Fundamentals of Programming

Python short course

Samuel Rivera

Department of Psychology
The Ohio State University

Fall, 2015

Outline

Programming

I'm the boss now, computer.

- *Programs*: a set of computer instructions
- *Programming*: defining the instructions
- *Algorithm*: set of instructions for some task

Why learn to program?

- Automation
  - Productivity!
  - Reduces human error
- Flexibility and Control
  - Custom experiments
  - Custom analysis

Requirements

1. Text editor: for writing code
   - Lets you type out instructions for the computer to follow
2. Programming language (Python)
   - Syntax and rules for writing instructions
3. interpreter/compiler (Python)
   - Converts our code to code the computer can understand

Text Editor

Not a word processor

Text editor: a program that lets you edit text

- notepad (notepad++)
- VI (VIM)

Some text editors are made for writing code

- syntax highlighting
- auto-completion

Example text file

Created with a text editor

```
This is an example text file.  The writing
inside is plain
text. Whatever text is saved in the text
file is the only thing saved
in the text file, including return keys,
tabs, and spaces.
No extra characters are added.
```

Improper text file

Created with MS word

```
\  00\00\00\00\00\00\00\00\00\00\00\00\00\00\00\00\00\
\00\00\00\00\00\\00\00\00\00\00\00\00\00\00\ 00\00\00
\00\00\00\00\00\0000\00\00\00\00\0  ...
```

Word processors add all types of things to files. They are not meant for writing computer code.

Programming language

Python (From Monty Python)

Programming language: Formal language specifying rules for communicating instructions to the computer

- store a value
- do something multiple times
- save something in a text file
- show something on the display

Interpreter/Compiler

Python

Interpreter: A program that takes the instructions you wrote in the python language, and converts it to computer code

- Python interpreter runs code line by line (like MATLAB)
- Compiled languages (like C++) must be processed (compiled) in their entirety before being run by computer

Get Python

It's Free

Anaconda: A free Python distribution for scientific computing

- https://www.continuum.io/downloads

Hello World

helloWorld.py

```
print('hello world')
```

Hello World 2.0

helloWorld2.py

```
# S Rivera
# Oct 2015
# Simple hello world program for lecture 1.
# <- means comment to ignore
print('hello world')
```

Simple Example

Store, manipulate, retrieve

```
x = 5   # Create a variable called x
        # That variable has memory storage
        # Store the value 5 there
x = x + 2   # add 2 to value in x, store it in x
print( x ) # show the value of x on screen
```

What you will learn

All about the algorithms

Any program is built from only 3 control structures

1. Do something
2. Do something multiple times
3. Do something if something else is True

Be clever and combine them to perform any task

Do Something

Computer, store some value.

```
x = 'something'  # a string
xx = 4.0 # a float value
y = [1, 2, 3] # a list of integers
z = True # (or False) a boolean value
zz = None # a None data type
```

Do Something

Computer, do math.

```
x = 5*1 # multipy
y = 3/2 # divide and truncate
z = 3/2.0 # divide  properly
zz = 2**10 # exponential
xyz = 5 % 2 # modulus
```

```
from math import sqrt  # load sqrt code
x = sqrt(2)  # store result of squareroot 2 in x
```

Do Something

Computer, compare values.

```
x = (5 > 2 )
y = (5 <= 2 )
z = ( 5 == 2)
zz = ( 5!= 2)
```

Do Something

Computer, use predefined *functions*.

```
x = range( 5 )  # Call the 'range' function.
                # Pass it the value 5, which
                # returns the list [0,1,2,3,4].
                # Store it in x.

help( range ) # Explain the range function.
              # Call from command prompt

print( 'print is also a function')

len( x ) # Return number of items in x.
         # Don't save result
```

Loop

Do something 5 times.

```
for x in [1,2,3,4,5]:
    print( x ) # show x every time in loop
```

Loop

Do something 5 times.

```
for x in range(5): # range function trick
    print( x )
```

Conditional

Do something if something else.

```
x = True
if x :
    print( 'x is true' )
else:
    print( 'x is not true' )
```

Exercise 1

Show the values from 1 to 10 on the screen

Pseudocode

```
Loop through a list of numbers from 1 to 10
    show the number
```

Exercise 1

Show the integers from 1 to 10 on the screen.

```
for x in range(10):
    print( x+1)
```

Exercise 2

Show the *EVEN* values from 1 to 10 on the screen

Pseudocode

```
Loop through a list of numbers from 1 to 10
    if it is even, show the number
```

Exercise 2

Show the *EVEN* values from 1 to 10 on the screen.

```
for x in range(1,11):
    if (x % 2) == 0:   # modulus trick
        print( x)
```

Additional resources

You are not alone.

- Google
- Free book: http://www.greenteapress.com/thinkpython/
- Online course: https://www.codecademy.com/
- Other students