CEFET/RJ Nova Friburgo
Programação de Clientes Web

Laboratório de ECMAScript 6

As atividades propostas devem ser entregues através da página do aluno no Github pages.

Arrays

Tópicos:

- Tipo de dados que permite armazenar vários valores em uma variável, e em várias dimensões.
- Criação e manipulação de Arrays no JavaScript.
- Comparação de dois arrays.
- Inserção e remoção de elementos de um array
- Iteração com `for .. of` e `for .. in`
- Uso das funções map, filter e reduce

Exercícios:

1. Crie um script que implemente as funções `min`, `max`, `uniq`, `sortNum`. Tais funções devem: retornar o valor mínimo (`min`) e máximo (`max`) de um array de números, retornar o array sem repetições (`uniq`) e ordenar o array numérico (`sortNum`).

Para testar seu programa, utilize as seguintes entradas:

Operação	Entrada	Saída
`min`	`(1, 4, 2, 6, 10, 3)`	`1`
`min`	`(1, 4, -1, 6, 10, 3)`	`-1`
`max`	`(1, 4, 2, 6, 10, 3)`	`10`
`uniq`	`(1, 2, 1, 4, 1, 3)`	`(1, 2, 3, 4)`
`uniq`	`(1, 2, 1, 3, 3)`	`(1, 2, 3)`
`sortNum`	`(1, 3, 2)`	`(1, 2, 3)`
`sortNum`	`(1, 2, 10, 3, 32)`	`(1, 2, 3, 10, 32)`

```
let meuArray = new Array(1, 4, 2, 4, 6, 10, 3);
meuArray.min(); // deve retornar 1
meuArray.max(); // deve retornar 10
meuArray.uniq(); // deve retornar (1, 2, 3, 4, 6, 10)
meuArray.sortNum(); // deve retornar (1, 2, 3, 4, 4, 6, 10)
```

2. Sequência de Fibonacci.
Na matemática, a série de Fibonacci é uma sucessão de números inteiros conforme esta sequência: `0, 1, 1, 2, 3, 5, 8, 13 ...`

Crie uma função para exibir a sequência de Fibonacci até a posição n, por exemplo, a série com os quatro primeiros elementos é `0, 1, 1, 2`. O valor da entrada e saída são respectivamente um número e um vetor. Você deve disponibilizar uma página para entrada do número n e sua respectiva saída.

Para analisar mais exemplos veja a tabela abaixo:

Entrada	Saída
0	
1	0
2	0, 1
4	0, 1, 1, 2
6	0, 1, 1, 2, 3, 5

3. Iterações em Arrays usando map, filter e reduce

considere um array: `const lista = [1, 2, 3, 4, 5, 6]`

a. Implemente uma função `double()`, que retorne o array com cada um dos elementos duplicado. A função deve usar `map`.
b. Implemente uma função `evenElements()`, que retorne um array com apenas os elementos pares do array original. A função deve usar `filter`.
c. Implemente uma função `sum()`, que retorne a soma de todos os elementos do array. A função deve usar `reduce`.