CS101: Intro to Computing
Fall 2015

Lecture 5

Administrivia

- Homework 3 is due ***tonight***
- Homework 4 assigned (due on Monday)
  - It's ***much*** longer!
  - **<u>Don't wait till the last minute!</u>**
- No lab this week
  - Labs resume next week

REVIEW

```
s="%"+"i"
i=3/6
x=float(s%i)*2
```

What is the value of x?

a) 0.0

b) "%i%i"

c) 1.0

d) "1.0"

e) None of the above.

```
s="TACO TUESDAY"[2:6]
t=int(3.7)
x=s[-1]+s[t-2]
```

What is the value of x?

a) "O "

b) "UO"

c) "TC"

d) "TO"

```
s="TACO TUESDAY"[2:6]
   0123456789...
s=   CO TU
s="CO T"
t=int(3.7)
t=3
x=s[-1]+s[t-2]
x="T"+s[1]
x="TO"
```

```
i=len(“TACO TUESDAY”)
c=(1.0+2.0j)*(-i)
x=abs(min(c.real,-13))
```

What is the value of x?

a) 0

b) 11

c) 12

d) 13

FUNCTIONS

Functions

- A small program we can run ***within*** Python
  - Saves us from having to rewrite code
  - Don’t reinvent the wheel!
- **<u>ANALOGY</u>**: Functions are ***verbs*** in Python.
- Also called a ***subroutine*** or ***procedure***

Function calls

- When we want to execute a function, we ***call*** it or ***invoke*** it
- Use name of the function with parentheses
  - Example: `help()`
- Many functions are part of the Python language
  - We call them ***built-in functions***

User input

- raw_input() is a built-in function
- Argument: string printed to user
- Return value: string user typed before hitting “ENTER”

Goal

- Purpose of a program is to ***achieve a goal!***
- Let’s write a quadratic equation solver!

```
print "QUADRATIC SOLVER"
print "ax^2+bx+c=0"

a_str=raw_input("Please enter a:")
b_str=raw_input("Please enter b:")
c_str=raw_input("Please enter c:")

a=float(a_str)
b=float(b_str)
c=float(c_str)

square_root=(b**2-4*a*c)**.5
denominator=2*a

answer1=(-b+square_root)/denominator
answer2=(-b-square_root)/denominator

print "Solution 1: %f" % answer1
print "Solution 2: %f" % answer2
```

METHODS

Methods

- Like attributes, ***functions*** can be stored inside the type, too.
- Use ***attribute operator*** on the value.

```
“STOP SHOUTING!”.lower()
(1+1j).conjugate()
```

Value is treated like an argument.

String methods

```
“GATTACA”.count(“A”)
“MVEMJSUN”.find(“J”)
“ABACAB”.replace(“AB”,”G”)
“  HAM   “.strip()
“clint barton”.title()
“wEiRd”.swapcase()
```

```
s="TACO TUESDAY"
x=s[0:s.find(" ")]].lower()
x=x.title().swapcase()
```

What is the value of x?

a) "tACO"

b) "tuesdaY"

c) "Taco "

d) "TUESDAY"

WRITING FUNCTIONS

Defining functions

- We ***define*** a function by typing:

1. the keyword ***def***
2. the name of our function
3. a pair of parentheses
4. a ***block*** of code

header

body

```
def greetings():
  print “Hola!”
  print “Bonjour!”
  print “Ni hao!”
  print “Hello!”
  print “Shalom!”
  print “Guten tag!”
  print “Konnichiwa!”
  print “As-salamu alaykum!”
```

header

body

Block

- A section of code grouped together
- Begins with a colon :
- Contents of the block are ***indented***
  - “Tabbed in”

```
def hello():
  print “hello”
```

Scope

- Variables declared ***inside*** a block can be independent of variables ***outside*** the block.
- Variables inside a block ***might not exist*** outside the block.
- Blocks are their own little world!
- Blocks are ***isolated*** from the rest of our code.

Return

- Our function can ***return*** a value (output).
- We use the keyword ***return***.

```
def three():
  return 3
```

- Return ***immediately*** exits the function.

```
def hello():
  return 0
  print “hello”
```

Parameters

- Our function can take ***input*** (arguments) as well.
- Parameters are variables declared in function header.

```
def print_message(message):
  print message
```

- Multiple parameter are separated by commas.

```
def quadratic(a,b,c):
  s=(b**2-4*a*c)**.5
  d=2*a
  return (-b+s)/d
```