schools Menu ▾ Log in

Java Booleans

❮ Previous Next ❯

Java Booleans

Very often, in programming, you will need a data type that can only have one of two values, like:

- YES / NO
- ON / OFF
- TRUE / FALSE

For this, Java has a `boolean` data type, which can store `true` or `false` values.

Boolean Values

A boolean type is declared with the `boolean` keyword and can only take the values `true` or `false`:

Example

```
boolean isJavaFun = true;
boolean isFishTasty = false;
```

□ Dark mode

Try it Yourself »

However, it is more common to return boolean values from boolean expressions, for conditional testing (see below).

Boolean Expression

A **Boolean expression** is a Java expression that **returns** a Boolean value: `true` or `false`.

This is useful when we want to compare values to find answers.

For example, you can use a comparison operator, such as the **greater than** (`>`) operator, to find out if an expression (or a variable) is `true` :

Example

```
int x = 10;
int y = 9;
System.out.println(x > y); // returns true, because 10 is higher than 9
```

Try it Yourself »

Or even easier:

Example

```
System.out.println(10 > 9); // returns true, because 10 is higher than 9
```

Try it Yourself »

□ Dark mode

In the examples below, we use the **equal to** (`==`) operator to evaluate an expression:

Example

```
int x = 10;
System.out.println(x == 10); // returns true, because the value of x is equal tc
```

Try it Yourself »

Example

```
System.out.println(10 == 15); // returns false, because 10 is not equal to 15
```

Try it Yourself »

Real Life Example

Let's think of a "real life example" where we need to find out if a person is old enough to vote.

In the example below, we use the `>=` comparison operator to find out if the age (`25`) is **greater than** OR **equal to** the voting age limit, which is set to `18` :

Example

```
int myAge = 25;
int votingAge = 18;
System.out.println(myAge >= votingAge);
```

□ Dark mode

The Boolean value of an expression is the basis for all Java comparisons and conditions.

You will learn more about conditions in the next chapter.

Test Yourself With Exercises

Exercise:

Fill in the missing parts to print the values `true` and `false` :

```
          isJavaFun = true;
          isFishTasty = false;
System.out.println(isJavaFun);
System.out.println(isFishTasty);
```

Submit Answer »

Start the Exercise

❮ Previous

Next ❯

□ Dark mode