Conditional and logical operators

Learn to use boolean values in Java, a necessary tool for loops and conditionals.

WE'LL COVER THE FOLLOWING

- Conditional operators: >, <, >=, <=, ==, !=
- Logical operators: `&&`, `||`, and `!`
- Exercise: even in a range

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.

```
class BooleanValues {
  public static void main(String[] args) {
    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:

```
class AssignmentError {
  public static void main(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.

Exercise: even in a range

To practice writing boolean expressions that test if a variable satisfies several conditions, complete the function `evenInRange` below so that the function returns true if the parameter `x` is an even number in the range 24...32 (inclusive of the values 24 and 32). You should only need to write a single line of code.

EvenInRange.java | Sample solution

```
class EvenInRange {

  public static boolean evenInRange(int x) {
     // you write this part (a one-line return statement)
  }

  public static void main(String[] args) {
    System.out.println(evenInRange(0));
    System.out.println(evenInRange(24));
    System.out.println(evenInRange(25));
    System.out.println(evenInRange(34));
  }
}
```