

# Introduction to Python

## Why Python?

D.S. Hwang

Department of Software Science  
Dankook University

# Outline

Why Python?

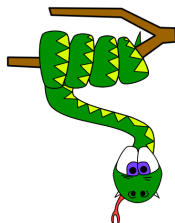
Program and Programming

Preparation

Summary























# What is a script? I

- ▶ Very level programming language
- ▶ Scripting language: Unix shells, Tcl, Perl, Python, Ruby, Scheme, REXX, JavaScript, Visual Basic, Matlab, R, ...
- ▶ This class deals with how to use Python and its **boosting tools**



# What is a script? II

src: <https://spectrum.ieee.org/computing/software/the-2017-top-programming-languages>

Language Rank	Types	Spectrum Ranking
1. Python	 	100.0
2. C	  	99.7
3. Java	  	99.5
4. C++	  	97.1
5. C#	  	87.7
6. R		87.7
7. JavaScript	 	85.6
8. PHP		81.2
9. Go	 	75.1
10. Swift	 	73.7

# Characteristics of a script

- ▶ Easily combined to other programming languages
- ▶ Provide extensive text processing
- ▶ Powerful in managing file and directory processing
- ▶ Many small scripts may yield a big system.
- ▶ Perhaps a special-purpose GUI on top
- ▶ Portable across Unix, Windows, Mac etc.
- ▶ Interpreted program without compilation and linkage

# Why we do not prefer C/C++ or Java ?

A job reading real numbers from *number.dat*:

```
.3 9.9 5.2  
1.6e-6 0.0 0.9 0.7  
9 8 7.7
```

Python code:

```
f = open('number.dat', 'r');  
val = f.read().split();
```

- Check this out if you are using C/C++!

# Why we are using C/C++?

- ▶ Speed up parts of our code
- ▶ A lot of legacy code
- ▶ Easy to integrate with an existing code
- ▶ Large community

# Program and Programming

The purposes of this teaching:

- ▶ How to make computers do what you want them to do

A program:

- ▶ consists of many instructions.
- ▶ is written in terms of a few basic operations.
- ▶ Each instruction should be understandable by computers.

Computers are similar but have a different set of operations.



# Program and Programming

## Computational thinking

- ▶ Conceptualizing, not programming
- ▶ A way that humans, not computers, think
- ▶ For everyone, everywhere

# Script versus Program

## Script

*"Scripts" are distinct from the core code of the application, which is usually written in a different language, and are often created or at least modified by the end-user. Scripts are often interpreted from source code or bytecode, whereas the applications they control are traditionally compiled to native machine code.*

## Program

*The program has an executable form that the computer can use directly to execute the instructions. The same program in its human-readable source code form, from which executable programs are derived (e.g., compiled)*

from Wikipedia

# Computational thinking

- ▶ Conceptualizing, not programming
  - ▶ Computer science is not computer programming. Thinking like a computer scientist means more than being able to program a computer. It requires thinking at multiple levels of abstraction.
- ▶ A way that humans, not computers, think
- ▶ For everyone, everywhere

# Computational thinking

- ▶ **Conceptualizing, not programming**
- ▶ A way that humans, not computers, think
  - ▶ **Computational thinking** is a way humans solve problems; it is not trying to get humans to think like computers. Computers are dull and boring; humans are clever and imaginative. We humans make computers exciting. Equipped with computing devices, we use our cleverness to tackle problems we would not dare take on before the age of computing and build systems with functionality limited only by our imaginations.
- ▶ For everyone, everywhere

# Computational thinking

- ▶ Conceptualizing, not programming
- ▶ A way that humans, not computers, think
- ▶ For everyone, everywhere
  - ▶ Computational thinking will be a reality when it is so integral to human endeavors it disappears as an explicit philosophy.

# What to install

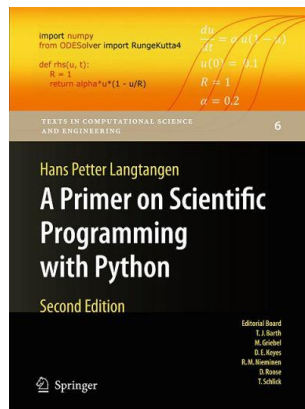
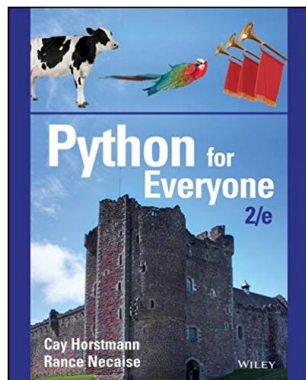
## Python URL

- ▶ <http://www.python.org>

## IDE

- ▶ PyCharm, <https://www.jetbrains.com/pycharm/>
- ▶ Spyder, <https://pythonhosted.org/spyder/>
- ▶ PyPE, <http://pype.sourceforge.net/index.shtml>
- ▶ Wing 101, <http://wingware.com/downloads/wingide-101>
- ▶ IDLE, [www.python.org/gedit](http://www.python.org/gedit)
- ▶ NINJA-IDE, [ninja-ide.org](http://ninja-ide.org)
- ▶ Eric, <http://eric-ide.python-projects.org/>
- ▶ and more ....

# References



# Summary

You should understand

- ▶ how to develop and use programs that solve real world problems
- ▶ the core features of a programming language with Python
- ▶ how to think methodically about programming
- ▶ some tools that make your programming more productive

Check out the web site!

- ▶ <https://money.usnews.com/careers/best-jobs/rankings/the-100-best-jobs>