

# Cartilha de Métodos de Array em Java

A classe `String` em Java é uma das mais utilizadas e possui uma grande variedade de métodos para manipulação de texto. A seguir, apresentamos os principais métodos com exemplos práticos.

## 1. `length()`

Retorna o número de caracteres da string.

```
String texto = "Java";  
System.out.println(texto.length()); // 4
```

## 2. `charAt(int index)`

Retorna o caractere na posição especificada.

```
String texto = "Java";  
System.out.println(texto.charAt(2)); // v
```

## 3. `substring(int beginIndex, int endIndex)`

Retorna uma nova string com os caracteres entre os índices especificados.

```
String texto = "Programação";  
System.out.println(texto.substring(0, 4)); // Prog
```

## 4. `toUpperCase()`

Converte todos os caracteres da string para maiúsculas.

```
String texto = "java";  
System.out.println(texto.toUpperCase()); // JAVA
```

## 5. `toLowerCase()`

Converte todos os caracteres da string para minúsculas.

```
String texto = "JAVA";  
System.out.println(texto.toLowerCase()); // java
```

## 6. `equals(String anotherString)`

Compara duas strings (sensível a maiúsculas/minúsculas).

```
String a = "Java";  
String b = "java";  
System.out.println(a.equals(b)); // false
```

## Cartilha de Métodos de Array em Java

### 7. equalsIgnoreCase(String anotherString)

Compara duas strings ignorando maiúsculas/minúsculas.

```
String a = "Java";  
String b = "java";  
System.out.println(a.equalsIgnoreCase(b)); // true
```

### 8. contains(CharSequence seq)

Verifica se a string contém a sequência de caracteres especificada.

```
String frase = "Estou aprendendo Java";  
System.out.println(frase.contains("Java")); // true
```

### 9. replace(char oldChar, char newChar)

Substitui caracteres na string.

```
String texto = "banana";  
System.out.println(texto.replace('a', 'o')); // bonono
```

### 10. trim()

Remove espaços em branco no início e no fim da string.

```
String texto = "  Java  ";  
System.out.println(texto.trim()); // "Java" (sem espaços)
```

- Strings em Java são imutáveis: qualquer alteração cria um novo objeto.
- Use equals() e não == para comparar conteúdos de strings.
- Para muitas concatenações, prefira StringBuilder por performance.

A classe String oferece métodos poderosos para manipulação textual. Entender e saber aplicá-los no momento certo faz toda a diferença no desenvolvimento de aplicações Java.