

# Unidade 1

## Exercício 1: Anagramas

Enunciado:

Escreva uma função que determine se duas palavras são anagramas uma da outra. Duas palavras são consideradas anagramas se contiverem as mesmas letras, independentemente da ordem.

Solução:

```
function saoAnagramas(palavra1, palavra2) {  
  // Remover espaços em branco e converter para minúsculas  
  palavra1 = palavra1.replace(/\s/g, "").toLowerCase();  
  palavra2 = palavra2.replace(/\s/g, "").toLowerCase();  
  
  // Verificar se as palavras têm o mesmo comprimento  
  if (palavra1.length !== palavra2.length) {  
    return false;  
  }  
  
  // Contar a ocorrência de cada letra em ambas as palavras  
  const contagem1 = {};  
  const contagem2 = {};  
  
  for (let letra of palavra1) {  
    contagem1[letra] = (contagem1[letra] || 0) + 1;  
  }  
  
  for (let letra of palavra2) {  
    contagem2[letra] = (contagem2[letra] || 0) + 1;  
  }  
  
  // Verificar se as contagens são iguais  
  for (let chave in contagem1) {  
    if (contagem1[chave] !== contagem2[chave]) {  
      return false;  
    }  
  }  
}
```

```

    return true;
}

// Exemplo de uso
const palavra1 = "roma";
const palavra2 = "amor";
if (saoAnagramas(palavra1, palavra2)) {
    console.log(` ${palavra1} e ${palavra2} são anagramas.`);
} else {
    console.log(` ${palavra1} e ${palavra2} não são anagramas.`);
}

```

## Unidade 2

### Exercício 2: Sequência de Fibonacci

Enunciado:

Implemente uma função que gere os primeiros  $n$  termos da Sequência de Fibonacci, onde  $n$  é um número fornecido pelo usuário. A Sequência de Fibonacci é uma série de números em que cada número é a soma dos dois anteriores (ex: 0, 1, 1, 2, 3, 5, 8, 13, ...).

Solução:

```

function fibonacci(n) {
    const sequencia = [0, 1]; // Começa com os dois primeiros termos: 0 e 1

    // Gera os próximos termos da sequência
    for (let i = 2; i < n; i++) {
        const proximoTermo = sequencia[i - 1] + sequencia[i - 2];
        sequencia.push(proximoTermo);
    }

    return sequencia;
}

// Função para solicitar entrada do usuário e gerar a sequência de Fibonacci
function gerarSequenciaFibonacci() {
    const n = parseInt(prompt("Digite o número de termos da Sequência de Fibonacci que deseja gerar:"));

    // Verificar se a entrada é válida
    if (isNaN(n) || n <= 0) {

```

```

    alert("Por favor, digite um número inteiro positivo válido.");
    return;
}

// Gerar a sequência de Fibonacci
const sequencia = fibonacci(n);
console.log(`Os primeiros ${n} termos da Sequência de Fibonacci são:
${sequencia.join(', ')}`);
}

// Chamar a função para gerar a sequência de Fibonacci
gerarSequenciaFibonacci();

```

## Unidade 3

Enunciado:

Crie uma função que recebe uma string como entrada e retorna o número de vogais (A, E, I, O, U) presentes nela. A função deve ser case-insensitive, ou seja, deve contar vogais tanto em letras maiúsculas quanto minúsculas.

Solução:

```

function contarVogais(string) {
    // Converter a string para minúsculas para tornar a comparação case-insensitive
    string = string.toLowerCase();

    // Inicializar contador de vogais
    let contadorVogais = 0;

    // Percorrer a string e contar vogais
    for (let i = 0; i < string.length; i++) {
        const caractere = string.charAt(i);

        if (caractere === 'a' || caractere === 'e' || caractere === 'i' || caractere === 'o' ||
            caractere === 'u') {
            contadorVogais++;
        }
    }
}

```

```
    return contadorVogais;
}

// Exemplo de uso
const palavra = "Olá, Mundo!";
const numVogais = contarVogais(palavra);
console.log(`A string "${palavra}" contém ${numVogais} vogais.`);
```

## Unidade 4

### Exercício 4: Jokempo

Enunciado:

Crie um jogo de jokempo onde o usuário tem que escolher entre A (pedra), B(papel) e C (tesoura) e após escolher, será sorteado qual escolha o inimigo fez, e será mostrado num alert o resultado do jogo.

Solução:

```
function jogoJokempo(escolhaUsuario) {
    // Escolhas possíveis do usuário e do inimigo
    const opcoes = ['A', 'B', 'C']; // A: pedra, B: papel, C: tesoura

    // Escolha aleatória do inimigo
    const escolhaInimigo = opcoes[Math.floor(Math.random() * opcoes.length)];

    // Traduzir escolhas para textos compreensíveis
    let escolhaUsuarioTexto, escolhaInimigoTexto;
    switch (escolhaUsuario) {
        case 'A':
            escolhaUsuarioTexto = 'Pedra';
            break;
        case 'B':
            escolhaUsuarioTexto = 'Papel';
```

```

        break;
    case 'C':
        escolhaUsuarioTexto = 'Tesoura';
        break;
    default:
        return "Escolha inválida. Por favor, escolha entre A, B ou C.";
}

switch (escolhaInimigo) {
    case 'A':
        escolhaInimigoTexto = 'Pedra';
        break;
    case 'B':
        escolhaInimigoTexto = 'Papel';
        break;
    case 'C':
        escolhaInimigoTexto = 'Tesoura';
        break;
}

// Determinar o resultado do jogo
let resultado;

if (escolhaUsuario === escolhaInimigo) {
    resultado = "Empate!";
} else if ((escolhaUsuario === 'A' && escolhaInimigo === 'C') ||
    (escolhaUsuario === 'B' && escolhaInimigo === 'A') ||
    (escolhaUsuario === 'C' && escolhaInimigo === 'B')) {
    resultado = "Você ganhou!";
} else {
    resultado = "Você perdeu!";
}

```

```
// Exibir resultado em um alerta

    alert(`Você escolheu ${escolhaUsuarioTexto}. O inimigo escolheu
    ${escolhaInimigoTexto}.\n${resultado}`);
}
```

```
// Exemplo de uso
```

```
const escolhaUsuario = prompt("Escolha uma opção:\nA - Pedra\nB - Papel\nC -
Tesoura").toUpperCase();
jogoJokempo(escolhaUsuario);
```