

Descrição da atividade: fazer os exercícios para praticar. Não precisa entregar as respostas.

Objetivo:

- Arrays;
- Funções
- Estruturas de decisão e repetição;
- String;
- JSON.

Veja o vídeo se tiver dúvidas nos exercícios: <https://youtu.be/I5221kq8KkQ>

Exercício 1: Codificar uma função que recebe uma string contendo números inteiros delimitados por vírgula e retorna um array de números.

Passos do algoritmo:

- Declarar a função converter: a função recebe como parâmetro uma string na variável entrada;
- Usar o método split, do objeto string que está na variável entrada, para obter um array de string e guardar o resultado na variável textos;
- Declarar a variável nros e inicializar com um array vazio;
- Percorrer os elementos do array textos usando uma estrutura de repetição for:
 - Converter cada elemento do array textos para inteiro e colocar o resultado na mesma posição no array nros;
- Retornar o array nros.

Código para testar a função converter:

```
const teste = "30,60,50,10,20,40";  
const resposta = converter(teste);  
console.log(resposta);
```

Resultado esperado:

```
[ 30, 60, 50, 10, 20, 40 ]
```

Dica: método split do objeto string (https://www.w3schools.com/jsref/jsref_split.asp).

Exercício 2: Codificar uma função que recebe uma string contendo um endereço com os campos delimitados por ponto e vírgula e retorna um objeto JSON (JavaScript Object Notation) com as propriedades do endereço.

Passos do algoritmo:

- Declarar a função formatar: a função recebe como parâmetro uma string na variável entrada;

- Usar o método `split`, do objeto `string` que está na variável entrada, para obter um array de `string` e guardar o resultado na variável textos;
- Declarar a variável obj com um objeto onde cada propriedade é um elemento do array textos:
 - Remover os espaços de cada elemento do array antes de atribuir na propriedade do obj;
- Retornar o conteúdo da variável obj.

Código para testar a função `formatar`:

```
const endereco = "R. Faria Lima, 155; Jardim Santa Maria; Jacareí; SP; 12328-070";
const resposta = formatar(endereco);
console.log(resposta);
```

Resultado esperado:

```
{
  logradouro: 'R. Faria Lima, 155',
  bairro: 'Jardim Santa Maria',
  municipio: 'Jacareí',
  uf: 'SP',
  cep: '12328-070'
}
```

Dicas:

- Método `split` do objeto `string` (https://www.w3schools.com/jsref/jsref_split.asp);
- Método `trim` do objeto `string` (https://www.w3schools.com/jsref/jsref_trim_string.asp) para remover espaços no início e fim da `string`.

Exercício 3: Codificar uma função que recebe uma quantidade e retorna um array contendo essa quantidade de números aleatórios no intervalo `[0,99]`.

Passos do algoritmo:

- Declarar a função aleatorio: a função recebe como parâmetro um inteiro na variável quantidade;
- Declarar a variável itens e inicializar com um array vazio;
- Usar uma estrutura de repetição `for` com a variável contadora `i` indo de 0 a quantidade. Para cada repetição:
 - Gerar um número aleatório e colocar na posição `i` do array `itens`;
- Retornar o array `itens`.

Código para testar a função `aleatorio`:

```
const nros = aleatorio(10);
console.log(nros);
```

Resultado esperado:

[45, 2, 46, 62, 74, 23, 69, 43, 26, 16]

Dicas:

- Método random da classe Math (https://www.w3schools.com/jsref/jsref_random.asp);
- Método floor da classe Math (https://www.w3schools.com/jsref/jsref_floor.asp).

Exercício 4: A função aleatorio do Exercício 3 gera números aleatórios no intervalo [0,99]. Alterar a função do Exercício 3 para receber 3 parâmetros, sendo o 1º parâmetro a quantidade de números a serem sorteados, e o 2º e 3º parâmetros o intervalo dos valores sorteados.

Código para testar a função aleatorio:

```
// gerar 10 números aleatórios no intervalo [40,69]
const nros = aleatorio(10,40,70);
console.log(nros);
```

Resultado esperado:

[44, 41, 47, 54, 59, 46, 64, 51, 58, 40]

Dica:

- Na função do Exercício 3 o intervalo era 100, mas aqui o intervalo será a diferença entre o 3º e 2º parâmetros;
- Na função do Exercício 3 o intervalo começava em zero, mas aqui o início do intervalo será deslocado para a posição do 2º parâmetro.

Exercício 5: Alterar a função do Exercício 4 para retornar o array com os elementos ordenados.

Código para testar a função aleatorio:

```
// gerar 10 números aleatórios no intervalo [40,69]
const nros = aleatorio(10,40,70);
console.log(nros);
```

Resultado esperado:

[42, 50, 50, 57, 58, 60, 63, 65, 68, 69]

Dica:

- Chame o método sort na variável itens antes de retornar. O método sort, do objeto Array, retorna um array com os elementos ordenados (https://www.w3schools.com/jsref/jsref_sort.asp).

Exercício 6: Codificar uma função que recebe um array de inteiros e retorna um array contendo apenas os números pares do array de entrada.

Passos do algoritmo:

- Declarar a função pares: a função recebe como parâmetro um array inteiro na variável elementos;
- Declarar a variável itens e inicializar com um array vazio;

- Percorrer os elementos do array elementos usando uma estrutura de repetição for. Para cada repetição:
 - Verificar se o elemento, que está na posição i do array elementos, possui o resto da divisão igual a zero quando dividido por 2. Em caso de verdadeiro:
 - Colocar o elemento, que está na posição i do array elementos, no final do array itens;
- Retornar o array itens.

Código para testar a função pares:

```
const nros = [21,12,18,15,28,19,23,14];  
const resultado = pares(nros);  
console.log(resultado);
```

Resultado esperado:

```
[12, 18, 28, 14]
```

Dica:

- Método push do objeto Array (https://www.w3schools.com/jsref/jsref_push.asp) adiciona um elemento no final do array.

Exercício 7: A função pares do Exercício 6 retorna apenas os números pares. Alterar a função do Exercício 6 para receber como 2º parâmetro um número inteiro, e a função retornará apenas os elementos do array que são múltiplos do número recebido como 2º parâmetro.

Código para testar a função multiplo:

```
const nros = [21,12,18,15,28,19,23,14];  
const resultado = multiplo(nros, 4);  
console.log(resultado);
```

Resultado esperado:

```
[12, 28]
```

Dica:

- Na função do Exercício 6 cada elemento é dividido por 2, agora esse valor precisará ser o número que está no 2º parâmetro da função.

Exercício 8: A função multiplo do Exercício 7 retorna apenas os números múltiplos do 2º parâmetro. Alterar a função do Exercício 7 para retornar os elementos do array que são múltiplos do número recebido no 2º ou 3º parâmetro.

Código para testar a função multiplo:

```
const nros = [21,12,18,15,28,19,23,14];  
const resultado = multiplo(nros, 3, 4);  
console.log(resultado);
```

Resultado esperado:

```
[21, 12, 18, 15, 28]
```

Dica:

- Use um ou lógico || no teste da estrutura de decisão if.

Exercício 9: Alterar a função do Exercício 8 para retornar a quantidade de elementos.

Código para testar a função contamultiplo:

```
const nros = [21,12,18,15,28,19,23,14];  
const resultado = contamultiplo(nros , 3, 4);  
console.log(resultado);
```

Resultado esperado:

```
5
```

Dica:

- Ao invés de guardar os valores na variável itens. Crie uma variável contadora e retorne o seu conteúdo no final.

Exercício 10: Alterar a função do Exercício 9 para retornar a quantidade elementos no formato de um objeto JSON.

Código para testar a função contamultiplo:

```
const nros = [21,12,18,15,28,19,23,14];  
const resultado = contamultiplo(nros , 3, 4);  
console.log(resultado);
```

Resultado esperado:

```
{ quantidade: 5 }
```

Dica:

- À direita do return, envolva a variável quantidade por chaves. Essa instrução fará a estruturação da variável como propriedade JSON.