

Questão 01

R- Basta declarar a variável usando o termo `let` seguido do nome da variável e em seguida atribuir um valor a ela dentro do `{}` em qualquer lugar no código.

Questão 02

R - Para definir uma variável global usando **var**, basta que eu declare ela usando **var** seguido do nome da minha variável fora de qualquer função e atribua um valor a essa variável. Perceba, declarar é dizer “`let/var/const minha_notas;`” definir é atribuir um valor a essa variável “`minha_notas = 10;`” Muitas vezes quando declaramos nós já definimos, como por exemplo: “`let/var/const minha_notas = 10;`”

```
var glob = 10; // N ta dentro de nenhuma func
function setup(){...}
function draw(){...}
function inventada(){...}
```

Poderia ser usado `var` ou `let` caso eu queira que a variável seja modificada, ou poderia ter usado `const` para não ser possível alterar, como elas foram declaradas fora das funções e blocos, todas elas estariam em um escopo global, ou seja, acessível para todo meu programa.

Questão 03

R- O escopo de bloco como o próprio nome já diz, indica que uma variável só vai existir enquanto ela estiver dentro do escopo `{}`, ou seja, entre esses dois bloquinhos/chaves `ai`. O escopo de função é semelhante com o de bloco, mas ao invés de a variável existir somente entre esse `{}`, ele vai existir somente entre os `{}` que vier acompanhado da função.

```
function setup(){ // o { ta vindo acompanhado da
                  // função então aqui é o escopo
                  // de função, mais precisamente
                  // o escopo da função setup
                  // toda variável ou constante
                  // definida aqui só existirá
                  // aqui dentro. N posso usá-la
                  // em draw() ou em inventada()
                  // ou em qualquer outro lugar
}
{
  // Criei um bloco {} do nada, tudo que eu defini
  // n'í aqui com let/const vai estar no meu escopo
}
```

```

    // de bloco, desse bloco especificamente.
    // os lets/consts que forem definidos aqui
    // não morrer qndo eu passar do }
    // OBS: Declarar com o var dentro do escopo de
    // bloco vai tornar var acessível no escopo
    // imediatamente superior ao qual o bloco está
    // inserido
}

function draw(){...}
function inventada(){...}

```

Questão 4

R- Em qualquer lugar do código eu posso escrever “const nome_da_variavel = valor_qualquer” e essa variável será constante, não poderá sofrer alterações e estará no escopo do local que eu declarei ela. O que muda em let e var é que essas variáveis, quando declaradas com let/var poderão ser reatribuídas/redefinidas ao longo do código

Questão 5

R - Variáveis declaradas com var podem ser usadas antes mesmo da definição, pois o javascript “Iça” a declaração dessas variáveis para o topo do escopo no qual ela foi declarada. É importante notar que se ela nunca foi definida até o momento de uso, o javascript identificará seu valor como undefined, mas deixará seu programa rodar mesmo assim, o que pode causar muitos problemas. Tudo isso citado com var não ocorre com let.

Questão 6

R - Sim, é possível. Ao definir uma variável dentro de uma função f1 com let, eu indico para meu código que dentro do escopo daquela função, minha variável vai estar acessível para modificação. Ai, eu declaro uma outra função interna f2 (ela está dentro da função f1), essa função interna f2, por estar dentro do meu escopo da função externa f1 vai ter acesso a tudo que foi declarado pela função externa f1.

Questão 7

R -

```

let slider_tamanho; // Declarei
let slider_px; // Declarei

```

```

let slider_py; // Declarei

function setup(){
  createCanvas(800,800);
  slider_px = createSlider(0,width); // Definir
  slider_py = createSlider(0,height); // Definir = Atribuir
  slider_tamanho = createSlider(0,50); // Definir = Atribuir
  slider_px.position(50,50);
  slider_py.position(50,100);
  slider_tamanho.position(50,150);
}
function draw(){
  background(0);
  fill(122);
  ellipse(slider_px.value(),slider_py.value(),slider_tamanho.value(),
  slider_tamanho.value());
}

```

Questão 8

```

let slider_tamanho; // Declarei
let slider_px; // Declarei
let slider_py; // Declarei
let color_picker; // Declarei

function setup(){
  createCanvas(800,800);
  slider_px = createSlider(0,width); // Definir
  slider_py = createSlider(0,height); // Definir = Atribuir
  slider_tamanho = createSlider(0,50); // Definir = Atribuir
  color_picker = createColorPicker('#ffffff');
  slider_px.position(50,50);
  slider_py.position(50,100);
  slider_tamanho.position(50,150);
}
function draw(){
  background(0);
  fill(color_picker.color());
  rect(slider_px.value(),slider_py.value(),slider_tamanho.value(),
  slider_tamanho.value());
}

```

Questão 9

```
let c1 = {}
let c2 = {}
function setup(){
  createCanvas(400,400);
  c1= createColorPicker("black");
  c2= createColorPicker("green");

}
function draw(){
  background(255);
  textSize(25);
  noStroke();
  text("Q09", 0, 25);
  text("Cor 1", 12, 100);
  c1.position(25,50);

  text("Cor 2", 88, 100);
  c2.position(100,50);

  for (let i = 0; i < 300; i++){
    let a = lerpColor(c1.color(),c2.color(), i*1/(300));
    stroke(a);
    line(i,height/2-20,i,height/2)
  }

}
```

Questão 10

```
let pointsX = [];
let pointsY = [];
let sliders = [];
function setup(){
  createCanvas(600,600)
  pointsX = [500, 510, 520, 530, 540];
  pointsY = [ 0, 100, 80 , 300, 100]
  for (let i = 0; i < pointsX.length; i++){
    sliders.push(createSlider(0,width,pointsX[i]));
    sliders[i].position(25, 80 + 15 * i);
  }
}
```

```

    for (let i = 0; i < pointsY.length; i++){
        let pos = pointsX.length + i;
        sliders.push(createSlider(0,height,pointsY[i]));
        sliders[pos].position(200, 80 + 15 * i);
    }
}

function draw(){

    background(255);
    fill(0);
    textSize(10);
    text("Q10", 0, 25);
    for (let i = 0; i < sliders.length/2; i++) {
        text(sliders[i].value() + "x", 25 + 150, 92 + 15 * (i));
        text(sliders[i+pointsY.length].value() + "y", 200 + 150, 92 + 15 * i);
    }

    fill(0);
    beginShape();
    for (let i = 0; i < sliders.length/2; i++) {
        vertex(sliders[i].value(), sliders[i+pointsY.length].value());
    }
    endShape(CLOSE);
}

```