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 take the values `true` or `false`.

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

Run example »

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`.

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
```

Run example »

Or even easier:

Example

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

Run example »

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 to 10
```

Run example »

Example

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

Run example »

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.

‹ Previous

Next ›