**REVISÃO**

**IF TERNÁRIO**

**Questão 1**

A) let idade = 36;

let resposta = ‘’;

idade >= 18 ? resposta = ‘Maior de idade pode tirar carteira’ : resposta = ‘Menor de idade pode não tirar carteira’

alert(resposta);

B) let temperatura = 18;

let resposta = ‘’;

temperatura >= 0 ? resposta = ‘Temperatura Maior ou igual a zero’’ : resposta = ‘ Temperatura negativa’

C) let err = 4;

err > 0 ? ((err == 1) ? "1 erro detectado." : @err + "erros detectados") : "0 nenhum detectado.";

**SINTAXE ARROW FUNCTION**

1. Já está em uma sintaxe simplificada.
2. A)- imc = (peso, altura) => {

return peso / (altura \* altura);

}

B)- jurosSimples = (C, i, t) => {

return C i t;

}

C)- palavra = (palavraA, palavraB) => {

return palavraA.lenght => palavraB.lenght ? ((palavraA.lenght == palavraB.lenght) ? 0 : 1) : 2;

}

D) var palavra = (palavraA) => {

return palavraA.split('').reverse().join('');

}

let numbers = [ 1, 16, 36, 49, 81, 100];

let raizQuadrada = numbers.map(num => Math.sqrt(num))

console.log(raizQuadrada);

let numbers = [ 1, 16, 36, 49, 81, 100];

function isBigEnough(value) {

return value >= 50;

}

let raizQuadrada = numbers.filter(isBigEnough)

console.log(raizQuadrada);

const numbers = [ 1, 16, 36, 49, 81, 100];

const reducer = (accumalator, currentValue) => accumalator + currentValue;

console.log(numbers.reduce(reducer));

**ESCOPO DE VARIÁVEIS**

1. Em poucas palavras, escopo é a propriedade que determina onde uma variável pode ser utilizada dentro de um programa.
2. Quando você declara uma variável com a palavra *var,* ela pode corresponder ao escopo global e local, já quando você faz isso com a palavra *let*, ela pode corresponder ao escopo global, local e de bloco. Quando você declara uma variável fora de qualquer função, seja ela *var* ou *let*, ela tem o escopo global, pois qualquer função no script consegue utilizar esta variável.

Exemplo:

var minhaVariavelGlobal = 10;

function func1(){

console.log(minhaVariavelGlobal);

}

function func2(){

console.log(minhaVariavelGlobal);

}function func3(){

console.log(minhaVariavelGlobal);

}

console.log(minhaVariavelGlobal);

Exemplo let

let minhaVariavelGlobal = 10;

function func1(){

console.log(minhaVariavelGlobal);

}

function func2(){

console.log(minhaVariavelGlobal);

}

function func3(){

console.log(minhaVariavelGlobal);

}

console.log(minhaVariavelGlobal);

O resultado é exatamente o mesmo, então isso significa que o escopo global serve tanto para var, como para let.

let permite que você declare variáveis limitando seu escopo no bloco, instrução, ou em uma expressão na qual ela é usada. Isso é inverso da keyword var, que define uma variável globalmente ou no escopo inteiro de uma função, independentemente do escopo de bloco.

1. Não há diferença.

**TIPAGEM COM TYPESCRIPT**

1. V-V-V-V-V
2. a) Verdadeiro

b) Verdadeiro

c) Falso

d) Falso

e) Verdadeiro

3- <!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>Calculadora</title>

</head>

<body>

<form>

<label for="lpeso">Peso:</label>

<input type="text" id="tpeso" name="tpeso"><br>

<label for="lvelocidade">Velocidade:</label>

<input type="text" id="tvelocidade" name="tvelocidade"><br>

<label for="ltempo">Tempo:</label>

<input type="text" id="ttempo" name="ttempo">

</form>

<input type="button" value="Calcular" onclick="calcular();" name="">

<input type="text" name="result" id='result'>

</body>

<script type="text/javascript">

calcular = (peso, velocidade, tempo) => peso\*velocidade\*tempo;

function calcular(){

} = (peso, velocidade, tempo) => peso\*velocidade\*tempo;

console.log

(

calcular(parseInt(document.getElementById("tpeso").value)

,(parseInt(document.getElementById("tvelocidade").value)

,(parseInt(document.getElementById("ttempo").value)))));

</script>

</html>