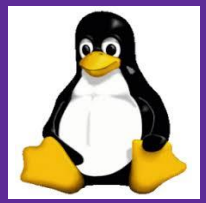
Introduction to Computation

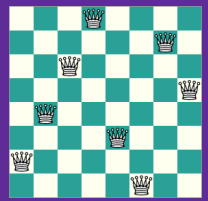
Autumn, 2023

Prof. Fan Cheng

Shanghai Jiao Tong University

chengfan85@gmail.com
https://github.com/ichengfan/itc





1

Outline

- Course Information
- Introduction to Computation
- Introduction to Python

Course Information

Course Motivation

Entry-Level Course for Beginners with zero background in Computer Science

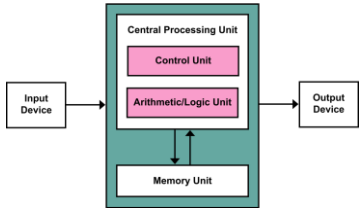
闻道有先后，术业有专攻

——韩愈《师说》

NO Prerequisites

Basic professional training

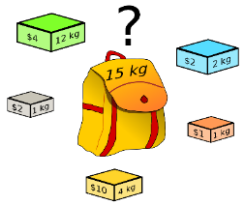
Background variance



Principles

Tools

Language



Problem solving

If the course is too easy for you, just have a chat with me!!!

Course Syllabus

Section A: Lecture (32 lecture hours)

- Python Programming and Principal Problem-Solving Methods
  - Brief Introduction to Python
  - Basic Data and Variable Types
  - Basic Data Structure
  - Control and loop Statements
  - Functions and Recursion
  - Principal Problem-Solving Methods
  - Class, Module, Exception
  - File, Random, Complexity
  - Pythonic: List comprehension, Functional programming, Lambda, Closure, Namespace

Section B: Experiment (32 lecture hours)

- Part I: Weekly Exercise
  - Exercises for each lecture (Leet code 300)
  - Introduction to Linux/Ubuntu
- Part II: Individual Projects
  - Recursion
  - Prime number
  - Big integer
- Part III: Team Project
  - A simplified version of “numpy”

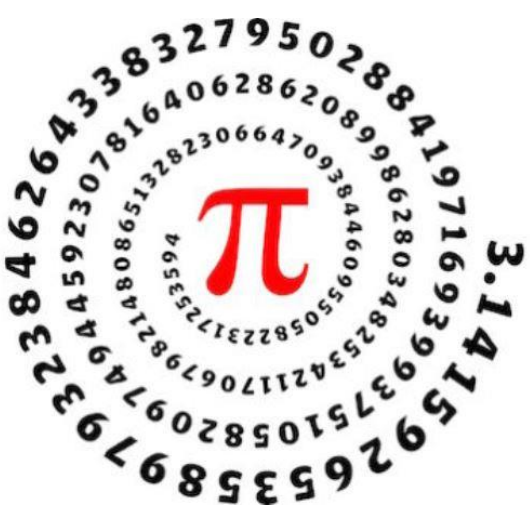
Binary Number

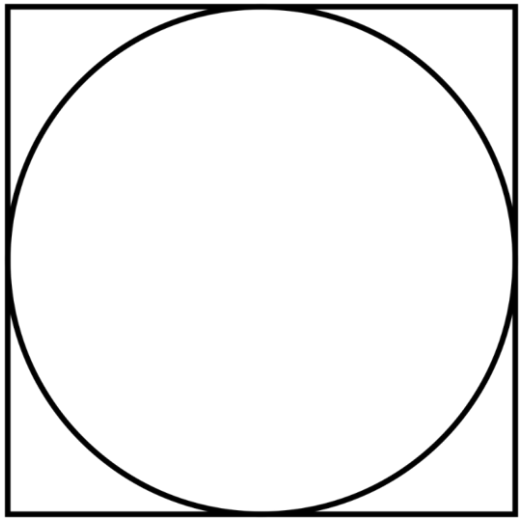
Everyday, we use the decimal numeral system
In mathematics and digital electronics, a binary number is a number expressed in the base-2 numeral system or binary numeral system, which uses only two symbols: typically 0 (zero) and 1 (one).
Decimal number and Binary Number
96 = 64 + 32 = (1100000)2
2018 = ?

Conversion: Transform an arbitrary integer N from decimal to binary and vice versa.
Check the last bit of N:
if it is 1, then
if it is 0, then
N/2

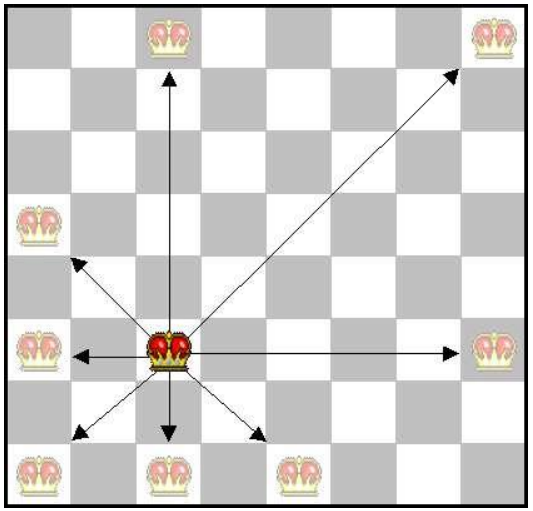
An example of Algorithm

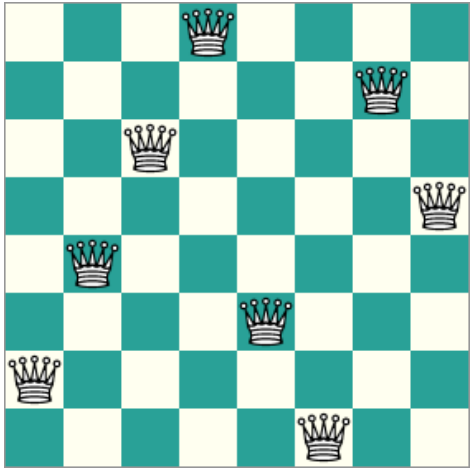
How to Compute π





Eight Queens Problem





Course Schedule

- Lecture Time:

  Tuesday: 10:00-12:00 p.m. (1st - 2nd), 1st -16th week
- Location:

  Upper Hall 215 (上院215)

- Experiment Time:

  Tuesday, 4:00-6:00 p.m. (9th - 10th)
- Location:

  上院215

人来务必带电脑

Grade Policy

Grade Distribution

- Homework & individual projects (10% + 15%)
- Team Project (25%)
  - Last Year: A simplified version of “numpy”
  - Group: 2-3
  - Requirement: Demo + Report
  - In discussion with TA’s, to be announced later
- Final exam (50%)

Textbook

- Course lecture notes
  - A textbook will be written by the lecture notes
- Complements
  - Learning Python: powerful object-oriented programming, Mark Lutz, Publisher: O‘Reilly Media, Year: 2013 (5th) 比较全面基础的教材
  - SICP: Structure and Interpretation of Computer Programs, Gerald Jay Sussman and Hal Abelson, 2nd. (有相当经验，学有余力)

Instructors

- Instructors
  - Prof. Fan Cheng, chengfan@sjtu.edu.cn, https://www.cs.sjtu.edu.cn/~chengfan/
  - 8 TAs
- Q&A sessions（答疑）
  - Please contact the TAs through their email and make a face-to-face appointment with them
- Official website: https://oc.sjtu.edu.cn/courses/61620
  - 所有的课程相关的材料都可以找到，包括课件、教材、练习等等
- Course history
  - 秋季学期, 2017—2023
  - 2023年入选 上海交通大学一流本科课程

Checklist

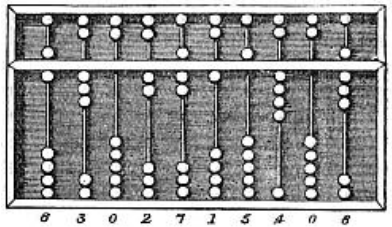
- Basic computer operations: emails, office, etc…
- Use search engines to find useful information
- Google, Stackoverflow, github, wikipedia
- Know how to control your computer via simple commands
- Write some simple programs
- Linux etc…

Survey QR Code

Introduction to Computation

计算机简史—古代

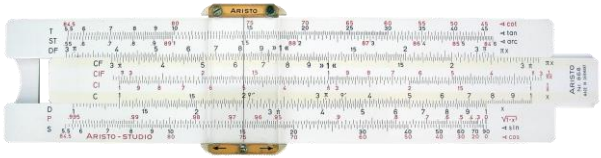
A computer is a machine that can be programmed to carry out sequences of arithmetic or logical operations (computation) automatically.



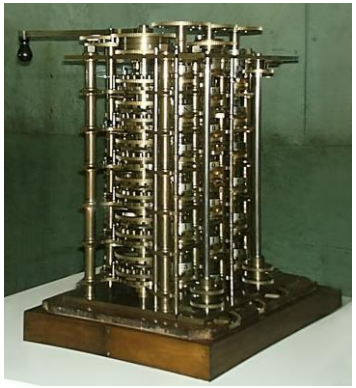
Abacus, Babylonia, 2400BCE

The Antikythera mechanism. Ancient Greece, 100 BCE.
安提基特拉机械: 计算天体在天空中的位置而制造的青铜机器

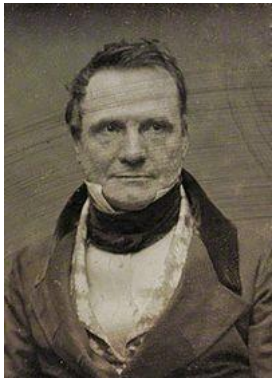
https://en.wikipedia.org/wiki/Computer



Slide rule for multiplication and division and for functions such as exponents, roots, logarithms, and trigonometry.



Babbage's Difference engine



Charles Babbage c. 1850

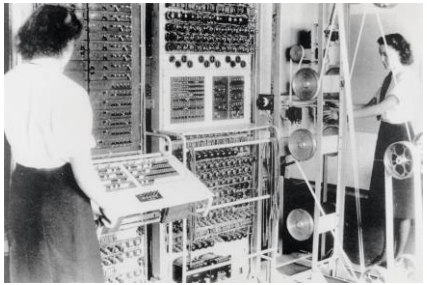
Considered the "father of the computer", he conceptualized and invented the first **mechanical computer** in the early 19th century.

差分机：巴贝奇设计计算机器的基本想法是利用“机器”将计算到印刷的过程全部自动化，全面去除人为疏失．差分机就是一台多项式求值机，只要将欲求多项式方程的前3个初始值输入到机器里，机器每运转一轮，就能产生出一个值来．用蒸汽机为动力，驱动大量的齿轮机构运转．未能完全建造完成！

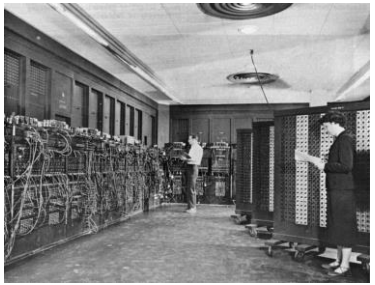
计算机简史—现代

Digital computers: Electromechanical, vacuum tubes, and digital electronic circuits

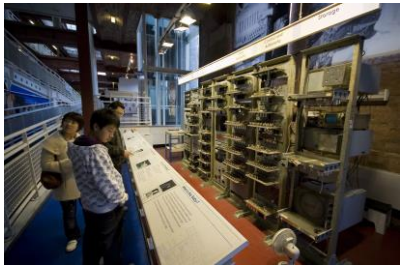
Replica of Konrad Zuse's Z3, the first fully automatic, digital (**electromechanical**) computer. (German, 1941)



Colossus, the first **electronic digital** programmable computing device, was used to break German ciphers during World War II. It is seen here in use at Bletchley Park in 1943.



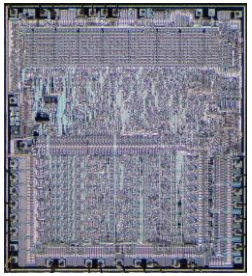
ENIAC (1946) was the first electronic, **Turing-complete device**, and performed ballistics trajectory calculations for the United States Army.



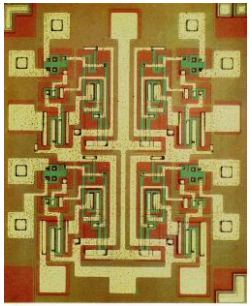
The University of Manchester's experimental **Transistor Computer** was first operational in November 1953 and it is widely believed to be the first transistor computer to come into operation anywhere in the world.

计算机简史—晶体管与集成电路

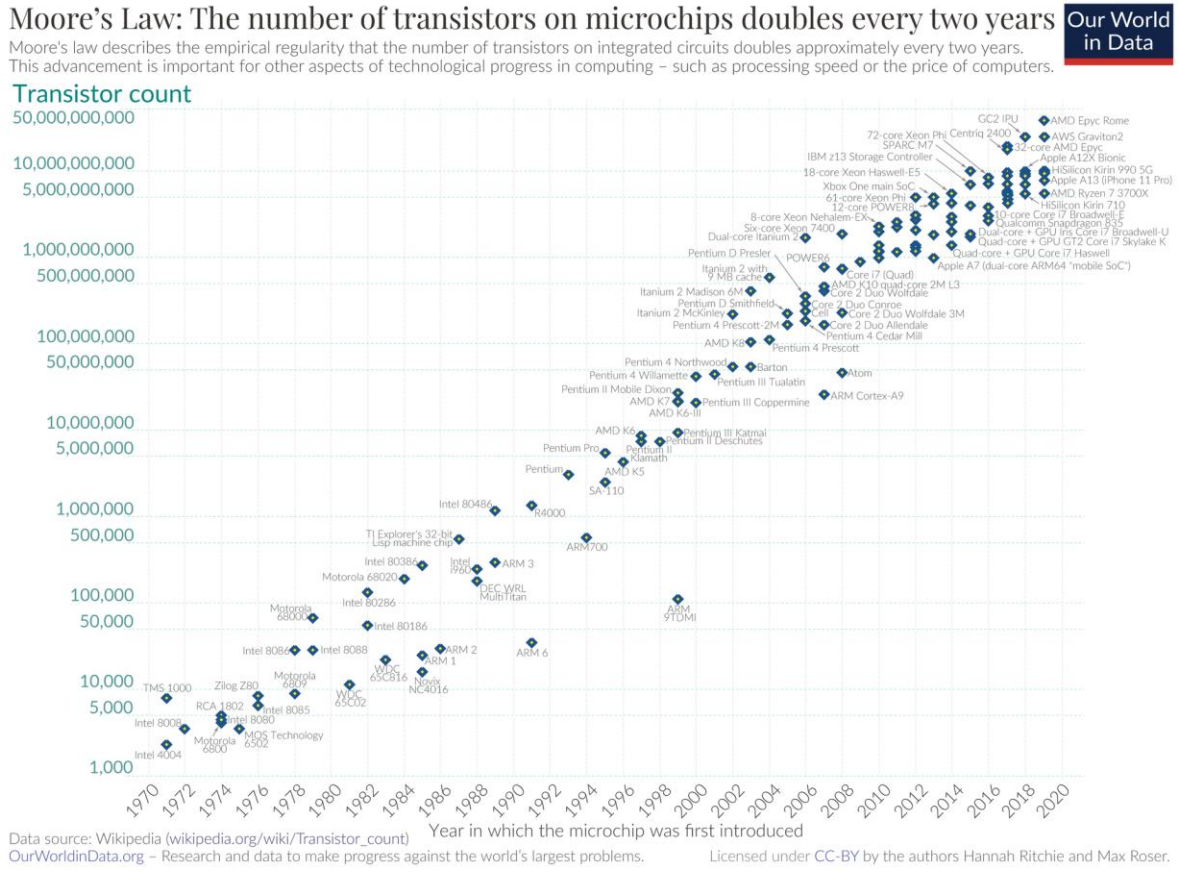
晶体管(Transistor)
AT&T's Bell Labs,
1947



MOS 6502:
3500 transistors
on a single chip,
1970



10μm – 1971
3nm – 2022



计算的力量——摩尔定律的启示

摩尔定律(Moore's law)：集成电路上可容纳的晶体管(transistor)数目，约每隔18-24个月便会增加一倍

1. 三种解读，性能、价格、尺寸
   - 集成电路上可容纳的晶体管数目，约每隔18个月便增加一倍。
   - 微处理器的性能每隔18个月提高一倍，或价格下降一半。
   - 相同价格所买的电脑，性能每隔18个月增加一倍。
2. 计算是科学理论转化为现实的必经之路
   - 科学范式：数学模型—解决方案—计算方法
   - 对于极其复杂的问题，推理难以奏效，通过计算可以提供一个解决方案（AI4Science）
   - 计算机科学还能再火20年 ☺
3. 熟练使用科学计算资源是现代大学生所必须具备的能力
   - 计算思维—每隔20年，1000倍计算能力的提升
   - 降维优势—对比牛顿、高斯等历史上的科学巨人
   - 并不仅仅是更多的工作机会 ☺

如果你反过来看摩尔定律，一个IT公司如果今天和18个月前卖掉同样多的、同样的产品，它的营业额就要降一半。IT界把它称为反摩尔定律。

Google前CEO Eric Schmidt

https://zh.wikipedia.org/zh-hans/%E6%91%A9%E5%B0%94%E5%AE%9A%E5%BE%8B

里程碑—人机大战的胜利

1997 Deep Blue（国际象棋）、2016 AlphaGo(围棋)

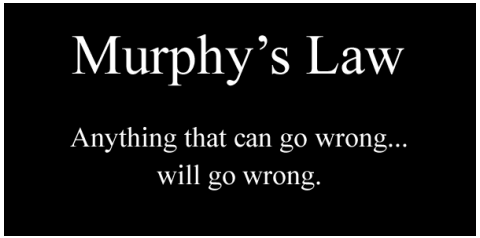
1997 Deep Blue (国际象棋)

2016 AlphaGo (围棋)

20年(1000倍) 以后??

1. How IBM’s Deep Blue Beat World Champion Chess Player Garry Kasparov (https://spectrum.ieee.org/how-ibms-deep-blue-beat-world-champion-chess-player-garry-kasparov)
2. AlphaGo (https://www.deepmind.com/research/highlighted-research/alphago)

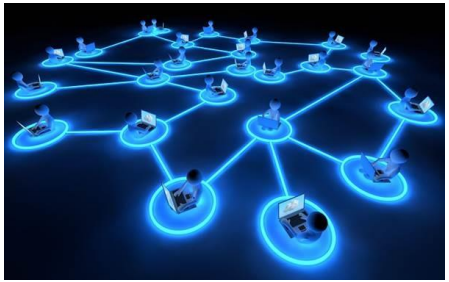
More Laws in Computation



墨菲定律（英语：Murphy's Law）

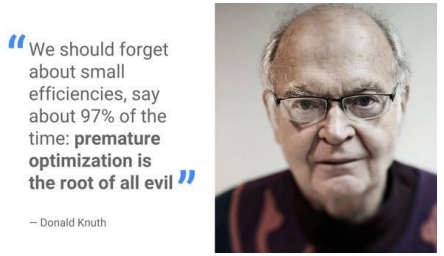
如果有多过一种方式去做某事，而其中一种方式将导致灾难，则必定有人会这样选择

“面包落地的时候，永远是抹黄油的一面着地。”



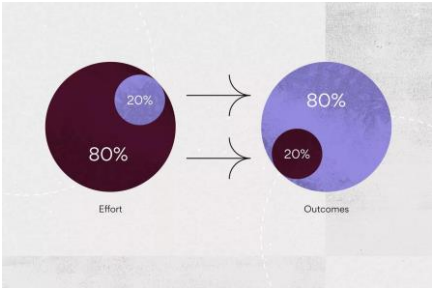
梅特卡夫定律（Metcalfe's law）

一个网络的价值等于该网络内的节点数的平方，而且该网络的价值与联网的用户数的平方成正比



Knuth 定律

“在（至少大部分）编程中，过早优化是万恶之源。”



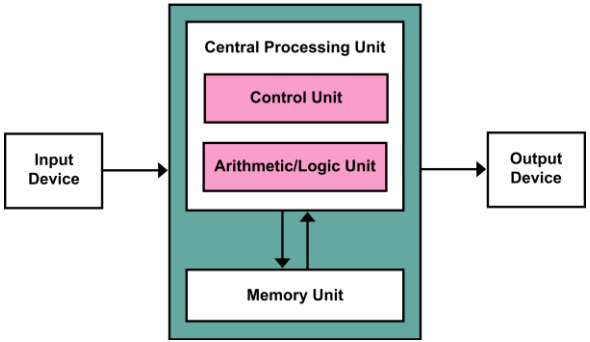
帕累托法则（Pareto principle, 二八法则）

约仅有20%的因素影响80%的结果

“20%的代码有80%的错误，只要找到它们并修复即可”

计算机架构:冯诺依曼体系

Process unit, Control unit, Memory, External mass storage, Input and output



The von Neumann architecture — also known as the von Neumann model or Princeton architecture — is a computer architecture based on a 1945 description by John von Neumann and others in the First Draft of a Report on the EDVAC.

John von Neumann (1903-1957) Hungarian-American mathematician, physicist, inventor, computer scientist, and polymath.

He made major contributions to a number of fields, including mathematics, physics, economics, computing (Von Neumann architecture, linear programming, self-replicating machines, stochastic computing), and statistics.

1. https://en.wikipedia.org/wiki/Von_Neumann_architecture
2. https://en.wikipedia.org/wiki/John_von_Neumann

Hardware

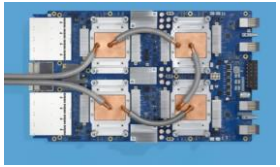
树莓派

Input

Output

Memory storage

Process unit



Software

Software is a set of instructions, data or programs used to operate computers and execute specific tasks.

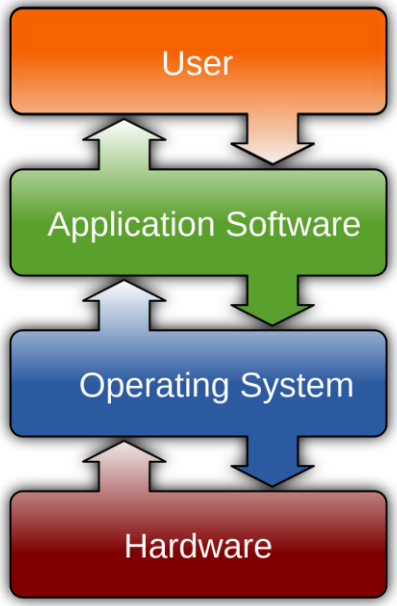
社会里面的各个组织：提供服务

OS

1. Unix
2. Mac OS
3. Windows
4. Linux
5. iOS
6. Android

Appl.

1. Text editor
2. Programming tool (C++, Python)
3. Communication tool: (wechat)
4. Games: lol, Warcraft
5. Movie player
6. Music player
7. Matlab
8. VS Code



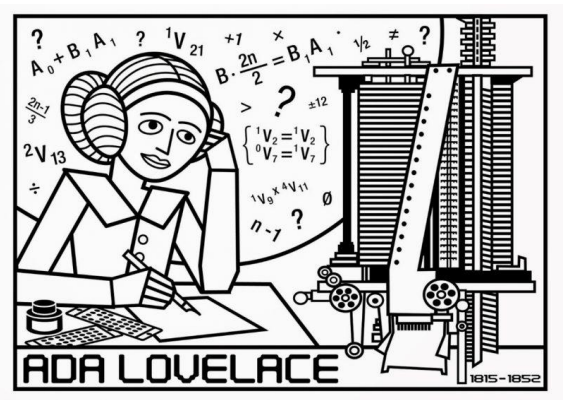
装机顺序

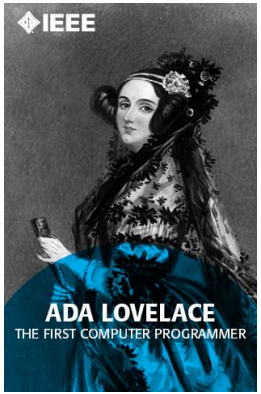
- 操作系统
  - Windows, macOS
- 开发环境
  - C++, Python
- 开发工具
  - vscode, xcode
- 应用(办公)工具
  - Office, qq
- 自定义软件
  - Hello world

https://en.wikipedia.org/wiki/Software

Ada Lovelace

English Computer Programmer, Mathematician (1815–1852). A gifted mathematician, Ada Lovelace is considered to have written instructions for the first computer program in the mid-1800s.





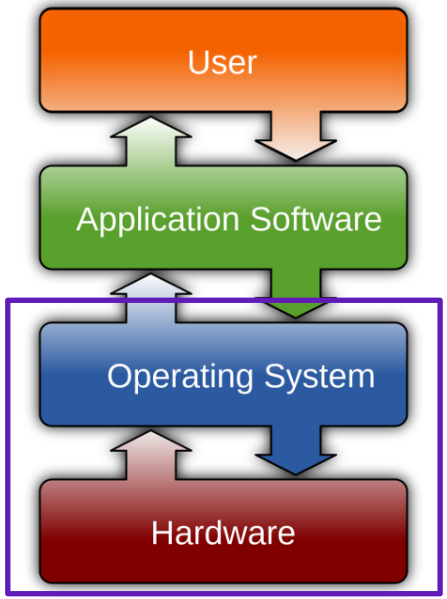
埃达・洛夫莱斯

Ada Lovelace championed **Charles Babbage's** work by, among other things, writing the first computer algorithm for his unbuilt Analytical Engine.

1. https://en.wikipedia.org/wiki/Ada_Lovelace
2. https://spectrum.ieee.org/charles-babbage-difference-engine
3. https://ieeexplore.ieee.org/document/7790972

Operating System

An operating system (OS) is system software that manages computer hardware and software resources and provides common services for computer programs.



操作系统：公共市政建设

A bridge between hardware and users

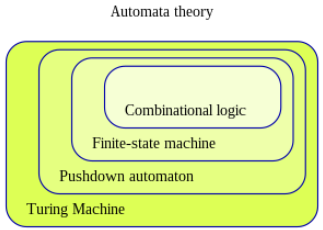
- 进程管理（Processing management）
- 内存管理（Memory management）
- 文件系统（File system）
- 网络通信（Networking）
- 安全机制（Security）
- 用户界面（User interface）
- 驱动程序（Device drivers）

- Unix系统
  - FreeBSD、OpenBSD
  - SUN的Solaris
  - **Linux**
    - Debian/Fedora/Ubuntu
    - Android
    - Chrome OS
- macOS
- windows

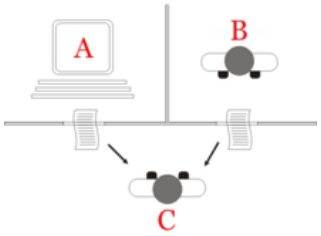
https://en.wikipedia.org/wiki/Operating_system

Alan Turing

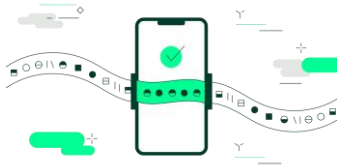
Alan Turing (1912-1954), was an English computer scientist, mathematician, logician, cryptanalyst, philosopher and theoretical biologist.



Turing Machine



Turing Test



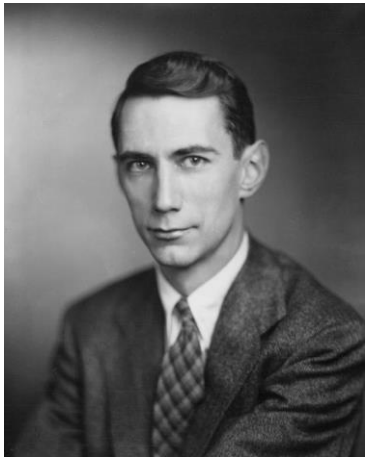
Turing Complete

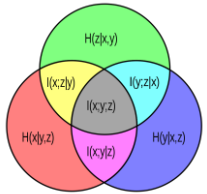
Turing Award

1. https://en.wikipedia.org/wiki/Alan_Turing
2. https://en.wikipedia.org/wiki/Turing_machine
3. https://en.wikipedia.org/wiki/Turing_test
4. https://en.wikipedia.org/wiki/Turing_completeness

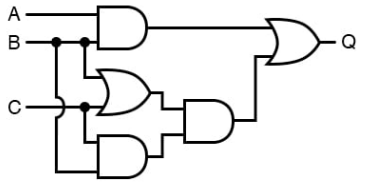
Claude Shannon

Claude Elwood Shannon (April 30, 1916 – February 24, 2001) was an American mathematician, electrical engineer, and cryptographer known as “the father of information theory”

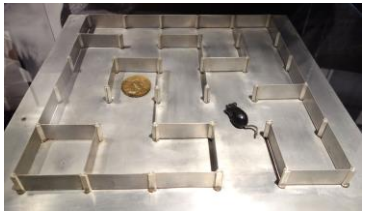




Information Theory



Logic Circuits



Artificial Intelligence

Cryptography

https://en.wikipedia.org/wiki/Claude_Shannon

Kurt Gödel

Kurt Friedrich Gödel was an Austrian, and later American, logician, mathematician, and philosopher. Gödel Prize is an annual prize for outstanding papers in the area of theoretical computer science.

Gödel's Incompleteness Theorem

任何逻辑自洽的形式系统，只要蕴涵皮亚诺算术公理，它就不能用于证明其本身的自洽性。

哥德尔不完备定理破坏了希尔伯特计划的哲学企图。大卫·希尔伯特提出，像实分析那样较为复杂的体系的兼容性，可以用较为简单的体系中的手段来证明。最终，全部数学的兼容性都可以归结为基本算术的兼容性。但哥德尔的第二条定理证明了基本算术的兼容性不能在自身内部证明，因此当然就不能用来证明比它更强的系统的兼容性了。

Who are here?

https://en.wikipedia.org/wiki/Kurt_G%C3%B6del

And also, You generation in the future

https://en.wikipedia.org/wiki/Turing_Award

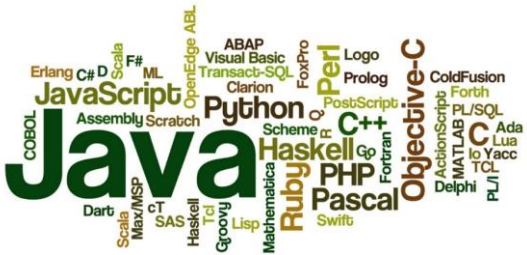
https://en.wikipedia.org/wiki/Claude_E._Shannon_Award

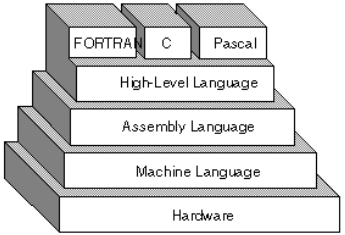
https://en.wikipedia.org/wiki/Fields_Medal

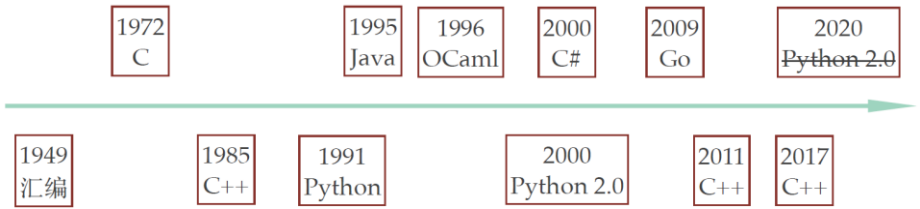
https://www.nobelprize.org/

Introduction to Python

Programming Language: Brief History







语言本质: 人和机器的沟通的桥梁
发展趋势：距离机器越来越远，距离人越来越近 、易学
语言之间：与时俱进、互相学习、吸取经验、相似
未来：人性化、容易使用、开发与运行速度极大提升

Best Programming Language

Wars: Best Programming language / text editor / IDE / programmer

- How critical is performance? Is it acceptable to write a slow solution quickly?
- What kinds of libraries would I need?
- What's the platform? OS? Web? Mobile devices?
- What's the nature of the problem? Regex-type string processing? Mathematical with Matlab/Maple/etc?
- Graphics? Text?
- How reusable do I want the solution to be?

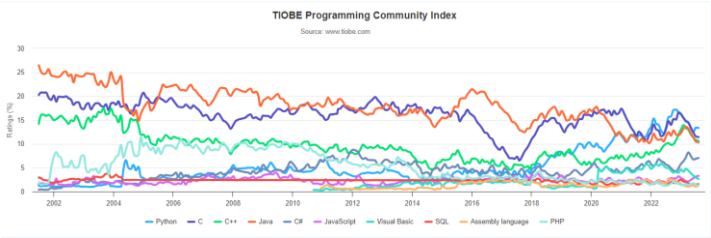
1. 熟练掌握2-3门语言，根据需要选择合适的语言
2. 现代的语言有很多相似的语法，学一门新语言代价很低
3. 大学阶段可能会涉及到 Python, C++, C, Java, Go, HTML, JS, Shell, Latex

https://stackoverflow.com/questions/3017363/picking-the-right-language

Why Python?

未来10年内（2016预测）

Github Language Stats



TIOBE Index for August 2023

- ✓ 实用性：未来学习中会广泛使用，支持各种任务，快速入手，不重复造轮子 (对比C)
- ✓ 初学者友好：对初学者要友好，学习曲线不要太陡，细节不要太多 (对比C, C++, Java)
- ✓ 开放性：开源语言、不受商业公司控制 (Java, C#)

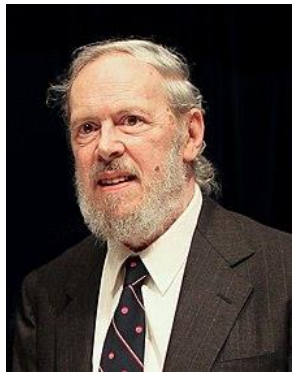
Learn how to program! Not Programming Language

1. https://madnight.github.io/githut/#/pull_requests/2023/2
2. https://www.tiobe.com/tiobe-index/

Brief History of C

C is a general purpose, procedural computer programming language supporting structured programming, lexical variable scope, and recursion, while a static type system prevents unintended operations. By design, C provides constructs that map efficiently to typical machine instructions and has found lasting use in applications previously coded in assembly language. Such applications include operating systems and various application software for computers, from supercomputers to embedded systems.

C was originally developed at Bell Labs by Dennis Ritchie between 1972 and 1973 to make utilities running on Unix. Later, it was applied to re-implementing the kernel of the Unix operating system. During the 1980s, C gradually gained popularity. Nowadays, it is one of the most widely used programming languages, with C compilers from various vendors available for the majority of existing computer architectures and operating systems. C has been standardized by the ANSI since 1989 (see ANSI C) and by the International Organization for Standardization.



Dennis Ritchie

Brief History of C++

C++ is a general-purpose programming language created by Bjarne Stroustrup as an extension of the C programming language, or **"C with Classes".** The language has expanded significantly over time, and modern C++ has object-oriented, generic, and function features in addition to facilities for low-level memory manipulation. It is almost always implemented as a compiled language, and many vendors provide C++ compilers, including the Free Software Foundation, LLVM, Microsoft, Intel, and IBM, so it is available on many platforms.

C++ was designed with a bias toward system programming and embedded, resource-constrained software and large systems, with performance, efficiency and flexibility of use as its design highlights. C++ has also been found useful in many other contexts, with key strengths being software infrastructure and resource-constrained applications, including desktop applications, servers (e.g. e-commerce, Web search or SQL servers), and performance-critical applications (e.g. telephone switches or space probes).

Bjarne Stroustrup

Programming Language: Hello World

Machine Language

```
01001000 01100101
01101100 01101100
01101111 00100000
01010111 01101111
01110010 01101100
01100100
```

Assembly Language

```
section     .text
global      _start

_start:
    mov     edx,len
    mov     ecx,msg
    mov     ebx,1
    mov     eax,4
    int     0x80
    mov     eax,1
    int     0x80

section     .data

msg     db  'Hello, World!', 0xa
len     equ $ - msg
```

C

```
#include<stdio.h>
printf("Hello World");
```

C++

```
#include<iostream>
using namespace std;

cout << "Hello World" << endl;
```

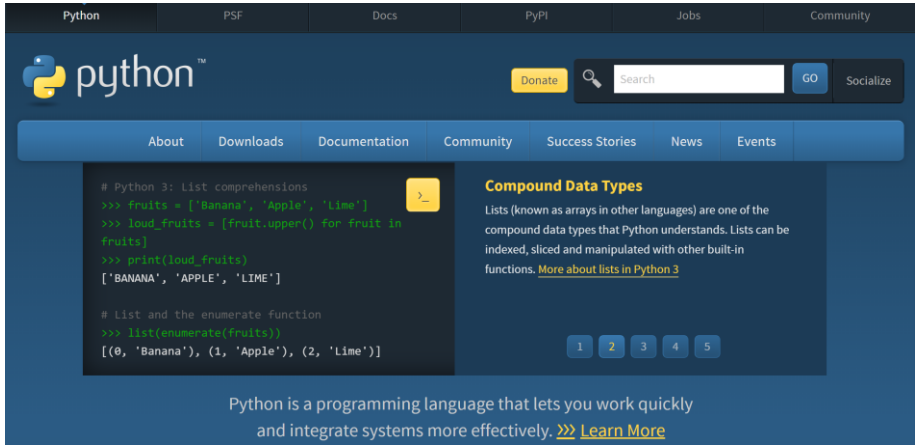
Python

```
print("Hello World")
```

灵魂拷问：选汇编还是选python？
本科阶段最广泛使用的语言

Python Homepage

Welcome to Python: **https://www.python.org/**

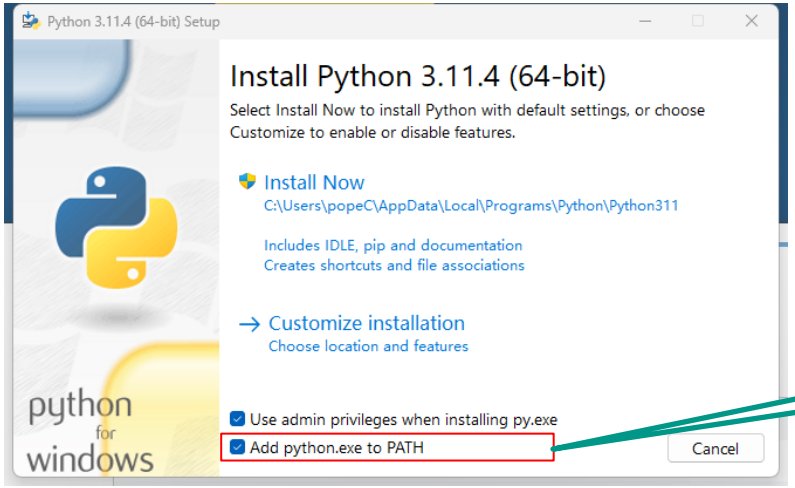


最新版3. 11， 各种平台

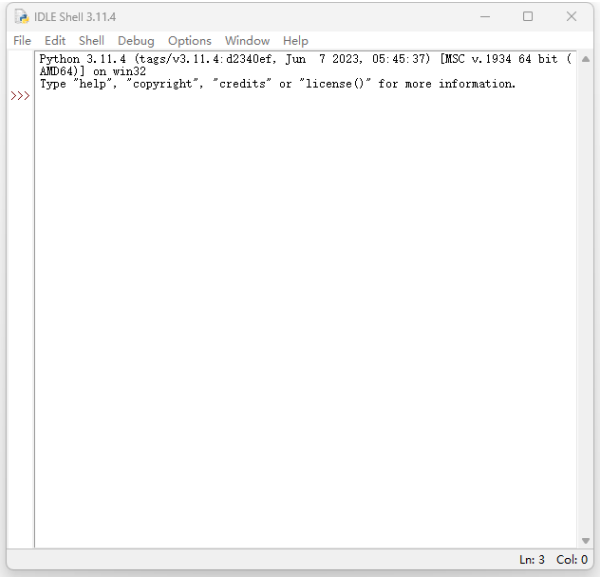
Python: Setup

具体安装攻略，请参考OC附件

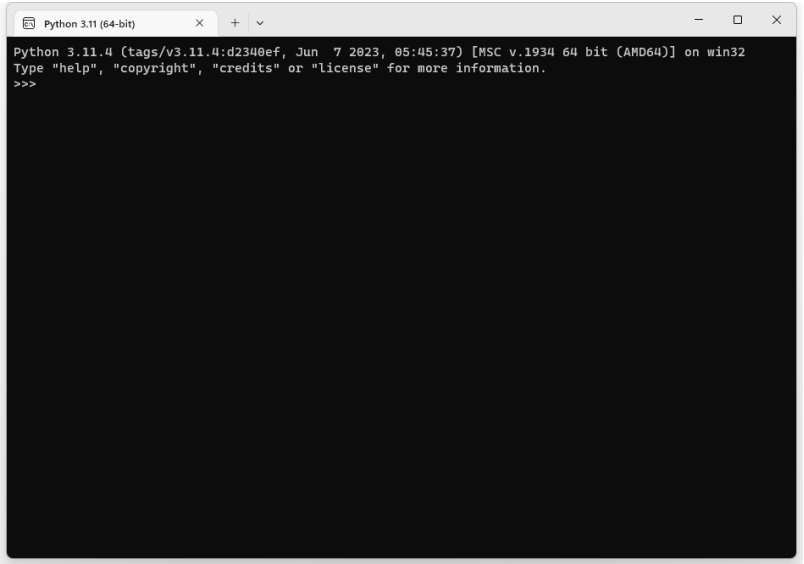
- Homepage->Downloads->python3.11.4
  - If you use mac/linux, please choose the corresponding version with 64 bits.
  - You may choose python 3.11 or python 3.10, but it should be 3.x
  - When install it, do remember to choose "**Add python.exe to PATH**"



Python: Setup



IDLE



Console (控制台)

After >>>, type:
print("Hello world")

IDLE和Console仅供测试，
不推荐实际使用

Hello world

- **Hello World** is a simple program that, when run, displays the message: Hello World. Fittingly, the Hello World program has long been the new programmer's induction into a myriad of programming languages. A project at the Association for Computing Machinery (ACM) of Louisiana Tech collected all the Hello World program examples they could find -- and discovered 204.
- No one knows exactly why Hello World has stood the test of time so well. The most likely historical explanation is simply that a short program like Hello World once allowed the programmer to make sure that a language's compiler, development environment, and run-time environment were **correctly installed**. Because this involved a lot of work, a very simple program was used first to test things out.
- According to Wikipedia, the tradition of using the phrase "Hello, world" as the test message was influenced by an example program in the book **"The C Programming Language",** by Brian Kernighan and Dennis Ritchie.

Hello world：环境安装正确

某程序员对书法十分感兴趣，退休后决定在这方面有所建树。于是花重金购买了上等的文房四宝。

一日，饭后突生雅兴，一番磨墨拟纸，并点上了上好的檀香，颇有王羲之风范，又具颜真卿气势，定神片刻，泼墨挥毫，郑重地写下一行字：hello world

（冷笑话—注意保暖）

程序员练书法？

“Hello world”

- print(x): 输出 x
- Try to type the following
  - print(“Shanghai Jiao Tong University”)
  - print(“Hello SJTU! Hello IEEE!”)
  - print(“中华人民共和国”)
  - print(“上海交通大学”)
  - print(“上海市闵行区东川路800号”)
  - print(“苟利国家生死以”)
  - print(“1+2+3=6”)
- What if we change “” to ‘’
- Be Cautious
  - “” should not be ignored
  - “” should under English mode, not Chinese mode
  - Syntax Error: invalid character in identifier
  - 语法错误，常见错误
- print(1+2+3+4)
- print(2*3/5)

```
Shanghai Jiao Tong University
Hello SJTU! Hello IEEE!
中华人民共和国
上海交通大学
上海市闵行区东川路800号
苟利国家生死以
1+2+3=6
```

```
>>> print(“Hello world”)
  File "<stdin>", line 1
    print(“Hello world”)
                 ^
SyntaxError: invalid character in identifier
>>>
```

Zen (禅) of Python

>>> import this

- Beautiful is better than ugly.
- Explicit is better than implicit.
- Simple is better than complex.
- Complex is better than complicated.
- Flat is better than nested.
- Sparse is better than dense.
- Readability counts.
- Special cases aren't special enough to break the rules.
- Although practicality beats purity.
- Errors should never pass silently.
- Unless explicitly silenced.
- In the face of ambiguity, refuse the temptation to guess.
- There should be one-- and preferably only one --obvious way to do it.
- Although that way may not be obvious at first unless you're Dutch.
- Now is better than never.
- Although never is often better than *right* now.
- If the implementation is hard to explain, it's a bad idea.
- If the implementation is easy to explain, it may be a good idea.
- Namespaces are one honking great idea -- let's do more of those!

Life is short, use python

Introduction to Python

What is Python Programming Language?

- Python is a **general-purpose high-level** programming language whose design philosophy emphasizes code readability.
- Python supports **multiple programming paradigms**, primarily but not limited to object oriented, imperative and, to a lesser extent, functional programming styles.
- Python aims to combine "remarkable power with very clear syntax", and its standard **library** is large and **comprehensive**.
- Its use of **indentation** (缩进格式) for block delimiters (块分隔符) is unusual among popular programming languages.

Development of Python

Python was conceived in the late 1980s and its implementation was started in December 1989 by Guido van Rossum at CWI (Centrum Wiskunde & Informatica) in the Netherlands. Note: 荷兰国家数学和计算机科学研究所

Guido van Rossum

python™

- Python 2.0, 2000
- Python 3.0, 2008

This Python license plate belongs to Guido van Rossum, creator of the Python language.

https://gvanrossum.github.io/

Benefit of Python

C语言之父--丹尼斯·里奇

Java语言之父--詹姆斯·高斯林

吉多・范罗苏姆

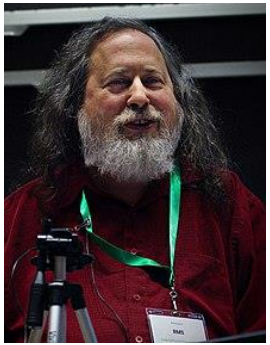
C++语言之父--
本贾尼·斯特劳斯特卢普

PHP之父——拉斯马斯·勒德尔夫

Life is short, use Python

Open Source

Open-source software (OSS) is a type of computer software whose source code is released under a license in which the copyright holder grants users the rights to study, change, and distribute the software to anyone and for any purpose



Richard Matthew Stallman
(born March 16, 1953)

- In the 1960s, companies make profits for hardware and software were provided for free. But as hardware prices continue to fall and profits from selling hardware shrink, manufacturers begin to look to software to bring in additional revenue.
- Stallman launched the GNU Project in September 1983 to write a Unix-like computer operating system composed entirely of free software. With this, he also launched the free software movement. He has been the GNU project's lead architect and organizer and developed a number of pieces of widely used GNU software including, among others, the GNU Compiler Collection, GNU Debugger, and GNU Emacs text editor.
- Stallman pioneered the concept of copyleft (反版权), which uses the principles of copyright law to preserve the right to use, modify, and distribute free software. He is the main author of free software licenses which describe those terms, most notably the GNU General Public License (GPL), the most widely used free software license.

https://en.wikipedia.org/wiki/Richard_Stallman

Open Source

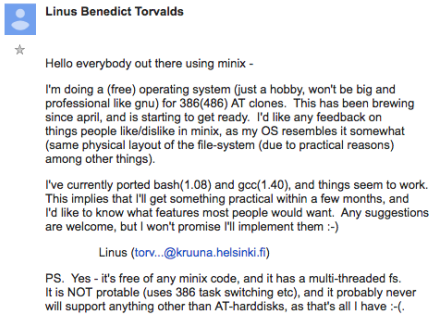
- Open-source software licensing:
  - Examples of free software license / open-source licenses include Apache License, BSD license, GNU General Public License, GNU Lesser General Public License, MIT License, Eclipse Public License and Mozilla Public License.
  - GNU is a recursive acronym for "GNU's Not Unix!",
- Open source is not free!
  - Nearly all open source software is free software, but there are exceptions.
  - First, some open source licenses are too restrictive, so they do not qualify as free licenses. For example, “Open Watcom” is nonfree because its license does not allow making a modified version and using it privately.
- Why Open Source Misses the Point of Free Software - GNU Project ...
  - https://www.gnu.org/philosophy/open-source-misses-the-point.en.html
- Open source for undergraduates: **Learn from the most experienced people in software**

GNU

软件需要运行操作系统之上，但是，创造一个自由的操作系统的所必不可少的部分——自由的内核，却还没有开发出来

Linus Torvalds

I'm doing a (free) operating system (just a hobby, won't be big and professional like gnu) for 386(486) AT clones.
–Torvalds, Linus (1991-08-25).



Linus Benedict Torvalds

Hello everybody out there using minix -

I'm doing a (free) operating system (just a hobby, won't be big and professional like gnu) for 386(486) AT clones. This has been brewing since april, and is starting to get ready. I'd like any feedback on things people like/dislike in minix, as my OS resembles it somewhat (same physical layout of the file-system (due to practical reasons) among other things).

I've currently ported bash(1.08) and gcc(1.40), and things seem to work. This implies that I'll get something practical within a few months, and I'd like to know what features most people would want. Any suggestions are welcome, but I won't promise I'll implement them :-)

Linus (torv...@kruuna.helsinki.fi)

PS. Yes - it's free of any minix code, and it has a multi-threaded fs. It is NOT protable (uses 386 task switching etc), and it probably never will support anything other than AT-harddisks, as that's all I have :-(.

- Linus Benedict Torvalds (born December 28, 1969) is a Finnish-American software engineer who is the creator and, historically, the principal developer of the Linux kernel, which is the kernel for Linux operating systems (distributions) and other operating systems such as **Android** and Chrome OS.
- He also created the distributed version control system **Git** and the scuba dive logging and planning software Subsurface.
- Because of the dominance of the Linux-based Android on smartphones, Linux, including Android, has the **largest installed** base of all general-purpose operating systems, as of May 2022.

https://en.wikipedia.org/wiki/Linus_Torvalds

Have Questions?

Google, stackoverflow, github, Wikipedia, python.org

Python Evolvement

- Python 2.0 was released on 16 October 2000, with many major new features including a full garbage collector and support for Unicode.
- Python 3.0, a major, backwards-incompatible (向后不兼容) release with Python 2.x, was released on 3 December 2008 after a long period of testing.
- Many of its major features have been backported (相反移植) to the backwards-compatible Python 2.7.
- Now the latest version is 3.6/3.7, which will be adopted in this course
- On January 1, 2020, the 2.x branch of the Python programming language will no longer be supported by its creators, the Python Software Foundation.

Python 2.x已经被抛弃，务必使用3.x

Python: Advantage

- It‘s Object-Oriented: Python is an object-oriented language, from the ground up.
- It‘s Free: Python is freeware — something which has lately been come to be called open source software.
- It‘s Portable (易于移植的): Python is written in portable ANSI C, and compiles and runs on virtually every major platform in use today.
- It‘s Powerful: Its tool set places it between traditional scripting languages (脚本语言) (such as Tcl, Scheme, and Perl), and systems languages (such as C, C++, and Java).
- It‘s Mixable (可混合的): Python programs can be easily "glued" to components written in other languages.
- It's Easy to Use: as with other interpreted languages, Python executes programs immediately, which makes for both an interactive programming experience and rapid turnaround (周转时间) after program changes.
- It's Easy to Learn: In fact, you can expect to be coding significant Python programs in a matter of days (and perhaps in just hours, if you're already an experienced programmer).

Life is short, you need python

IDE: 集成开发环境

- An integrated development environment (IDE) is a software application that provides comprehensive facilities to computer programmers for software development.
  - VS Code, Visual studio, Eclipse, IDLE, VS code, Pycharm, Atom, ect.
- An IDE normally consists of a source code editor, build automation tools, and a debugger. Most modern IDEs have intelligent code completion.
  - Some IDEs, such as NetBeans and Eclipse, contain a compiler, interpreter, or both; others, such as SharpDevelop and Lazarus, do not.
- Sometimes a version control system, or various tools to simplify the construction of a graphical user interface (GUI), are integrated.
- Many modern IDEs also have a class browser, an object browser, and a class hierarchy diagram, for use in object-oriented software development.

组装机 和 一体机 的区别

https://en.wikipedia.org/wiki/Integrated_development_environment

Common Python IDE

课程推荐

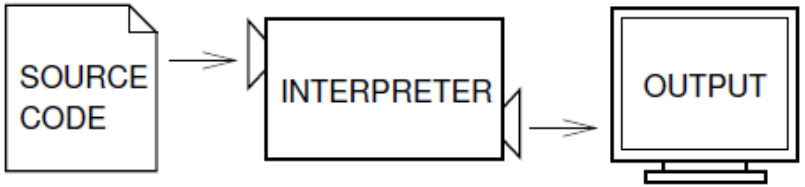
- Visual Studio Code (VS Code) 推荐，微软大力支持
- PyCharm: free for education, best for python
  - https://www.jetbrains.com/pycharm/
  - The free version for students, use your edu email: https://www.jetbrains.com/student/

其他选择

- Sublime text + default python3
- Anaconda: https://www.anaconda.com/
- Linux: Emacs, Vim ect
- Text editor: sublime text, notepad, notepad++, ect.

VS code: 通用性好，C++等等
PyCharm: 写python最专业
安装攻略：见OC

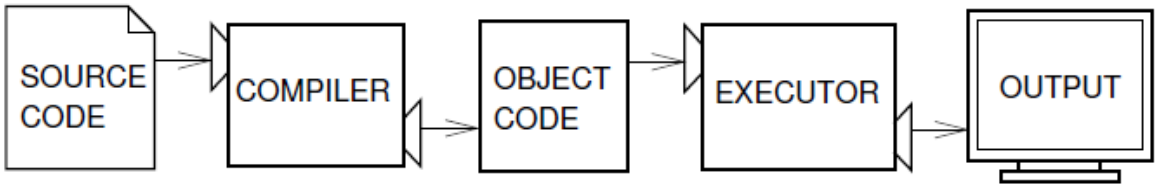
Python Interpreter (解释器)



- In computer science, an interpreter is a computer program that directly executes, i.e. performs, instructions written in a programming or scripting language, without requiring them previously to have been compiled into a machine language program. An interpreter generally uses one of the following strategies for program execution:
- parse the source code and perform its behavior directly;
- translate source code into some efficient intermediate representation and immediately execute this;
- explicitly execute stored precompiled code made by a compiler which is part of the interpreter system.

https://en.wikipedia.org/wiki/Interpreter_(computing)

Compiler 编译器



- A compiler is computer software that transforms computer code written in one programming language (the source language) into another programming language (the target language). Compilers are a type of translator that support digital devices, primarily computers. The name compiler is primarily used for programs that translate source code from a high-level programming language to a lower level language (e.g., assembly language, object code, or machine code) to create an executable program

One by one VS. one pass

https://en.wikipedia.org/wiki/Compiler

Documentation (文档)

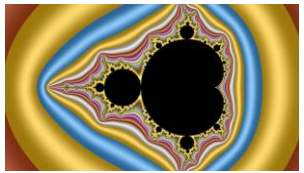
Python documentation：
Tutorial, Library Reference, Language Reference etc. for Python 3 can be found in the following website: https://docs.python.org/3/

- Software documentation is written text or illustration that accompanies computer software or is embedded in the source code. It either explains how it operates or how to use it, and may mean different things to people in different roles.
- Documentation is an important part of software engineering. Types of documentation include:
  - Requirements – Statements that identify attributes, capabilities, characteristics, or qualities of a system. This is the foundation for what will be or has been implemented.
  - Architecture/Design – Overview of software. Includes relations to an environment and construction principles to be used in design of software components.
  - Technical – Documentation of code, algorithms, interfaces, and APIs.
  - End user – Manuals for the end-user, system administrators and support staff.
  - Marketing – How to market the product and analysis of the market demand.

When you need help, turn to documentation

A complicated example

A fractal (分形) is a never-ending pattern. Fractals are infinitely complex patterns that are self-similar across different scales. They are created by repeating a simple process over and over in an ongoing feedback loop.



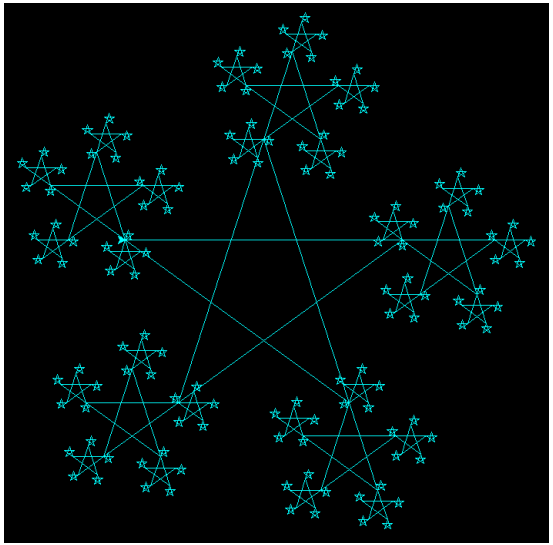
```
from turtle import *
import turtle

tur = turtle.Turtle()
tur.speed(6)
tur.getscreen().bgcolor("black")
tur.color("cyan")
tur.penup()
tur.goto((-200, 50))
tur.pendown()

def star(turtle, size):
    if size <= 10:
        return
    else:
        for i in range(5):
            turtle.forward(size)
            star(turtle, size/3)
            turtle.left(216)

star(tur, 360)
turtle.done()
```

Copy and paste in IDLE or VS code



https://pythonguides.com/fractal-python-turtle/

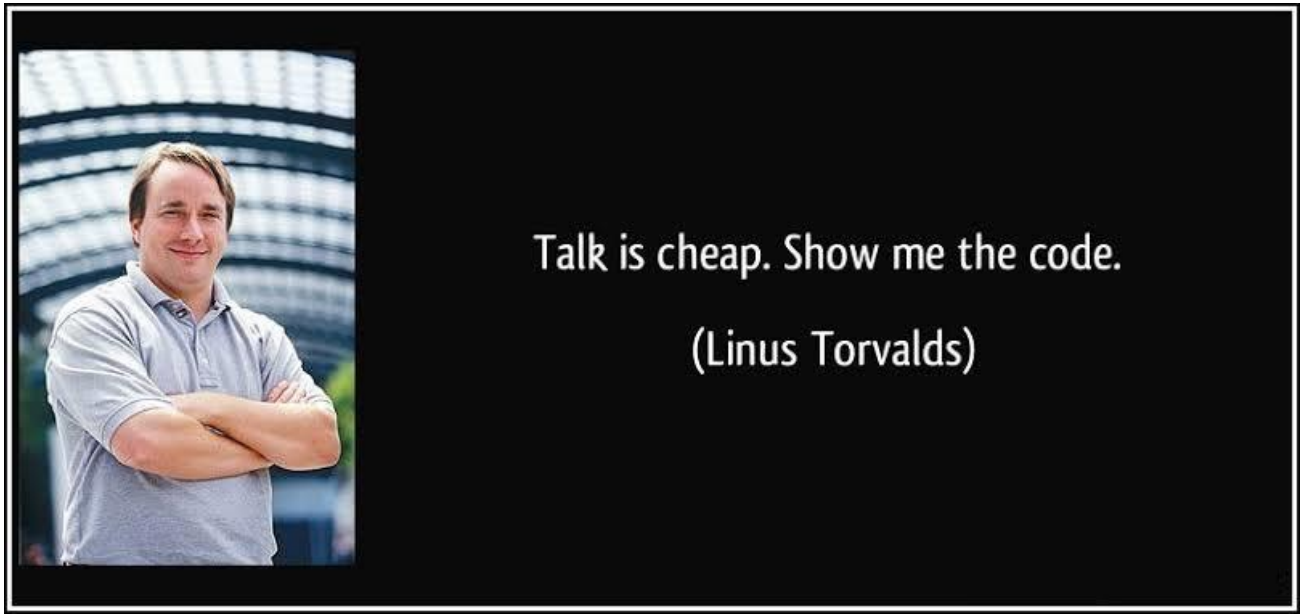
Game In Python

An example to play games in Python:

pip install pygame

To see if it works, run one of the included examples:

python -m pygame.examples.aliens



Ready For Coding