

**Instruções para a entrega:** entregar as respostas na folha de respostas na aula do dia 27/ago.

**1** – A linguagem Javascript possui tipagem dinâmica, isto é, o tipo da variável depende do conteúdo atual da variável. O operador **typeof** retorna o tipo de dado da variável ou expressão. Esse operador pode retornar um dos seguintes tipos primitivos:

- string
- number
- boolean
- undefined

e os tipos complexos:

- function
- object

Para mais detalhes acesse:

[https://www.w3schools.com/js/js\\_datatypes.asp](https://www.w3schools.com/js/js_datatypes.asp)

Analise o trecho de código a seguir e marque a alternativa que retorna somente tipos complexos.

```
let a = 12;
let b = 12.5;
let c = 'oi';
let d = "oi";
let e = true;
let f = {};
let g = undefined;
let h = null;
let i = function(){};
let j = [1,2];
```

- I. console.log( typeof a );
- II. console.log( typeof b );
- III. console.log( typeof c );
- IV. console.log( typeof d );
- V. console.log( typeof e );
- VI. console.log( typeof f );
- VII. console.log( typeof g );
- VIII. console.log( typeof h );
- IX. console.log( typeof i );
- X. console.log( typeof j );

- (a) IV, V, IX, X
- (b) VI, VII, VIII, IX, X
- (c) VI, VIII, IX, X
- (d) VI, VII, VIII, IX

(e) I a X

**2** – As funções na linguagem Javascript são definidas usando a palavra reservada function. As funções podem ter nomes ou serem anônimas, isto é, não ter um nome na sua definição. Para mais detalhes acesse:

[https://www.w3schools.com/js/js\\_functions.asp](https://www.w3schools.com/js/js_functions.asp)

Analise o código a seguir e marque a alternativa incorreta.

```
function somar(a,b){
    return a + b;
}

let dif = function(a,b){
    return a - b;
};

function sum(lista){
    let s = 0;
    for(let i = 0; i < lista.length; i++){
        s += lista[i];
    }
    return s;
}
```

```
let diferenca = dif;
let nros = [9, 4, 3, 8, 5];
```

- (a) console.log( somar(2,3) );
- (b) console.log( dif(5,2) );
- (c) console.log( sum(nros) );
- (d) console.log( diferenca(8,3) );
- (e) console.log( dif );

**3** – As palavras reservadas var, let e const são usadas para declarar variáveis na linguagem Javascript:

- let: a variável declarada usando let tem o escopo do seu bloco. As variáveis criadas com let só podem ser utilizadas após sua declaração;
- var: variáveis declaradas usando var podem ser utilizadas mesmo antes de sua declaração;
- const: ela será constante e só pode ser usada após a sua declaração.

Analise o trecho de código e marque a alternativa que contém apenas nomes de variáveis que foram utilizadas corretamente, isto é, elas não produzem erro. Lembre-se que undefined é um valor.

```
for(var j = 0; j < 5; j++){  
  console.log('j:' + j);
```

```
for(k = 0; k < 3; k++){  
  console.log('k:' + k);
```

```
for(let i = 0; i < 5; i++){  
  console.log('i:' + i);
```

```
if( false ){  
  const m = 9;  
}  
console.log('m:' + m);
```

```
console.log('n:' + n);  
var n = 10;
```

```
console.log('p:' + p);  
let p = 10;
```

```
console.log('q:' + q);  
q = 10;
```

```
console.log('w:' + w);  
const w = 10;
```

- (a) todas as variáveis estão erradas
- (b) j, k, i, n, q, w
- (c) j, k, i, n, w
- (d) j, k, n
- (e) j, k, n, q, w

**4 –** Os arrays no Javascript podem ser redimensionados após a sua criação e podem receber valores de diferentes tipos de dados. Analise o trecho de código a seguir e marque a alternativa que contém os valores a serem impressos.

```
let v = [];  
v[2] = 5;  
console.log( v.length );  
v[4] = true;  
console.log( v.length );
```

- (a) 0 e 0

- (b) 1 e 2
- (c) 2 e 2
- (d) 3 e 5
- (e) Irá ocorrer erro.

**5 –** Na linguagem Javascript os objetos são definidos usando chaves. A variável obj a seguir recebe um objeto sem membros:

```
var obj = {};
```

Marque a alternativa que possui erro na criação do objeto.

- (a) let a = {}  
a.nome = 'Paula';  
a.idade = 22;  
console.log( a );
- (b) let b = {nome:'Pedro'};  
console.log( b );
- (c) let c = {nome:'José', irmao:{nome:'Pedro'}};  
console.log( c );
- (d) let d = {nome:'Carla', {nome:'Pedro'}};  
console.log( d );
- (e) let e = {x:10,y:20};  
console.log( e );

**6 –** Marque a alternativa que possui uma função que retorna o somatório da propriedade valor dos elementos do array lista.

```
let lista = [{codigo:1,valor:2.5},  
             {codigo:2,valor:1.5},  
             {codigo:3,valor:5}];
```

- (a) function somar(v){  
 let s = 0;  
 for(let i = 0; i < v.length; i++){  
 s += v.valor[i];  
 }  
 return s;  
}
- (b) function somar(v){  
 let s = 0;  
 for(let i = 0; i < v.length; i++){  
 s += valor[i];  
 }  
 return s;  
}
- (c) function somar(v){

```

    let s = 0;
    for(let i = 0; i < v.length; i++){
        s += valor;
    }
    return s;
}
(d) function somar(v){
    let s = 0;
    for(let i = 0; i < v.length; i++){
        s += v[i];
    }
    return s;
}
(e) function somar(v){
    let s = 0;
    for(let i = 0; i < v.length; i++){
        s += v[i].valor;
    }
    return s;
}

```

**7** – Os objetos Javascript podem ter propriedades e métodos, onde os métodos são armazenados como propriedades. Marque a alternativa cujo objeto esteja correto.

- (a) 

```
let a = {
  x: 1,
  y: 2,
  print: function(){
    console.log(x + "," + y);
  }
};
```
- (b) 

```
let a = {
  x: 1,
  y: 2,
  print: function(){
    console.log(this.x + "," + this.y);
  }
};
```
- (c) 

```
let a = {
  x: ,
  y: ,
  print: function(){
    console.log(this.x + "," + this.y);
  }
};
```
- (d) 

```
let a = {
  x: undefined,
  y: undefined,
  print: function(){
    console.log(x + "," + y);
  }
}
```

```

};
(e) let a = {
  x: ,
  y: ,
  print(){
    console.log(x + "," + y);
  }
};

```

**8** – Marque a alternativa que possui a chamada correta do método print do objeto do Exercício 7.

- (a) a.print();
- (b) a.print;
- (c) this.print;
- (d) print();
- (e) this.print();

**9** – Hoisting é o processo utilizado pelo Javascript para mover as declarações da variáveis para o início do script ou função, desta forma, é possível usar uma variável antes dela ser declarada no corpo do script ou função. Para mais detalhes acesse:

[https://www.w3schools.com/js/js\\_hoisting.asp](https://www.w3schools.com/js/js_hoisting.asp)

Marque a alternativa que exibe o valor 12 na mensagem de alerta.

- (a) 

```
<script>
  x = 12;
  const x;
  alert(x);
</script>
```
- (b) 

```
<script>
  x = 12;
  let x;
  alert(x);
</script>
```
- (c) 

```
<script>
  x = 12;
  alert(x);
  var x;
</script>
```
- (d) 

```
function teste(){
  w = 12;
  alert(w);
  const w;
}
teste();
```
- (e) 

```
function teste(){
```

```
w = 12;  
alert(w);  
let w;  
}  
teste();
```

**10** – Arrow Function nos permite escrever funções usando uma sintaxe enxuta/curta. Para mais detalhes acesse:

[https://www.w3schools.com/js/js\\_arrow\\_function.asp](https://www.w3schools.com/js/js_arrow_function.asp)

Marque a alternativa que possui uma arrow function que produz o resultado esperado.

- (a) 

```
let a = (x, y) => {  
  return x + y;  
}  
console.log( a(2,8) );
```
- (b) 

```
let b = () => alert('oi');  
b();
```
- (c) 

```
let c = (x, y) => return x - y;  
console.log( c(8,3) );
```
- (d) 

```
let d = (x, y) => x * y;  
console.log( d(8,3) );
```
- (e) 

```
let e = (x, y) => {x + y};  
console.log( e(8,3) );
```