Lecture 2: Basics: Introduction to Python

BT 3051 – Data Structures and Algorithms for Biology

Karthik Raman
Department of Biotechnology
Indian Institute of Technology Madras

August 4, 2014

Programming Languages

- Computers are great at following *explicit* instructions
- Programming languages are our interface with the CPU
- Every programming language helps us in various ways
- C/C++/FORTRAN: Raw computational power (low-level languages)
- Java: Inter-operability and improved readability
- Python/MATLAB: High-level languages

Why Python?

- We need a method to tell the CPU what to do
- A computer cannot understand English/Hindi/Tamil (at least if you want to program it)
- A computer can also not understand Python (as is)!
- Any code written in any language needs to be translated to machine code for the computer to run it (compilation)
- "A computer can no more understand a Python program than it can understand a program written in plain English!"
- Python, however, is more human-friendly – powerful constructs render code more readable
- Python is an interpreted language (like MATLAB, but unlike C/C++/Java)

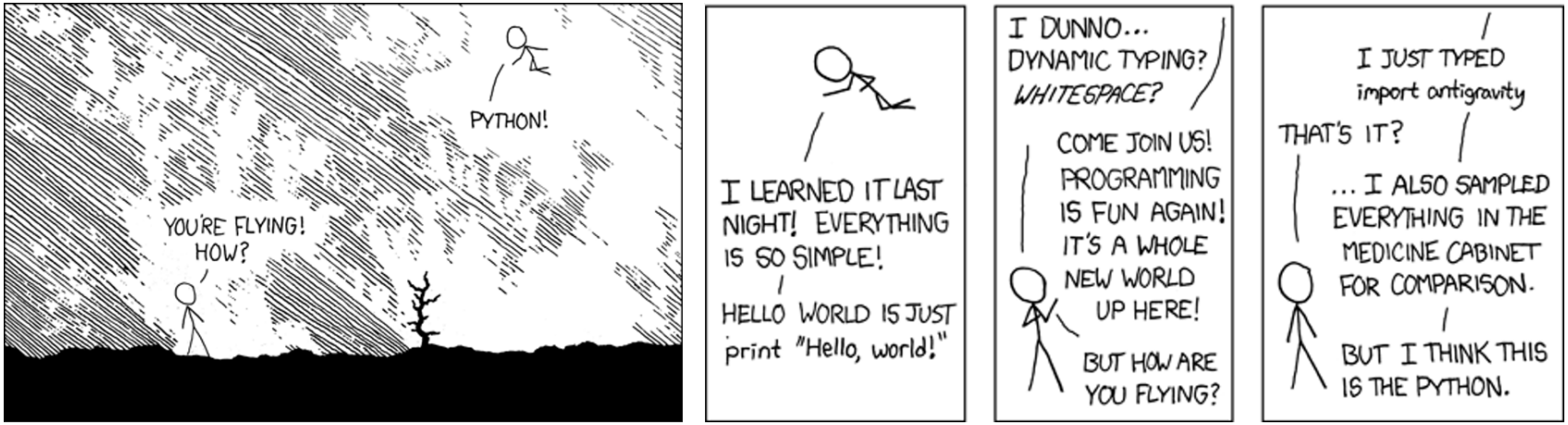
Why Python?

http://en.wikipedia.org/wiki/Python_(programming_language)

- Python is a widely used general-purpose, high-level programming language
- Its design philosophy emphasises **code readability**, and its syntax allows programmers to express concepts in fewer lines of code than would be possible in languages such as C
- The language provides constructs intended to enable clear programs on both a small and large scale

- Python is heavily used in the CS/IT/scientific computing
- Python has a very large number of useful packages for science, math, biology, <you name it>
- Can also interface with C/C++/Java libraries etc. for raw speed!

Why Python?



"I wrote 20 short programs in Python yesterday. It was wonderful. Perl, I'm leaving you."

Courtesy: http://xkcd.com/353/

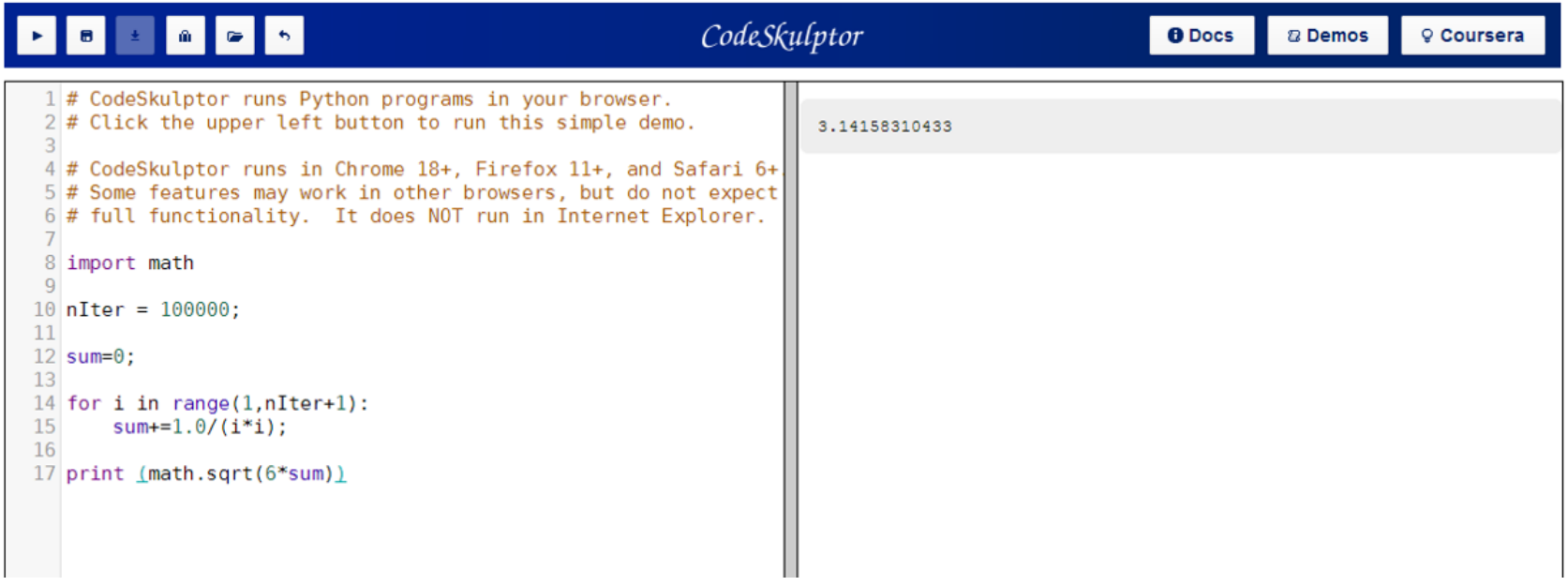
See also: The Homogenization Of Scientific Computing Or Why Python Is Steadily Eating Other Languages' Lunch

iPython

- http://ipython.org/
- Fantastic and powerful interaction shell for Python

codeskulptor.org

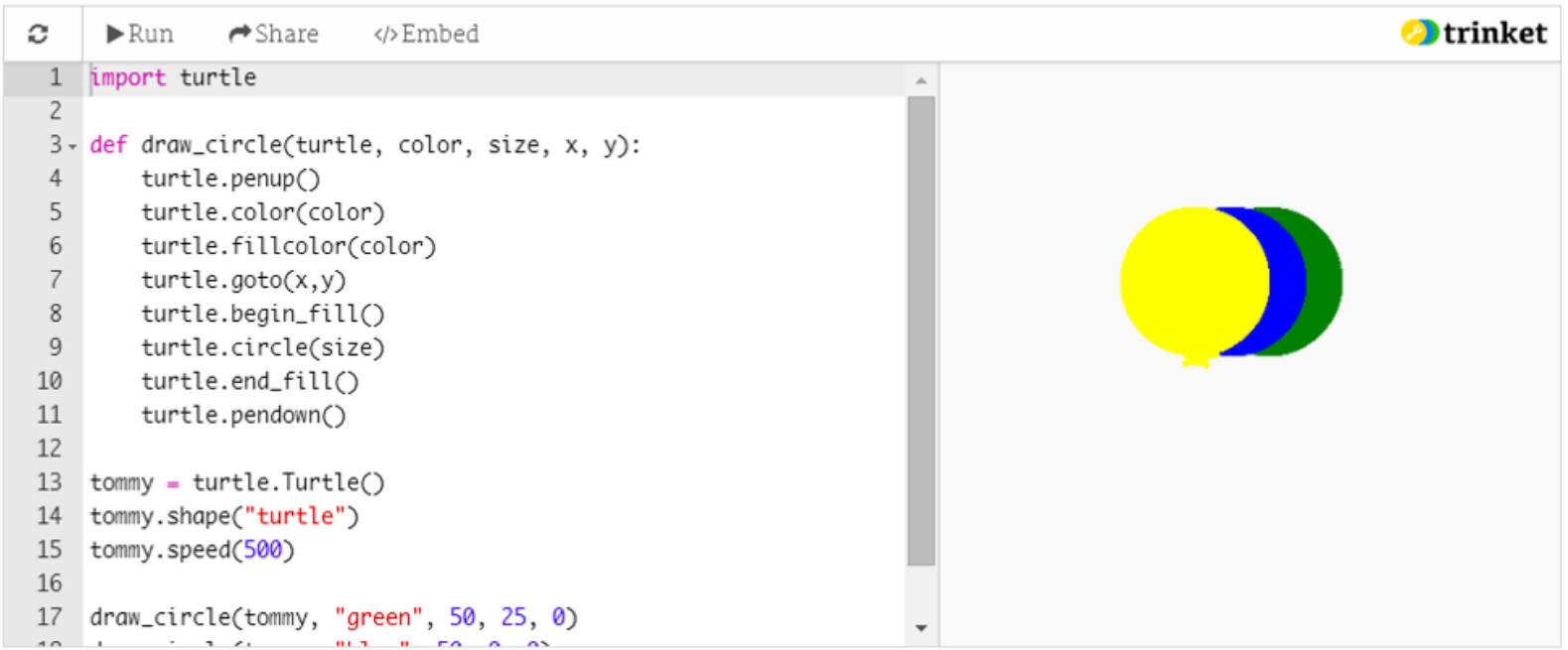
- Thanks, Scott Rixner (http://www.cs.rice.edu/~rixner/)!



trinket.io

Teach Python Straight from Your Class Website

Skip the installs and get back to teaching with **trinket's** interactive Python:



Coding Style Guidelines

Courtesy: Scott Rixner, Coursera Principles of Computing

- Use *docstrings*: check out `__doc__` in Python
- Comment appropriately
- Do not use global variables
- Indent your code, or obviously, Python will throw a fit!
- Use [4] spaces; NEVER USE TABS!
- Name variables/functions/classes descriptively
- Class fields should be private; can only be accessed by a method, if at all
- Design your functions to do a single thing but do it well
- Use pylint to clean your code to conform to accepted style guidelines!

Questions?

Also remember