

                  var img = new Image();
                  img.onload = function(){
                  ctx.drawImage(img,0,0,600,300);
                  } // close img.onload function

                  img.src = 'images/LCD.png';

                  ctx.fillStyle="blue";
                  ctx.fillRect(0,0,640,480);

                  ctx.moveTo(0,0);      // set start position
                  ctx.lineTo(320,240); //accept finish point and draw line
                  ctx.stroke();

            }
}
```

Image displayed successfully at (0,0)

Try changing image size
ctx.drawImage(img,100,100,200,200);



- Read Concept of viewport from chap 03 of book

Step 5 : Image animation Task breakdown

1) Write a function myInit() for Initialization i.e. loading background and foreground images

1) Write Keyboard handler function to start animation when user press enter

3) Call an update function at regular intervals by calling it on timer at key press

4) Update animation variable inside update function which calls repeatedly once the timer invokes it.

5) Detect Collision to stop moving object inside update function

Modify .html file to accept keyboard_timer.js

Exercise.html | Excercise.js | keyboard_timer.js

```
1
2
3   <head>
4
5   <title> HT Javascript Lab 01 </title>
6
7
8   </head>
9
10  <body>
11
12  <h1> Hill Chapter 02,03 Exercise 01 </h1>
13  <canvas  id="mycanvas" width="640" height="480" style="border:1px solid"
14
15  <p>  Enter default content here </p>
16
17  </canvas>
18
19  <!-- <p> Above this is canvas area </p> -->
20   <p> Press Enter to play your first animation </p>
21
22
23  <!--<script  src="scripts/Excercise.js"> </script> -->
24  <script  src="scripts/keyboard_timer.js"> </script>
25
26  </body>
```

Check comment tag

See Error

Press Enter to play your first animation

Inspector Console Debugger Style Editor Performance Memory Network Storage Acc

Filter output

Errors Warnings Logs Info Debug CSS XHR Requests

Loading failed for the <script> with source "file:///D:/___CG%20WEB%20SEARCH/Lab%2001/scripts/keyboard%20and%20timer.js".

The character encoding of the HTML document was not declared. The document will render with garbled text in some browser confi
The character encoding of the page must be declared in the document or in the transfer protocol.

Add new script file namely keyboard_timer.js

click event handling

keyboard_timer.js | keyboard_timer.html | Excercise.js

```
1
2   var bgImage = new Image(); //create empty 2D matrix for loading background image
3   var fgImage = new Image(); //create empty 2D matrix for loading foreground image
4
5   window.onLoad = myInit(); // First peform initialization
6
7   function myInit(){
8
9           bgImage.src = 'images/basket.bmp';    // loading
10          fgImage.src = 'images/basketball.bmp'; //loading
11
12          myDisplay();
13  }
14
15  function myDisplay() {
16
17        alert("inside myDisplay function"); // short cut to avoid onload ??????
18
19          var cvs = document.getElementById("mycanvas")
20          var ctx = cvs.getContext('2d');
21          ctx.drawImage(bgImage,0,0);  // draws background image
22
23          cvs.onclick = function(e){  // event object e knows current position of cursor
24              var x = e.clientX-24;
25              var y = e.clientY-12;
26
27              ctx.drawImage(fgImage,x,y,64,48); // draws foreground image
28          }
29  }
```

click event handling output

Further Modify keyboard_timer.js
Timer Event Handling

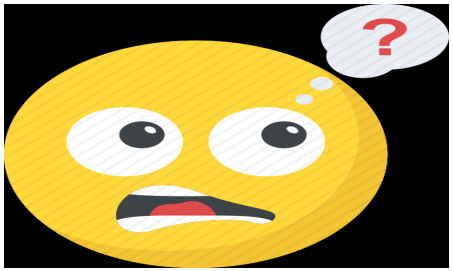
keyboard_timer.js | keyboard_timer.html | Excercise.js

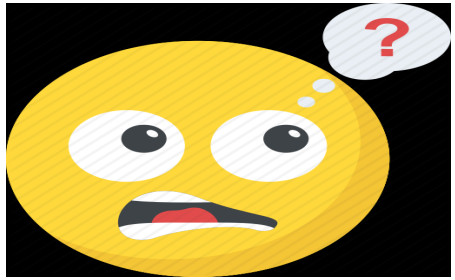
```
1
2   var bgImage = new Image(); //create empty 2D matrix for loading background image
3   var fgImage = new Image(); //create empty 2D matrix for loading foreground image
4   var timer ;
5   var ball = {
6               xx: 320,
7               yy: 200,
8               w: fgImage.width,
9               h: fgImage.height
10          }
11
12
13  window.onLoad = myInit(); // First peform initialization
14
15
16  function myInit(){
17
18          bgImage.src = 'images/basket.bmp';    // loading
19          fgImage.src = 'images/basketball.bmp'; //loading
20          myDisplay();
21  }
```

Carefully observe braces { and paranthesis (management while adding Event Handlers or listeners

```
function myDisplay() {

    alert("inside myDisplay function"); // short cut to avoid onload ??????

     var cvs = document.getElementById("mycanvas")
     var ctx = cvs.getContext('2d');
     ctx.drawImage(bgImage,0,0);  // draws background image

     cvs.onclick = function(e){  // event object e knows current position of cursor
                                 // listner for cursor
          var x = e.clientX-24;
          var y = e.clientY-12;

          ctx.drawImage(fgImage,x,y,64,48); // draws foreground image
     }

     //document.body.add = function(e){  // tried add but listner not listen ???
     document.body.addEventListener('keydown',function(e) { // callback registration

     e = event || window.event;                // any kind of event
     alert(String.fromCharCode(e.keyCode)+ "-->" + e.keyCode);  // checking callback
     console.log("you press some key");                         // checking callback


     var keycode = e.charCode || e.keyCode;
     if(keycode === 13) {  // code for return key

        // set the timer to repeatedly call update function
        console.log("you invoke timer callback function");
        timer = setInterval(moveBall,10) // deltaT = 10ms
         }
     }
     )
```

myDisplay() is still in continuation

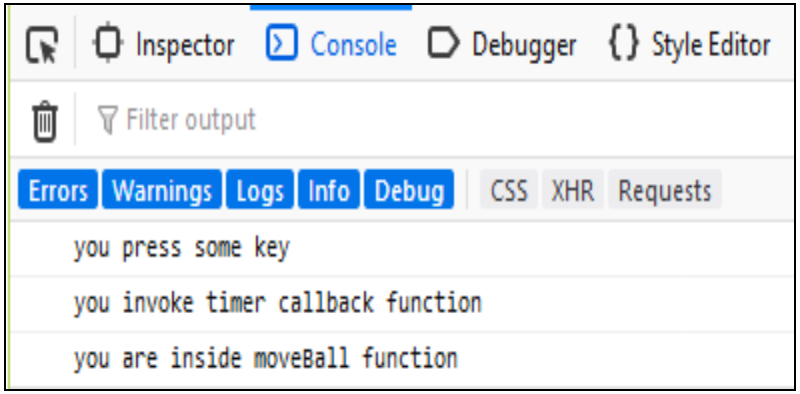


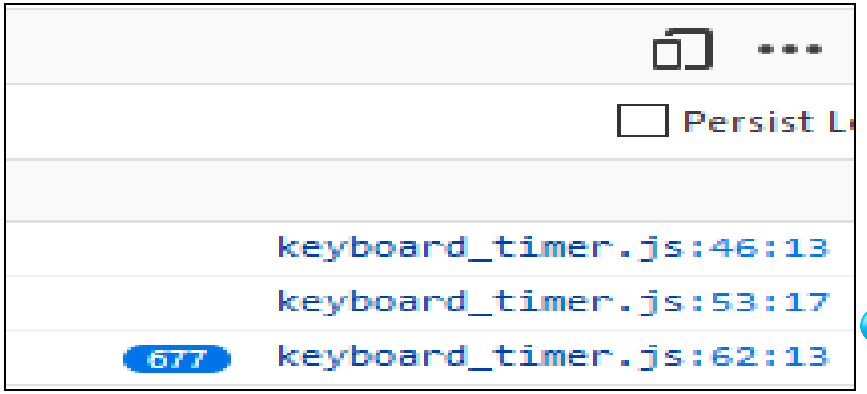


```
        //var moveBall  = function(){
         function moveBall() {
           //alert("inside moveBall function"); // alert is dangerous inside timer ???
           console.log("you are inside moveBall function");

           // Redrawing Preparations
           ctx.drawImage(bgImage,320,200);  // draws background image

           ctx.drawImage(fgImage,ball.xx,ball.yy,ball.w,ball.h); // draws fg image at initial position
           //ball.xx +=10;    // update ball.y variables to animate it
           //ball.yy -= 2.5;  // update ball.x variable to animate it


        }  // end moveBall()

}  // end myDisplay()
```

Output of moveBall ()
Timer Event Handling

Hill Chapter 02,03 Exercise 01





Lab Practice 01

- Correct or modify code so that ball will change its position
- Stop the timer
- Attempt to draw some drawings from chap 02 and chap 03 of recommended Text Book i.e. F.S. Hill and show me your efforts in next lab.

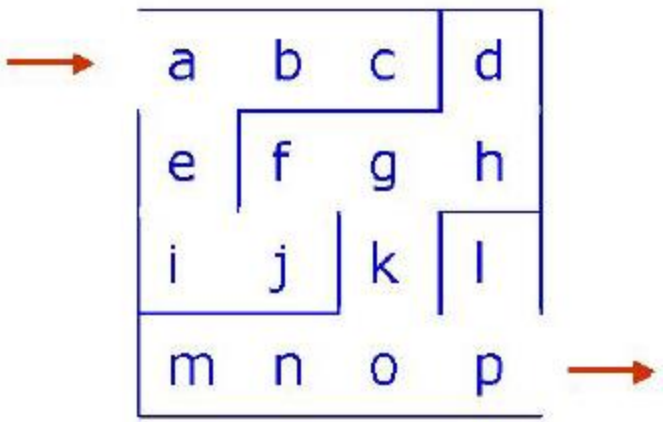
Major Assignment 01

Case Study 2.7 Building and Running Mazes

Due Week: Last week of August 2019

Note: Submit in your lab timings

A maze is a graph with two special nodes.



$$\left\langle \left\{ \begin{array}{l} \langle a,e\rangle, \langle e,i\rangle, \langle i,j\rangle, \\ \langle f,j\rangle, \langle f,g\rangle, \langle g,h\rangle, \\ \langle d,h\rangle, \langle g,k\rangle, \langle a,b\rangle \\ \langle m,n\rangle, \langle n,o\rangle, \langle b,c\rangle \\ \langle k,o\rangle, \langle o,p\rangle, \langle l,p\rangle \end{array} \right\}, a, p \right\rangle$$