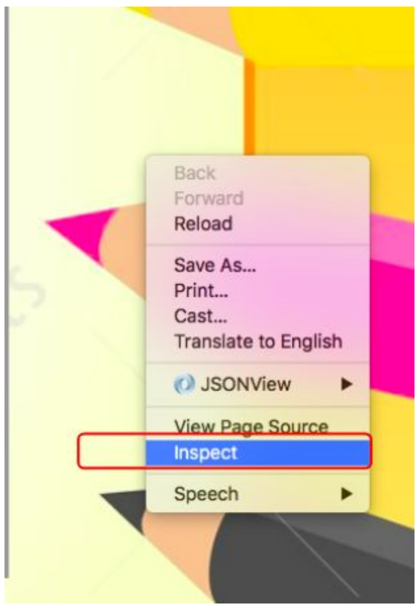
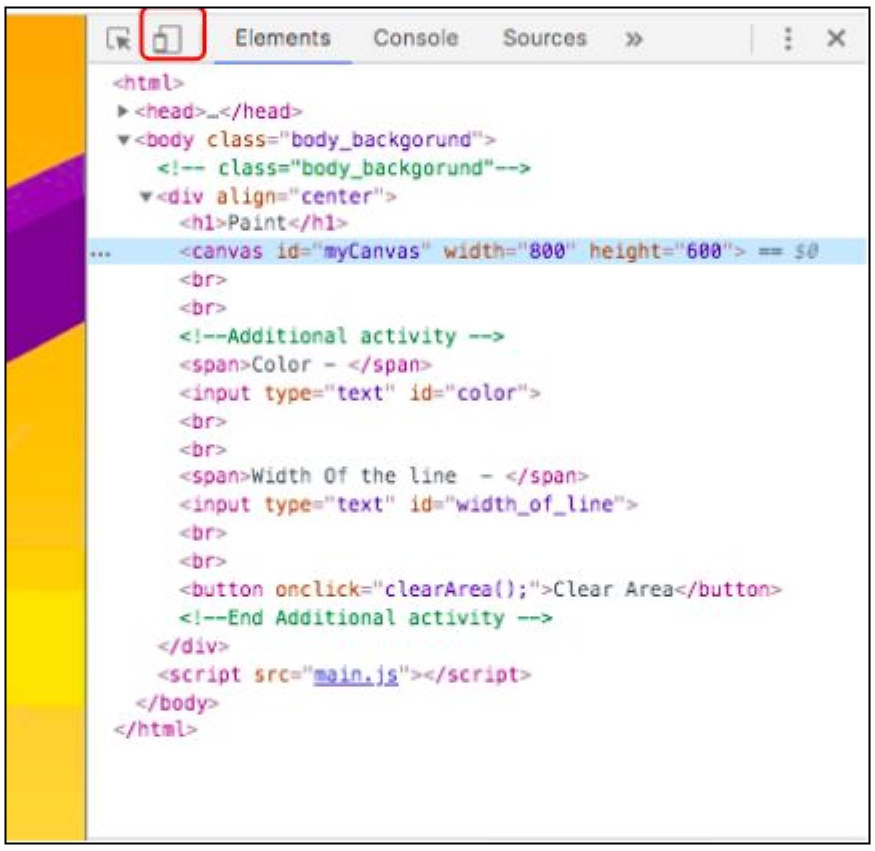
- **Steps are for testing how the canvas will behave when it's in mobile.**

1. Right click on the screen and click Inspect element option-



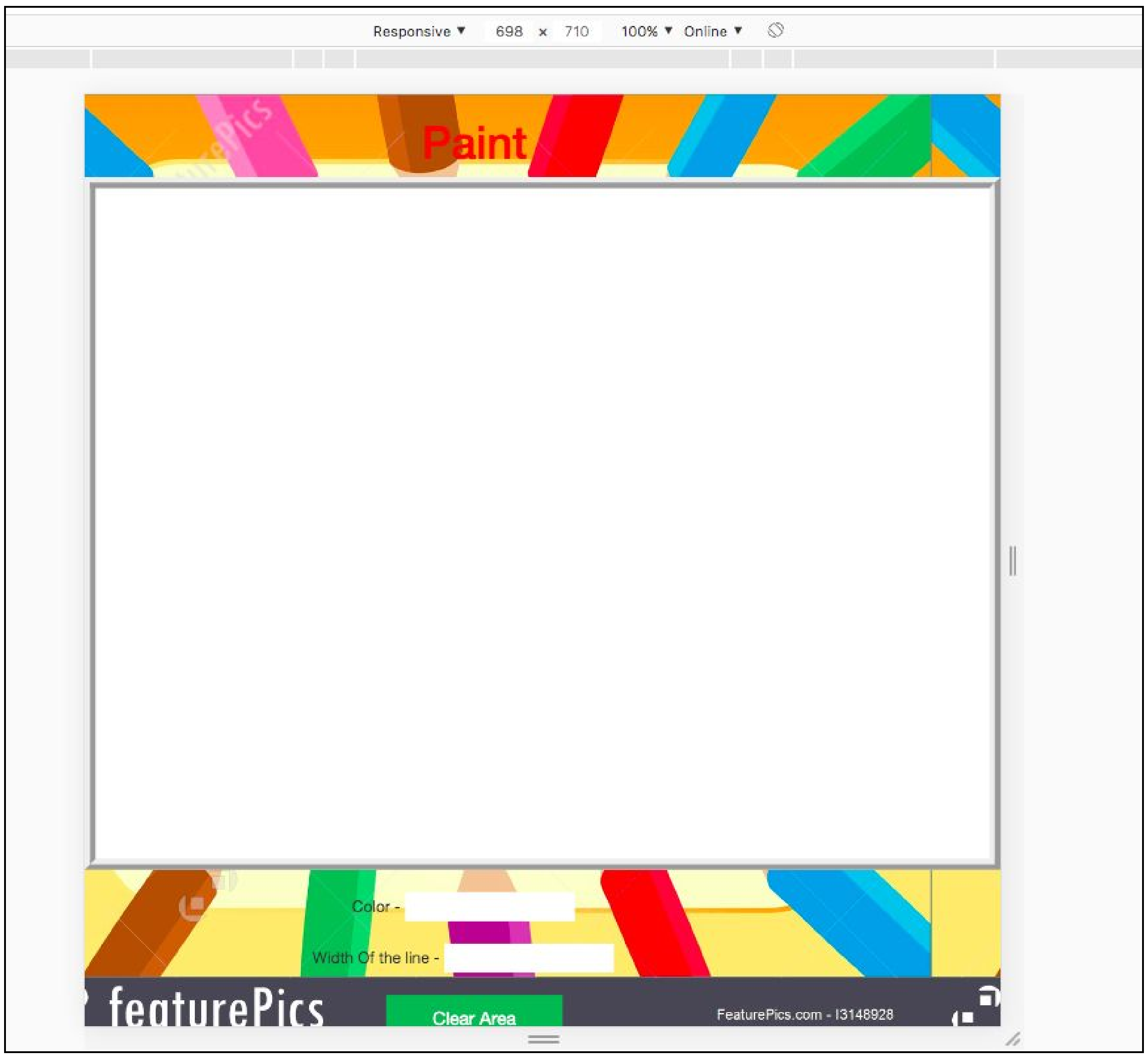
2. Now click on the mobile/tab icon as per the below image



```
<html>
 ▶<head>…</head>
 ▼<body class="body_backgorund">
    <!-- class="body_backgorund"-->
   ▼<div align="center">
      <h1>Paint</h1>
...   <canvas id="myCanvas" width="800" height="600"> == $0
      <br>
      <br>
      <!--Additional activity -->
      <span>Color - </span>
      <input type="text" id="color">
      <br>
      <br>
      <span>Width Of the line  - </span>
      <input type="text" id="width_of_line">
      <br>
      <br>
      <button onclick="clearArea();">Clear Area</button>
      <!--End Additional activity -->
    </div>
    <script src="main.js"></script>
  </body>
</html>
```

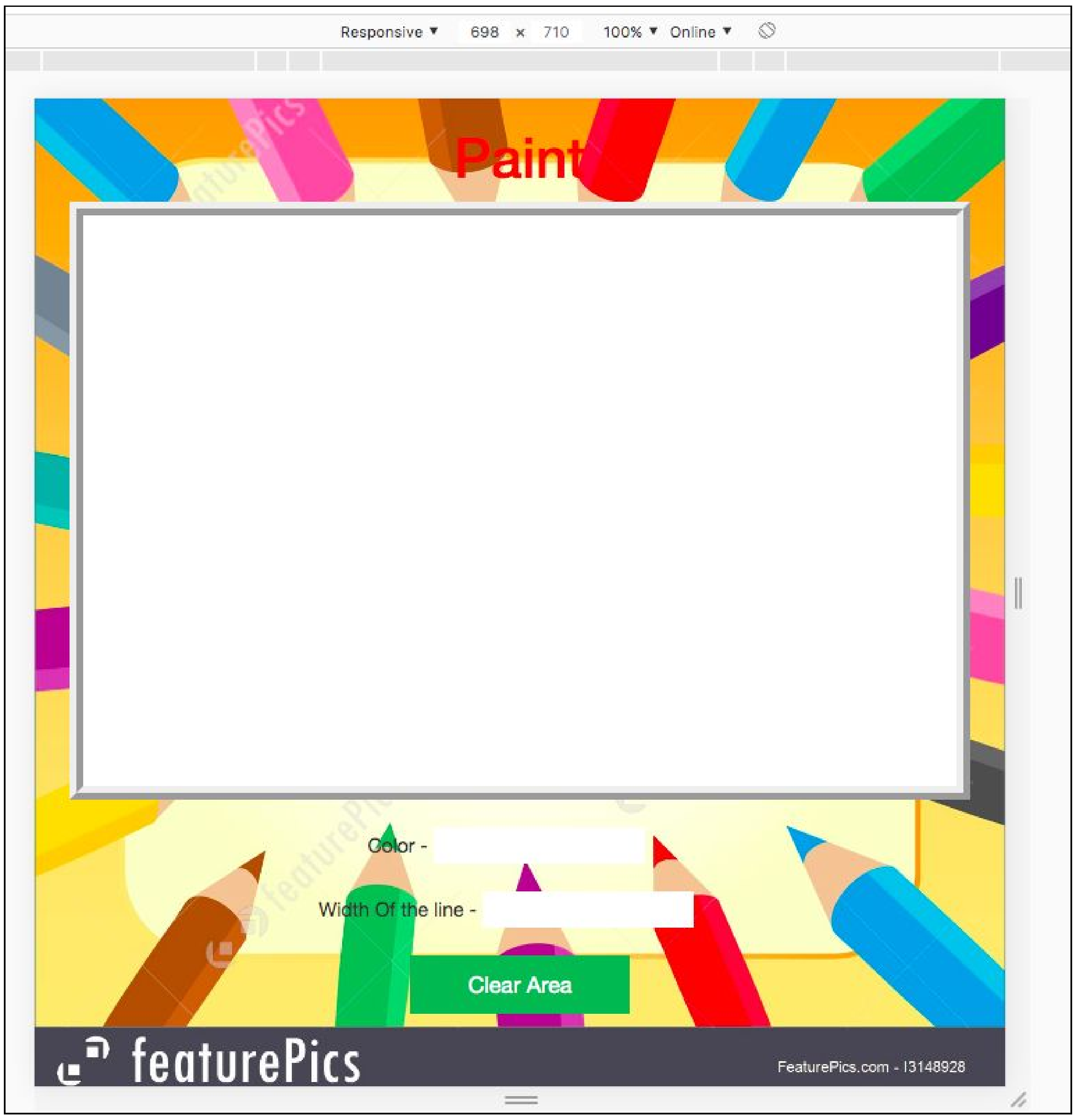
This will get the website to a mobile screen view

Like this -



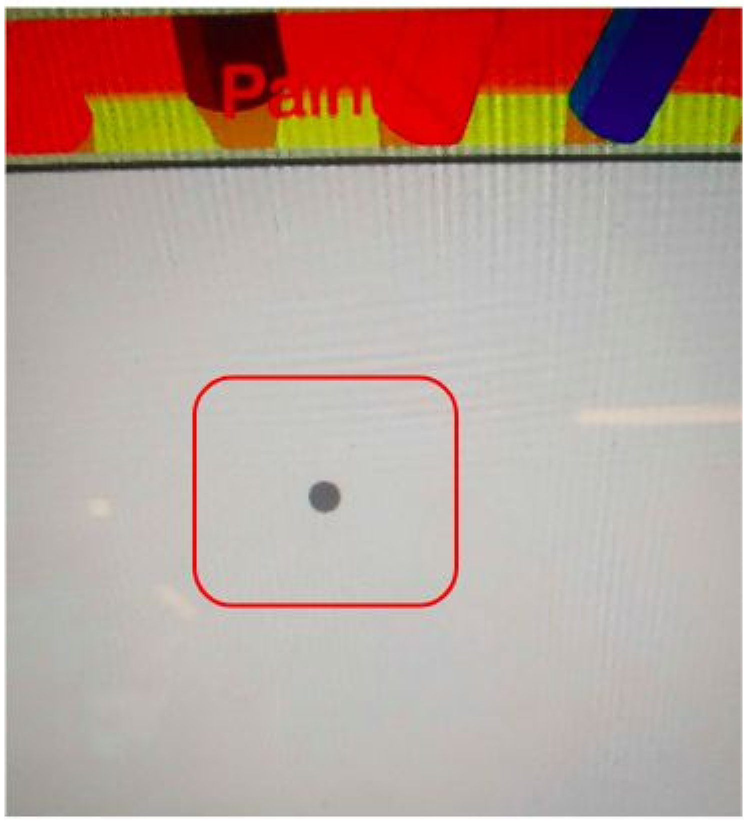
Responsive ▾ 698 × 710 100% ▾ Online ▾
Paint
Color -
Width Of the line -
Clear Area
featurePics
FeaturePics.com - I3148928

3. Now do refresh, you will notice the canvas has resized itself as per the screen size

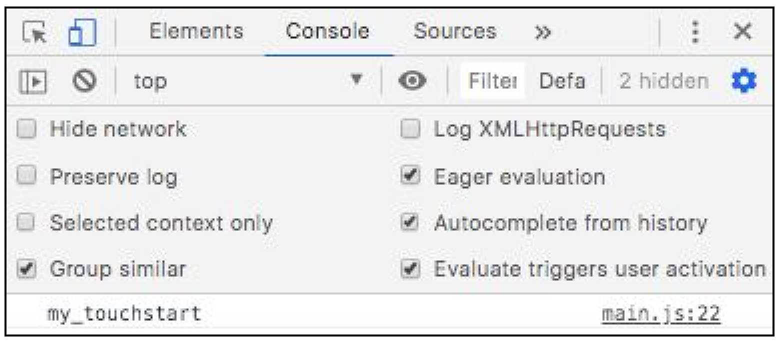


Responsive ▾ 698 × 710 100% ▾ Online ▾
Paint
Color -
Width Of the line -
Clear Area
featurePics
FeaturePics.com - I3148928

4. So when you move the mouse inside the website the mouse cursor will change to the below image. This icon will act the same as your finger moving on the screen, so when we code and test , we will use this as our finger on screen.



5. Just click inside the canvas and leave the click, and see on the console screen a message will come saying **my_touchstart-**

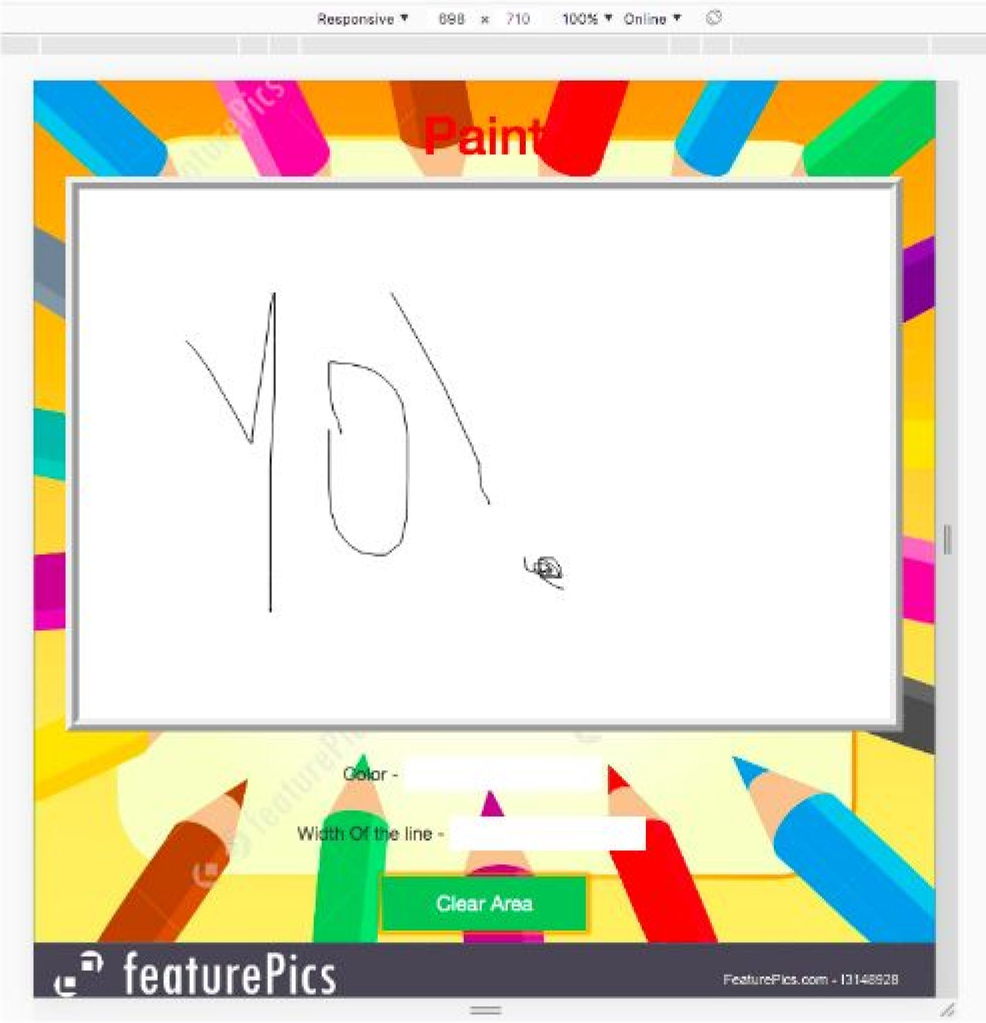


□ Hide network
□ Preserve log
□ Selected context only
☑ Group similar
□ Log XMLHttpRequests
☑ Eager evaluation
☑ Autocomplete from history
☑ Evaluate triggers user activation

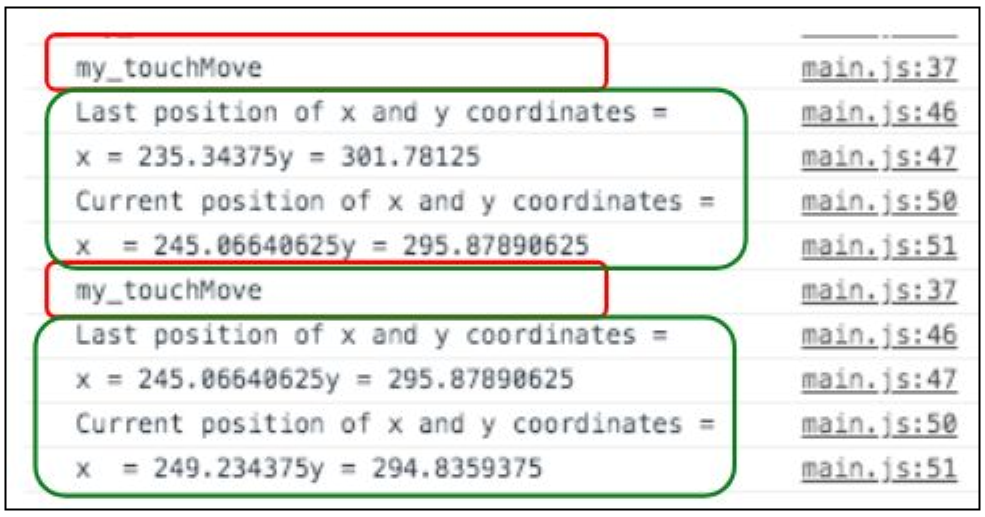
```
my_touchstart                                        main.js:22
```

This means the touch is working

6. Now click and move the mouse inside the canvas and it will paint, this mouse click and moving means, touching on the mobile screen and moving the finger.

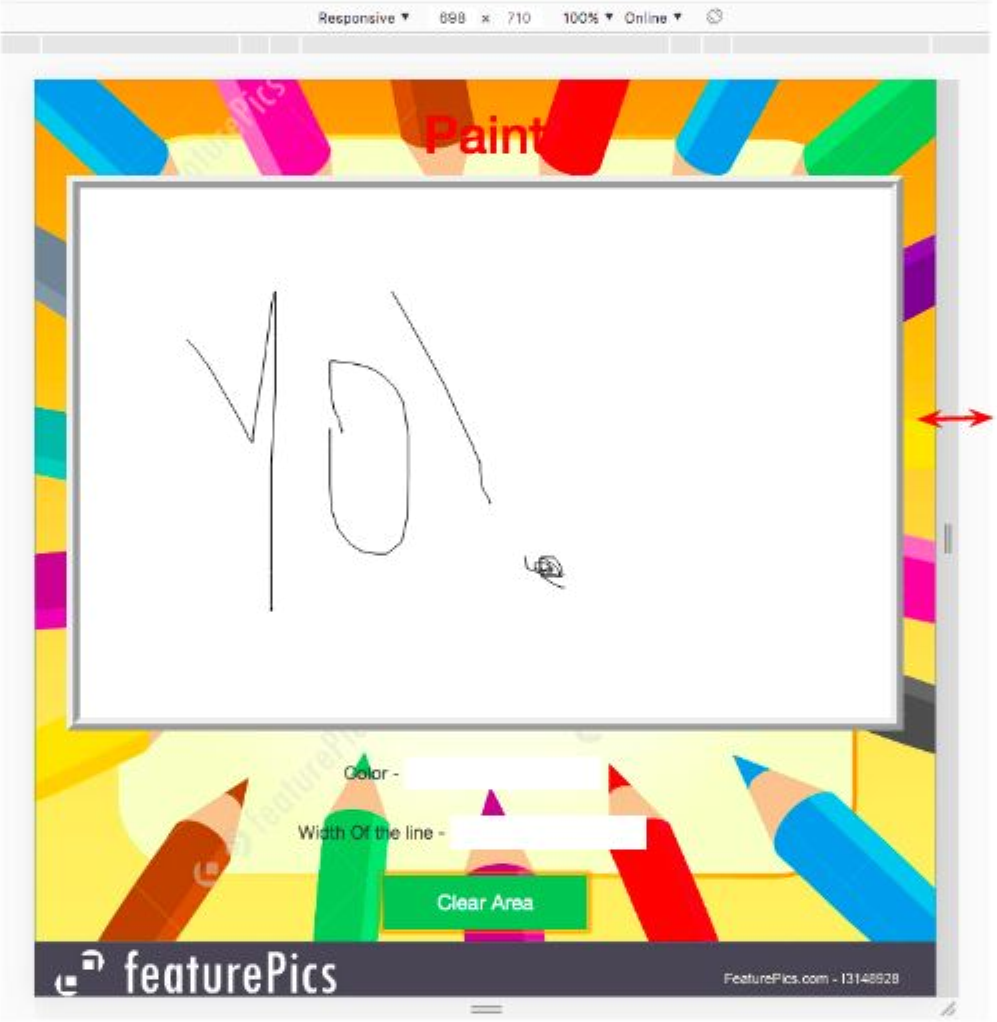


7. As you draw you will see on the console screen that messages are printed saying **my_touchmove** and the **last** and **current** x and y and coordinates, same as the previous class.



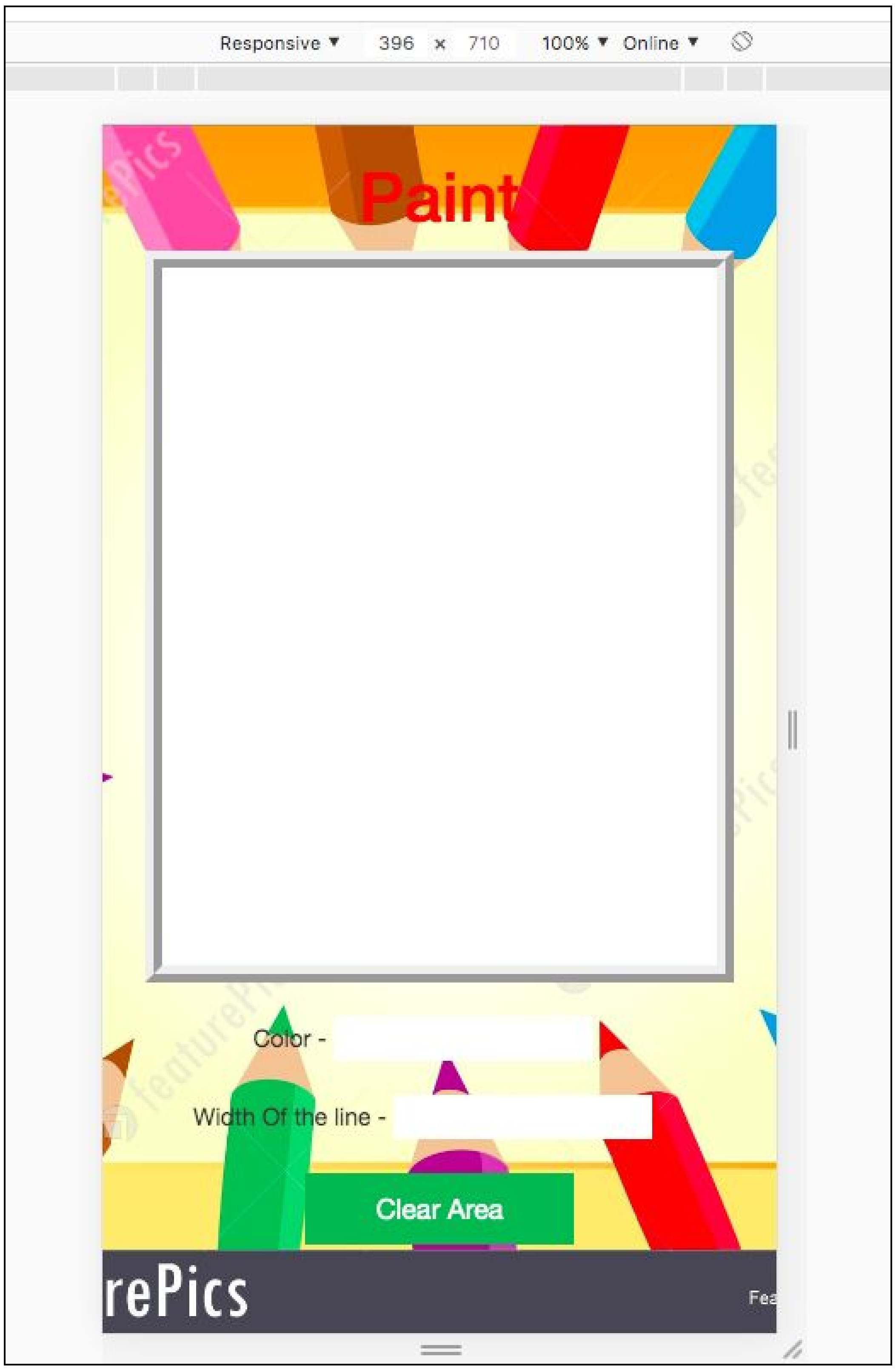
```
my_touchMove                                        main.js:37
Last position of x and y coordinates =              main.js:46
x = 235.34375y = 301.78125                          main.js:47
Current position of x and y coordinates =           main.js:50
x  = 245.06640625y = 295.87890625                   main.js:51
my_touchMove                                        main.js:37
Last position of x and y coordinates =              main.js:46
x = 245.06640625y = 295.87890625                    main.js:47
Current position of x and y coordinates =           main.js:50
x  = 249.234375y = 294.8359375                      main.js:51
```

8. Now reduce the width on the website as per the following image-



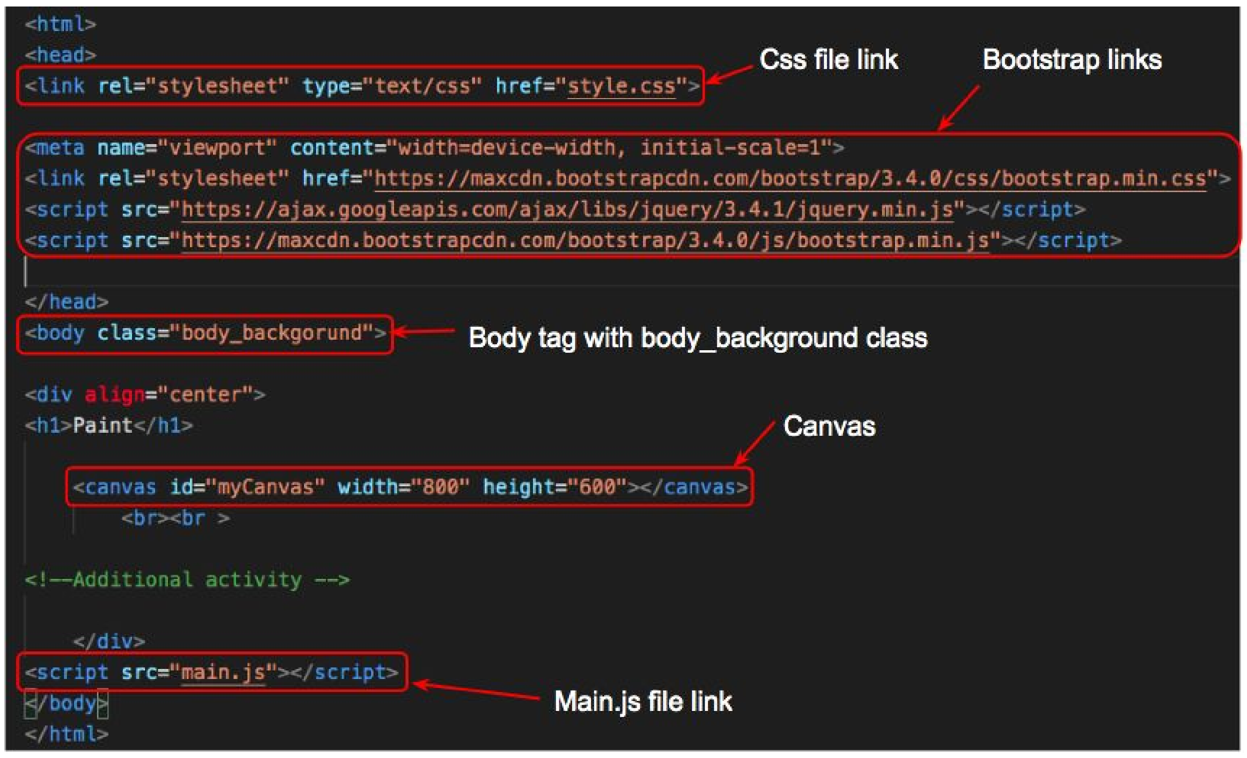
Responsive ▼ 898 × 710 100% ▼ Online ▼

9. Now again refresh the page, and you will see that again canvas has resized as per the screen size



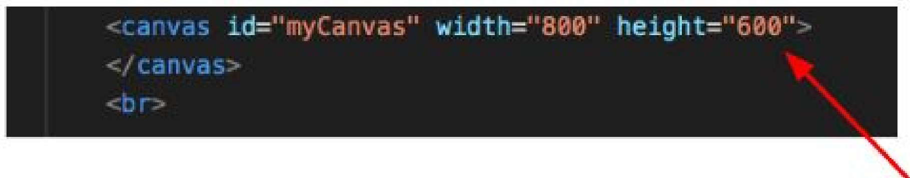
So we will follow the above steps while we write the code and while testing the code

- **HTML code prewritten**



```
<html>
<head>
<link rel="stylesheet" type="text/css" href="style.css">

<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/css/bootstrap.min.css">
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/js/bootstrap.min.js"></script>

</head>
<body class="body_backgorund">

<div align="center">
<h1>Paint</h1>

    <canvas id="myCanvas" width="800" height="600"></canvas>
        <br><br >

<!--Additional activity -->

    </div>
<script src="main.js"></script>
</body>
</html>
```

Output -

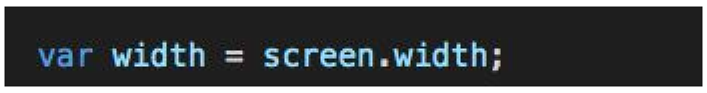


```
<canvas id="myCanvas" width="800" height="600">
</canvas>
<br>
```

The above HTML code had -

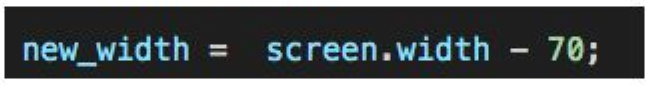
- Link for the style sheet
- Bootstrap links
- Body tag with body_background class
- Canvas
- Link for main.js code

- **Fetching the width of the screen**



```
var width = screen.width;
```

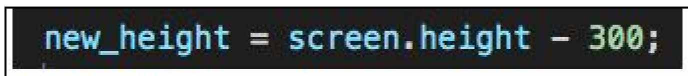
- **Setting the values to new_width and new_height variable**



```
new_width =  screen.width - 70;
```

For example -

- The width of the screen is 800px
  new_width = screen.width - 70 // **800 - 70**
  So **new_width** = 730

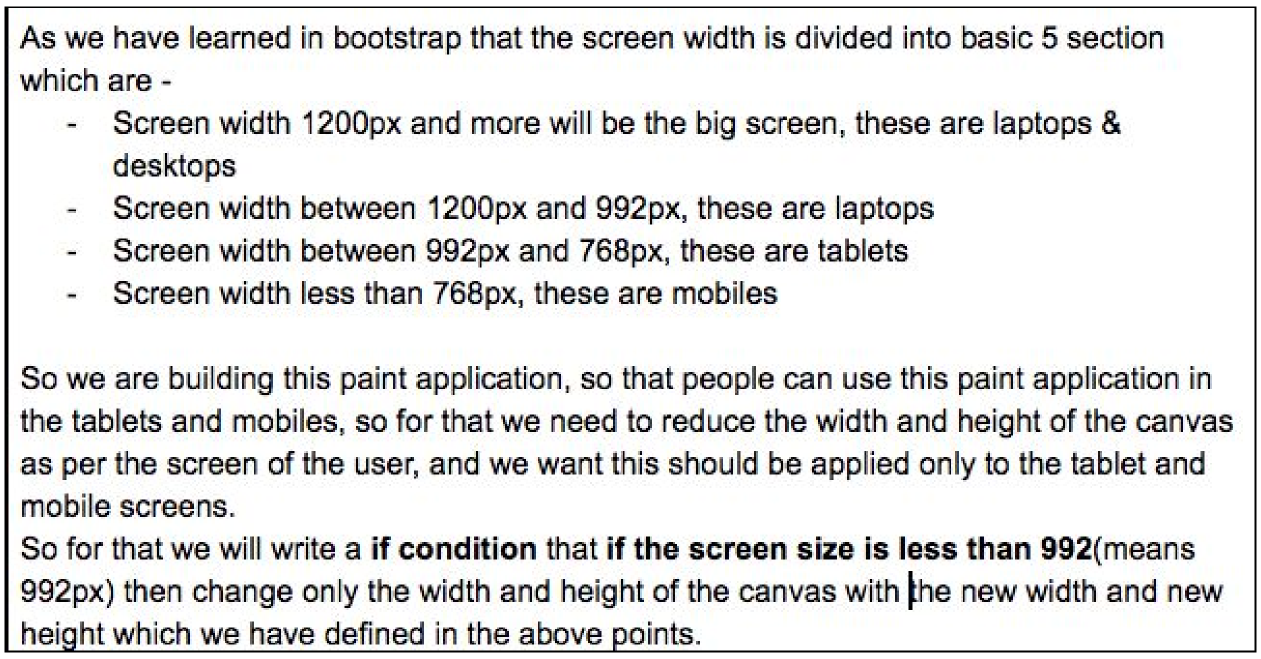


```
new_height = screen.height - 300;
```

For example -

- The height of the screen is 700px
  new_height = screen.height - 300 // **700 - 300**
  So **new_height** = 400

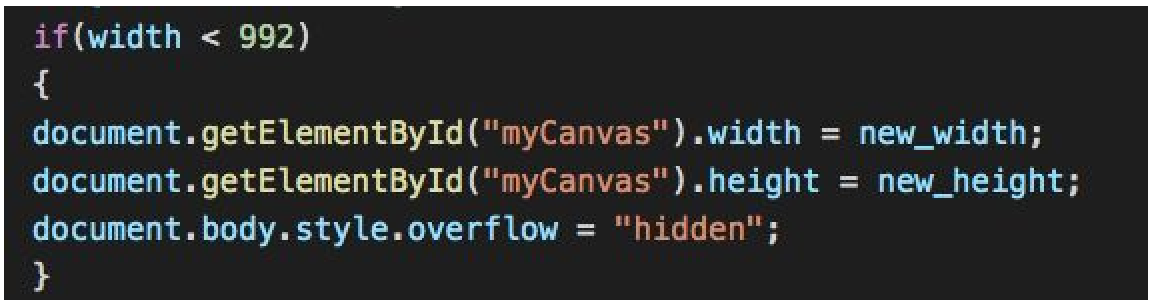
- **Changing the width and height of the canvas if the screen with is less than 992px**



As we have learned in bootstrap that the screen width is divided into basic 5 section which are -

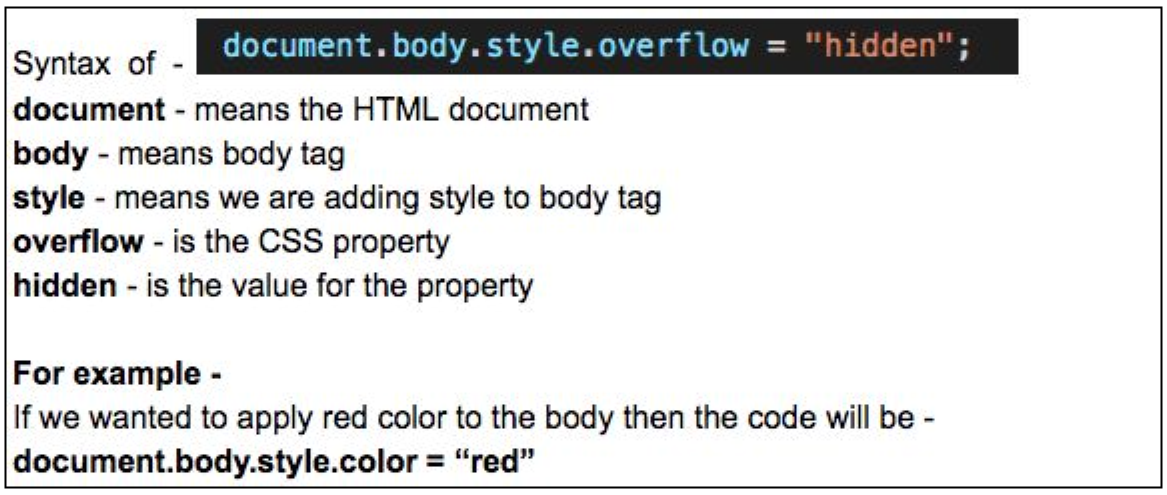
- Screen width 1200px and more will be the big screen, these are laptops & desktops
- Screen width between 1200px and 992px, these are laptops
- Screen width between 992px and 768px, these are tablets
- Screen width less than 768px, these are mobiles

So we are building this paint application, so that people can use this paint application in the tablets and mobiles, so for that we need to reduce the width and height of the canvas as per the screen of the user, and we want this should be applied only to the tablet and mobile screens.
So for that we will write a **if condition** that **if the screen size is less than 992**(means 992px) then change only the width and height of the canvas with the new width and new height which we have defined in the above points.



```
if(width < 992)
{
document.getElementById("myCanvas").width = new_width;
document.getElementById("myCanvas").height = new_height;
document.body.style.overflow = "hidden";
}
```

- **Explaining overflow**

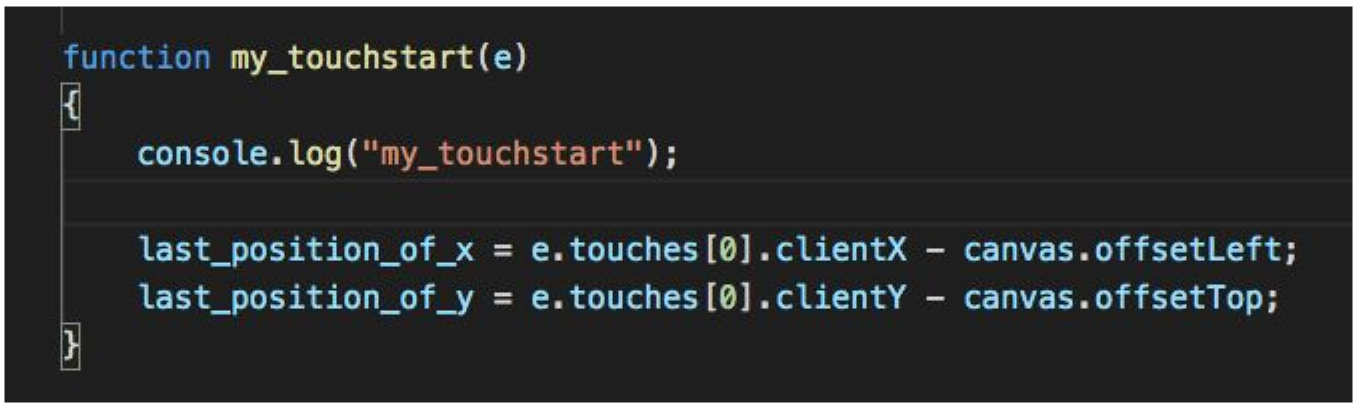


Syntax of - `document.body.style.overflow = "hidden";`

document - means the HTML document
body - means body tag
style - means we are adding style to body tag
overflow - is the CSS property
hidden - is the value for the property

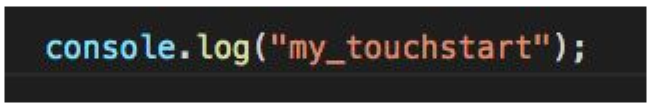
For example -
If we wanted to apply red color to the body then the code will be -
document.body.style.color = “red”

- **Code for function my_touchstart()**

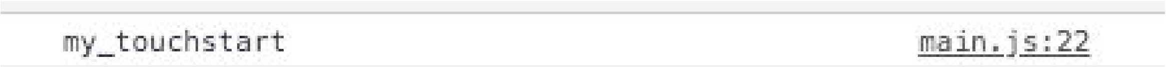


```
function my_touchstart(e)
{
    console.log("my_touchstart");

    last_position_of_x = e.touches[0].clientX - canvas.offsetLeft;
    last_position_of_y = e.touches[0].clientY - canvas.offsetTop;
}
```

Output -

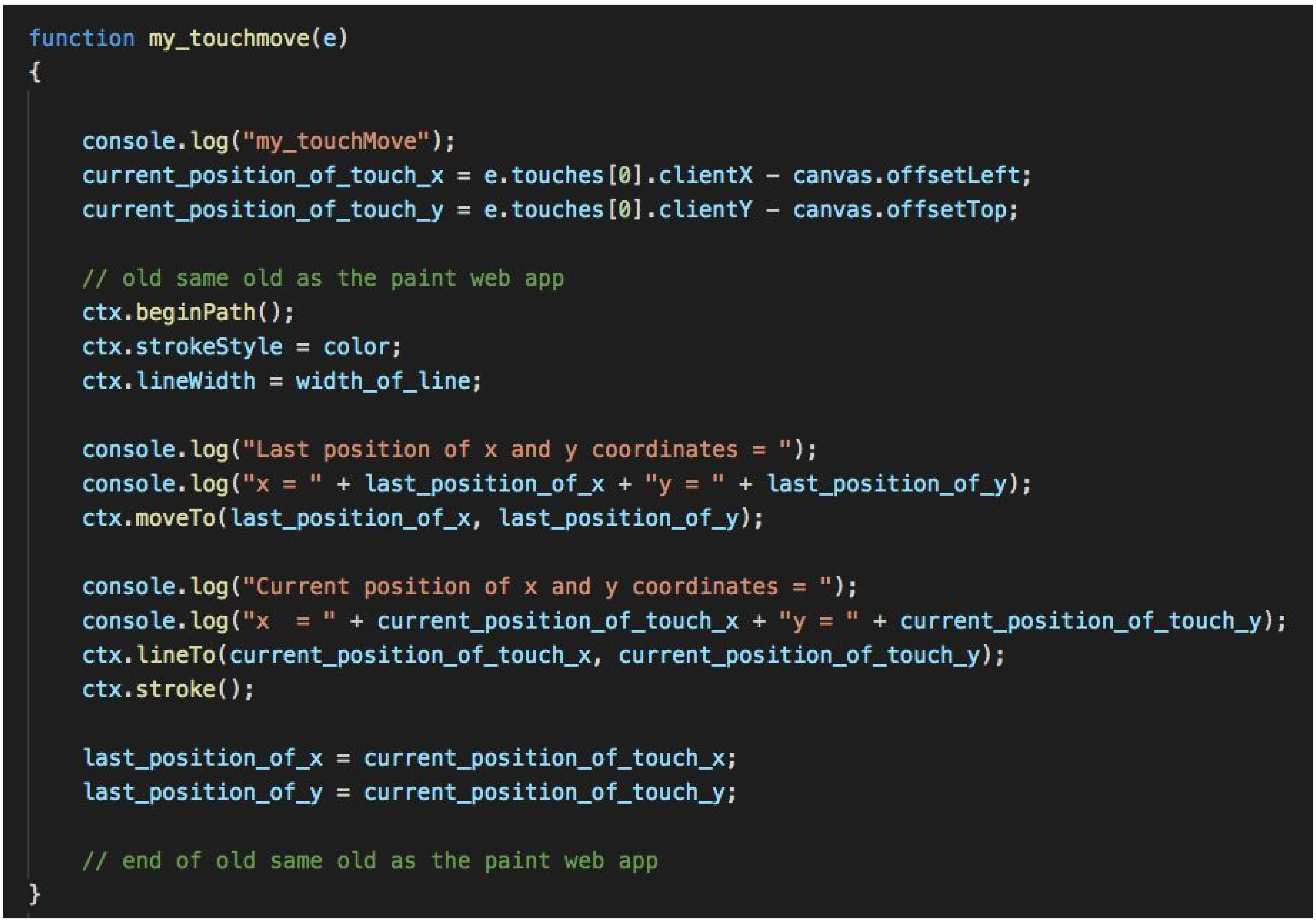


```
console.log("my_touchstart");
```



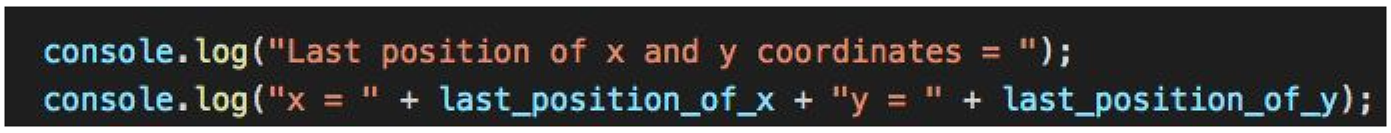
```
my_touchstart                                        main.js:22
```

- **Code for function my_touchmove()**

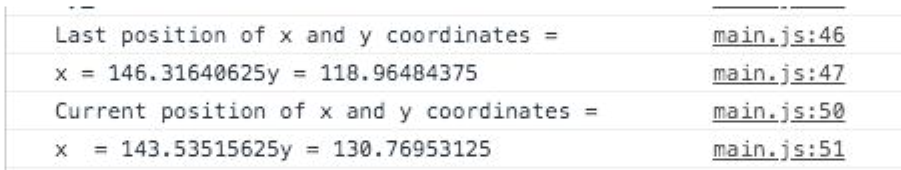


```
function my_touchmove(e)
{

    console.log("my_touchMove");
    current_position_of_touch_x = e.touches[0].clientX - canvas.offsetLeft;
    current_position_of_touch_y = e.touches[0].clientY - canvas.offsetTop;

    // old same old as the paint web app
    ctx.beginPath();
    ctx.strokeStyle = color;
    ctx.lineWidth = width_of_line;

    console.log("Last position of x and y coordinates = ");
    console.log("x = " + last_position_of_x + "y = " + last_position_of_y);
    ctx.moveTo(last_position_of_x, last_position_of_y);

    console.log("Current position of x and y coordinates = ");
    console.log("x  = " + current_position_of_touch_x + "y = " + current_position_of_touch_y);
    ctx.lineTo(current_position_of_touch_x, current_position_of_touch_y);
    ctx.stroke();

    last_position_of_x = current_position_of_touch_x;
    last_position_of_y = current_position_of_touch_y;

    // end of old same old as the paint web app
}
```

Output of -

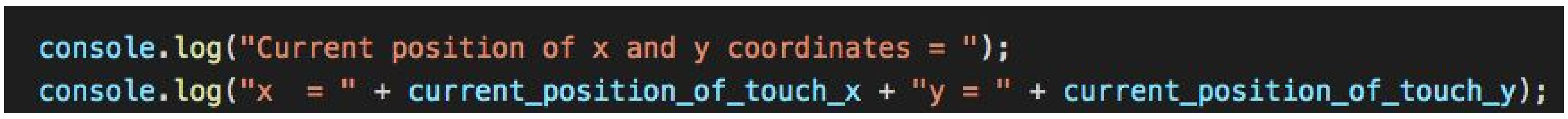


```
console.log("Last position of x and y coordinates = ");
console.log("x = " + last_position_of_x + "y = " + last_position_of_y);
```

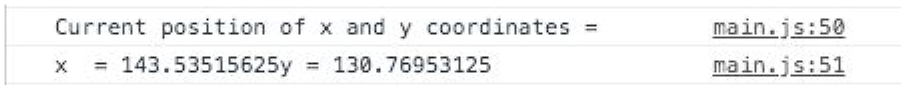


```
Last position of x and y coordinates =              main.js:46
x = 146.31640625y = 118.96484375                    main.js:47
Current position of x and y coordinates =           main.js:50
x  = 143.53515625y = 130.76953125                   main.js:51
```

Output of -



```
console.log("Current position of x and y coordinates = ");
console.log("x  = " + current_position_of_touch_x + "y = " + current_position_of_touch_y);
```



```
Current position of x and y coordinates =           main.js:50
x  = 143.53515625y = 130.76953125                   main.js:51
```