Lecture 3: Conditionals and Loops

Eleanor Birrell

September 11, 2019

Review: Conditional Statements

- if statement syntax


```
if x == 13 :
      print("that's my favorite number too!")
```

- if-else statement syntax

```
if x == 13:
    print("that's my favorite number too!")
else:
    print("mine is 13")
```

- condition must be an expression that evaluates to True or False (type bool)
  - Booleans: True, False
  - relational operators: ==, !=, >, <, >=, <=
  - logical operators: and, or, not
  - functions that evaluate to type bool

What about?

```
what's your favorite positive integer?
      13
that's my favorite number too!
```

```
what's your favorite positive integer?
      20
my favorite number is less than that.
```

```
what's your favorite positive integer?
      10
my favorite number is more than that.
```

Attempt #1: If statements

```
if x == 13:
    print("that's my favorite number too!")
if x > 13:
    print("my favorite number is less than that.")
if x < 13:
    print("my favorite number is more than that.")
```

Attempt #2: nested if-statements

```
if x == 13:
    print("that's my favorite number too!")
else:
    if x < 13:
         print("my favorite number is less than that.")
    else:
        print("my favorite number is more than that.")
```

Attempt #3: using elif

```
if x == 13:
    print("that's my favorite number too!")
elif x > 13:
    print("my favorite number is less than that.")
else:
    print("my favorite number is more than that.")
```

Exercise

Convert the following program to a program with the same behavior that doesn't use nested if-statements

```
x = int(input("pos int?\n\t"))
if x == 13:
    print("mine too!")
else:
    if x > 30 or x < 10:
        print("mine is 13")
    else:
        if x == 19:
            print("19!")
        else:
            print("?")
print("!")
```

What about…

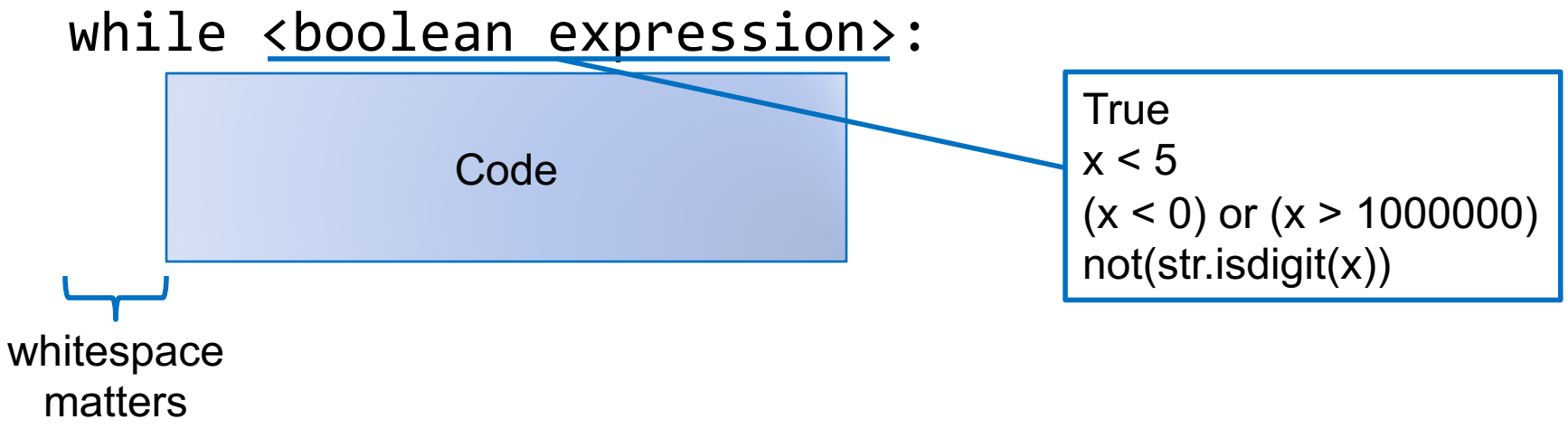
```
what's your favorite positive integer?
     absdfa
that's not good input!
```

```
x = input("what's your favorite " +
          "positive integer?")
if not str.isdigit(x):
     print("that's not good input")
```

- The condition can be any expression that evaluates to a Boolean value
  - Boolean values (e.g., True), expressions with relational operators (e.g., x < 5), expressions with logical operators (e.g., True or False), or functions that return a Boolean value

while loops

- When you want some set of statements to execute repeatedly . . . until some stopping criteria is met.



Example

Write a program that prompts user for a password, repeating until the correct password is entered, then prints "got it!"

Assume that the correct password is "123456"

Exercise

- Write a program that asks the user for a positive number and keeps asking until the user enters a positive int, then prints "Thanks!"

- Example run

```
Enter a positive integer:
   -10
That's not a positive integer! Try again:
   hello
That's not a positive integer! Try again:
   13
Thanks!
```

Example

Write a program that asks the user for a positive integer and then counts down from that value to `1` (all on one line!) and then prints "GO!" on the next line. For example, if the user enters 5, it should print:

```
5, 4, 3, 2, 1
GO!
```

Exercise (try this at home!)

Write a program that asks the user for a positive integer and then prints the value $1^2 + 2^2 + \cdots + n^2$

For example, if the user enters 5, it would print 55 (since 1+4+9+16+25 == 55)