

Métodos

A classe List possui alguns métodos, os quais podemos utilizar para manipular o valor do objeto. Vejamos quais são eles e como podemos utilizá-los.

Método add

O método add, como o nome sugere, adiciona um novo elemento ao final da lista.

Parâmetro	Descrição
elemento	Elemento a ser adicionado à lista

```
1 void main() {  
2     List<String> vingadores = List.empty(growable: true);  
3  
4     //Imprimimos a lista  
5     print(vingadores);  
6  
7     // Removemos o elemento da lista  
8     vingadores.add("Homem de Ferro");  
9  
10    //Imprimimos a lista novamente  
11    print(vingadores);  
12 }  
13
```

É importante ressaltar que no exemplo anterior, criamos uma lista vazia utilizando o construtor empty da própria classe List. O mesmo possui uma propriedade chamada growable. Se growable for false, que é o padrão, a lista criada terá comprimento fixo de comprimento zero e não poderá ter elementos adicionados. Se growable for true, a lista é “cultivável” e equivalente a <Tipo> [] e poderá ter elementos adicionados a ela.

Método remove

O método remove, como o nome sugere, remove um elemento da lista. Caso seja possível remover, um valor true será retornado, caso contrário será retornado um valor falso.

Parâmetro	Descrição
-----------	-----------

elemento	Elemento a ser removido da lista
----------	----------------------------------

```

1 void main() {
2     List<String> vingadores = [
3         "Homem de ferro",
4         "Capitão América",
5         "Thor",
6         "Hulk"
7     ];
8
9     //Imprimimos a lista
10    print(vingadores);
11
12    // Removemos o elemento da lista
13    vingadores.remove("Thor");
14
15    //Imprimimos a lista novamente
16    print(vingadores);
17 }
18

```

Método removeAt

O método removeAt é muito parecido com o método remove, diferenciando-se pelo fato de remover o elemento com base em seu índice na lista. Caso seja possível remover, um valor true será retornado, caso contrário será retornado um valor falso.

Parâmetro	Descrição
índice	Índice do elemento a ser removido

```
1 void main() {  
2     List<String> vingadores = [  
3         "Homem de ferro",  
4         "Capitão América",  
5         "Thor",  
6         "Hulk"  
7     ];  
8  
9     //Imprimimos a lista  
10    print(vingadores);  
11  
12    // Removemos o elemento da lista  
13    vingadores.removeAt(1);  
14  
15    //Imprimimos a lista novamente  
16    print(vingadores);  
17 }  
18
```

É importante ressaltar que os índices de uma lista iniciam em 0 (zero). Desta forma o primeiro elemento da lista é o de índice 0 (zero), o segundo de índice 1 e assim sucessivamente.

Método clear

O método clear, como o nome sugere, limpa a lista. Ou seja, remove todos os elementos presentes na mesma.

```

1 void main() {
2     List<String> vingadores = [
3         "Homem de ferro",
4         "Capitão América",
5         "Thor",
6         "Hulk"
7     ];
8     // Imprimimos a lista
9     print(vingadores);
10
11    // Limpando a lista
12    vingadores.clear();
13
14    // Imprimimos a lista
15    print(vingadores);
16 }

```

Método contains

O método contains, como o próprio nome já sugere, verifica se a lista contém um determinado elemento. Caso o elemento esteja na lista, um valor true será retornado, caso contrário será retornado um valor falso.

Parâmetro	Descrição
elemento	Elemento ao qual estamos procurando

```

1 void main() {
2     List<String> vingadores = [
3         "Homem de ferro",
4         "Capitão América",
5         "Thor",
6         "Hulk"
7     ];
8     // Procuraremos pelo Capitão América
9     print(vingadores.contains("Capitão América"));
10 }

```