Lista de Exercícios - Teórico JavaScript

1. Um certo documento HTML contém o seguinte elemento h1:

```
<h1 id="meuId">Bem-Vindo</h1>
```

Deseja-se alterar o conteúdo desse h1 via JavaScript. A instrução que atinge esse objetivo é:

A - document.getElement(“h1”). innerHTML = “Alterado”;
B - document.getElementById(“h1”). innerHTML = “Alterado”;
C - document.getElementByName(“h1”). innerHTML = “Alterado”;
D - document.getElementById(“meuId”). innerHTML = “Alterado”;
E - document.getElementByName(“meuId”). innerHTML = “Alterado”;

2. A função fetch() do JavaScript serve para

A - percorrer cursores de consultas de banco de dados.

B - acessar e manipular partes do pipeline HTTP.

C - realizar a conversão de respostas para array de objetos.

D - filtrar dados da memória de acordo com os parâmetros.

E - realizar requisições XMLHttpRequest e extensões do HTTP.

3. Considere o código JavaScript a seguir quanto à sua sintaxe, execução e retorno:

```
< p id="funcao"></p>
<script>
 function computa()
 {
  document.getElementById("funcao").innerHTML =
  Math.abs(-12.12);
 }
</script>
```

A saída retornará:

A - 12

B - -12

C - 12,12

D - -12,12

E - Erro

4. No contexto da linguagem JavaScript, analise o trecho a seguir.

```
hello = (val) => "Hello " + val;
```

Essa definição é equivalente ao trecho:

A - function hello(val) { return 'Hello ' + val; }

B - ((val == ' ') ? 'Hello ' + val : val)

C - function hello(val) { if (val != '') { return 'Hello ' + val } else { return '' } }

D - function hello(val) { return val; }

E - ((val != ' ') ? val : '')

5. Analise o código JavaScript exibido a seguir.

```
function xpto(x) {
  return x % 2 == 0;
 }
const numeros = [22, 0, 18, 1];
alert (numeros.every(xpto));
```

O valor exibido na execução desse código é:

A - 1

B - 3

C - false

D - true

E - undefined

6.Considere o código JavaScript a seguir.

```
var x1
var x2
var x3
var y = 11
var z = 12
x1= z++ == ++y
x2= 10 + "casa"
x3= y+1 === z && 10 != "20"
```

Após a execução dessas operações, os valores de *x1, x2* e *x3* são, respectivamente:

A - false 10casa true

B - false 10casa false

C - false undefined false

D - true 10casa true

E - true undefined false

7. Qual caractere é o operador de concatenação de strings em Java?

A - .

B - |

C - +

D - &

E - &&

8. Qual método Javscript permite o agrupamento de todos os elementos de um Array em uma string com a possibilidade de definir o caractere de concatenação?

A - join()

B - ToString()

C - concatenate()

D - ToChar()

E - plus()

9.Assinale a alternativa **correta** com relação à Template Literals.

A - Bibliotecas Javascript de back-end voltadas primariamente à implementação de APIs REST.

B - Recurso Javascript que possibilita a interpolação e a substituição de variáveis em strings.

C - Modelos extensíveis CSS que proveem maior produtividade em projetos com uso intenso das tags e

D - Bibliotecas Bootstrap voltadas à aceleração da construção de sites responsivos e voltados à mobile.

E - São bibliotecas Javascript voltadas à geração de código HTML formatado com as respectivas definições CSS autogeradas.

10. Em Javascript existem diferentes tipos de operadores, marque a alternativa que contém somente operadores relacionais?

A - Typeof, In

B - <=, >=, <<, >>

C - In, Instanceof

D - ==, !=

E - &, ^

11. Javascript é uma linguagem que tem bastante problemas de compatibilidade entre navegadores, e é de comum conhecimento que existem navegadores que se encaixam melhor à linguagem. Qual dos navegadores abaixo tem o maior problema de compatibilidade com Javascript?

A - Google Chrome.

B - Mozilla Firefox.

C - Internet Explorer.

D - Safari.

E - Microsoft Edge.

12. No contexto do JavaScript, considere as seguintes afirmativas sobre a declaração de variáveis com *let* ou *var*.

I. Variáveis declaradas com *let* não podem ser redeclaradas no mesmo { } bloco.
II. Variáveis declaradas com *let* podem ser utilizadas em qualquer trecho do código (escopo global).
III. Variáveis declaradas com *var* no interior de um { } bloco podem ser utilizadas fora do bloco de origem.

Está correto o que se afirma apenas em

A - I.

B - II.

C - I e II.

D - I e III.

E - II e III.

13. Considere o código JavaScript a seguir.

```
let txt = "";

function funcao(value, index, array) {
    if (index % 2 == 0) {txt += value};
}

function xpto (x) {
            x.forEach(funcao);
            return txt;
}

alert (xpto([0, 1, 1, 2, 3, 5]));
```

A execução desse código provoca a exibição de:

A - 011235

B - 013

C - 125

D - 532110

E - null

14. Considere o comando JavaScript a seguir.

```
document.getElementById('demo').innerHTML = Date()
```

Numa página web na qual esse código seja aplicado, o elemento que é compatível com a estrutura do comando para receber a data corrente é:

A - <p "demo">H

B - H

C - H

D - H

E - H

15. Analise o script JS a seguir.

```
<script>
function xpto() {
  let n = 0;
  return {next:
           function()
           {
             if (n < 10) {
               n += 2;
               return {value:n/2, done:false}}
             else {
               return {value: -1, done: true}}
           }
  }
}

saida = ""
const n = xpto();
x = n.next();
while (x.done == false) {
  saida += x.value + "/";
  x = n.next();
}
alert(saida);
</script>
```

O resultado da execução desse código é:

A - -1/-2/-3/-4/-5/

B - 0/1/2/3/4/5/

C - 1/2/3/4/5/

D - 5/4/3/2/1/

E - 5/4/3/2/1/0

16. Na linguagem *JavaScript*, ao invocar o método *getElementsByClassName*, do objeto *document*, será retornado:

A - Um objeto.

B - Uma função.

C - Um valor numérico.

D - Um *array.*

E - Um *string.*

17. JavaScript é uma linguagem que sofre muito com compatibilidade entre navegadores. A jQuery sofre com o mesmo problema. Animações, manipulação de DOM e outra tarefas corriqueiras são mais complexas e menos produtivas ao usar o jQuery.

A - Certo

B - Errado

18. A linguagem *JavaScript* provê uma série de métodos que facilitam a manipulação de *arrays.* Sobre o método de manipulação de *array of*, é correto afirmar que

A - cria um novo *array* a partir de um *array* existente.

B - preenche o *array* com um valor estático.

C - devolve @@*iterator*, contendo os valores do *array.*

D - cria um novo *array* a partir dos argumentos passados para o método.

19. Considere o seguinte código *JavaScript*, sabendo que o usuário irá digitar corretamente os valores solicitados via *prompt:*

```
var v1 = 3; var v2, v3, v4; v2 = prompt("Digite o número 3:"); v2 =
prompt("Digite a palavra true:"); v4 = false;
console.log(v1===v2); console.log(v2==v3);
console.log(v1%=v2); console.log(v1);
```

Ao final da execução, quais valores serão impressos?

A - false, false, 0, 0

B - false, false, NaN, NaN

C - true, false, NaN, 3

D - true, false, 0, 3

20. Considere o seguinte código *JavaScript:*

```
let o = {one:1,two:2,three:3};
for(let p in o) console.log(p);
```

Ao final da execução, quais valores serão impressos?

A - 1, 2, 3

B - one:1, two:2, three:3

C - p, p, p

D - 'one', 'two', 'three'