```
//SET:

//O set, assim como os arrays e objects, é uma estrutura indexada. Ele é muito semelhante as arrays, pois são uma estrutura onde
podemos conter diversos valores, porém, o set possuí uma diferença essencial, ele não pode repetir valores, e nem indexa os valores,
ele simplesmente serve para guardar valores que não se repetirão.

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

//Vejamos como utilizá-los:

//GERANDO UM SET:
//Assim como as instâncias comuns, podemos gerar um set por através do operador "new"...
const times = new Set() //Perceba que por através da função construtora Set() já geramos um set...




//INSERINDO VALORES A UM SET():
//Existem 2 formas de inserir valores num set, podemos fazer isso durante a inicialização, ou por através do método "add()" do set,
da seguinte forma:
//Atribuíndo valores por set():
times.add('flamengo') //Veja que só tivemos que armazenar um valor como parâmetro do método "add()"...
times.add(01) //Podemos armazenar valores de tipos diferentes dentro de um set...
times.add('palmeiras').add('corinthians') //Podemos encadear atribuições de valores ao set...
times.add('flamengo') //Note que o set não aceita valores repetidos...
console.log('1)', times) //Note que o javascript identifica o elemento como sendo um "set" e ainda mostra quantos registros existem
dentro dele por através de um número dentro dos parênteses...

//Atribuindo valores por inicialização:
const valoresInicializados = new Set([ //Veja que para inicializar um set temos usar um array que receberá outros arrays com seus
valores...
    'Oi', //Veja que estamos usando valores de tipos diferentes...
    123,
    false,
    'Oi' //Tentamos repetir um, mas veja que ele não entra no set...
```

```
])
console.log('\n2)', valoresInicializados)




//VENDO DE UM SET POSSUÍ UM DETERMINADO VALOR:
//O set possuí o método "has()" que percorre um set verificando se ele possuí um determinado valor que passarmos como parâmetro de "has()", se a chave existir, ele retorna "true", se não existir, ele retorna "false"...
console.log('\n3)', times.has('flamengo'))




//DELETANDO UM DETERMINADO VALOR DENTRO DE UM SET:
//Para deletar um valor dentro de um set usamos o método "delete()", colocando o nome do valor como parâmetro de "delete()"...
valoresInicializados.delete(123)
console.log('\n4)', valoresInicializados) //Perceba que a chave 123 foi deletada.




//CAPTURANDO TAMANHO TOTAL DE UM SET:
//Podemos capturar o tamanho total de um set usando o atributo "size", da seguinte forma...
console.log('\n5)', valoresInicializados.size)
console.log('\n5)', times.size)
```

RESULTADO NO CONSOLE...

```
[Running] node "c:\Users\Almoxarifado\Documents\javascript\arquivos_das_aulas\154-set__Estrutura_de_Conjunto_de_Valores_Nao_Indexados_e_Nao_Repetitivos.js"
1) Set(4) { 'flamengo', 1, 'palmeiras', 'corinthians' }

2) Set(3) { 'Oi', 123, false }

3) true
```

```
4) Set(2) { 'Oi', false }

5) 2

5) 4

[Done] exited with code=0 in 0.113 seconds
```