

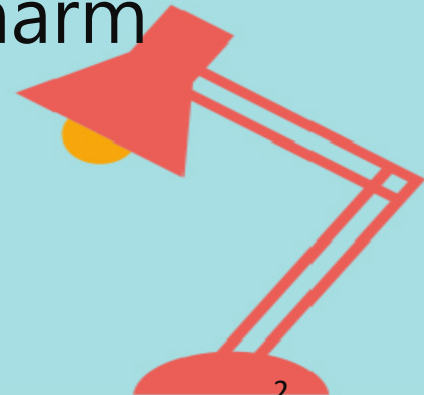


110-1基礎程式設計(1)

亞大資工系

課程大綱

- Essential-基本的
 - 啟思博的概念(Kissipo Learning)
 - Anaconda 和Jupyter notebook
 - Hello World程式
- Advanced-進階的
 - 關於Python 程式語言
 - Python編輯器:IDLE, Spyder, Visual Studio Code, PyCharm
 - Python程式的執行 (Run)
 - Python程式的偵錯 (Debug)



(A) ESSENTIAL-基本的



(B) ADVANCED-進階的





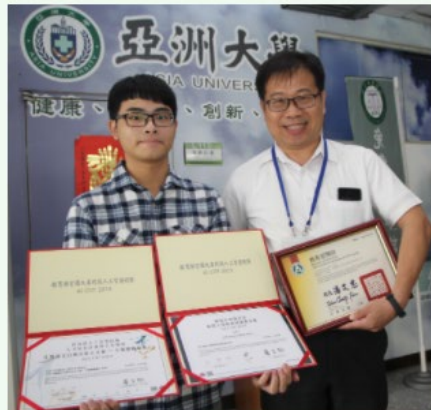
<上圖：亞大活動剪影——點按可閱相關報導、置上可觀靜止圖像>

ASIA Hot News

2020-08-24

資工系林映辰同學獲AI競賽6個獎項---

林映辰同學參加工研院Aldea平台AI競賽，共獲台灣牧場乳量預測第三名等6個獎項。



圖說：資工系朱學亭老師（右）與資工系林映辰同學（左）到人工智慧學校台中分校分享AI程式經驗。

亞洲大學（Asia University, Taiwan）資工系林映辰同學去年初開始研修「AI人工智慧」學程，學習AI技巧，開始參加工研院Aldea人工智慧共創平台競賽，一年中，先後獲Aldea平台AI競賽得6項比賽獎項。

這6項獎項，包括兩個教育部委辦的全國大專校院AI競賽：生醫論文自動分析正式比賽前標、新聞立場檢索技術競賽佳作獎，以及工研院所辦臺灣COVID-19確診人數預測比賽第二名、展場民眾特徵分析比賽第三名、台灣牧場乳量預測比賽第三名以及臺灣海洋廢棄物預測比賽第五名等4項產業議題獎項。

牧場乳量預測是目前精準農業的熱門議題之一，林映辰同學透過AI人工



ASIA Hot News [2019-12-13]

資工系學生獲AI技術研發競賽第二名---

資工系學生參加科技部的「科技大擂台與AI對話」Fun Cup比賽，脫穎而出，榮獲第二名。



圖說：科技部楊琇雅司長(右)頒獎給亞大資工系李衡(左)和李柏翰同學。

考驗。

「透過教育部補助的AI系列課程計畫，比賽同學都已上了兩個學期的AI課程。」朱學亭老師說，亞大推動AI人工智慧系列課程，同學都必須完成工研院Aldea平台上的專題。所以，同學都學會如何使用Python語言撰寫分析程式，輸入比賽資料，訓練深度學習模型來



科技部前瞻司委託國家實驗研究院主辦的「科技大擂台與AI對話」Fun Cup比賽，11月16日舉辦Hands-on課程，邀請北科大廖元甫教授、臺大李宏毅教授實驗室成員，帶領參賽同學們實作AI語音技術，11月30日進行線上比賽。亞洲大學（Asia University, Taiwan）資工系林映辰、李柏翰、李衡及劉子源四位同學組隊參賽，從50隊參賽隊伍中，脫穎而出，獲得第二名，12月7日北上受獎。

指導同學參賽的資工系朱學亭老師指出，「科技大擂台與AI對話」是由科技部AI技術研發競賽等。「Fun Cup」是為了使我國青年學子了解並親近AI技術，運用最新AI語音技術，進行人機對話的



亞大朱學亭老師(左)指導資工系與科技部「科技大擂台與AI對話」Fun Cup比賽。

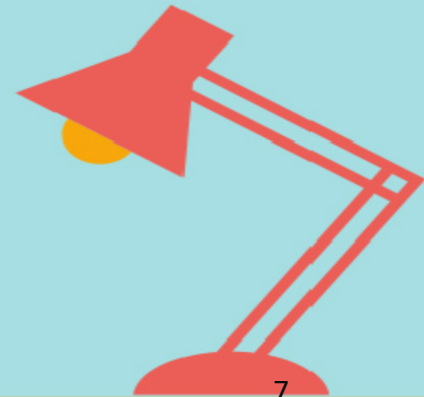
讓他們進行AI語音技術的實作，分工跑程式，完成全部100題語音辨識，更是開心！」

工研院AIDEA平台

[About Us](#)[Topics](#)[Projects](#)[Competition](#)[Career](#)[Login](#)

ITSA 線上程式設計競賽

- 教育部智慧創新跨域人才培育計畫舉辦之每兩月的競賽
- 每次比賽為九題為原則，以中文為主。
- 設有程式自評平臺(E-tutor)，提供線上程式解題與題庫



ITSA程式自學平台



首頁 線上題庫 老師專區 關於平臺 程式能力線上自我評量



- 【分項活動】延長報名-2020年9月ITSA程式能力線上自我...[平臺訊息](#) 2020-08-24
- 【平臺消息】平臺將於8/18(二) 07:00進行伺服器維護...[平臺訊息](#) 2020-08-17
- 【平臺消息】題目錯誤修正說明[平臺訊息](#) 2020-05-28
- 【分項活動】2020年5月ITSA程式能力線上自我評量報名說...[平臺訊息](#) 2020-04-15
- 【活動訊息】緊急通知：ITSA程式能力線上自我評量暫停一次[競賽訊息](#) 2020-02-07

[more](#)

ITSA 線上程式設計競賽

- ITSA 線上程式設計競賽每次比賽九題
- 設有程式自評平臺(E-tutor)，提供題庫



程式自學平臺

首頁 線上題庫 老師專區 關於平臺 程式能力線上自我評量

◦ 【分項活動】延長報名-2020年9月ITSA程式能力線上自我... 平臺訊息 2020-08-24

◦ 【平臺消息】平臺將於8/18(二) 07:00進行伺服器維護... 平臺訊息 2020-08-17

◦ 【平臺消息】題目錯誤修正說明 平臺訊息 2020-05-28

◦ 【分項活動】2020年5月ITSA程式能力線上自我評量報名說... 平臺訊息 2020-04-15

◦ 【活動訊息】緊急通知：ITSA程式能力線上自我評量暫停一次 競賽訊息 2020-02-07

more

<https://sites.google.com/site/itsancku/>

ITSA 線上程式設計競賽

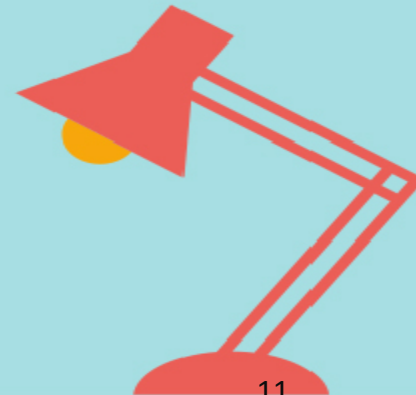


APCS (大學程式設計先修檢測)

- APCS-Advanced Placement Computer Science
- 程式設計觀念題: 50題選擇題。
- 程式設計實作題: 4個非選擇題測驗。

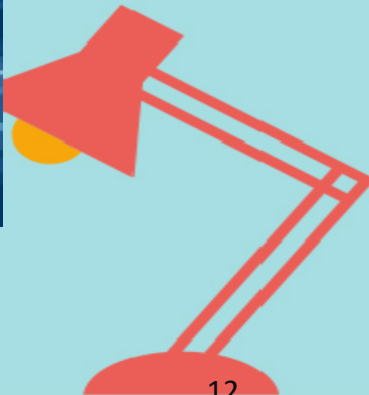


<https://apcs.csie.ntnu.edu.tw/>



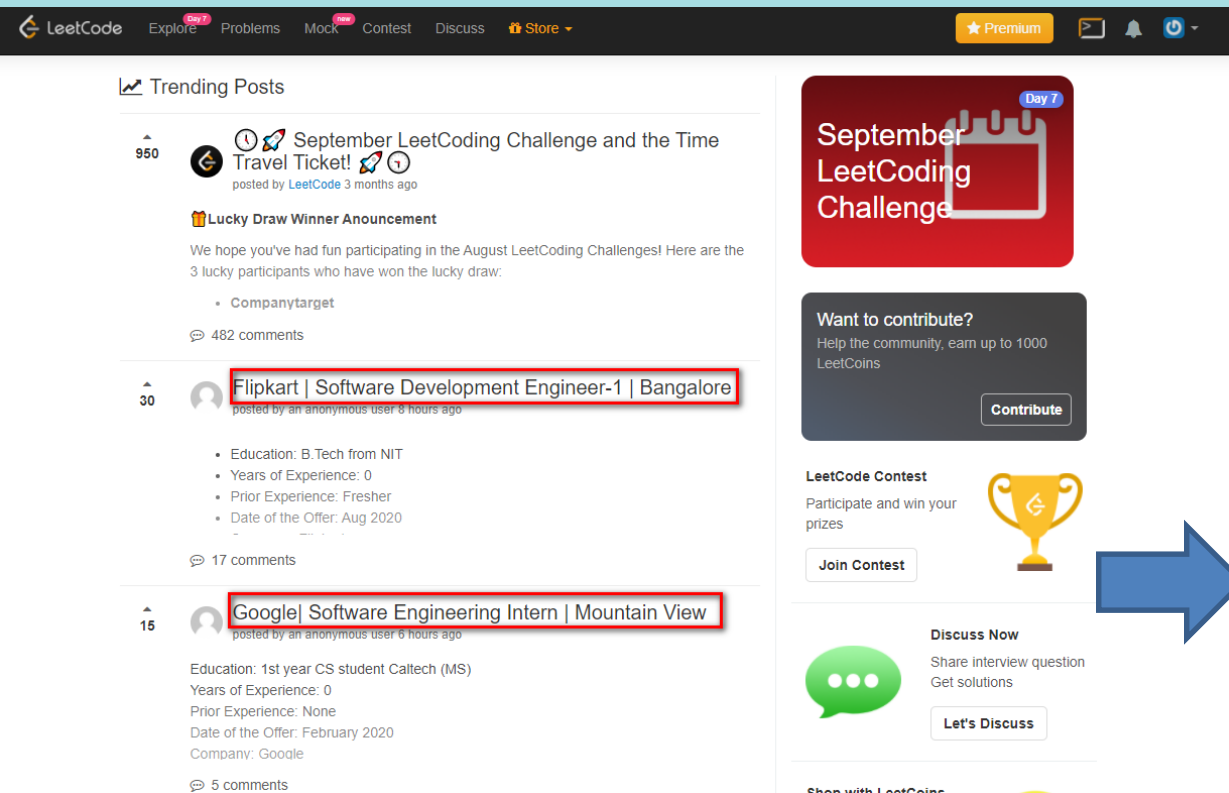
CPE(大學程式能力檢定)

- 線上程式設計，利用電腦自動評判
- CPE題目來自UVA on-line judge網站 (4000題)
- CPE 採用「瘋狂程設」評判系統
- ACM-ICPC國際大學校際程式設計競賽



LeetCode

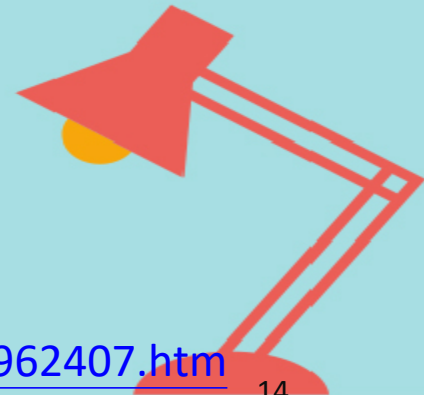
- The World's Leading Online Programming Learning Platform



Google | Software Engineering Intern | Mountain View
Education: 1st year CS student Caltech (MS)
Years of Experience: 0
Prior Experience: None
Date of the Offer: February 2020
Company: Google
Title/Level: Software Engineering Intern
Location: Mountain View
Salary: \$7500/month (USD)
Other details: Remote internship

Google 資深軟件工程師 LeetCode 刷題 筆記首次公開

- 有人說寫代碼就像我們平時開車，僅憑經驗你就可以將車開走；但當有一天，這輛車出問題跑不起來的時候，你 **不懂汽車的運行機制，你要怎麼排除和解決問題？** 所以擁有紮實的數據結構和算法，才能開好編程這輛車。
- 作為程序員立身之根本， **算法更是面試必不可少的環節**。像 Google、Facebook、Apple、Nvidia 等等這些硅谷公司，都會把數據結構和算法作為入門門檻。甚至 **90% 的面試時間** 都是圍繞數據結構和算法
- 建議大家在力扣（ LeetCode ）上刷題的，Leetcode 是注重算法的刷題工具，在考算法的筆試面試中，碰見原題的概率非常大。關鍵是你可以看到其他人留下的 **思考過程和解題思路**，比你啃死板的書籍更有收穫。



Leetcode 美國和中國網站

- 美國Leetcode <https://leetcode.com/>
- 中國Leetcode <https://leetcode-cn.com/>
- 美國Leetcode帳號可以用在中國Leetcode網站
- 題目：演算法, 資料庫, Shell, 多執行緒

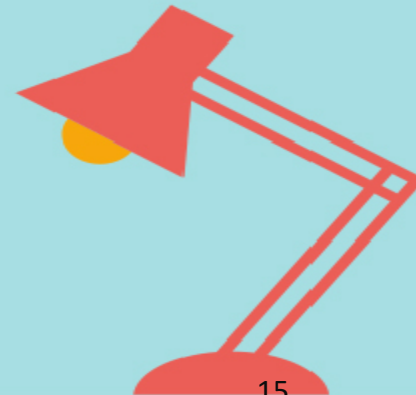
力扣 (LeetCode) - 全部

算法 数据库 Shell 多线程 全新 独家授权 官方授权

已解决 1/1779 - 简单 1 中等 0 困难 0 随机开始

搜索题目 名称、内容 或 编号 难度 状态 列表 标签

#	题名	题解	通过率	难度	出现频率
每日 1 题	77 组合	383	74.8%	中等	
?	1 两数之和	6315	49.4%	简单	



力扣第一题

 探索 **题库** 圈子 竞赛 面试 职位  商店

题目描述

评论 (5.8k)

题解(6315)

提交记录

1. 两数之和

难度 **简单** 9056 收藏 分享 切换为英文 关注 反馈

给定一个整数数组 `nums` 和一个目标值 `target`，请你在该数组中找出和为目标值的那 **两个** 整数，并返回他们的数组下标。

你可以假设每种输入只会对应一个答案。但是，数组中同一个元素不能使用两遍。

示例:

给定 `nums = [2, 7, 11, 15]`, `target = 9`

因为 `nums[0] + nums[1] = 2 + 7 = 9`

所以返回 `[0, 1]`

通过次数 1,355,370 | 提交次数 2,745,965

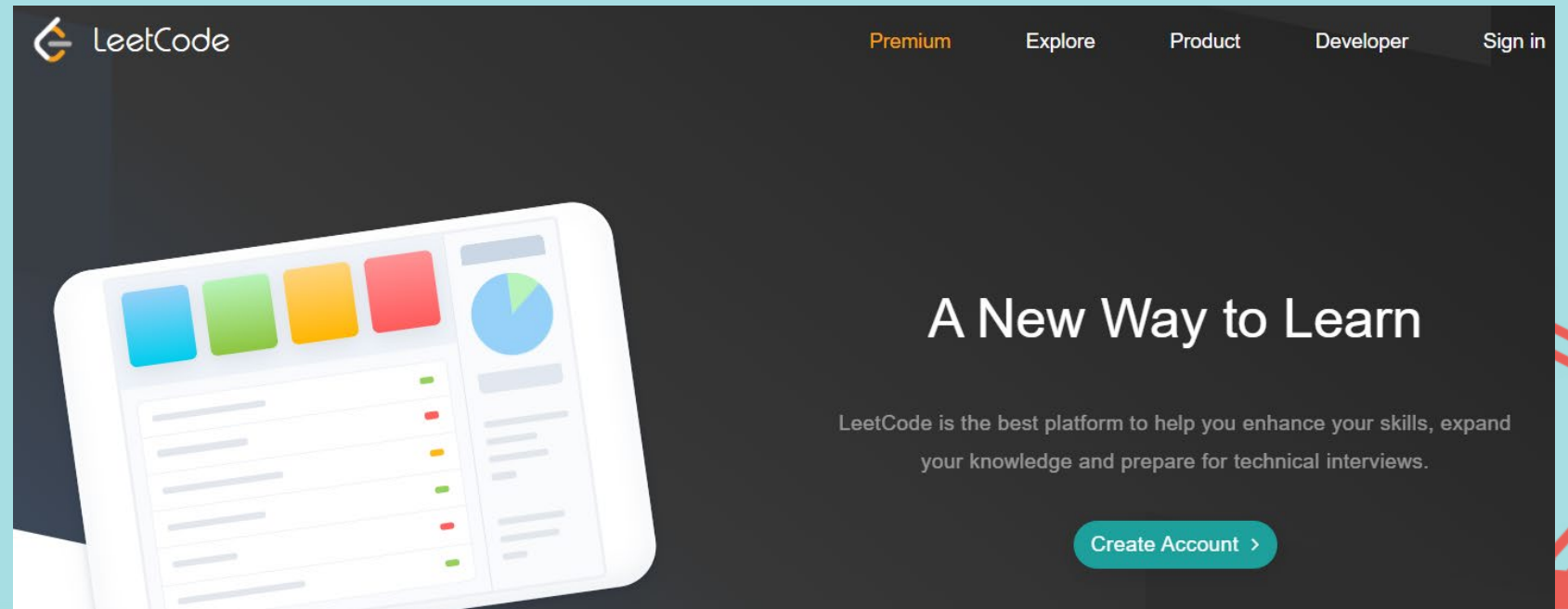
C++

智能模式

```
1 class Solution {
2 public:
3     vector<int> twoSum(vector<int>& nums, int target) {
4
5     }
6 };
```

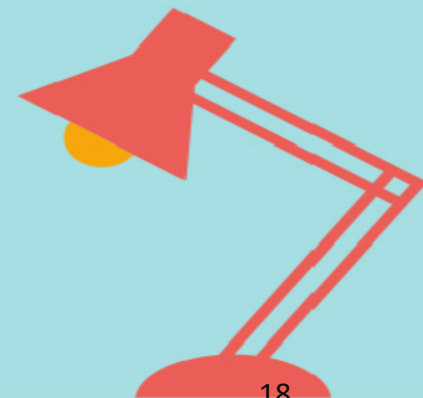
註冊LeetCode帳號

- 登入美國Leetcode <https://leetcode.com/>
- 請同學申請一個LeetCode帳號
- 登入中國Leetcode <https://leetcode-cn.com/>



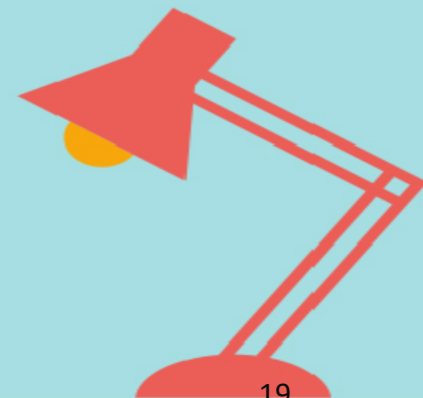
(1)各種程式語言介紹

- 低階程式語言
 - 機械碼
 - 組合語言
- 高階程式語言
 - 直譯語言
 - 編譯語言



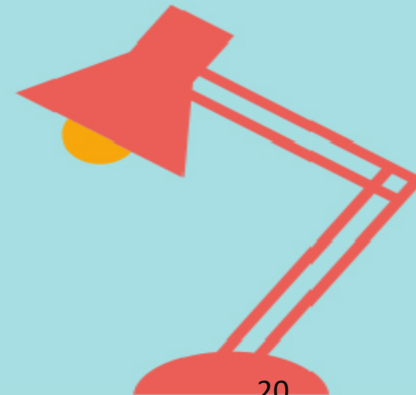
程式語言介紹

- 低階程式語言
 - 機械碼
 - 組合語言
- 高階程式語言
 - 直譯語言
 - 編譯語言



機械碼 (Machine Code)

- i386:
10110000 00000011
11111110 11000000
11000000 11100000 00000010
- MIPS:
00100000 00001000 00000000 00000011
00100001 00001000 00000000 00000001
00000000 00001000 01000000 10000000
- 他們都在說 $(3 + 1) * 4$!



i386

10110000 00000011

$X = 3$

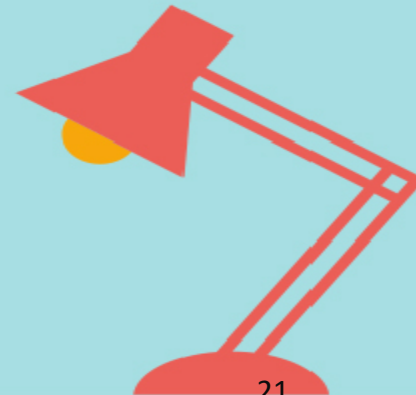
11111110 11000000

$X = X + 1$

11000000 11100000 00000010

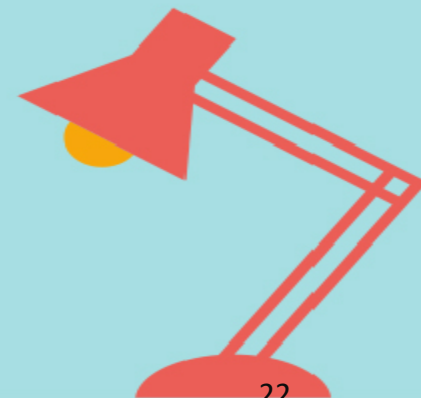
$X = X * 4$

$X = (3 + 1) * 4$



組合語言

- 用單字取代 0 和 1
- 給暫存器 (register) 名字
- 動作 (通常) 很單純
- MOV, JMP, ADD
- mov al, 3
inc al
shl al, 2
 - addi \$t0, \$0, 3
addi \$t0, \$t0, 1
sll \$t0, \$t0, 2



組合語言 (cont.)

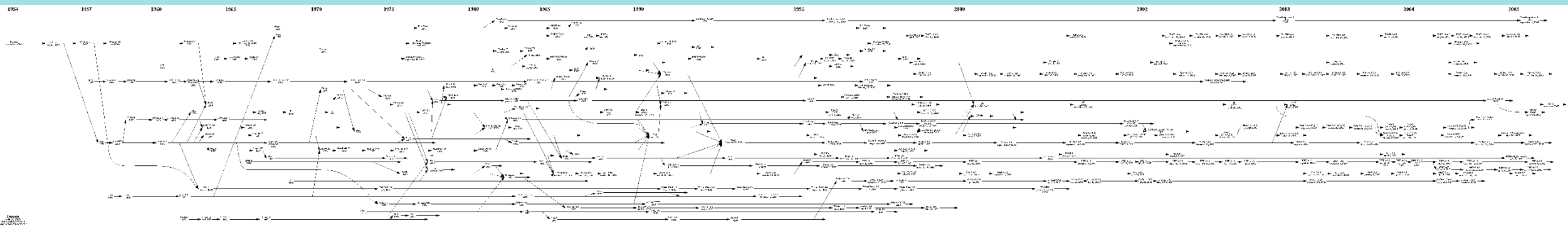
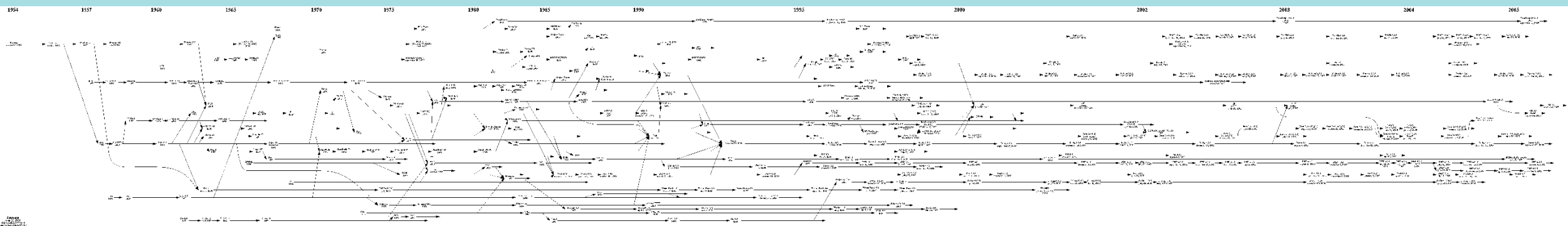
- 不允許一些危險行為

```
10110000 00000000 11111111 10101100 00000000
11000000 11100000 00000010 11101011 11101011
11110110
```

MOV	al, 3
ADD	al, 1
SHL	al, 2
JMP	-8
ADD	edi, esi
ROL	al, e0h

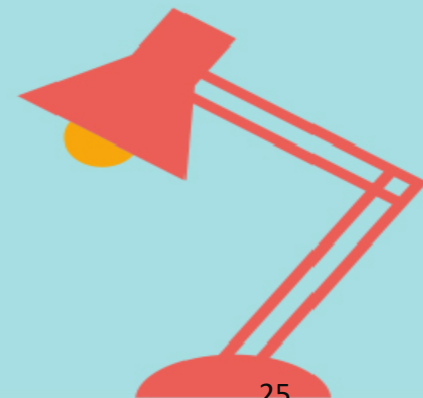


高階語言



高階語言 (cont.)

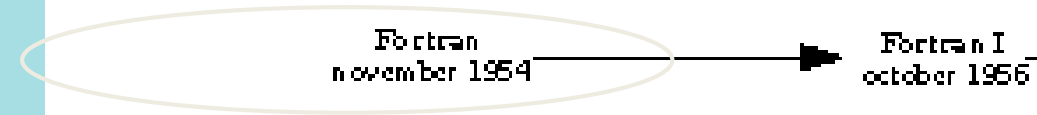
- 變數
 - 不用自己管資料放在哪裡
 - 可以取個有意義的名字
- 指派 (assignment)
 - $X = 3; X = 2;$
- 運算式
 - $c = \text{sqrt}(a * a + b * b)$
- 控制結構
 - 如果... 就 ... 要不 ...: if ... then ... else ...
 - 如果 ... 就重複 ...: while ... do ...
- 函式
 - 內建函式: $\text{sin}()$, $\text{cos}()$, ...
 - 使用者自訂



1954

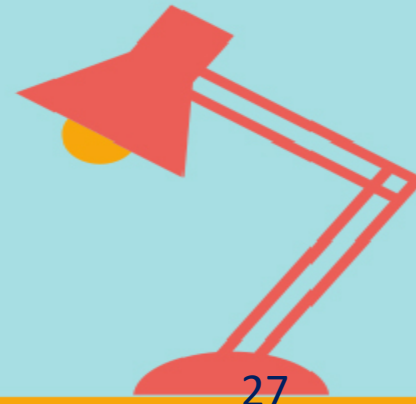
FORTRAN

- 第一個高階語言！
- 還在用的老古董
- Mathematical **F**ormula **T**ranslating System
- IBM 1954-57 完成
- 1977 年訂定 F77 標準
 - 結構化
 - 多種內建資料型態
- 科學計算
 - 效率



FORTRAN (cont.)

```
PROGRAM CMPLXD
  IMPLICIT COMPLEX(X)
  PARAMETER (PI = 3.141592653589793, XJ = (0, 1))
  DO 1, I = 0, 7
    X = EXP(XJ * I * PI / 4)
    IF (AIMAG(X).LT.0) THEN
      PRINT 2, 'e**(j*', I, '*pi/4) = ', REAL(X), ' - j', -AIMAG(X)
    ELSE
      PRINT 2, 'e**(j*', I, '*pi/4) = ', REAL(X), ' + j', AIMAG(X)
    END IF
  2 FORMAT (A, I1, A, F10.7, A, F9.7)
  1 CONTINUE
  STOP
END
```



Pascal

- 高度結構化、嚴格的語法
 - 別想惡搞
 - 安全
- 自訂結構
- 巢狀函式
- 第一個有「中介碼」的編譯器

1970

1

Prolog
1970

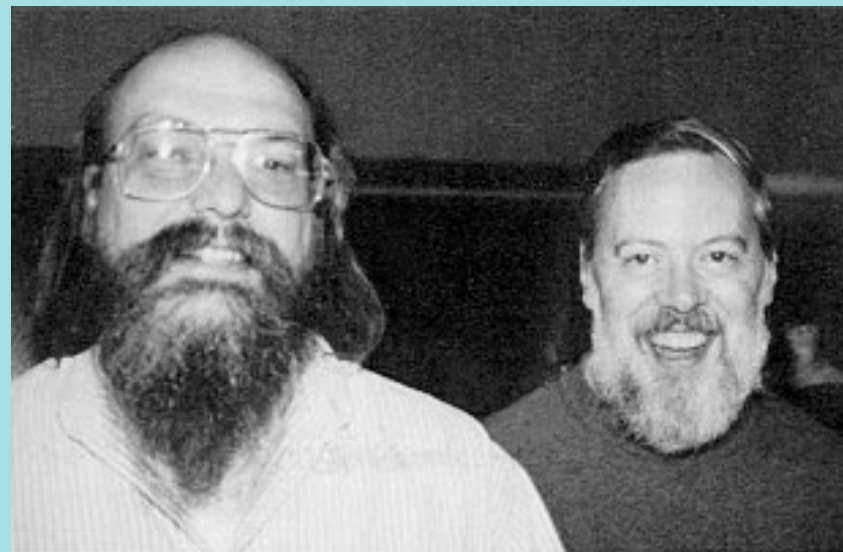
COBOL 74 ANSI
1974

Pascal
1970

C

- AT&T Bell Labs
- 因 Unix 而生，因 Unix 而壯大
- 語法自由
- 效率高
- 現今最強勢語言的源頭 (Java, C#, ...)
- 提供指標但沒有自動記憶體管理 (危險！)

```
void strcpy(char *s, char *t) {  
    while (*s++ = *t++);  
}
```



肯·汤普逊（左）和丹尼斯·里奇

C++

- AT&T Bell Labs
- 勝過 Objective-C 成為 C 加上 OO 的標準語言
- 和 C (幾乎) 向下相容
- 同樣注重效率
- 包含泛型程式設計
 - 因其實作方式，使用之也成為一種技術
 - 編譯期可以做「很多事」



Java & C#

- Sun Micro. & Microsoft
- 編譯成虛擬碼再由虛擬機器執行
- 執行時及時編譯成原生碼
- 超大函式庫

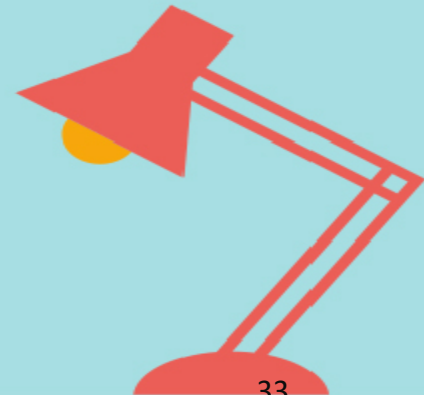
Go 程式語言

- Go (又稱Golang) 是Google開發的一種靜態強型別、編譯型、並發型，並具有垃圾回收功能的程式語言。
- 作者：Robert Griesemer, Rob Pike, Ken Thompson
- 湯普遜在一次採訪中表達了對C++的負面評價，說：
「C++是一個由各種互相衝突的想法組成的垃圾堆。」



腳本語言 (Scripting Language)

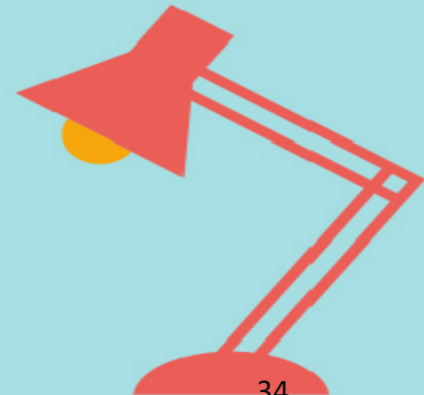
- Shell script
- awk/sed
- Tk/Tcl
- Perl
- Python
- Ruby
- Javascript



Shell script + awk + sed

```
f=`echo $1 | sed -e 's|\\(. *\\)\\.^[^/]*$|\\1|'`  
autob5 -i utf8 -o big5 < $1 | bg5conv > $f.cjk \  
    && latex $f.cjk
```

```
ps x | grep bbs | awk '{print $1}' | xargs kill
```



Perl

- 處理文字資料非常方便
- CPAN 上有上萬個模組
- 滿地 \$, %, @ 和奇怪符號
- Perl 1 到 Perl 5 完全向下相容
 - 造成不少混淆和麻煩
- 語法自由，可以寫到完全無法理解



Python

- 「ABC programming language」
 - 用縮排當語法！
 - 高度可讀
 - 龐大的標準函式庫
 - 物件導向
-
- Your challenge is powered by Python!
 - 作者：Guido van Rossum



Ruby

- 強調和其他語言很像
 - Perl
 - Python
 - Smalltalk
 - Eiffel
- OO

```
ash = {:water => 'wet', :fire => 'hot'}  
puts hash[:fire] # Prints: hot
```

```
hash.each_pair do |key, value|  
  puts "#{key} is #{value}"  
end
```



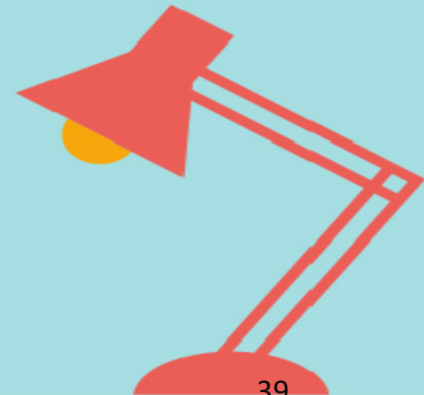
Javascript

- 控制網頁上的物件
- AJAX !



函數式程式語言

- 寫出函數的定義就是程式
- 不用「assignment」
- 函數和其他所有資料有同等地位
 - 匿名函數
- Closure



lisp

- 變成一種通稱
- clisp, scheme 等「方言」
- 一堆括號
- 允許 assignment
- 非常動態

```
(defun factorial (n)
  (if (<= n 1)
      1
      (* n (factorial (- n 1)))))
```



Haskell

- 靜態
- 型別檢查
- 自動型別推論
- 完全不用 `assignment`
- 重要的程式語言理論實作平台



Prolog

- 邏輯程式語言
- 寫出邏輯，然後問問題

```
sibling(X,Y) :- parent(Z,X), parent(Z,Y).  
parent(X,Y) :- father(X,Y).  
parent(X,Y) :- mother(X,Y).  
mother(trude, sally).  
father(tom, sally).  
father(tom, erica).  
father(mike, tom).
```

```
?- sibling(sally, erica).  
yes.
```



PHP/ASP/JSP

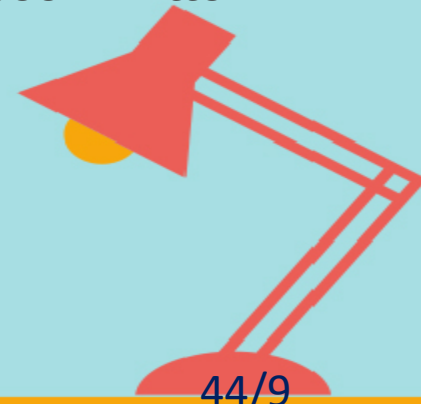
- 在 HTML 裡直接寫程式
- `<?php`
- `echo 'Hello World!';`
- `?>`



TIOBE Index for September 2020

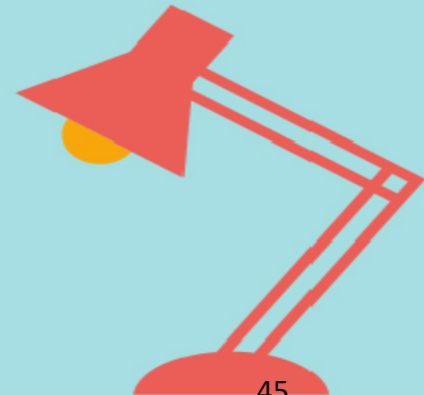
Sep 2020	Sep 2019	Change	Programming Language	Ratings	Change
1	2	▲	C	15.95%	+0.74%
2	1	▼	Java	13.48%	-3.18%
3	3		Python	10.47%	+0.59%
4	4		C++	7.11%	+1.48%
5	5		C#	4.58%	+1.18%
6	6		Visual Basic	4.12%	+0.83%
7	7		JavaScript	2.54%	+0.41%
8	9	▲	PHP	2.49%	+0.62%
9	19	▲▲	R	2.37%	+1.33%
10	8	▼	SQL	1.76%	-0.19%
11	14	▲	Go	1.46%	+0.24%
12	16	▲▲	Swift	1.38%	+0.28%
13	20	▲▲	Perl	1.30%	+0.26%
14	12	▼	Assembly language	1.30%	-0.08%
15	15		Ruby	1.24%	+0.03%
16	18	▲	MATLAB	1.10%	+0.04%
17	11	▼▼	Groovy	0.99%	-0.52%
18	33	▲▲	Rust	0.92%	+0.55%
19	10	▼▼	Objective-C	0.85%	-0.99%
20	24	▲▲	Dart	0.77%	+0.13%

The ratings are based on the number of skilled engineers world-wide, courses and third party vendors. Popular search engines such as Google, Bing, Yahoo!, Wikipedia, Amazon, YouTube and Baidu are used to calculate the ratings. It is important to note that the TIOBE index is not about the *best* programming language or the language in which *most lines of code* have been written.

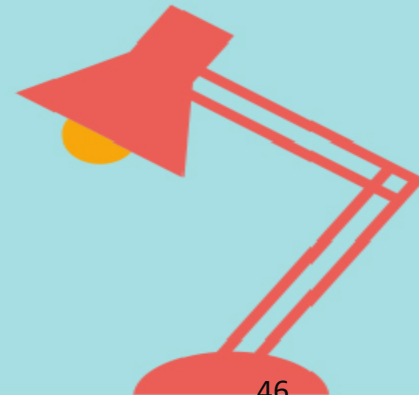


編譯器、直譯器

- 直譯器: 邊讀程式邊執行
 - 可以邊寫邊執行，方便除錯
 - 效率較差
 - 程式碼要帶著走
- 編譯器: 編譯成執行檔再執行
 - 寫完才能編譯，錯誤比較難找
 - 效率好；程式碼可以藏起來

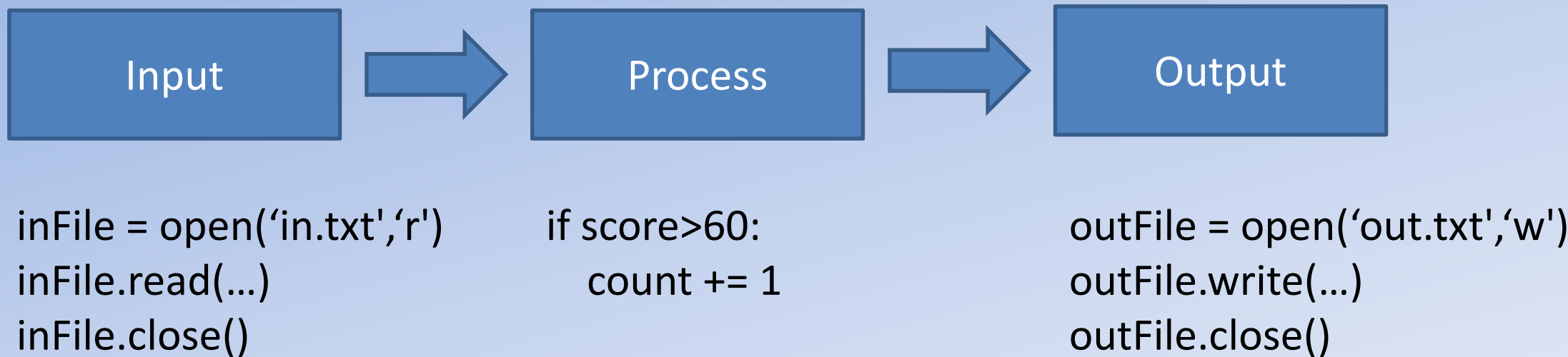


- Learning Python with Examples
- <https://htchutw.gitbooks.io/learning-python-with-examples/content/>
- Study of Deep Learning with Examples
- <https://htchutw.gitbooks.io/study-of-deep-learning-with-examples/content/>



(2)寫程式的基本模型

Input-Process-Output 程式模型



(2a)程式要素：資料結構和演算法

演算法；具體計算步驟的序列，可以用紙筆計算。

計算240和96的最大公因數gcd，使用輾轉相除法

紙筆計算：

$240 \div 96 = 2 \dots 48$

$96 \div 48 = 2 \dots 0$

當餘數是0時，除數就是gcd

所以，240和96的gcd=48

演算法：

`a=240; b=96`

`r=a%b`

`while r!=0:`

`a=b;b=r`

`r=a%b`

`print("gcd=",b)`

當餘數是0時，除數就是gcd

寫成函數：

`def gcd(a, b):`

`r=a%b`

`while r!=0:`

`a=b;b=r`

`r=a%b`

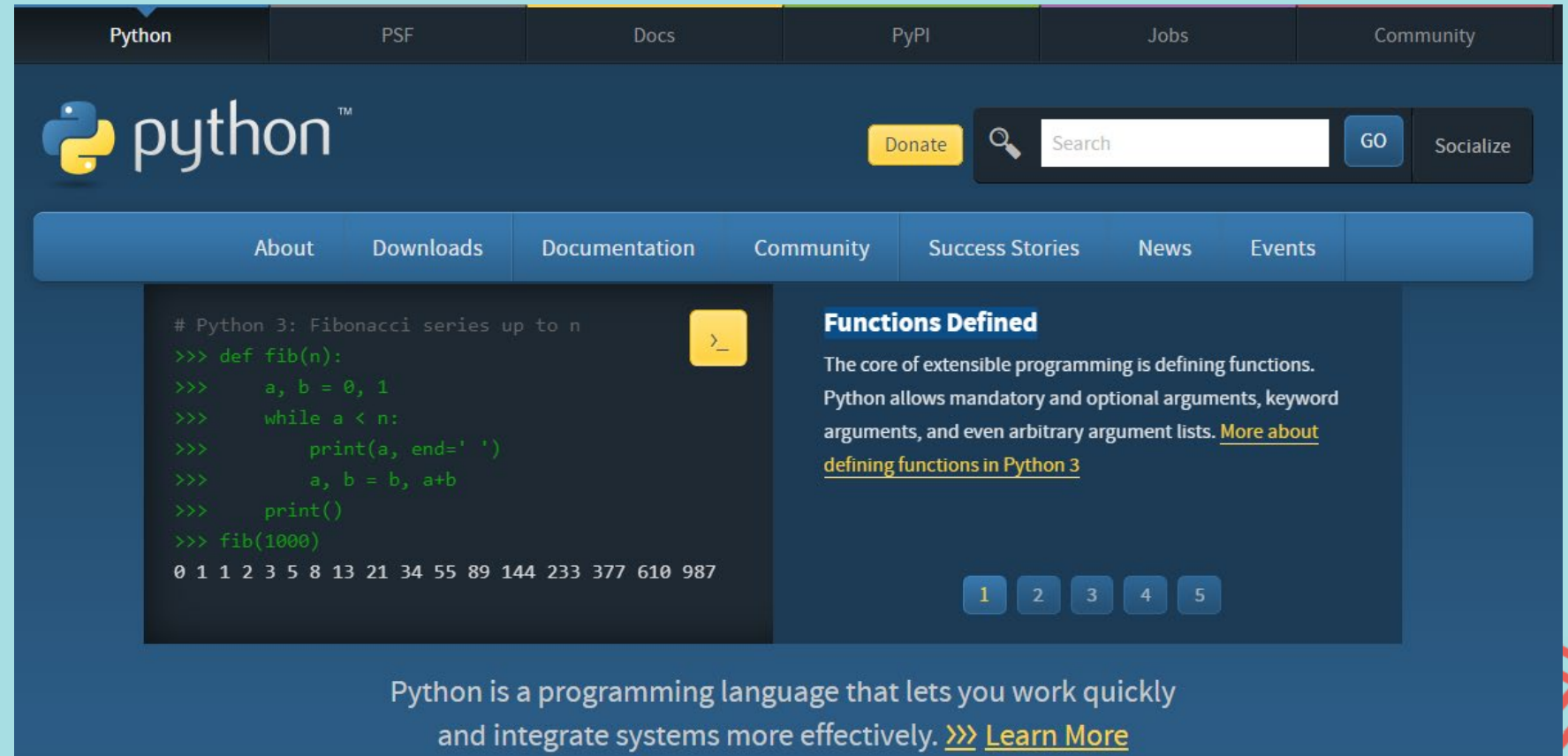
`return b`

(3.a) 寫Python程式前的準備

- Python的官網
- Python程式的編輯工具
- Jupyter notebook



(3.b) Python官網的介紹



The screenshot displays the Python.org homepage with a dark blue header and navigation bar. The navigation bar includes links for Python, PSF, Docs, PyPI, Jobs, and Community. Below this, the Python logo is prominently displayed. To the right of the logo is a search bar with a magnifying glass icon and a 'GO' button, along with a 'Donate' button and a 'Socialize' button. A secondary navigation bar below the header contains links for About, Downloads, Documentation, Community, Success Stories, News, and Events. The main content area features a code editor on the left with a Python 3 Fibonacci function example, a 'Functions Defined' section on the right explaining the core of extensible programming, and a footer with the text 'Python is a programming language that lets you work quickly and integrate systems more effectively. >>> [Learn More](#)'.

Python

PSF

Docs

PyPI

Jobs

Community

python™

Donate

Search

GO

Socialize

About

Downloads

Documentation

Community

Success Stories

News

Events

```
# Python 3: Fibonacci series up to n
>>> def fib(n):
>>>     a, b = 0, 1
>>>     while a < n:
>>>         print(a, end=' ')
>>>         a, b = b, a+b
>>>     print()
>>> fib(1000)
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987
```

Functions Defined

The core of extensible programming is defining functions. Python allows mandatory and optional arguments, keyword arguments, and even arbitrary argument lists. [More about defining functions in Python 3](#)

1 2 3 4 5

Python is a programming language that lets you work quickly and integrate systems more effectively. [>>> Learn More](#)

(3.c) Python的版本

- 目前版本~~Python 2.7.x~~和Python 3.8.x
- **print 函数**
- **Unicode**
- **除法运算**



Sunsetting Python 2



Donate



Search

GO

About

Downloads

Documentation

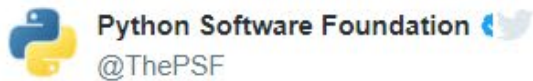
Community

Success Stories

News

Events

Tweets by @ThePSF



Python Software Foundation

@ThePSF

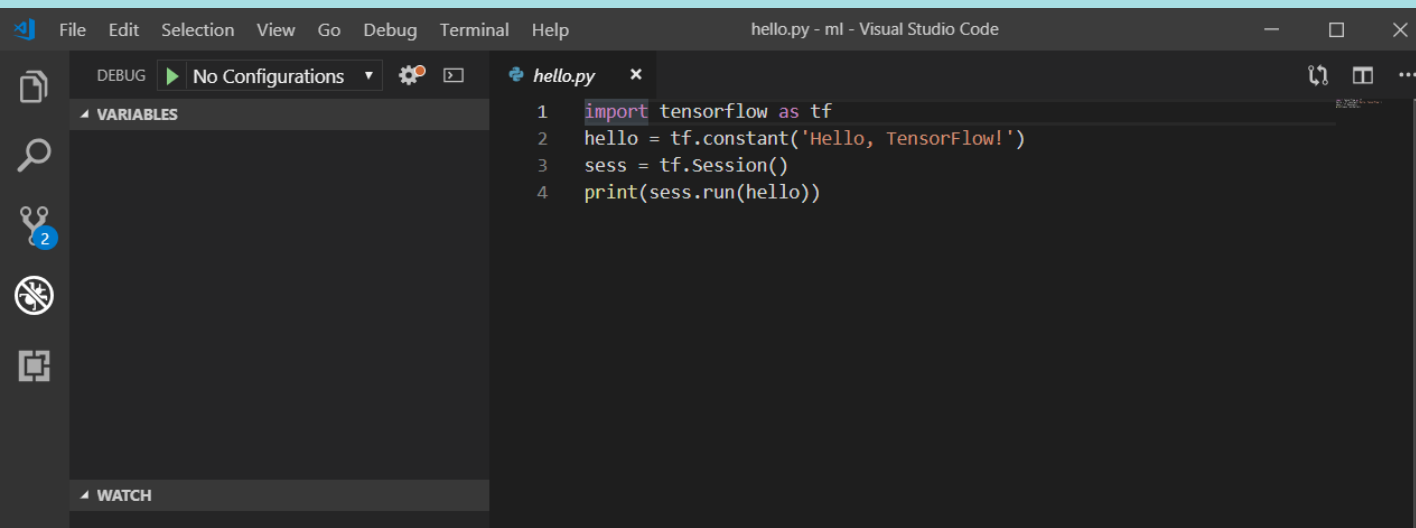
CUDA in Your Python: Effective Parallel Programming on the GPU by William Horton. Learn how to speed up your Python programs using Nvidia's CUDA platform. pyvideo.org/pycolorado-201...

Sunsetting Python 2

We are volunteers who make and take care of the Python programming language. We have decided that **January 1, 2020**, was the day that we sunset Python 2. That means that we will not improve it anymore after that day, even if someone finds a security problem in it. You should upgrade to Python 3 as soon as you can.

(3.d) Python程式的編輯工具

- Python IDLE工具
- Visual Studio Code
- PyCharm
- Colab



Visual Studio Codespaces



Microsoft | Visual Studio 其他 ▾ 免費 Visual Studio 所有 Microsoft ▾ 🔍 👤

Visual Studio Codespaces

(前身為 Visual Studio Online)

隨處皆可存取的雲端託管開發環境

立即在您的瀏覽器或 Visual Studio Code 中試用

在 Visual Studio 2019 試用

開始使用 >

註冊非公開預覽版 >

意見反應



e) Jupyter notebook

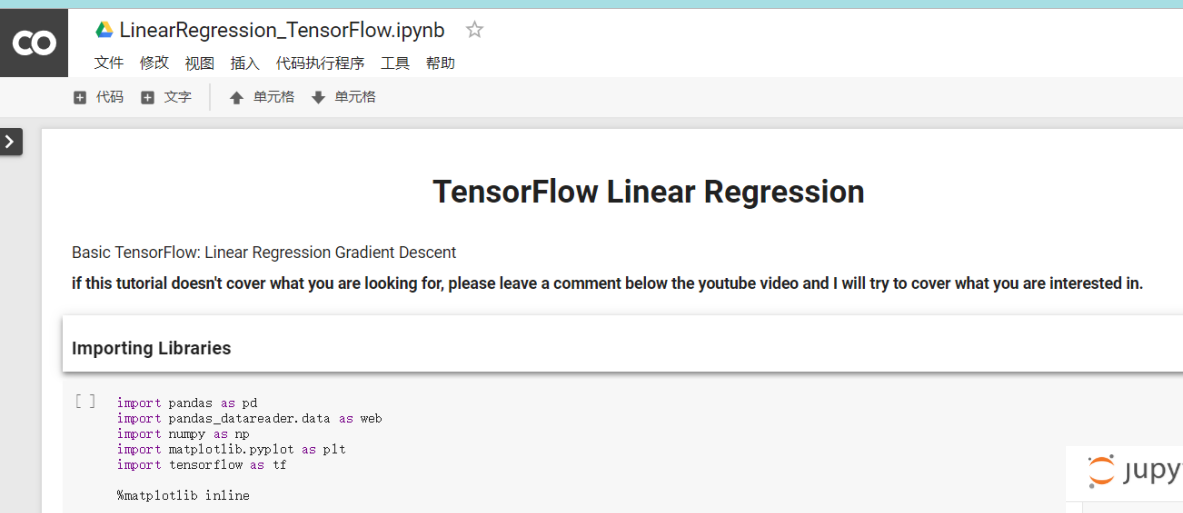
notebook是可以在網頁寫程式碼的工具。
接在網頁執行並且可直接將輸出呈現

- 整個網頁可以匯出成notebook的形式分享。
- 資料分析、機器學習的平台(如 Kaggle)上面的討論都是以Jupyter notebook的形式交流。
- 支援多種主流的直譯語言像是Python, R, Julia。



(3.f) Jupyter notebook的雲端平台

- Google Colab



(4) Python 標準函式庫(Standard Library)

- <https://docs.python.org/3/library/>

Text Processing Services

- `string` — Common string operations
- `re` — Regular expression operations
- `difflib` — Helpers for computing deltas
- `textwrap` — Text wrapping and filling
- `unicodedata` — Unicode Database
- `stringprep` — Internet String Preparation
- `readline` — GNU readline interface
- `rlcompleter` — Completion function for GNU readline

Binary Data Services

- `struct` — Interpret bytes as packed binary data
- `codecs` — Codec registry and base classes

Data Types

- `datetime` — Basic date and time types
- `calendar` — General calendar-related functions
- `collections` — Container datatypes
- `collections.abc` — Abstract Base Classes for Containers
- `heapq` — Heap queue algorithm
- `bisect` — Array bisection algorithm
- `array` — Efficient arrays of numeric values
- `weakref` — Weak references
- `types` — Dynamic type creation and names for built-in types
- `copy` — Shallow and deep copy operations
- `pprint` — Data pretty printer
- `reprlib` — Alternate `repr()` implementation
- `enum` — Support for enumerations

Numeric and Mathematical Modules

- `numbers` — Numeric abstract base classes
- `math` — Mathematical functions
- `cmath` — Mathematical functions for complex numbers
- `decimal` — Decimal fixed point and floating point arithmetic
- `fractions` — Rational numbers
- `random` — Generate pseudo-random numbers
- `statistics` — Mathematical statistics functions

Functional Programming Modules

- `itertools` — Functions creating iterators for efficient looping
- `functools` — Higher-order functions and operations on callable objects
- `operator` — Standard operators as functions

File and Directory Access

- `pathlib` — Object-oriented filesystem paths
- `os.path` — Common pathname manipulations
- `fileinput` — Iterate over lines from multiple input streams
- `stat` — Interpreting `stat()` results
- `filecmp` — File and Directory Comparisons
- `tempfile` — Generate temporary files and directories
- `glob` — Unix style pathname pattern expansion
- `fnmatch` — Unix filename pattern matching
- `linecache` — Random access to text lines
- `shutil` — High-level file operations
- `macpath` — Mac OS 9 path manipulation functions

Python Runtime Services

- `sys` — System-specific parameters and functions
- `sysconfig` — Provide access to Python's configuration information
- `builtins` — Built-in objects
- `__main__` — Top-level script environment
- `warnings` — Warning control
- `dataclasses` — Data Classes
- `contextlib` — Utilities for with-statement contexts
- `abc` — Abstract Base Classes
- `atexit` — Exit handlers
- `traceback` — Print or retrieve a stack traceback
- `__future__` — Future statement definitions
- `gc` — Garbage Collector interface
- `inspect` — Inspect live objects
- `site` — Site-specific configuration hook

Custom Python Interpreters

- `code` — Interpreter base classes
- `codeop` — Compile Python code

Importing Modules

- `zipimport` — Import modules from Zip archives
- `pkgutil` — Package extension utility
- `modulefinder` — Find modules used by a script
- `runpy` — Locating and executing Python modules
- `importlib` — The implementation of `import`

(4.a)re正規運算式

```
import re
```

```
m = re.search('(?<=abc)def', 'abcdef')
```

```
import re
```

```
m = re.search(r'(?<=-)\w+', 'spam-egg')
```

```
m.group(0)
```

(?<=...) 匹配字符串前面匹配 ... 的内容。

(?<!...) 匹配字符串前面不配 ... 的内容。

```
[1] #re — Regular expression operations
import re
m = re.search('(?<=abc)def', 'abcdef')
m.group(0)
```

↳ 'def'

```
▶ m = re.search(r'(?<=-)\w+', 'spam-egg')
m.group(0)
```

↳ 'egg'

(4.b) random --- 生成亂數

```
import random as rd  
a=rd.random()
```

#random --- 生成亂數

```
import random as rd
```

```
a=rd.random()
```

```
b=rd.uniform(2.5, 10.0)
```

```
c=rd.randrange(10)
```

```
d=rd.choice(['win', 'lose', 'draw'])
```

```
print(a,b,c,d)。
```

Random float: 0.0 <=

Random float: 2.5

Integer from 0 to 9

Single random element from a sequence

```
[1] #re — Regular expression operations  
import re  
m = re.search('(?<=abc)def', 'abcdef')  
m.group(0)
```

↳ 'def'

```
m = re.search(r'(?<=-)\w+', 'spam-egg')  
m.group(0)
```

↳ 'egg'

(4.c) sqlite3 --- SQLite 數據庫

```
import sqlite3
```

```
conn = sqlite3.connect('example.db')
```

```
conn = sqlite3.connect('example.db')  
c = conn.cursor()  
c.execute()  
conn.commit()  
conn.close()
```

```
[17] #sqlite3  
import sqlite3  
conn = sqlite3.connect('example.db')  
c = conn.cursor()  
  
# Create table  
c.execute('''CREATE TABLE stocks  
          (date text, trans text, symbol text, qty real, price real)''')  
  
# Insert a row of data  
c.execute("INSERT INTO stocks VALUES ('2006-01-05','BUY','RHAT',100,35.14)")  
c.execute("INSERT INTO stocks VALUES ('2006-04-05','BUY','MSFT',1000,72.00)")  
  
# Save (commit) the changes  
conn.commit()  
  
# We can also close the connection if we are done with it.  
# Just be sure any changes have been committed or they will be lost.  
conn.close()
```

```
[19] conn = sqlite3.connect('example.db')  
c = conn.cursor()  
for row in c.execute('SELECT * FROM stocks ORDER BY price'):  
    print(row)  
conn.close()
```

```
➞ ('2006-01-05', 'BUY', 'RHAT', 100.0, 35.14)  
   ('2006-04-05', 'BUY', 'MSFT', 1000.0, 72.0)
```

(4.d) urllib --- URL 處理模組

```
from urllib import request
```

```
request.urlopen('http://...')
```

```
[20] from urllib import request  
with request.urlopen('http://www.asia.edu.tw/news1.php') as response:  
    html = response.read().decode('utf-8')  
    print(html)
```



```
<html>  
<head>  
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">  
<meta name="viewport" content="width=device-width">  
<meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1" />  
<meta name="verify-v1" content="IAHYK7vmrRk/KviiF+L04ShubFzyQ71KR7NERk1Km/Q=" >  
<META name="y_key" content="1d3ca36296b6703a">  
<meta name="msvalidate.01" content="65BDB355D02C416670B1099A28F087CF" />  
<meta name="majestic-site-verification" content="MJ12_529e78b6-6269-4f07-bdaa-4e
```

```
from urllib import request
```

```
with request.urlopen('http://www.asia.edu.tw/news1.php') as response:
```

```
    html = response.read().decode('utf-8')
```

```
    print(html)
```

(5) Python常用外部函數庫

- NumPy/SciPy
- Pandas
- Matplotlib
- Scikit-Learn
- LightGBM/XGBoost
- Tensorflow/Keras
- PyTorch

- Scrapy/BeautifulSoup
- Seaborn/Plotly/Bokeh
- Gensim
- Folium
- Pillow/Scikit-image/cv2
- Django/Flask/Tornado
- Jieba

(5.a) NumPy/SciPy

- SciPy是一個開源的Python演算法庫和數學工具包。
- NumPy是Python語言的一個擴充程式庫。支援高階大量的維度陣列與矩陣運算，此外也針對陣列運算提供大量的數學函式函式庫。



(5.b) Pandas

- pandas是為Python編程語言用於數據操作和分析的軟件庫。特別是，它提供了用於操作數值表和時間序列的數據結構和操作。



(5.c) Matplotlib

- matplotlib是Python程式語言及其數值數學擴展包 NumPy的可視化操作界面。



(5.d) Scikit-Learn

- Scikit-learn是機器學習庫。與NumPy和SciPy進行互操作。



(5.e) LightGBM/XGBoost

- LightGBM/XGBoost是決策樹算法的機器學習框架



(5.f) Tensorflow/Keras

- Tensorflow是由Google開發的機器學習庫。
- tf.keras 是 TensorFlow 的高階 API，用於建構及訓練深度學習模型。



(5.g) PyTorch

- PyTorch是由Facebook開發的機器學習庫。



(5.h) Scrapy/BeautifulSoup

- Python 爬蟲的工具



(5.i) Seaborn/Plotly/Bokeh



(5.j) Gensim

- Gensim是針對無監督式機器學習和自然語言處理的軟件包。



(5.k) Folium-地圖互動

- folium是基於leaflet.js的python地圖套件，讓使用者可以很快速的使用python產生一些可互動的地圖。



(5.1) Pillow/Scikit-image/cv2

- Pillow/ Scikit-image 是 Python強大的圖像處理庫，功能包括：基本圖像操作（創建縮略圖、幾何變換、圖像裁剪、圖像分離與合併、粘貼圖片）圖像存儲、圖像顯示、格式轉換、截屏操作、圖像繪製功能、圖像濾鏡功能等。
- cv2是OpenCV的Python庫



(5.m) Django/Flask/Tornado

- Django/Flask/Tornado是三個最常用的Python Web框架



(5.n) Jieba

- Jieba是中文自然語言處理基礎套件。





Thanks!

Q&A

