

## VARIÁVEIS, ESCOPO, TIPOS DE DADOS e OPERADORES

1) Supondo o seguinte código, informe o que será impresso em cada alert?

```
<script>
var a;
var b=15;
var c=0;
var d=15/b;
var e=15/c;
var f= c/0;
var g=null;
alert(a);    →
alert(b);    →
alert(c);    →
alert(d);    →
alert(e);    →
</script>
```

2) No código abaixo, **escolha a opção** que representa o resultado **dos consoles**:

```
<script>
var numero1 =10;
var numero2 ="10";
var numero3 = new String("10");
var numero4 = 20/2;
console.log(numero1===numero2);
console.log(numero2===numero3);
console.log(numero2===numero3);
console.log(numero1==numero4);
</script>
```

- a) true, true, true, true
- b) true, false, false, true
- c) false, false, false, false
- d) false, true, false, true
- e) false, true, false, false
- f) nenhuma das anteriores

3) Qual será o resultado **impresso na tela**?

```
<script>
var i = 1 + "2" + 3;
for (j = 0; j < 10; j++) {
  alert(`${i}${j}`);
  if (j === 5)
    break; }
</script>
```

4) Informar qual o **resultado em cada console**:

```
let x = new Number(50);
let obj1 = {};
let obj2 = new Object();
let nome = "José";
let nome1 = new String("José");
let frutas = new Array("laranja", "banana", "maçã");
let y = null;
let z;

//Operador Condicional (?)
let resultado1 = (x > 40) ? "Maior que 40" : "Menor ou igual a 40";
```

## LISTA DE EXERCÍCIOS PARA ESTUDO PWEB – 05/2025

```
console.log(resultado1); →

// Operador Vírgula (,)
let function a()
  return "a";

let function b()
  return "b";

let c=(a(),b());
console.log(c); →

//delete
let objTeste = { propriedade1: "valor1", propriedade2: "valor2" };
delete objTeste.propriedade1;
console.log(objTeste); →

// in
let resultadoInObj = "propriedade2" in objTeste;
let resultadoInArray = 2 in frutas;
console.log(resultadoInObj); →
console.log(resultadoInArray); →

//instanceof
console.log(x instanceof(Number)); →
console.log(obj1 instanceof(Object)); →
console.log(obj2 instanceof(Object)); →
console.log(nome instanceof(String)); →
console.log(nome1 instanceof(String)); →
console.log(frutas instanceof(Array)); →

//this
let objetoThis = {
  valor: 10,
  obterValor: function() {
    return this.valor;
  }
};
console.log(objetoThis.obterValor());→

// typeof
console.log(x typeof(Number)); →
console.log(obj1 typeof(Object)); →
console.log(obj2 typeof(Object)); →
console.log(nome typeof(String)); →
console.log(nome1 typeof(String)); →
console.log(frutas typeof(Array)); →
console.log(x typeof(Number)); →

// void
let resultadoVoid = void(0);
console.log(resultadoVoid); →

// Coalescência Nula (??)
let resultadoNullish1 = y ?? "Valor padrão para y";
let resultadoNullish2 = z ?? "Valor padrão para z";
let resultadoNullish3 = nome ?? "Valor padrão para nome";
console.log(resultadoNullish1); →
console.log(resultadoNullish2); →
```

## LISTA DE EXERCÍCIOS PARA ESTUDO PWEB – 05/2025

```
console.log(resultadoNullish3); →
```

5) Supondo o código JavaScript:

```
let x=30;
let y=null;
let z;

alert(String(x));
alert(String(y));
alert(String(z));
alert(x.toString());
alert(y.toString());
alert(z.toString());
```

Será **impresso na tela**:

- a) 30, null, undefined, 30, erro, erro
- b) 30, 0, 0, 30, 0, 0
- c) 30 undefined, null, 30, undefined, null
- d) 30, null, undefined, 30, null, undefined
- e) erro, erro, erro, 30, null, undefined
- f) 30, null, undefined, 30, null, erro

6) Qual alternativa apresenta o **resultado no console das conversões** abaixo?

```
<script>
  var num1 = Number("123Teste");
  var num2 = Number(true);
  var num3 = Number("2.52");
  var num4 = parseInt("123Teste");
  var num5 = parseFloat("123.5678").toFixed(2);
  var num6 = String(null);
  console.log(`{num1},{num2},{num3},{num4},{num5},{num6}`);
</script>
```

- a) NaN,1,2.52,NaN,123.57,"null"
- b) 123,1,2.52,123,123.57,"null"
- c) NaN,1,2.52,123,123.57,"null"
- d) 123,1,2.52,NaN,123.57,"null"
- e) NaN,1,2.52,123,13,"null"

7) Dado o bloco de código JavaScript:

```
<script>
  var i = 5;
  var j = 10;
  var count = 0;
  while (count < 5) {
    document.write(i > j ? "mais" : "menos");
    document.write(" ");
    i++;
    count++;
  }
</script>
```

Quando o bloco for inserido no corpo de uma página HTML e executado no navegador Internet Explorer, sem considerar aspas, exibirá na tela:

- a) "mais mais mais mais mais"
- b) "menos mais menos mais"
- c) "menos menos menos menos menos"
- d) "mais mais mais mais"

## LISTA DE EXERCÍCIOS PARA ESTUDO PWEB – 05/2025

e) “mais mais mais mais mais mais”

8) Verifique serão os **resultado dos alerts**, no código apresentado:

→ **Observar escopo do var, let e const**

```
<script>
  var x = 1;
  var y = 2;
  let j = 3;
  let i = 4;
  const k1 = 5;

  function principal() {
    let z = 3;
    w = 4;
    const k2 = 7;

    x = x * 10;
    y = y * 10;
    return `x=${x} y=${y} i=${i} j=${j} z=${z} w=${w} k1=${k1} k2=${k2}`;
  }
  alert(principal()); //primeiro alert
  alert(`x=${x} y=${y} i=${i} j=${j} z=${z} w=${w} k1=${k1} k2=${k2}`);
</script>
```

## FUNÇÕES

9) No JavaScript existem vários tipos de funções, entre elas:

- *Functions declaration* (Função de declaração)
- *Functions expression* (Função de expressão)
- *Arrow Functions* (Função de flecha)

Crie uma função que soma 2 números **para cada um dos tipos** dessas funções.

- Criar uma **função** em JavaScript que **receba uma frase** e retorne **quantas letras 'A'** (maiúsculas ou minúsculas) foram encontradas.
- Criar uma **função** em JavaScript que **receba 5 números** e **retorne-os** em ordem **decrescente**.
- Criar uma **função** em JavaScript que **receba um array de números** e **retorne-os** em ordem decrescente.

10) Escreva uma função (declarada, expressão de função ou arrow function) em JavaScript, para gerar a tabuada de qualquer número inteiro entre 1 e 10. A função **recebe o número** que se deseja ver a tabuada, e a **saída (string)** deve ser conforme o exemplo abaixo:

## LISTA DE EXERCÍCIOS PARA ESTUDO PWEB – 05/2025

<b>** Tabuada do 1 **</b>	
1 x 0 = 0	
1 x 1 = 1	
1 x 2 = 2	
1 x 3 = 3	
1 x 4 = 4	
1 x 5 = 5	
1 x 6 = 6	
1 x 7 = 7	
1 x 8 = 8	
1 x 9 = 9	
1 x 10 = 10	

11) Considerando como parâmetros da **função3**, o **penúltimo** e o **último** dígito do seu ra, o alert resultará: \_\_\_\_\_

```
<script>
  funcao1 = function(n) {
    return (Math.pow(n, 3));
  }
  funcao2 = function(n1, n2) {
    return funcao1(Math.min(n1, n2));
  }
  funcao3 = function(x,y) {
    return funcao2(x,y);
  }

  var penultimo = prompt("Digite o penúltimo dígito do seu ra");
  var ultimo = prompt("Digite o último dígito do seu ra");
  alert(funcao3(penultimo, ultimo));
</script>
```

12) Informe o que será impresso na execução da função Imprimir.

```
<script>
  function Imprimir(n) {
    for (var i = 0; i <= 10; i++) {
      var linha = n + " * " + i + " = " + n * i;
      console.log(linha);
    }
  }

  Imprimir(10);
</script>
```

13) Se o arquivo index.html for executado em um navegador, qual será o **texto mostrado no input do resultado**, quando o botão for clicado?

Arquivo: index.html

```
<!DOCTYPE html>
<html lang="pt-br">

<head>
  <meta charset="UTF-8">
  <title>Exercício formulário com evento </title>
</head>

<body>
  <form id="formulario1">
    <br>
```

## LISTA DE EXERCÍCIOS PARA ESTUDO PWEB – 05/2025

```
<label>resultado</label>
<input type="text" name="resultadoout" id="resultado" disabled>
<br>
<input type="button" value="Executa função" onclick="teste();">
</form>
<script>
  function teste() {
    const x = [2, 4, 6, 8, 10];
    let s = " ";
    x.forEach(f);
    console.log(s);

    function f(value, index, array) {
      s += index + " " + value; //observar espaço em branco
    }

    document.forms.formulario1.resultadoout.value = s;
  }
</script>
</body>
</html>
```

14) No código abaixo, se o **número de entrada for 4**, quando clicado o botão, o **resultado** mostrará: \_\_\_\_\_

```
<body>
  <h1>Operação</h1>
  <form id="formulario1">
    <label>Escolha um número</label>
    <input type="number" id="numero">
    <br>
    <label>resultado</label>
    <input type="text" id="resultado" disabled>
    <br>
    <input type="button" value="Executa função" onclick="funcao3();">
  </form>
  <script>
    funcao1 = function(n) {
      return (n * 3);
    }
    funcao2 = function(n) {
      return funcao1(n * 2);
    }
    funcao3 = function() {
      document.forms.formulario1.elements[1].value =
        funcao2(document.forms.formulario1.elements[0].value);
    }
  </script>
</body>
```

## ARRAYS

15) Considerando o código abaixo, o que será mostrado no **console**?

→ **Observar como o sort organiza**

```
<script>
  var novoArray = [3, 1, 24, 32, 10, 15, 2];
  novoArray.sort();
```

## LISTA DE EXERCÍCIOS PARA ESTUDO PWEB – 05/2025

```
    novoArray.reverse();  
    console.log(novoArray);  
</script>
```

- a) 1,10,15,2,24,3,32
- b) 1,2,3,10,15,24,32
- c) 32,24,15,10,3,2,1
- d) 32,3,24,2,15,10,1
- e) Nenhuma das anteriores

16) Considerando o código abaixo, o que será mostrado no **alert**?

```
var numeros = [40, 100, 1, 5, 25, 10];  
numeros.sort(function(a, b){return a-b});  
alert(numeros);
```

17) Considerando o código abaixo, o que será mostrado no **alert**?

```
var nomes = ["ana", "Ana", "Beatriz", "Pedro", "Matheus",  
    "Mateus", "mateus"];  
nomes.sort();  
alert(nomes);
```

18) Supondo o seguinte código, informe o que será **impresso no alert**?

```
function teste() {  
    var auxiliar = 0;  
    for (var i = 0; i < arguments.length; i++) {  
        auxiliar += arguments[i]*2;  
    }  
    return auxiliar;  
}  
var arr1 = [1, 2,3, 4];  
alert(teste.apply(null, arr1));
```

19) Considerando o código abaixo, qual será o resultado da variável total?

```
<script>  
    var alunos = ["Antonio", "Tobias", "Helio", "Daniel", "Junior", "Leonardo", "Jose", "Livia",  
    "Pedro"];  
    var total = 0;  
    alunos.sort();  
    alunos.pop();  
    alunos.pop();  
  
    for (i = 0; i <= alunos.length - 1; i++) {  
        total = total + alunos[i].length;  
    }  
    document.write("O total é: " + total);  
</script>
```

20) No código abaixo, considere que os **números que serão inseridos no prompt são: 12, 20, 8, 19,14,1,4,12**. O resultado da **variável palavraFinal** será:

```
<script>  
    const alfabeto = 'abcdefghijklmnopqrstuvwxyz';  
    var palavraFinal = "";  
    var i=0;  
    while (i<8)  
    {  
        var numero=prompt("Digite um número:");  
        palavraFinal+=alfabeto[numero];  
        i++;  
    }  
}
```

## LISTA DE EXERCÍCIOS PARA ESTUDO PWEB – 05/2025

```
    alert(palavraFinal);  
</script>
```

### OBJETOS, FUNÇÃO CONSTRUTORA, CLASSES, HERANÇA

21) Criar dois **objetos** (objLivro1 e objLivro2) com **2 propriedades** (Titulo e Autor). Usar **duas formas** diferentes (entre aquelas vistas em aula).

22) Observe as formas em que foram criados os objetos abaixo, e **assinale as que estão corretas**:

a)

```
var aluno1 = new Object(); // forma 1  
aluno1.ra = "00001234";  
aluno1.nome = "Pedro da Silva";
```

b)

```
var aluno2 = {}; // forma 2  
aluno2.ra = "1234";  
aluno2.nome = "José da Silva";
```

c)

```
var aluno3 = { // forma 3  
  ra: "1234567",  
  nome: "João da Silva",  
};
```

d)

```
function Aluno(_ra, _nome) {  
  this.ra = _ra;  
  this.nome = _nome;  
}  
var aluno4 = new Aluno(123, 'Carlos'); // forma 4
```

e)

```
function Aluno2()  
{  
  var _ra;  
  var _nome;  
  
  this.getRa = function() {  
    return this._ra;  
  };  
  
  this.setRa = function(value) {  
    this._ra = value;  
  };  
  
  this.getNome = function() {  
    return this._nome;  
  };  
  
  this.setNome = function(value) {  
    this._nome = value;  
  };  
}  
var aluno5 = new Aluno2(); // forma 5  
aluno5.setRa(124);  
aluno5.setNome("Helio");
```

f)



## LISTA DE EXERCÍCIOS PARA ESTUDO PWEB – 05/2025

```
class Aluno3 {
    constructor(ra, nome) {
        this._ra = ra;
        this._nome = nome;
    }
}
var aluno6 = new Aluno3(123, "Ana");
alert(aluno6._ra + " " + aluno6._nome);
```

g)

```
class Aluno4 {
    constructor() {
        this._ra;
        this._nome;
    }

    setNome(value) {
        this._nome = value;
    }

    getNome() {
        return this._nome;
    }

    setRa(value) {
        this._ra = value;
    }

    getRa() {
        return this._ra;
    }
}
var aluno7 = new Aluno4();
aluno7.setNome("Anna");
aluno7.setRa('123');
alert(aluno7.getNome() + '-' + aluno7.getRa());
```

h)

```
class Aluno {
    #ra; // Campos privados
    #nome;

    constructor(ra, nome) {
        this.#ra = ra;
        this.#nome = nome;
    }

    get ra() {
        return this.#ra;
    }

    set ra(value) {
        this.#ra = value;
    }

    get nome() {
        return this.#nome;
    }

    set nome(value) {
```

## LISTA DE EXERCÍCIOS PARA ESTUDO PWEB – 05/2025

```
        this.#nome = value;
    }
}

const aluno8 = new Aluno(124, "Andrea");
console.log(aluno8.ra);
```

i)

```
class Aluno {
    constructor(ra, nome) {
        this._ra = ra; // Convenção de nome para indicar "privado"
        this._nome = nome;
    }

    getRa() {
        return this._ra;
    }

    setRa(value) {
        this._ra = value;
    }

    getNome() {
        return this._nome;
    }

    setNome(value) {
        this._nome = value;
    }
}

const aluno9 = new Aluno(124, "Beatriz");
aluno9.setRa(124);
aluno9.setNome("Beatriz");
```

23) Criar a **classe** Médico com os atributos nome e CRM. Criar as **subclasses** (“filhas” da classe Médico) Pediatra com atributo idade máxima paciente e Geriatria com atributo idade do paciente. Criar um objeto da classe Pediatra e outro objeto da classe Geriatria. Fazer **1º usando função construtora** e depois repetir o exercício **usando classes**.

24) O \_\_\_\_\_ é um **formato simples e popular para armazenar e transferir dados aninhados ou hierárquicos (ver exemplos)**. É bastante comum na web, usado para enviar dados de um servidor para um navegador da web ou para transferir dados entre sistemas. Veja exemplo:

```
[
  {
    "id": 1,
    "nome": "Camiseta Algodão",
    "preco": 35.50,
    "cores_disponiveis": ["Azul", "Verde", "Branco"],
    "tamanhos": ["P", "M", "G"]
  },
  {
    "id": 2,
    "nome": "Calça Jeans",
    "preco": 89.99,
    "cores_disponiveis": ["Azul Claro", "Azul Escuro", "Preto"],
    "tamanhos": ["38", "40", "42", "44"]
  }
]
```

## LISTA DE EXERCÍCIOS PARA ESTUDO PWEB – 05/2025

25) Informe se o identificador está sendo criado de forma **literal** (L) ou como **objeto** (O).

```
( ) var x = new Number(50);  
( ) var obj1 = {};  
( ) var obj2 = new Object();  
( ) var nome = "José";  
( ) var nome1 = new String("José");  
( ) var notas = ["do", "ré", "mi", "fá", "sol", "lá", "si"];  
( ) var frutas = new Array("laranja", "banana", "maçã");
```

---

### Gabarito

1)  
undefined  
15  
0  
1  
infinity  
NaN  
null

2) d

3)  
1230  
1231  
1232  
1233  
1234  
1235

4)  
condicional - "maior que 40"  
virgula - "b"  
delete - { propriedade2: 'valor2' }  
in - true, true  
instanceof - true, true, true, false, true, true  
this - 10  
typeof - object, object, object, string, object, object  
void - undefined  
coalescência nula - "valor padrão para y", "valor padrão para z", "José"

5) a

6) c

7) c

8)  
Primeiro Alert  
x=10 y=20 i=4 j=3 z=3 w=4 k1=5 k2=7

Segundo Alert - não vai mostrar porque está acessando variáveis que são de escopo de função a começar pelo z

9) criar as funções

10) criar a função

11) vai depender do seu RA  
Exemplo:  
Penúltimo 2  
Último 3

## LISTA DE EXERCÍCIOS PARA ESTUDO PWEB – 05/2025

resultado = 8

12) É um exemplo de função de tabuada no caso para o 10

13) 0 21 42 63 84 10

14) 24

15) d

16) 1,5,10,25,40,100

17) Ana,Beatriz,Mateus,Matheus,Pedro,ana,mateus

18) 20

19) 41

20) muitobem

21) criar como desejar

22) todas estão corretas

23) criar como desejar

24) JSON

25) objeto, literal, objeto, literal, objeto, literal, objeto

-->observe que no caso do obj1 embora ele tenha sido criado de maneira literal o typeof dele é object