



# Destructuring

Destructuring é uma forma conveniente de **extrair múltiplos valores** de arrays e objetos de uma só vez



Array Matching

1. `let a, b;`
2. `[a, b] = [1, 2];`

Atribuindo valores **diretamente**  
a variáveis já existentes

```
1. let [a, b] = [1, 2];
```

Criando variáveis e atribuindo valores  
de uma só vez

```
1. var [a, ,b] = [1, 2, 3];
```

**Ignorando** valores na criação de variáveis

```
1. var [a = 1, b = 2, c = 3] = [10];
```

Criando **default value**, caso algum valor não esteja disponível

```
1. var [a, b, ...c] = [1, 2, 3, 4, 5, 6];
```

Criando variáveis e atribuindo valores com  
**spread operator**



1. `var a = 1;`
2. `var b = 2;`
3. `[a, b] = [b, a]`

Realizando **troca de valores** sem a necessidade de uma variável temporária



Object Matching

```
1. var {a, c} = {a: 10, b: 20, c: 30};
```

**Atribuindo valores diretamente** a variáveis  
por meio de um objeto

```
1. var {d: a, f: c} = {d: 10, e: 20, f: 30};
```

Atribuindo valores diretamente a variáveis  
com conversão do nome da propriedade

```
1. var {d: a, f: {g: c}} = {d: 10, e: 20, f: {g: 30}};
```

Atribuindo valores diretamente a variáveis  
com conversão do nome da propriedade

```
1.  var books = [{  
2.    title: "Clean Code",  
3.    pages: 240  
4.  },  
5.  {  
6.    title: "Refactoring",  
7.    pages: 342  
8.  }];  
9.  
10. for(var {title, pages} of books) {  
11.   console.log(`O ${title} tem ${pages} páginas`);  
12. }
```

Utilizando object matching **dentro de um for/of** para criar variáveis

```
1.  var books = [{
2.    title: "Clean Code",
3.    pages: 240
4.  },
5.  {
6.    title: "Refactoring"
7.  }];
8.
9.  for(var {title, pages = "-"} of books) {
10.    console.log(`O ${title} tem ${pages} páginas`);
11.  }
```

Utilizando object matching dentro de um for/of para criar variáveis com **default value**