

```
//FOR(OF):  
//O laço "for(of)" é um laço semelhante ao "for(in)", serve para iterar sobre um elemento iterável, mas enquanto o "for in" tem como foco o array, trazendo como resultado os índices de um array, o "for of" pode ser utilizado em qualquer elemento iterável: strings, arrays, maps e sets, com a única exceção de não iterar sobre Objects. E invés de trazer o valor índice, ele traz o valor que existe dentro destes conjuntos.
```

```
//Vejam os como utilizar o laço for of...
```

```
//USANDO FOR(OF) EM STRINGS:
```

```
const string = 'String'
```

```
for(let letra of string) //Se formos usar só uma linha após o laço "for of" não há necessidade de colocar os colchetes...  
  console.log('1)', letra)
```

```
//USANDO FOR(OF) EM ARRAY:
```

```
const array = ['A', 'r', 'r', 'a', 'y']  
console.log('\n')
```

```
for(let valor of array)  
  console.log('2)', valor)
```

```
//USANDO FOR(OF) EM OBJECT:
```

```
const objeto = {  
  o: 'O',  
  b: 'b',  
  j: 'j',  
  e: 'e',  
  c: 'c',  
  t: 't'  
}  
console.log('\n')
```

```
for(let valor of Object.keys(objeto)) //Perceba que o "for of" não itera sobre objetos diretamente, temos que retirar os nomes das  
chaves por através de um Object.keys() ou os valores pelo Object.values(), para que ele possa iterar sobre um array de nomes de  
chaves...
```

```
console.log('3)', valor)
```

```
//USANDO FOR(OF) EM MAP:
```

```
const map = new Map([ //Perceba que no map, assim como no object temos que trabalhar com chaves e valores...
```

```
  ['Map', {abordado: true}],
```

```
  ['Set', {abordado: true}],
```

```
  ['Promise', {abordado: false}]
```

```
])
```

```
console.log('\n')
```

```
for(let valor of map.keys()) //Como os valores ficam dentro de um array, mas precisamos chamar o construtor "Map" para retirar  
somente as chaves...
```

```
console.log('4)', valor)
```

```
for(let valor of map.values()) //Como os valores ficam dentro de um array, mas precisamos chamar o construtor "Map" para retirar  
somente os valores...
```

```
console.log('4)', valor)
```

```
for(let valor of map.entries()) //Como os valores ficam dentro de um array, mas precisamos chamar o construtor "Map" para retirar  
chaves valores...
```

```
console.log('4)', valor)
```

```
//USANDO FOR(OF) EM SET:
```

```
const set = new Set(['S', 'e', 't'])
```

```
console.log('\n')
```

```
for(let valor of set)
```

```

console.log('5)', valor)

//USANDO LAÇO FOR(OF) PARA CAPTURAR SOMENTE VALORES QUE POSSUEM UM PREFIXO..
//Perceba que criamos uma função que pega uma string para um assunto, se existe algum assunto relacionado dentro de um array de assuntos...
const pesquisar = (assunto, arrayDeAssuntos) => {
    let assuntosEncontrados = []
    for(let palavra of arrayDeAssuntos){ //Veja que dentro do for(of) criamos uma variável "palavra" que irá iterar sobre todos os valores do array "arrayDeAssuntos"
        if (palavra.includes(assunto)) //Usando "includes()" verificamos se o valor armazenado na variável "palavra" durante a iteração é incluso entre os valores do array iterado
            assuntosEncontrados.push(palavra) //Para cada iteração onde o includes() é verdadeiro o valor é adicionado ao array criado dentro da função e que será retornado...
        }

        return console.log('\n6)', assuntosEncontrados)
    }
}

pesquisar('pro', ['programação', 'mobile', 'profissional'])

```

RESULTADO NO CONSOLE...

```

[Running] node "c:\Users\Almoxarifado\Documents\javascript\arquivos_das_aulas\126-for_of.js"
1) S
1) t
1) r
1) i
1) n
1) g

2) A

```

2) r  
2) r  
2) a  
2) y

3) o  
3) b  
3) j  
3) e  
3) c  
3) t

4) Map  
4) Set  
4) Promise  
4) { abordado: true }  
4) { abordado: true }  
4) { abordado: false }  
4) [ 'Map', { abordado: true } ]  
4) [ 'Set', { abordado: true } ]  
4) [ 'Promise', { abordado: false } ]

5) S  
5) e  
5) t

6) [ 'programação', 'profissional' ]

[Done] exited with code=0 in 0.158 seconds