## **Java ArrayList**

A ArrayListclasse é uma [matriz](https://www.w3schools.com/java/java_arrays.asp) redimensionável , que pode ser encontrada no java.utilpacote.

A diferença entre uma matriz interna e uma ArrayListem Java é que o tamanho de uma matriz não pode ser modificado (se você deseja adicionar ou remover elementos de / para uma matriz, é necessário criar um novo). Os elementos while podem ser adicionados e removidos de um ArrayListquando você quiser. A sintaxe também é um pouco diferente:

### Exemplo

Crie um ArrayListobjeto chamado **cars** que armazenará strings:

import java.util.ArrayList; // import the ArrayList class  
  
ArrayList<String> cars = new ArrayList<String>(); // Create an ArrayList object

Se você não sabe o que é um pacote, leia nosso [Tutorial de Pacotes Java](https://www.w3schools.com/java/java_packages.asp) .

## **Adicionar itens**

A ArrayListclasse tem muitos métodos úteis. Por exemplo, para adicionar elementos ao ArrayList, use o add()método:

### Exemplo

import java.util.ArrayList;  
  
public class MyClass {  
  public static void main(String[] args) {  
    ArrayList<String> cars = new ArrayList<String>();  
    cars.add("Volvo");  
    cars.add("BMW");  
    cars.add("Ford");  
    cars.add("Mazda");  
    System.out.println(cars);  
  }  
}

## **Acessar um item**

Para acessar um elemento no ArrayList, use o get()método e consulte o número do índice:

### Exemplo

cars.get(0);

**Lembre-se: os** índices de matriz começam com 0: [0] é o primeiro elemento. [1] é o segundo elemento, etc.

## **Alterar um item**

Para modificar um elemento, use o set()método e consulte o número do índice:

### Exemplo

cars.set(0, "Opel");

## **Remover um item**

Para remover um elemento, use o remove()método e consulte o número do índice:

### Exemplo

cars.remove(0);

Para remover todos os elementos no ArrayList, use o clear()método:

### Exemplo

cars.clear();

## **Tamanho da lista**

Para descobrir quantos elementos um ArrayList possui, use o sizemétodo:

### Exemplo

cars.size();

## **Loop através de um ArrayList**

Faça um loop pelos elementos de um ArrayListcom um forloop e use o size()método para especificar quantas vezes o loop deve ser executado:

### Exemplo

public class MyClass {  
  public static void main(String[] args) {  
    ArrayList<String> cars = new ArrayList<String>();  
    cars.add("Volvo");  
    cars.add("BMW");  
    cars.add("Ford");  
    cars.add("Mazda");  
    for (int i = 0; i < cars.size(); i++) {  
      System.out.println(cars.get(i));  
    }  
  }  
}

Você também pode percorrer um loop ArrayListcom o **for-each** :

### Exemplo

public class MyClass {  
  public static void main(String[] args) {  
    ArrayList<String> cars = new ArrayList<String>();  
    cars.add("Volvo");  
    cars.add("BMW");  
    cars.add("Ford");  
    cars.add("Mazda");  
    for (String i : cars) {  
      System.out.println(i);  
    }  
  }  
}

## **Outros tipos**

Elementos em um ArrayList são realmente objetos. Nos exemplos acima, criamos elementos (objetos) do tipo "String". Lembre-se de que uma String em Java é um objeto (não um tipo primitivo). Para usar outros tipos, como int, você deve especificar um equivalente [classe de mensagens publicitárias](https://www.w3schools.com/java/java_wrapper_classes.asp) : Integer. Para outros tipos primitivos, use: Booleanpara booleano, Characterchar, Doubledouble, etc:

### Exemplo

Crie um ArrayListnúmero para armazenar (adicione elementos do tipo Integer):

import java.util.ArrayList;  
  
public class MyClass {  
  public static void main(String[] args) {  
    ArrayList<Integer> myNumbers = new ArrayList<Integer>();  
    myNumbers.add(10);  
    myNumbers.add(15);  
    myNumbers.add(20);  
    myNumbers.add(25);  
    for (int i : myNumbers) {  
      System.out.println(i);  
    }  
  }  
}

## **Classificar um ArrayList**

Outra classe útil no java.utilpacote é a Collectionsclasse, que inclui o sort()método para classificar as listas alfabeticamente ou numericamente:

### Exemplo

Classificar uma matriz de strings:

import java.util.ArrayList;  
import java.util.Collections;  // Import the Collections class  
  
public class MyClass {  
  public static void main(String[] args) {  
    ArrayList<String> cars = new ArrayList<String>();  
    cars.add("Volvo");  
    cars.add("BMW");  
    cars.add("Ford");  
    cars.add("Mazda");  
  
    Collections.sort(cars);  // Sort cars  
  
    for (String i : cars) {  
      System.out.println(i);  
    }  
  }  
}

### Exemplo

Classifique uma ArrayList de números inteiros:

import java.util.ArrayList;  
import java.util.Collections;  // Import the Collections class  
  
public class MyClass {  
  public static void main(String[] args) {  
    ArrayList<Integer> myNumbers = new ArrayList<Integer>();  
    myNumbers.add(33);  
    myNumbers.add(15);  
    myNumbers.add(20);  
    myNumbers.add(34);  
    myNumbers.add(8);  
    myNumbers.add(12);  
  
    Collections.sort(myNumbers);  // Sort myNumbers  
  
    for (int i : myNumbers) {  
      System.out.println(i);  
    }  
  }  
}