



python :  
powered



python

# Administration

- Instructor:
  - 曾學文 資工系副教授
  - Office: Room 908
  - Email: hwtseng@nchu.edu.tw
  - Tel: 04-22840497 ext. 908
  - <http://wccclab.cs.nchu.edu.tw/www/index.php/course>
- Office Hours:
  - (Wednesday) 14:00~16:00
- Grade:
  - Homework 40%
  - Computer-based Test 30%
  - Final Project 30 %

# Outline

1. Python簡介與使用操作
2. 敘述句(statements)與資料結構(data structures)解說、練習
3. 程式模組(modules)解說、練習
4. 輸入與輸出(input and output)應用練習
5. 錯誤與例外處理(errors and exception)觀念簡介、練習
6. 物件與類別(objects and classes)觀念介紹、練習
7. 標準函數庫(standard libraries)應用練習
8. Python 應用於Mechanical Learning
9. 期末成果展示



# Introductory

- Raise your hand is always welcome!
- No phone, walk, sleep, and late during the lecture time.
- Slides are not enough. To master the materials, page-by-page reading is necessary.
- Do not copy the homework.

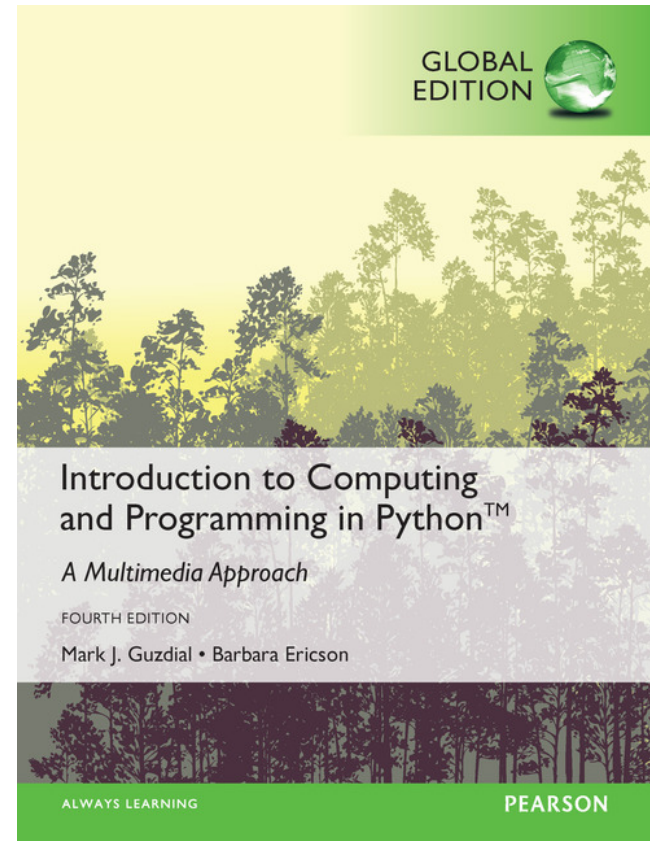
# Reference Book

- Fundamentals of Python: First Programs, “Kenneth A. Lambert”, International Edition, ISBN: 1111822700



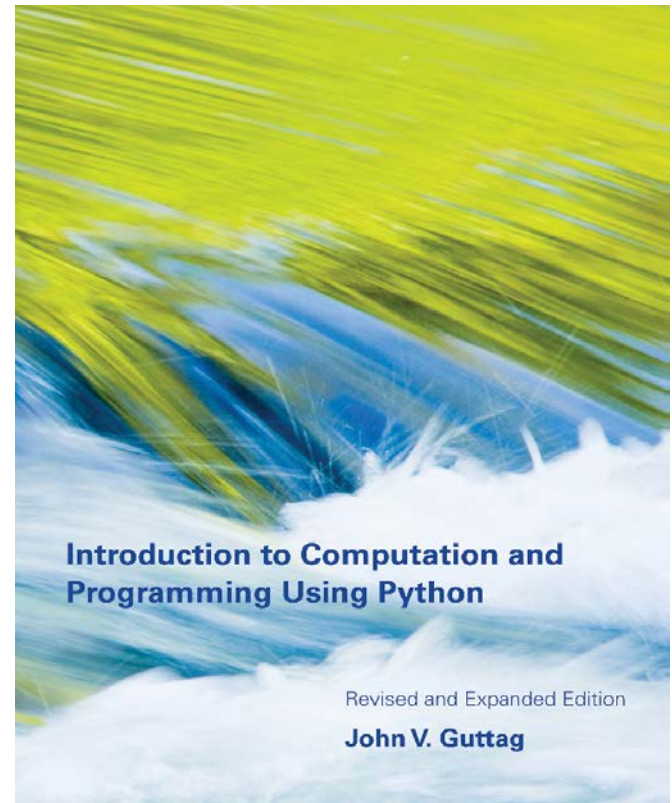
# Reference Book

- Introduction to Computing and Programming in Python, “Mark J. Guzdial, Barbara Ericson”, Global Edition (4e), ISBN: 9781292109862



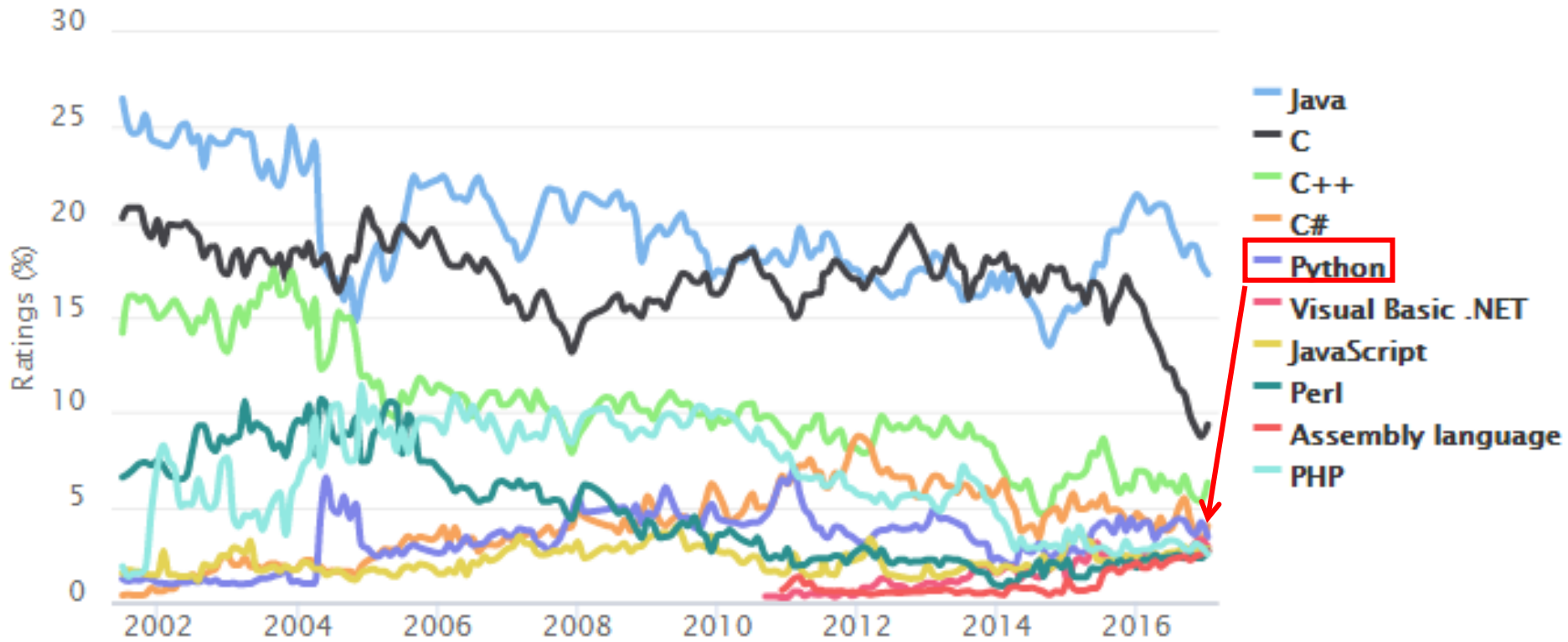
# Reference Book

- Introduction to Computation and Programming Using Python, " John V. Guttag ", Revised And Expanded Edition, ISBN: 9780262316644



# TIOBE Programming Community Index

Source: [www.tiobe.com](http://www.tiobe.com)





Jan 2017	Jan 2016	Change	Programming Language	Ratings	Change
1	1		Java	17.278%	-4.19%
2	2		C	9.349%	-6.69%
3	3		C++	6.301%	-0.61%
4	4		C#	4.039%	-0.67%
5	5		Python	3.465%	-0.39%
6	7	▲	Visual Basic .NET	2.960%	+0.38%
7	8	▲	JavaScript	2.850%	+0.29%
8	11	▲	Perl	2.750%	+0.91%
9	9		Assembly language	2.701%	+0.61%
10	6	▼	PHP	2.564%	-0.14%
11	12	▲	Delphi/Object Pascal	2.561%	+0.78%
12	10	▼	Ruby	2.546%	+0.50%
13	54	▲	Go	2.325%	+2.16%
14	14		Swift	1.932%	+0.57%
15	13	▼	Visual Basic	1.912%	+0.23%
16	19	▲	R	1.787%	+0.73%
17	26	▲	Dart	1.720%	+0.95%
18	18		Objective-C	1.617%	+0.54%

# History

Programming Language	2017	2012	2007	2002	1997	1992	1987
Java	1	1	1	2	17	-	-
C	2	2	2	1	1	1	1
C++	3	3	3	3	2	2	4
C#	4	4	6	10	-	-	-
Python	5	7	7	16	27	-	-
PHP	6	5	4	6	-	-	-
JavaScript	7	9	8	7	20	-	-
Visual Basic .NET	8	24	-	-	-	-	-
Perl	9	8	5	4	3	-	-
Assembly language	10	-	-	-	-	-	-
Lisp	28	12	12	9	7	11	2
Ada	30	15	16	15	10	3	13
Prolog	33	43	27	27	14	9	3



# Python

- Simple
  - Python is a simple and minimalistic language in nature
  - Reading a **good** python program should be like reading English
  - Its Pseudo-code nature allows one to concentrate on the problem rather than the language
- Easy to Learn
- Free & Open source
  - Freely distributed and Open source
  - Maintained by the Python community  
<http://www.python.org/community/>
- High Level Language – memory management
- Portable – runs on anywhere and combine with c code



# Python

- Interpreted
  - You run the program straight from the source code.
  - Python program → Bytecode → a platform's native language
  - You can just copy over your code to another system and it will automatically work with python platform
- Object-Oriented
  - Simple and additionally supports procedural programming
- Extensible – easily import other code
- Embeddable – easily place your code in non-python programs
- Extensive libraries
  - (i.e. reg. expressions, doc generation, CGI, ftp, web browsers, ZIP, WAV, cryptography, etc...) (wxPython, Twisted, Python Imaging library)



# Python Timeline/History



- Python was conceived in the late 1980s.
  - Guido van Rossum (吉多·范羅蘇姆),
  - Benevolent Dictator For Life (仁慈獨裁者)
  - Rossum is Dutch, born in Netherlands
  - Descendant of ABC, he wrote glob() func in UNIX
  - M.D. @ U of Amsterdam, worked for CWI, NIST, CNRI, **Google**
  - Also, helped develop the ABC programming language
  - Monty Python's Flying Circus (蒙提·派森的飛行馬戲團)
- In 1991 python 0.9.0 was published and reached the masses through alt.sources
  - The alt.sources newsgroup is intended to be a repository for source-code of all sorts that people wish to distribute and share with other people.



# Python Timeline/History

- In January of 1994 python 1.0 was released
  - Functional programming tools like lambda, map, filter, and reduce
  - comp.lang.python formed, greatly increasing python's user base
- In 1995, python 1.2 was released.
- By version 1.4 python had several new features
  - Keyword arguments (similar to those of common lisp)
  - Built-in support for complex numbers
  - Basic form of data-hiding through name mangling (easily bypassed)
    - private, protected, public
- Computer Programming for Everybody initiative
  - Make programming accessible to more people, with basic “literacy” similar to those required for English and math skills for some jobs.
  - Project was funded by DARPA (Defense Advanced Research Projects Agency)



# Python Timeline/History

- In 2000, Python 2.0 was released.
  - Introduced list comprehensions similar to Haskell
    - Haskell is a modern functional language (like lisp)
  - Introduced garbage collection
- In 2001, Python 2.2 was released.
  - Included unification of types and classes into one hierarchy, making python's object model purely object-oriented
  - Generators were added (function-like iterator behavior)
    - **iterator** is an object that enables a programmer to traverse a container.
- Standards

# Version Release Dates

- Python 1.0 - January 1994
  - Python 1.5 - December 31, 1997
  - Python 1.6 - September 5, 2000
- Python 2.0 - October 16, 2000
  - Python 2.1 - April 17, 2001
  - Python 2.2 - December 21, 2001
  - Python 2.3 - July 29, 2003
  - Python 2.4 - November 30, 2004
  - Python 2.5 - September 19, 2006
  - Python 2.6 - October 1, 2008
  - Python 2.7 - July 3, 2010
- Python 3.0 - December 3, 2008
  - Python 3.1 - June 27, 2009
  - Python 3.2 - February 20, 2011
  - Python 3.3 - September 29, 2012
  - Python 3.4 - March 16, 2014
  - Python 3.5 - September 13, 2015
  - Python 3.6 - December 23, 2016

Python Taiwan

<https://www.facebook.com/groups/pythontw/10152295869513438/>



# Running Python

- There are **three** different ways to start Python:

## (1) Interactive Interpreter:

- You can enter **python** and start coding right away in the interactive interpreter by starting it from the command line.

```
$python          # Unix/Linux  
  
or  
  
python$        # Unix/Linux  
  
or  
  
C:>python      # Windows/DOS
```

# Interactive Interpreter

- Here is the list of all the available command line options:

Option	Description
-d	provide debug output
-O	generate optimized bytecode (resulting in .pyo files)
-S	do not run import site to look for Python paths on startup
-v	verbose output (detailed trace on import statements)
-X	disable class-based built-in exceptions (just use strings); obsolete starting with version 1.6
-c cmd	run Python script sent in as cmd string
file	run Python script from given file

# Script from the Command-line

- A Python script can be executed at **command line** by invoking the interpreter on your application, as in the following:

```
$python script.py          # Unix/Linux
```

```
or
```

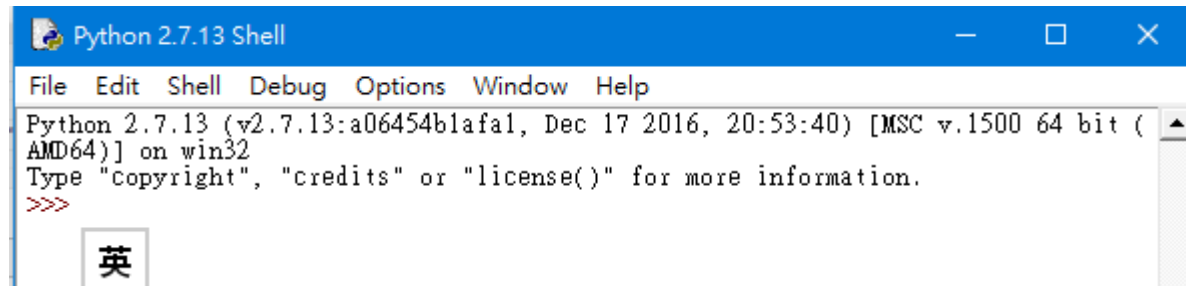
```
python% script.py         # Unix/Linux
```

```
or
```

```
C:>python script.py       # Windows/DOS
```

# Integrated Development Environment (IDE)

- You can run Python from a graphical user interface (GUI) environment as well.
  - All you need is a GUI application on your system that supports Python.
- **Unix:** IDLE is the very first Unix IDE for Python.
- **Windows:** PythonWin is the first Windows interface for Python and is an IDE with a GUI.
- **Macintosh:** The Macintosh version of Python along with the IDLE IDE is available from the main website, downloadable as either MacBinary or BinHex'd files.



```
Python 2.7.13 Shell
File Edit Shell Debug Options Window Help
Python 2.7.13 (v2.7.13:a06454b1afa1, Dec 17 2016, 20:53:40) [MSC v.1500 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
```

英