if-then Statement

In the last video, we created our first class in Java using the IntelliJ IDE. So now it's time to get back to studying Java.

In this video, we're going to take a look at some more operators, but before that, I need to talk about the **if-then** statement first.

if-then statements in Java

The **if-then** statement is the most basic of all the control flow statements. It tells your program to execute a certain section of code, only if a particular tests evaluates to **true**.

This is known as **conditional logic**.

Conditional Logic

Conditional logic uses specific statements in Java to allow us to check a condition, and execute certain code based on whether that condition (the expression) is **true** or **false**.

Let's see how this works in practice.

Assignment Operator (=)

The assignment operator assigns the value of an expression, to the variable to the left of the operator.

```
boolean isAlien = false;
```

So, **`isAlien`** is the variable in this case, and it's been set to false, which is the value of our expression.

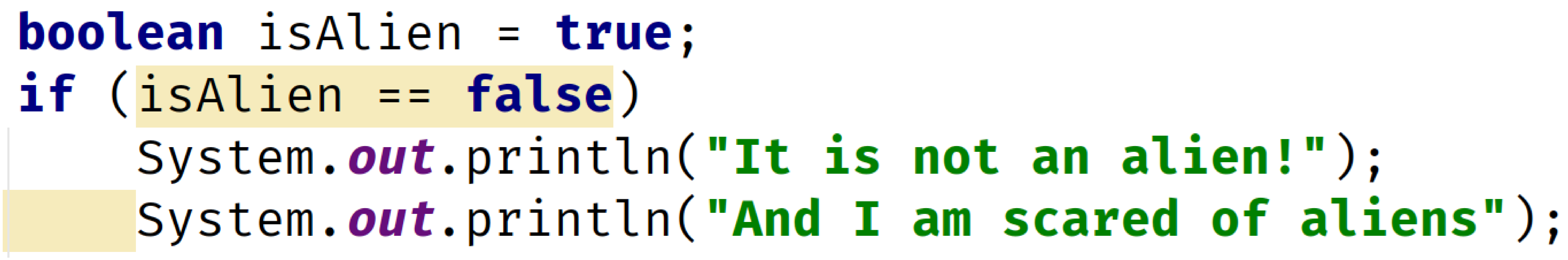
Equality Operator (==)

The equality operator tests to see if two operands are considered equal, and returns a boolean value.

```
if (isAlien == false) {
```

So, here **`isAlien`** is being tested against the value false.

Best Practice Rule - Always use a Code Block for If-Then statements



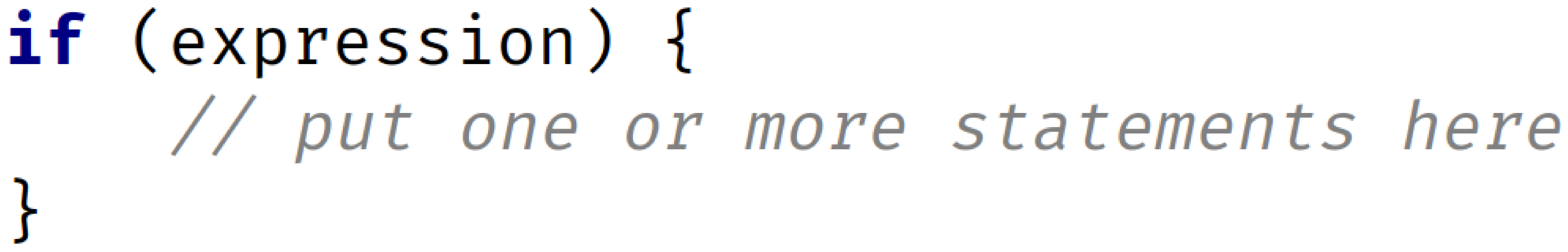
```java
boolean isAlien = true;
if (isAlien == false)
    System.out.println("It is not an alien!");
    System.out.println("And I am scared of aliens");
```

Instead of using the **if** statement as we can see here, we should instead use a code block.

The Code Block

A **code block** allows more than one statement to be executed, in other words, a block of code.

The format is:



```
if (expression) {
    // put one or more statements here
}
```