

BASE

OBJETIVO

Praticar os fundamentos

OBJETIVO

Praticar os fundamentos

Básico de javascript

Básico de testes unitários

OBJETIVO

Função

Soma

Multiplicação

Subtração

Vetor

Soma

Multiplicação

Subtração

Objeto

Soma

Multiplicação

Subtração

OBJETIVO

```
const assert = require("assert")
const soma = require("./soma")

describe("# SOMA", () => {
  describe("## caminho feliz", () => {
    it("efetua a soma de dois inteiros positivos", () => {
      const v1 = 3
      const v2 = 4
      const gabarito = 7
      const resultado = soma(v1, v2)
      assert.equal(gabarito, resultado, `a soma entre ${v1} e ${v2} deve ser ${gabarito}, mas foi ${resultado}`)
    })
  })
})
```

BONS TESTES

- 1) Testam comportamento, não implementação
- 2) Não dependem de orquestração
- 3) Facilitam a leitura direta

FUNÇÃO

```
function dobro (x) {  
  return 2 * x;  
}
```


FUNÇÃO

```
var dobro = function dobro (x) {  
  return 2 * x;  
}
```

FUNÇÃO

```
var dobro = function (x) {  
  return 2 * x;  
}
```

FUNÇÃO

```
let dobro = function (x) {  
  return 2 * x;  
}
```

FUNÇÃO

```
let dobro = (x) => {  
  return 2 * x;  
}
```

FUNÇÃO

```
const dobro = x => {  
  return 2 * x  
}
```

FUNÇÃO

```
const dobro = x => 2 * x
```

FUNÇÃO

```
// a/b/soma.js  
  
const soma = () => 7  
  
module.exports = soma
```

Implementar a função soma para que passe em todos os testes

VETORES

- Vetores são Objetos especializados para disposição linear dos seus elementos
- Possui o atributo não enumerável *length* que é automaticamente atualizado
- O seu protótipo possui *push*, *pop*, *sort*, *map*, *join*, etc
- Seu construtor é o *Array*

VETORES

```
const vogais = Array();  
  
vogais.push('a');  
vogais.push('e');  
vogais.push('i');  
vogais.push('o');  
vogais.push('u');
```

```
const vogais = [ 'a', 'e', 'i', 'o', 'u' ];
```

VETORES

```
const a = vogais[0];  
const e = vogais[1];  
const i = vogais[2];  
const o = vogais[3];  
const u = vogais[4];
```

```
const [a, e, i, o, u] = vogais;
```

VETORES

```
const alfabeto = [...vogais, ...consoantes]
```

```
const [a, ...eiou] = vogais
```

VETORES

```
const vogais = [ 'a', 'e', 'i', 'o', 'u' ];  
const VOGAIS = []  
let i  
  
for (i = 0; i < vogais.length; i++) {  
  VOGAIS[i] = vogais[i].toUpperCase()  
}
```

```
const vogais = [ 'a', 'e', 'i', 'o', 'u' ];  
const VOGAIS = vogais.map(v => v.toUpperCase())
```

VETORES

"Recebe 2 vetores e retorna a concatenação de ambos"
"Recebe 2 vetores retorna a soma de todos os elementos"
"Recebe 2 vetores e retorna o primeiro com todos os elementos somados com todo o segundo vetor"

```
function concat () {  
  
}  
function all () {  
  
}  
function parts () {  
  
}  
  
module.exports = function mk_soma(type) {  
  if (type === "concat") {  
    return concat  
  } else if (type === "all") {  
    return all  
  } else if (type === "parts") {  
    return parts  
  }  
}
```


OBJETOS

- Objetos são tipos primitivos do javascript
- São sempre passados por referência
- Há várias opções para configurar os atributos
- Utiliza esquema de dicionário [chave]: valor,
- Estabelece herança prototípica