Python: An Introduction

OFE 2021 Summer Workshop

Haiyong Liu
Department of Economics

Agenda

- Introduction
- Running Python
- Python Programming
  - Variables
  - Types
  - Arithmetic operators
  - Boolean logic
  - Strings
  - Printing
- Exercises

What is python?

- Object oriented language
- Interpreted language
- Supports dynamic data type
- Independent from platforms
- Focused on development time
- Simple and easy grammar
- High-level internal object data types
- Automatic memory management
- It's free (open source)!

Brief History of Python

- Invented in the Netherlands, early 90s by Guido van Rossum
- Named after Monty Python
- Open sourced from the beginning
- Considered a scripting language, but is much more
- Scalable, object oriented and functional from the beginning
- Used by Google from the beginning
- Increasingly popular

Python's Benevolent Dictator For Life

"Python is an experiment in how much freedom programmers need. Too much freedom and nobody can read another's code; too little and expressive-ness is endangered."

- Guido van Rossum

Language properties

- Everything is an object
- Modules, classes, functions
- Exception handling
- Dynamic typing, polymorphism
- Static scoping
- Operator overloading
- Indentation for block structure

High-level data types

- Numbers: int, long, float, complex
- Strings: immutable
- Lists and dictionaries: containers
- Other types for e.g. binary data, regular expressions, introspection
- Extension modules can define new “built-in” data types

Why learn python?

- Fun-to-use "Scripting language"
- Object-oriented
  - Highly educational
- Very easy to learn
- Powerful, scalable, easy to maintain
  - high productivity
  - Lots of libraries
- Glue language
  - Interactive front-end for FORTRAN/C/C++ code

Where to use python?

- System management (i.e., scripting)
- Graphic User Interface (GUI)
- Internet programming
- Database (DB) programming
- Text data processing
- Distributed processing
- Numerical operations
- Graphics
- And so on…

Why learn python? (cont.)

- Reduce development time
- Reduce code length
- Easy to learn and use as developers
- Easy to understand codes
- Easy to do team projects
- Easy to extend to other languages