


Name: Ahmed Hamada Soliman Mohamed

Section: 1

Number: 13

Screenshot

RGB Color

Resulting Color: 

Html code

```
<!DOCTYPE html>
<html>
  <head>
    <title>Color Genrator</title>
    <link rel="stylesheet" href="styles.css">
  </head>
  <body>
    <h1>RGB Color</h1>

    <div class="red">
      <label> Red: &emsp; &emsp; </label>
      <input type="range" max="255" min="0" value="0" step="1"
id="set_red" onchange="get_red()" >
      <a> &emsp; &emsp; &emsp; Value = </a>
      <output id="show_red">0</output>
    </div>
    <div class="green">
      <label> Green: &emsp; </label>
      <input type="range" max="255" min="0" value="0" id =
"set_green" onchange="get_green()" >
```

```

        <a>&emsp;&emsp;&emsp;Value = </a>
        <output id="show_green">0</output>
    </div>
    <div class="blue">
        <label> Blue:&emsp;&emsp;</label>
        <input type="range" max="255" min="0" value="0" id =
"set_blue" onchange="get_blue()">
        <a>&emsp;&emsp;&emsp;Value = </a>
        <output id="show_blue">0</output>
    </div>
    <br/>
    <a>Resulting Color:</a>
    <div class="color_box" id="result"></div>
</body>

<script src="main.js">

</script>

</html>

```

CSS Code

```

body {
    font-family: "Comic Sans MS", cursive, sans-serif;
}
.red {
    background-color:red;
    padding: 10px;
    margin-right: 70%;
    border-radius: 10px;
}
.blue {
    background-color:blue;
    padding: 10px;
    margin-right: 70%;
    border-radius: 10px;
}
.green {

    background-color:green;
    padding: 10px;
    margin-right: 70%;
    border-radius: 10px;
}
.color_box{
    display: inline-block;
    background-color: rgb(0,0,0);
    border: thin solid dimgray;
    width: 50px;
    height: 20px;
    text-align: right;
    line-height: 15px;
}

```

```

padding: 5px;
vertical-align: middle
}

input[type=range] {
width: 200px;
cursor: pointer;
position: relative;
}

input[type=range]::before,
input[type=range]::after {
display: inline-block;
position: absolute;
}

input[type=range]::before {
content: attr(min);
left: 0;
transform: translate(calc(-100% - 5px))
}

input[type=range]::after {
content: attr(max);
right: 0;
transform: translate(calc(100% + 5px))
}

```

JavaScript Code

```

function get_red()
{
var set = document.getElementById("set_red").value;
document.getElementById("show_red").value = set;
}

function get_blue()
{
var set = document.getElementById("set_blue").value;
document.getElementById("show_blue").value = set;
}

function get_green()
{
var set = document.getElementById("set_green").value;
document.getElementById("show_green").value = set;
}

var input = document.querySelectorAll("input");
for(var i = 0; i < input.length; i++){
input[i].addEventListener("input", function(){
var red = document.getElementById("set_red").value,
green = document.getElementById("set_green").value,

```

```
        blue = document.getElementById("set_blue").value;
        var display = document.getElementById("result");
        display.style.background = "rgb(" + red + ", " + green +
", " + blue + ")";
    });
}
```