

```
//MAP:
```

//O map, assim como os arrays e objects, é uma estrutura indexada. Ele é muito semelhante ao objeto, pois possui um conjunto de par "chave" e "valor", porém, o map possui uma diferença essencial, ele pode armazenar qualquer tipo de dado como chave, estamos falando de funções, números, boolean e etc. Qualquer tipo de dado pode ser uma chave, enquanto os objetos só podem armazenar como chave nomes de variáveis e strings quando usamos a notação por colchete.

//Os maps assim como os objetos também possuem a própria classe construtora, o que significa que podemos gerar maps por através de instanciamento e eles também vão ter os métodos e comportamentos próprios.

```
//Vejamos como utilizá-los:
```

```
//GERANDO UM MAP:
```

//Assim como as instâncias comuns, podemos gerar um map por através do operador "new"...

```
const tecnologias = new Map() //Perceba que por através da função construtora Map() já geramos um map...
```

```
//INSERINDO VALORES A UM MAP():
```

//Existem 2 formas de inserir valores num map, podemos fazer isso durante a inicialização, ou por através do método "set()" do map, da seguinte forma:

```
//Atribuindo valores por set():
```

```
tecnologias.set('react', {framework: false}) //Veja que não usamos os ":" como nos objects, separamos chave e valor por vírgula...
```

```
console.log('1', tecnologias) //Note que o javascript identifica o elemento como sendo um "map" e ainda mostra quantos registros existem dentro dele por através de um número dentro dos parênteses. Além disso, ao invés de separá-los por ":" a representação de serapação ao imprimí-los é por através da flecha "=>"...
```

```
//Atribuindo valores por inicialização:
```

```
const chavesVariadas = new Map([ //Veja que para inicializar um map temos usar um array que receberá outros arrays com suas chaves e valores...
```

```
  [function () {}, 'Função'], //Veja que uma função pode ser chave...
```

```
  [{}, 'Objeto'], //Um objeto pode ser chave...
```

```
  [123, 'number'] //Até um número qualquer pode ser chave...
```

```
])
```

```
console.log('\n2)', chavesVariadas) //Note que o número de registros em "chavesVariadas" é diferente do número de registros em "tecnologias"...
```

```
//SOBRESCREVENDO O VALOR DE UMA CHAVE:
```

```
//Para sobrescrever o valor de uma chave, basta nós usarmos o método "set()" referenciando a uma chave já existente e mudar o valor dela...
```

```
chavesVariadas.set(123, 'número') //Veja que usamos o mesmo nome da chave 123, mas trocamos o valor de "number" para "número"...  
console.log('\n3)', chavesVariadas)
```

```
//CAPTURANDO O VALOR DE UM MAP:
```

```
//Veja que podemos usar o método "get()" do map para capturar um valor específico de uma chave...
```

```
console.log('\n4)', tecnologias.get('react').framework) //Veja que usamos o método get() aplicando o nome da chave sobre ele, como esse elemento exclusivamente é um objeto, usamos a notação "ponto" para referenciar a chave "framework" para que pudessemos ter retornado só o valor de "framework"
```

```
//VENDO DE UM MAP POSSUÍ UMA DETERMINADA CHAVE:
```

```
//O map possui o método "has()" que percorre um map verificando se ele possui uma determinada chave que passarmos como parâmetro de "has()", se a chave existir, ele retorna "true", se não existir, ele retorna "false"...
```

```
console.log('\n5)', tecnologias.has('react'))
```

```
//DELETANDO UMA DETERMINADA CHAVE DENTRO DE UM MAP:
```

```
//Para deletar uma chave dentro de um map usamos o método "delete()", colocando o nome da chave como parâmetro de "delete()"...
```

```
chavesVariadas.delete(123)
```

```
chavesVariadas.delete({}) //Detalhe importante, se tentarmos deletar "{}" ou "function" não vamos conseguir, isso acontece por que o método "delete()" só pode deletar chaves que possuem um indentificador numérico ou string...
```

```
console.log('\n6)', chavesVariadas) //Perceba que a chave 123 foi deletada.
```

```
//CAPTURANDO TAMANHO TOTAL DE UM MAP:
```

```
//Podemos capturar o tamanho total de um map usando o atributo "size", da seguinte forma...
```

```
console.log('\n7)', chavesVariadas.size)
```

```
console.log('\n7)', chavesVariadas.size)
```

RESULTADO NO CONSOLE...

```
[Running] node "c:\Users\Almoxarifado\Documents\javascript\arquivos_das_aulas\153-  
map__Estrutura_Indexada_Por_Chaves_de_Diferentes_Tipos.js"
```

```
1) Map(1) { 'react' => { framework: false } }
```

```
2) Map(3) {  
  [Function (anonymous)] => 'Função',  
  {} => 'Objeto',  
  123 => 'number'  
}
```

```
3) Map(3) {  
  [Function (anonymous)] => 'Função',  
  {} => 'Objeto',  
  123 => 'número'  
}
```

```
4) false
```

```
5) true
```

```
6) Map(2) { [Function (anonymous)] => 'Função', {} => 'Objeto' }
```

```
7) 2
```

7) 2

[Done] exited with code=0 in 0.121 seconds