A `boolean` variable can hold either the value `true` or the value `false`. Boolean values are quite useful when directing a program to take repeated actions, or to take actions only under some circumstances.

Note that Java’s `true` and `false` values are written with lowercase, unlike Python’s.

```java
package com.github.akarazhev.jacademy.jprog.basics;

public final class BooleanValues {

    public static void main(final String[] args) {
        final boolean b = true;
        System.out.println(b);
        System.out.println(false);
    }
}
```

Conditional operators: >, <, >=, <=, ==, !=

Conditional operators work just like they do in most common programming languages. You can compare ints, doubles, and Strings, yielding a `true` or `false` value.

A very typical coding error is to type `=` when you meant `==`: assigning a value to a variable when you meant to test the value of that variable. Be careful – the Java compiler will not catch this error for you:

```java
package com.github.akarazhev.jacademy.jprog.basics;

public final class AssignmentError {

    public static void main(final String[] args) {
        int x = 5;
        System.out.println(x = 4);
    }
}
```

Logical operators: `&&`, `||`, and `!`

The *and* operator is written as `&&` in Java. It yields `true` if both operands are `true`. The *or* operator `||` works as you would expect from other languages, too. The *not* operator, `!` precedes a boolean `true` or `false` value that you would like to negate.