**JAVA.**



Java is a popular programming language.

Java is used to develop mobile apps, web apps, desktop apps, games and much more.

Example

public class Main {

public static void main(String[] args) {

System.out.println("Hello World");

}

}

## **What is Java?**

Java is a popular programming language, created in 1995.

It is owned by Oracle, and more than **3 billion** devices run Java.

It is used for:

* Mobile applications (specially Android apps)
* Desktop applications
* Web applications
* Web servers and application servers
* Games
* Database connection
* And much, much more!

## **Why Use Java?**

* Java works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc.)
* It is one of the most popular programming language in the world
* It is easy to learn and simple to use
* It is open-source and free
* It is secure, fast and powerful
* It has a huge community support (tens of millions of developers)
* Java is an object oriented language which gives a clear structure to programs and allows code to be reused, lowering development costs
* As Java is close to [C++](https://www.w3schools.com/cpp/default.asp) and [C#](https://www.w3schools.com/cs/default.asp), it makes it easy for programmers to switch to Java or vice versa

## **Java Variables**

Variables are containers for storing data values.

In Java, there are different **types** of variables, for example:

* String - stores text, such as "Hello". String values are surrounded by double quotes
* int - stores integers (whole numbers), without decimals, such as 123 or -123
* float - stores floating point numbers, with decimals, such as 19.99 or -19.99
* char - stores single characters, such as 'a' or 'B'. Char values are surrounded by single quotes
* boolean - stores values with two states: true or false

## **Declaring (Creating) Variables**

To create a variable, you must specify the type and assign it a value:

### Syntax

*type variableName = value;*

Where type is one of Java's types (such as int or String), and variableName is the name of the variable (such as **x** or **name**). The **equal sign** is used to assign values to the variable.

To create a variable that should store text, look at the following example:

### Example

Create a variable called **name** of type String and assign it the value "**John**":

String name = "John";

System.out.println(name);

## **Java Data Types**

As explained in the previous chapter, a variable in Java must be a specified data type:

### Example

int myNum = 5; // Integer (whole number)

float myFloatNum = 5.99f; // Floating point number

char myLetter = 'D'; // Character

boolean myBool = true; // Boolean

String myText = "Hello"; // String

## **Primitive Data Types**

A primitive data type specifies the size and type of variable values, and it has no additional methods.

There are eight primitive data types in Java:

|  |  |  |
| --- | --- | --- |
| **Data Type** | **Size** | **Description** |
| byte | 1 byte | Stores whole numbers from -128 to 127 |
| short | 2 bytes | Stores whole numbers from -32,768 to 32,767 |
| int | 4 bytes | Stores whole numbers from -2,147,483,648 to 2,147,483,647 |
| long | 8 bytes | Stores whole numbers from -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807 |
| float | 4 bytes | Stores fractional numbers. Sufficient for storing 6 to 7 decimal digits |
| double | 8 bytes | Stores fractional numbers. Sufficient for storing 15 decimal digits |
| boolean | 1 bit | Stores true or false values |
| char | 2 bytes | Stores a single character/letter or ASCII values |

## **Numbers**

Primitive number types are divided into two groups:

**Integer types** stores whole numbers, positive or negative (such as 123 or -456), without decimals. Valid types are byte, short, int and long. Which type you should use, depends on the numeric value.

**Floating point types** represents numbers with a fractional part, containing one or more decimals. There are two types: float and double.

# **Java Keywords**

|  |  |
| --- | --- |
| **Keyword** | **Description** |
| [abstract](https://www.w3schools.com/java/ref_keyword_abstract.asp) | A non-access modifier. Used for classes and methods: An abstract class cannot be used to create objects (to access it, it must be inherited from another class). An abstract method can only be used in an abstract class, and it does not have a body. The body is provided by the subclass (inherited from) |
| assert | For debugging |
| [boolean](https://www.w3schools.com/java/ref_keyword_boolean.asp) | A data type that can only store true and false values |
| [break](https://www.w3schools.com/java/ref_keyword_break.asp) | Breaks out of a loop or a switch block |
| [byte](https://www.w3schools.com/java/ref_keyword_byte.asp) | A data type that can store whole numbers from -128 and 127 |
| [case](https://www.w3schools.com/java/ref_keyword_case.asp) | Marks a block of code in switch statements |
| [catch](https://www.w3schools.com/java/ref_keyword_catch.asp) | Catches exceptions generated by try statements |
| [char](https://www.w3schools.com/java/ref_keyword_char.asp) | A data type that is used to store a single character |
| [class](https://www.w3schools.com/java/ref_keyword_class.asp) | Defines a class |
| [continue](https://www.w3schools.com/java/ref_keyword_break.asp) | Continues to the next iteration of a loop |
| const | Defines a constant. Not in use - use [final](https://www.w3schools.com/java/ref_keyword_final.asp) instead |
| [default](https://www.w3schools.com/java/ref_keyword_default.asp) | Specifies the default block of code in a switch statement |
| [do](https://www.w3schools.com/java/ref_keyword_do.asp) | Used together with while to create a do-while loop |
| [double](https://www.w3schools.com/java/ref_keyword_double.asp) | A data type that can store whole numbers from 1.7e−308 to 1.7e+308 |
| [else](https://www.w3schools.com/java/ref_keyword_else.asp) | Used in conditional statements |
| [enum](https://www.w3schools.com/java/ref_keyword_enum.asp) | Declares an enumerated (unchangeable) type |
| exports | Exports a package with a module. New in Java 9 |
| [extends](https://www.w3schools.com/java/ref_keyword_extends.asp) | Extends a class (indicates that a class is inherited from another class) |
| [final](https://www.w3schools.com/java/ref_keyword_final.asp) | A non-access modifier used for classes, attributes and methods, which makes them non-changeable (impossible to inherit or override) |
| [finally](https://www.w3schools.com/java/ref_keyword_finally.asp) | Used with exceptions, a block of code that will be executed no matter if there is an exception or not |
| [float](https://www.w3schools.com/java/ref_keyword_float.asp) | A data type that can store whole numbers from 3.4e−038 to 3.4e+038 |
| [for](https://www.w3schools.com/java/ref_keyword_for.asp) | Create a for loop |
| goto | Not in use, and has no function |
| [if](https://www.w3schools.com/java/ref_keyword_if.asp) | Makes a conditional statement |
| [implements](https://www.w3schools.com/java/ref_keyword_implements.asp) | Implements an interface |
| [import](https://www.w3schools.com/java/ref_keyword_import.asp) | Used to import a package, class or interface |
| [instanceof](https://www.w3schools.com/java/ref_keyword_instanceof.asp) | Checks whether an object is an instance of a specific class or an interface |
| [int](https://www.w3schools.com/java/ref_keyword_int.asp) | A data type that can store whole numbers from -2147483648 to 2147483647 |
| [interface](https://www.w3schools.com/java/ref_keyword_interface.asp) | Used to declare a special type of class that only contains abstract methods |
| [long](https://www.w3schools.com/java/ref_keyword_long.asp) | A data type that can store whole numbers from -9223372036854775808 to 9223372036854775808 |
| module | Declares a module. New in Java 9 |
| native | Specifies that a method is not implemented in the same Java source file (but in another language) |
| [new](https://www.w3schools.com/java/ref_keyword_new.asp) | Creates new objects |
| [package](https://www.w3schools.com/java/ref_keyword_package.asp) | Declares a package |
| [private](https://www.w3schools.com/java/ref_keyword_private.asp) | An access modifier used for attributes, methods and constructors, making them only accessible within the declared class |
| [protected](https://www.w3schools.com/java/ref_keyword_protected.asp) | An access modifier used for attributes, methods and constructors, making them accessible in the same package and subclasses |
| [public](https://www.w3schools.com/java/ref_keyword_public.asp) | An access modifier used for classes, attributes, methods and constructors, making them accessible by any other class |
| requires | Specifies required libraries inside a module. New in Java 9 |
| [return](https://www.w3schools.com/java/ref_keyword_return.asp) | Finished the execution of a method, and can be used to return a value from a method |
| [short](https://www.w3schools.com/java/ref_keyword_short.asp) | A data type that can store whole numbers from -32768 to 32767 |
| [static](https://www.w3schools.com/java/ref_keyword_static.asp) | A non-access modifier used for methods and attributes. Static methods/attributes can be accessed without creating an object of a class |
| strictfp | Restrict the precision and rounding of floating point calculations |
| [super](https://www.w3schools.com/java/ref_keyword_super.asp) | Refers to superclass (parent) objects |
| [switch](https://www.w3schools.com/java/ref_keyword_switch.asp) | Selects one of many code blocks to be executed |
| synchronized | A non-access modifier, which specifies that methods can only be accessed by one thread at a time |
| [this](https://www.w3schools.com/java/ref_keyword_this.asp) | Refers to the current object in a method or constructor |
| [throw](https://www.w3schools.com/java/ref_keyword_throw.asp) | Creates a custom error |
| [throws](https://www.w3schools.com/java/ref_keyword_throws.asp) | Indicates what exceptions may be thrown by a method |
| transient | A non-accesss modifier, which specifies that an attribute is not part of an object's persistent state |
| [try](https://www.w3schools.com/java/ref_keyword_try.asp) | Creates a try...catch statement |
| var | Declares a variable. New in Java 10 |
| [void](https://www.w3schools.com/java/ref_keyword_void.asp) | Specifies that a method should not have a return value |
| volatile | Indicates that an attribute is not cached thread-locally, and is always read from the "main memory" |
| [while](https://www.w3schools.com/java/ref_keyword_while.asp) | Creates a while loop |

**Note:** true, false, and null are not keywords, but they are literals and reserved words that cannot be used as identifiers.