**Métodos com retorno**

Reaproveitar o que já foi feito no código\*\*

Criar método -> Sistema

Pode receber variáveis como parâmetros e devolver uma variável como retorno da execução do método.

\*\*Declarar qual método de retorno\*\*  
Algo semelhante ao uso de classes em Portugol.

Exemplo 1 -> **public** **int** **metodo1()** **{**

**System.**out**.**println**(**"Chamou o metodo 1."**);**

**return** atributo1**;**

**}**

**-> Método utilizado para retornar o atributo1**

Exemplo 2 -> **public** **boolean** **metodo2()** **{**

**System.**out**.**println**(**"Chamou o metodo 2."**);**

**return** atributo1 **>=** 0**;**

**}**

**-> Método que verificar se o atributo1 é maior ou igual a 0**

**EXEMPLO DO CÓDIGO:**

**public** **class** **MetodoRetorno** **{**

**public** **int** atributo1**;**

**public** **int** **metodo1()** **{**

**System.**out**.**println**(**"Chamou o metodo 1."**);**

**return** atributo1**;**

**}**

**public** **boolean** **metodo2()** **{**

**System.**out**.**println**(**"Chamou o metodo 2."**);**

**return** atributo1 **>=** 0**;**

**}**

**}**

**USO DO INT COMO PALAVRA CHAVE**

Utilizado para mostrar que esperamos o retorno do método para o que chamou um valor do tipo inteiro.

Para identificar qual o valor de retorno -> **return** “ ”;

**Exemplo -> public** **int** **metodo1()** **{**

**System.**out**.**println**(**"Chamou o metodo 1."**);**

**return** atributo1**;**

**}**

(Nesse caso, método 1 irá fazer o retorno do valor equivalente de atributo1).

**OBS. Caso não aconteça o retorno para o método, pode acontecer erro na compilação dentro do programa.**

**Exemplo -> public** **boolean** **metodo2()** **{**

**System.**out**.**println**(**"Chamou o metodo 2."**);**

**return** atributo1 **>=** 0**;**

**}**

->Verificação se “atributo1” é um valor maior ou igual à zero, fazendo retorno da resposta se é verdadeiro ou falso (true or false).

**->Exemplo de saída no console:**

**C:**\**>**javac **MetodoRetorno.**java

**C:**\**>**javac **TesteMetodoRetorno.**java

**C:**\**>**java **TesteMetodoRetorno**

**Criando** um objeto da classe **MetodoRetorno**

**Chamou** o metodo 1**.**

30

**Chamou** o metodo 2**.**

**true**

Método com retorno

<http://www.universidadejava.com.br/java/java-metodos-retorno/>

**Métodos sem retorno**

Esse tipo de método executa apenas o código que tem dentro, não retornando nenhum resultado.

**USO DO VOID COMO PALAVRA CHAVE**

**EXEMPLO ->** class Metodo2{

public void escrever()

{

System.out.println("Método sem Retorno - VOID ");

}

}

public class Metodos\_Sem\_Retorno {

public static void main(String[] args) {

Metodo2 m = new Metodo2();

m.escrever();

}

}

**Metodos sem retorno**

<https://www.devmedia.com.br/trabalhando-com-metodos-em-java/25917#:~:text=M%C3%A9todos%20sem%20retorno%20em%20Java,com%20a%20palavra%2Dchave%20void>.

**DIFERENÇA ENTRE MÉTODO COM E SEM RETORNO**

**A diferença é que o método sem retorno funciona como um serviço e o método com retorno como uma função.**

O método com retorno irá trazer para outra parte do código um valor ou objeto que poderá ser usado. Dado seu exemplo:

public void somaDois (int a, int b){

System.out.println(a+b);

}

Esse método não terá retorno: ele terá apenas a função de somar dois numeros. Portanto, o resultado da soma não está sendo utilizado em nenhuma parte do código, apenas está sendo printado no console.

public int somaDois (int a, int b){

return a+b;

}

Com o retorno desse método, você obterá um dado do tipo int contendo a soma de a+b. Você poderia usá-lo assim:

int resultadoDaSoma = somaDois(1,3);

System.out.println(resultadoDaSoma);

**- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -**

**RESUMO**

Um método que retorna algo é necessário quando você quer buscar algum valor, como por exemplo um método getNome, onde está trazendo o nome do objeto.

Um método "sem retorno", é usado quando você tem a intenção de alterar algum dado no sistema, mas não quer ter que dar retorno de nada, como por exemplo um método setNome. Na realidade todos os métodos possuem algum retorno, porém métodos void retornam vazio, ou seja, nada.

**Métodos String**

Método embutidos uteis para String

Exemplos -> encontrar o comprimento, separa-la ou uni-la com algo.

**EXEMPLO COMPRIMENTO**

public class JavaEstudos {

public static void main(String[] args) {

String teste = "HENRIQUE";

int testeresultado;

testeresultado = teste.length();

System.out.println(testeresultado);

}

}

**OBS. Pode ser utilizado para classificar uma ordem de nomes de acordo com o comprimento deles, ou fazer o bloqueio para o usuário caso o nome que ele tente inserir seja maior que o limite de um campo de registro, por exemplo.**

**EXEMPLO ENCONTRAR CARACTER ESPECIFICO**

public class JavaEstudos {

public static void main(String[] args) {

String teste = "HENRIQUE";

System.out.println(teste.charAt(0));

}

}

**OBS. Utilizado para encontrar um caráter especifica da String, no exemplo, zero é equivalente ao “H”.**

**EXEMPLO CODEPOINTAT**

Retorna o valor em UNICODE do caractere especificado no index do parâmetro.

* String nome = "Dinossauro";

System.out.println(nome.codePointAt(0));

Saída: 68

Note que a codificação da letra 'D' correspondente a posição 0 na nossa String é igual a 68, considerando a codificação UNICODE.

**EXEMPLOS SUBSTRING E SUBSEQUENCE**

Ambos retornam uma parte especifica da String, porém o SubString retorna uma nova String enquanto o método SubSequence retorna uma sequência de Char (CharSequence).

**OBS. É importante ressaltar que o CharSequence é uma interface e a String é a implementação dessa interface.**

String nome = "HENRIQUE";

System.out.println (nome.subSequence(0, 5));

System.out.println (nome.substring(0, 5));

Saída:

HENRI

HENRI

## **EXEMPLOS REVERSE (StringBuilder)**

->Utilizado para fazer a inversão de uma String.

public class JavaEstudos {

public static String reverse(String str) {

return new StringBuilder(str).reverse().toString();

}

public static void main(String[] args) {

String nome = "HENRIQUE";

nome = reverse (nome);

System.out.println("Meu nome ao inverso é { " + nome + " }");

}

}

**EXEMPLOS REPLACE**

**->**Método utilizado para substituir caracteres dentro de uma String,

public class JavaEstudos {

public static void main(String[] args) {

String texto = "hello world";

String leitura = texto.replace('o', 'a');

System.out.println(leitura);

}

}

OBS. O programa realizou a troca dos caracteres ‘o’ pelos caracteres ‘a’.

**EXEMPLOS TRIM**

-> Método utilizado para retirar espaços vazios do início e do fim de uma String.

public class JavaEstudos {

public static void main(String[] args) {

String texto = " HENRIQUE ";

texto = texto.trim();

System.out.println("-" + texto + "-");

}

}