CompSci 101
Introduction to Computer Science

Sept 8, 2016

Prof. Rodger
(lecture by Barrett Ames)
Whose this guy?
Announcements

• Reading and RQ 4 due next time
• Asgn 2 out, APT 1 is due Tuesday
• Lab 2 this week
• To add class or change sections – see:
  – www.cs.duke.edu/courses/compsci101/fall16
• Today
  – more APT practice
  – functions, parameters
  – Names, types and values
Python Functions

• Answer these questions based on thinking, don't run any code

• Why do we need functions?
  – Manage complexity of large programs
  – Test and develop code independently
  – Reuse code in new contexts: create APIs!
Functions return values

• Most functions return values
  – Sometimes used to make things simpler, but returning values is a good idea

```python
def inch2centi(inches):
    return 2.54*inches

xh = inch2centi(72)
```

```python
def pluralize(word):
    return word + "es"

pf = pluralize("fish")
```
What is an APT? BMI APT

• Automated/Algorithmic Problem Testing
  – Write one function, 2-30 lines, solve a problem
  – Tested automagically in Eclipse or the browser
  – Lots of test cases – test test test

• Start simple, build toward more complex
  – What is a function? A function call?
  – What is a parameter? Argument?
  – How do you run/execute a program

comsci 101, fall 2016
Demo Solving APT BMI

• Write your code in Eclipse
  – Create python file
  – Name of file important – case matters
  – name of function important – cut and paste this
  – Write your code
  – Test a few examples in Eclipse

• Run online on using APT Tester
  – Tests on lots of examples, Debug, fix
  – Get all GREEN – Green dance!

• Submit on APT page
  – README form too
Function Detective

Results of Code Analysis

• For details on plurals: http://bit.ly/1N49u6b

• How did we call pluralize many times?
  – Loop. What is an alternative?

• What does the 'if' statement do?
  – Selects a code block to execute (more next week)

• If you have a question? Write and run code!
Organization matters

- https://www.youtube.com/watch?v=1ve57l3c19g
APT organization, Code organization

• You’ve written the BMI.py APT
  – Where is that module? How do you test it?
  – PyDev console, but then must import it
  – Adding print statements in BMI.py to test

• Putting sentences together in order…
  – “Once upon a time…” “It was the best of times…” “Aujourd’hui ma maman est morte”

• Putting code together in order
  – Takes judgment and experience
Python – Names and Types

• Names vs abstractions
  – What is http://152.3.140.1
  – What is http://www.amazon.com

• Types are important
  – What is foo.pdf, foo.mp4, foo.jpg, foo.wav
  – Do the file extensions guarantee file type?

• Python – what types are these?
  first = "Susan"
  x = 6
  y = 3.4
Strings

• Sequence of characters in quotes
  "I" + 'Love' + "'Python"

• String operators: concatenation (+), repeat(*)

• Precedence?
  "a" + "b" + "c" * 3

• Precedence?
  "a" + "b" "c" * 3
  'abcbbcabc'
Strings

- Sequence of characters in quotes (same result)
  
  "I" + 'Love' + "Python"
  
  'ILovePython'

- String operators: concatenation (+), repeat(*)

- Precedence?
  
  "a" + "b" + "c" * 3
  
  'abccc'

- Precedence?
  
  "a" + "b" "c" * 3
  
  'abcbbcbbc'
Function

• `def functionName(parameters):
  block of code`

• **Parameters** – place holder, will store value passed in

• **Arguments** – values in the call, that you pass to the function to use as input
Function – return or print?

• Example function that returns a value
  def sum(a, b):
    return a+b

• Example function that prints
  def hw(name):
    print "Hello " + name

• Call Functions
  print sum(4,7)
  answer = sum(4,7)
  hw("Sue")
  sum(4,7)
  print hw("a")
Function – return or print?

• Example function that returns a value
  
  ```python
def sum(a, b):
    return a + b
  ```

• Example function that prints
  
  ```python
def hw(name):
    print "Hello " + name
  ```

• Call Functions
  
  ```python
print sum(4,7)
answer = sum(4,7)
hw("Sue")
```
Old MacDonald had a farm, E-I-E-I-O
And on his farm he had a pig, E-I-E-I-O
With a Oink Oink here, and a Oink Oink there
Here a Oink, there a Oink everywhere a Oink Oink
Old MacDonald had a farm E-I-E-I-O

Old MacDonald had a farm, E-I-E-I-O
And on his farm he had a cow, E-I-E-I-O
With a Moo Moo here, and a Moo Moo there
Here a Moo, there a Moo everywhere a Moo Moo
Old MacDonald had a farm E-I-E-I-O

• Write a Program to print this song
def OldMacPig():
    print "Old MacDonald had a farm,",
    print "E-I-E-I-O"
    print "And on his farm he had a pig",
    print "E-I-E-I-O"
    print "With a Oink Oink here",
    print "and a Oink Oink there"
    print "Here a Oink, there a Oink",
    print "everywhere a Oink Oink"
    print "Old MacDonald had a farm",
    print "E-I-E-I-O"
Rest of Code

• Function OldMacCow
  – Replace “pig” with “cow”
  – Replace “Oink” with “Moo”

• Code to start:

```python
if __name__ == '__main__':
    OldMacPig()
    print
    OldMacCow()
```
Discuss how to make code better
bit.ly/101sp16-0126-1a

• Describe in words how you can make the code better? More efficient?
  – Fewer lines of code?
  – Use more functions?
  – Discuss your changes.

• What advantages do the changes you make have?
Demo – Old Mac improvements

- What does the horse say?
- What does the cow say?
- What does the fox say?
Assignment 2 out

• Totem poles
  – printing heads
  – functions
Names, Types and Values