|  |
| --- |
| HTML |
| <!DOCTYPE html>  <html lang="en">  <head>      <meta charset="UTF-8">      <meta name="viewport" content="width=device-width, initial-scale=1.0">      <link rel="stylesheet" href="./colorGame.css">      <link href="https://fonts.googleapis.com/css2?family=Montserrat:wght@100;300;900&display=swap" rel="stylesheet">      <title>Color Game</title>  </head>  <body>      <h1>          The Great          <br>          <span id="colorDisplay">RGB</span>          <br>          Color Game      </h1>      <div id="stripe">          <button id="reset">New Colors</button>          <span id="message"></span>          <button class="mode">Easy</button>          <button class="mode selected">Hard</button>      </div>      <div id="container">          <div class="square"></div>          <div class="square"></div>          <div class="square"></div>          <div class="square"></div>          <div class="square"></div>          <div class="square"></div>      </div>      <script src="./colorGame.js"></script>  </body>  </html> |
| CSS |
| body {      background-color: #232323;      margin: 0;      font-family: "Montserrat", "Avenir";}  .square {      width: 30%;      background: purple;      padding-bottom: 30%;      float: left;      margin: 1.66%;      border-radius: 15%;      transition: background 0.6s;      --webkit-transition: background 0.6s;/\* safari \*/      --moz-transition: background 0.6s;/\* mozilla \*/}  #container {      margin: 20px auto;      max-width: 600px;}  h1 {      text-align: center;      line-height: 1.1;      font-weight: normal;      color: white;      background: steelblue;      margin: 0;      text-transform: uppercase;      padding: 20px 0;}  #colorDisplay {      font-size: 200%;}  #message {      display: inline-block;      width: 20%;}  #stripe {      background: white;      height: 30px;      text-align: center;      color: black;}  .selected {      color: white;      background: steelblue;}  button {      border: none;      background: none;      text-transform: uppercase;      height: 100%;      font-weight: 700;      color: steelblue;      letter-spacing: 1px;      font-size: inherit;      transition: all 0.3s;      --webkit-transition: all 0.3s;/\* safari \*/      --moz-transition: all 0.3s;/\* mozilla \*/      outline: none;}  button:hover {      color: white;      background: steelblue;} |

|  |
| --- |
| javaScript |
| var game = {};  game.init = function () {      setupModeButton();      setupSquares();      reset();}  //add value to fix problem with mode game and quantity squares in mode game  var numSquares = 6;  //create arr with random color  var colors = [];  //add function pickColor()  var pickedColor;  //selectors for manipulation DOM  var squares = document.querySelectorAll(".square");  //add input to catch element in squares  var colorDisplay = document.getElementById("colorDisplay");  //add information to UX -> inform user for right choise  var messageDisplay = document.querySelector("#message");  //add information about color that we want to find is square  var h1 = document.querySelector("h1");  //button for start new game  var resetButton = document.querySelector("#reset");  //add two button for change level game(easy or hasr)  /\* var easyBtn = document.querySelector("#easyBtn");  var hardBtn = document.querySelector("#hardBtn"); \*/  var modeButtons = document.querySelectorAll(".mode");  game.init();  function setupModeButton () {      for(var i = 0; i < modeButtons.length; i++) {          modeButtons[i].addEventListener("click", function () {              modeButtons[0].classList.remove("selected");              modeButtons[1].classList.remove("selected");              this.classList.add("selected");              this.textContent === "Easy" ? numSquares = 3 : numSquares = 6;              reset();              });      }}  function setupSquares () {      for(var i = 0; i < squares.length; i++) {          //add initial colors to square          squares[i].style.backgroundColor = colors[i];          //add click listeners to squares          squares[i].addEventListener("click", function () {              // grab color of clicked square              var clickedColor = this.style.background;              //compare color to pickedColor              if(clickedColor === pickedColor) {                  messageDisplay.textContent = "Correct!";                  resetButton.textContent = "Play again?";                  changeColors(clickedColor);                  h1.style.background = pickedColor;              } else {                  //add method to change background color to empty space                  this.style.background = "#232323";                  messageDisplay.textContent = "Try again!";}});}}   function reset() {      //generate all new colors      colors = generateRandomColors(numSquares);      //pick a new randon color from array      pickedColor = pickColor();      //change color display to match picked Color      colorDisplay.textContent = pickedColor;      //change textContent to default value      resetButton.textContent = "New Colors";      messageDisplay.textContent = "";      //change colors of squares      for(var i = 0; i < squares.length; i++) {          //add initial colors to square          if(colors[i]) {              squares[i].style.display = "block";              squares[i].style.background = colors[i];          } else {              squares[i].style.display = "none";          }          //change display.style to default value(black)      }      h1.style.backgroundColor = "steelblue";};   resetButton.addEventListener("click", function () {      reset();})     //view the num of color in display   //colorDisplay.textContent = pickedColor;   //view -> change colors than user guess the color  function changeColors (color) {           //loop though all squares           for(var i = 0; i < squares.length; i++) {               //change each color to match given color               squares[i].style.background = color;           }}       //create a random color for user, what he must guess  function pickColor () {           var random =  Math.floor(Math.random() \* colors.length);           return colors[random];}       //random colors in square to give the use choise to guess  function generateRandomColors (num) {           //make an array           var arr = [];           //add num random colors to array           for (var i = 0; i < num; i++) {               arr.push(randomColor());               //get random color and push to arr}           //return that array           return arr;}       //create random colors for square  function randomColor () {           //pick a "red" 0 - 255           var r = Math.floor(Math.random() \* 256)           //pick a "green" 0 - 255           var g = Math.floor(Math.random() \* 256)           //pick a "blue" 0 - 255           var b = Math.floor(Math.random() \* 256)           return `rgb(${r}, ${g}, ${b})`;           //return "rgb(" + r + "," + g + "," + b + ")"; } |