LECTURE 4

BOOLEANS AND CONDITIONALS

MCS 260 Fall 2021

Emily Dumas

REMINDERS

- Project 1 description coming soon (probably end of this week)
- Project 1 due Fri Sep 17

TYPES WE'VE SEEN

We've seen a few types so far:

- int (integer), e.g. `260`
- float (floating point number), e.g. `260.0`
- str (string), e.g. `"260"`
  - We'll have more to say about strings soon!
- complex

BOOLEAN

There is a type in Python called a **boolean** or **bool** that has only two possible values:

- `True`
- `False`

These values are Python keywords.

(Boolean variables are also used in mathematics and CS, and most programming languages support them in some way.)

BOOLEAN EXPRESSIONS

Boolean values in Python programs are usually produced by operators such as:

`>`	is greater than
`<`	is less than
`==`	is equal to *note two equal signs!*
`!=`	is not equal to
`>=`	is greater than or equal to
`<=`	is less than or equal to

Examples of boolean expressions:

```
2 > 5           # False

11 == 9 + 2     # True

score >= 85.0   # True or False, depending on score
```

You can assign variables to boolean expressions:

```
x = 4>5   # Make bool variable x storing False

account_lockout = failed_logins >= 3
# now account_lockout is True or False depending on the value
# of failed_logins
```

CONTROL FLOW

So far, the programs we've written are executed line by line, starting from the top and moving down.

Most programs need to also make *decisions*, e.g. to say that something should happen only if certain criteria are met.

This is what **conditionals** do. They say that a certain section of code should only run if a boolean expression evaluates to True.

CONDITIONALS

```
if boolean expression:
    indented line 1
    indented line 2
    ...
non-indented line
```

The indented lines below the `if` form a **code block**.

If the boolean is `True`, the indented block runs.

If the boolean is `False`, the indented block is skipped.

(Recommended to use **four spaces** to indent a block.)

EXAMPLE PROGRAM

Let's write a program that will tell the user whether a quadratic polynomial

$$ax^2 + bx + c$$

is a perfect square or not.

ELSE

An `if` statement can be followed by `else:` and a code block to be executed if the condition is False.

```
if x == 100:
    print("x is equal to 100")
else:
    print("x is NOT equal to 100")
```

This is useful for handling dichotomies.

ELIF

An `if` statement can also be followed by **`elif`** (for "else if"), which begins a new conditional.

```
if height_cm >= 120:
    print("you can ride the roller coaster")
elif height_cm >= 100:
    print("you can only ride if accompanied by an adult")
else:
    print("you are not allowed to ride the roller coaster")
```

A chain of `if` / **`elif`** / **`elif`** / . . . is the typical way to compare a variable to multiple values or categories.

QUADRATIC ROOTS

Let's modify our perfect square program to tell us how many real roots a quadratic polynomial has.

BOOLEAN ALGEBRA

Expression	Condition to be True	In math
x `and` y	Both x and y are True	$x \wedge y$
x `or` y	At least one of x, y is True	$x \vee y$
`not` x	x is False	$\neg x$ $!x$ $\bar{x}$

QUESTION

What happens if you use a non-boolean value in a conditional? e.g.

```
if "walrus":
   print("Will this statement execute or not?")
```

Answer: The value will first be converted to boolean.

BOOLEAN COERCION

A few values convert to False (are "falsy"):

- Zero in any numeric type (0, 0.0, 0j)
- The empty string " "
- None (TBD)
- Empty containers like lists and tuples (TBD)

Anything else converts to True.

So if `x` is a variable of type int, then:

```
if x != 0:
    print("x is nonzero")
```

does exactly the same thing as

```
if x:
    print("x is nonzero")
```

REFERENCES

- In *Downey*:
  - Conditionals and booleans are discussed in sections 5.1 - 5.7.

REVISION HISTORY

- 2021-08-28 Initial publication