



Lista de Atividades 2 – Strings e Operadores

Disciplina: TI23J – Desenvolvimento Web 2

1. Considerando o código a seguir, qual será o resultado apresentado para z?

```
let x = 5;  
let y = '8';  
let z = x + y;
```

- a) undefined
 - b) 58 Resposta correta**
 - c) 13
 - d) Resultará em erro
 - e) Nenhuma das alternativas
2. Qual é a melhor descrição do problema do código abaixo?

```
var str = "Hello World";  
str[0] = "F";
```

- a) Depois que uma string é criada, ela não pode ser alterada, pois as strings são imutáveis em JavaScript. Resposta correta**
 - b) A alteração de caracteres de uma string só pode ser feita usando o método `setCharAt()`.
 - c) Não há problema no código.
3. Como podemos extrair "Pizza" da string "Pizza Lover"?
- a) `str.slice(0, 4)`
 - b) `str.slice(0, 5)` Resposta Correta**
 - c) `str.slice(4, 0)`
 - d) `str.slice(5, 0)`

4. As strings são imutáveis em JavaScript. Forneça um trecho de código que exemplifique essa característica?

Resposta: As strings são imutáveis, pois não podemos modificar um caractere diretamente em uma string já existente. Exemplo:

```
// EXERCÍCIO 4  
let str = "Banana";  
str[2] = "t"; // Tentando modificar o primeiro caractere  
console.log(str); // "Banana" (não muda!)
```

5. O que é precedência e associatividade de operadores? Exemplifique.

Resposta: Precedência de operadores, é a determinação de qual operador vai ser executado primeiro no código. A associatividade é a direção que os operadores vão ser lidos e executados, sendo alguns operadores “da esquerda para a direita” e outros “da direita para a esquerda”.

Exemplos:

```
//EXERCÍCIO 5
let conta = 15 + 3 * 5
console.log(conta) // Resultado: "30"

let resultado = 100 / 5 / 2;
console.log(resultado); // 10 (porque / a associatividade é da E p/ D)
```

6. Com suas palavras, explique o funcionamento do operador ternário e apresente um exemplo.

Resposta: Um operador ternário é em JS o único que opera sobre três operandos, ele é usado para expressões condicionais. Os seus três operandos são: ‘condição’, ‘true’ e ‘false’.

Exemplo:

```
//Exercício 6
let a = 10, b = 20;
let maior = (a > b) ? a : b;
console.log(maior); // 20
```

7. Crie um código que determine se um valor é par ou ímpar.

```
//Exercício 7
function parOuImpar(numero) {
  console.log("Esse número é ímpar ou par?")
  if (numero % 2 === 0) {
    console.log("par");
  } else {
    console.log("ímpar");
  }
}
parOuImpar(9); // 'ímpar'
parOuImpar(8); // 'par'
```

8. Crie um código que determina se um valor é número primo.

```
//EXERCÍCIO 8
function ePrimo(x) {
  if (x < 2) return console.log("Não é primo");

  for (let i = 2; i * i <= x; i++) {
    if (x % i == 0) {
      return console.log("Não é primo");
    }
  }

  console.log("É PRIMO!");
}
ePrimo(11); // "É PRIMO!"
ePrimo(6); // "Não é primo"
ePrimo(2); // "É PRIMO!"
```

9. Crie um código que converta uma medida de temperatura dada em Celsius (C) para Fahrenheit (F).

```
//EXERCÍCIO 9
function CelParaFah(c){
  let fahrenheit = c * 1.8 + 32
  return console.log(fahrenheit.toFixed(1))
}
CelParaFah(30)
CelParaFah(21)
```

10. Crie um código que conte a quantidade de palavras em um textos.

```
//EXERCÍCIO 10
function contarPalavras(frase) {
  let palavras = frase.trim().split(/\s+/)
  console.log(palavras.length)
}

contarPalavras("olá mundo")
contarPalavras("Este é um teste")
contarPalavras("Exemplo com      espaços")
```

11. Crie um programa que, dado um valor inteiro em segundos, converta e retorne o horário em horas, minutos e segundos.

Exemplos:

Entrada: 3600	Saída: 1:0:0
Entrada: 4508	Saída: 1:15:8
Entrada: 78568	Saída: 21:49:28

```
//EXERCÍCIO 11
function transformaSeg(segundos) {
  let horas = Math.round(segundos / 3600);
  let minutos = Math.round((segundos % 3600) / 60);
  let segs = segundos % 60;

  console.log(horas + ":" + minutos + ":" + segs);
}
transformaSeg(4508)
transformaSeg(3600)
transformaSeg(78568)
```

12. Utilizando a objeto Math, construa em JavaScript as seguintes equações.

a)
$$z = \frac{\sqrt{x^2 + y^3}}{|x + y|}$$

b)
$$z = \frac{1 + \sin(x)}{1 + \cos(x)}$$

c)
$$z = 1 + \frac{1}{x} + \frac{1}{x^2} + \frac{1}{x^3} + \frac{1}{x^4}$$

d)
$$z = \frac{x}{y} - \frac{x + \left(\frac{x}{y}\right)^2}{x - \left(\frac{x}{y}\right)^2}$$

e)
$$z = \sqrt{\pi + \sqrt{e^3 + \sqrt{4 + \sqrt{x}}}}$$

13.

```
//EXERCÍCIO 12
// a)
function equacaoA(x,y){
    let resultado = (Math.sqrt(Math.pow(x, 2) + Math.pow(y, 3)) / Math.abs(x + y)).toFixed(3)
    return console.log(resultado)
}
equacaoA(2, 3)

// b)
function equacaoB(x) {
    let resultado = ((1 + Math.sin(x)) / (1 + Math.cos(x))).toFixed(3)
    return console.log(resultado)
}
equacaoB(Math.PI / 4)
equacaoB(Math.PI / 6)

// c)
function equacaoC(x) {
    let resultado = (1 + (1 / x) + (1 / Math.pow(x, 2)) + (1 / Math.pow(x, 3)) + (1 / Math.pow(x, 4))).toFixed(3)
    return console.log(resultado)
}
equacaoC(2)
equacaoC(1)
```

```
// d)
function equacaoD(x, y) {
    let frac = x / y
    let numerador = x - Math.pow(frac, 2)
    let denominador = x - Math.pow(frac, 2)
    let resultado = (frac - (numerador / denominador)).toFixed(3)
    return console.log(resultado)
}
equacaoD(5, 6)
equacaoD(1, 2)

// e)
function equacaoE(x) {
    let resultado = Math.sqrt(Math.PI + Math.sqrt(Math.pow(Math.E, 3) + Math.sqrt(4 + Math.sqrt(x)))).toFixed(3)
    return console.log(resultado)
}

equacaoE(9)
equacaoE(1)
```