

Lect 1 – Introduction

Information Processing Applications

Rob Capra

Course Information

- Instructor: Dr. Robert Capra
- Office: Manning 210
- Email: r<lastname> at unc dot edu
- Prerequisites: COMP110 or equivalent
- Review syllabus & course policies

What is an algorithm?

What is an algorithm?

Are some algorithms better than others?

.... How do you define “better”?

What is a data structure?

Can you name some data structures?

What characteristics do they have?

How are data structures and algorithms related?

Formal and Natural Languages

- Natural languages
 - Examples: English, Spanish, Chinese
 - Evolved naturally
- Formal languages
 - Examples: mathematical notation, chemical formulas
 - Designed by people for a purpose
- Programming languages are formal languages that have been designed to express computations.
- Parsing
- Natural vs. Formal
 - Ambiguity, redundancy, literalness
 - Poetry vs. Legal contract vs. computer program

Python

- History
 - Guido van Rossum (Benevolent Dictator for Life)
 - Started implementation in 1989
- Philosophy (from PEP20 Zen of Python)
 - Beautiful is better than ugly.
 - Explicit is better than implicit
 - Simple is better than complex
 - Complex is better than complicated
 - Readability counts.
 - “To describe something as clever is not considered a compliment in the Python culture” – Alex Martelli
 - Python's philosophy rejects the Perl "there is more than one way to do it" approach to language design in favor of "there should be one—and preferably only one—obvious way to do it“
 - An important goal is making Python fun to use. This is reflected in the origin of the name which comes from Monty Python. (spam & eggs)



Python

- **Features** (from <http://www.python.org/about/>)
 - Very clear, readable syntax
 - Strong introspection capabilities
 - Intuitive OO
 - Natural expression of procedural code
 - Modular
 - Exception-based error handling
 - Very high level dynamic data types
 - Extensive standard library – “batteries included”
- **Main Python site**
 - <http://www.python.org/>

Python

- High-level languages
 - Python, C++, Java, PHP
 - Easier to code in, shorter, portable
 - Must be processed into low-level to run on hardware
- Low-level languages
 - Machine/assembly language
 - Run natively on a particular type of hardware, very fast

- To run high-level languages:

- Interpreters
- Compilers



Python

- Shell mode
 - Type expressions at a shell prompt
 - Interpreter immediately shows results

```
$ python3
Python 3.2 (r32:88445, Mar 25 2011, 19:28:28)
[GCC 4.5.2] on linux2
Type "help", "copyright", "credits" or "license"
>>> 2 + 3
5
>>>
```

“chevron” prompt

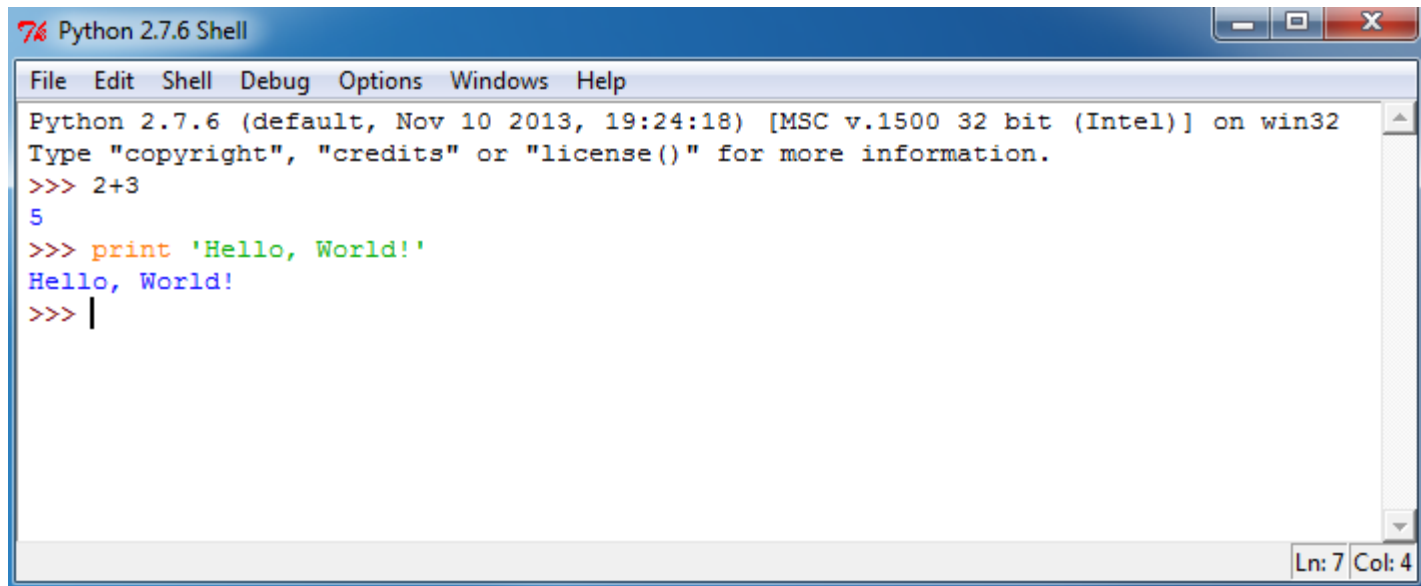
- Program mode
 - Write code in a file (call the contents a Python *script*)
 - Use the interpreter to execute the file as a whole

```
print("My first program adds two numbers, 2 and 3:")
print(2 + 3)
```

```
$ python firstprogram.py
My first program adds two numbers, 2 and 3:
5
```

Python 2.7 and IDLE

- Python 2.7
 - <http://www.python.org/download/>



The screenshot shows a window titled "Python 2.7.6 Shell" with a menu bar (File, Edit, Shell, Debug, Options, Windows, Help). The main text area displays the following content:

```
Python 2.7.6 (default, Nov 10 2013, 19:24:18) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> 2+3
5
>>> print 'Hello, World!'
Hello, World!
>>> |
```

The status bar at the bottom right indicates "Ln: 7 Col: 4".

- A note about Python 2.x versus 3.x

Interactive Python Text

- Run & change code in web page (ActiveCode)
- Step through code (CodeLens)

NOTE about Class Quizzes and the INTPY “Check your understanding” questions

INTPY: *How to Think Like a Computer Scientist*, interactive edition 2.0:
<http://interactivepython.org/runestone/static/thinkcspy/toc.html>

TPY: *Think Python: How to Think Like a Computer Scientist*. Downey, A.
<http://www.greenteapress.com/thinkpython/>

PSADS: *Problem Solving with Algorithms and Data Structures*
<http://interactivepython.org/runestone/static/pythonds/index.html>

PDA: *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython*.
McKinney, W. ISBN: 978-1449319793

Debugging

- Syntax errors
- Runtime errors (exceptions)
- Semantic errors
- Experimental debugging
- Blurry boundaries between debugging and programming

“When you have eliminated the impossible, whatever remains, however improbable, must be the truth” – A. Conan Doyle