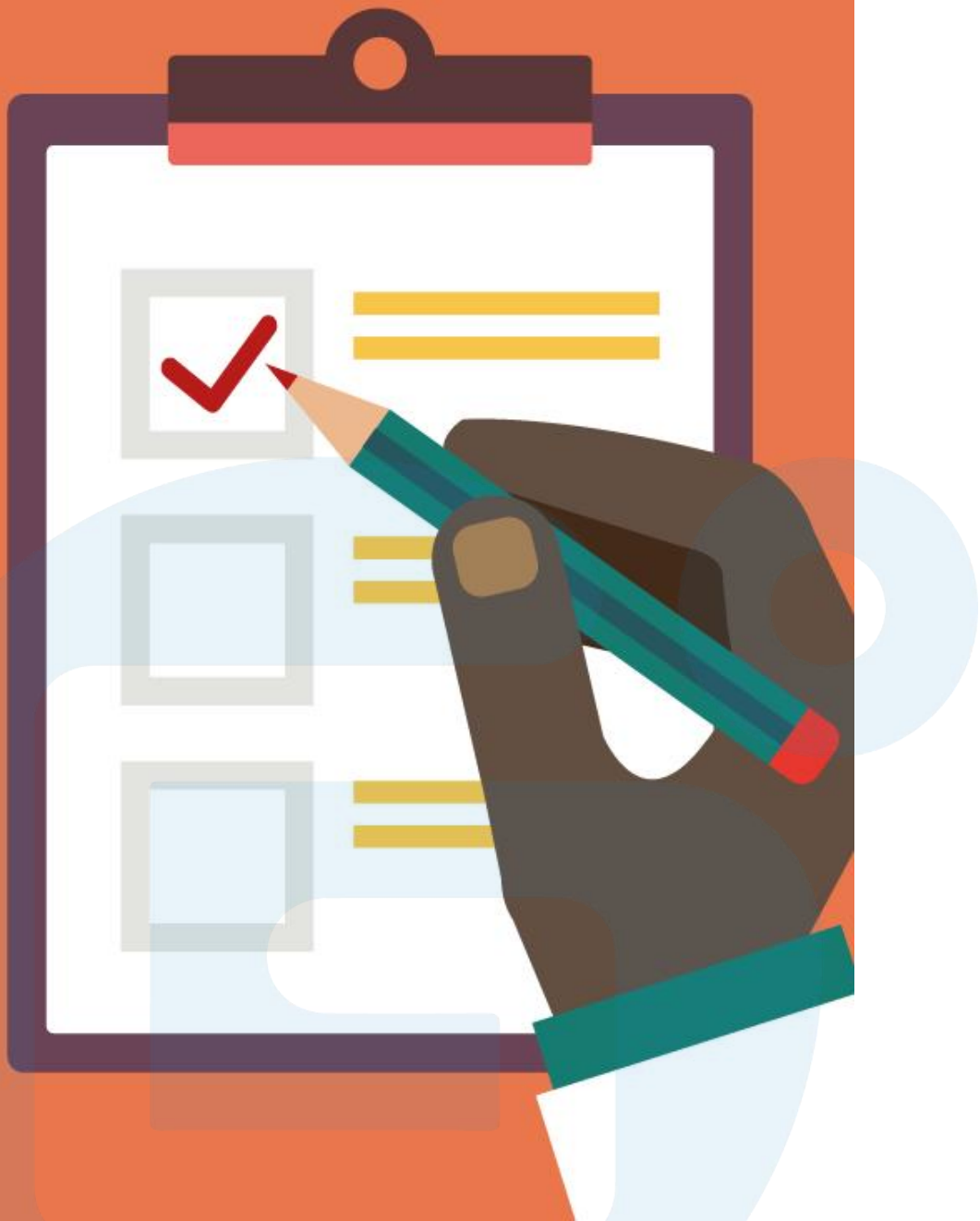


Talk is cheap, show me the code

第一课：Python 编程导论

Python初阶入门课程系列



OUTLINE

➤ 零基础编程入门

➤ Python背景知识

➤ 开发环境设置

➤ 课后作业

ME 常老师



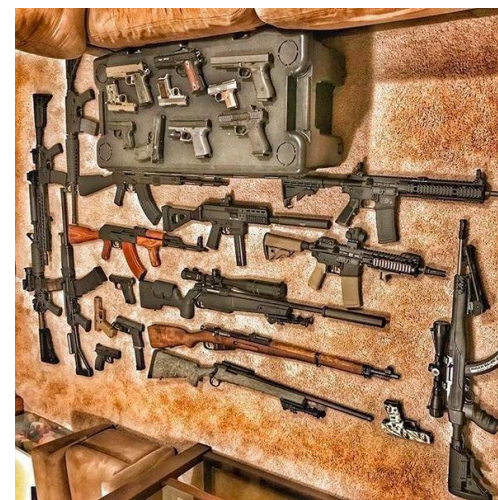
Talk is cheap, show me the code

- 北京航空航天大学 Ph.D
- 中科院高能物理研究所 北京正负电子对撞机国家实验室 副研究员
主要研究方向为计算物理，数据科学等，十年编程经验，代码被用于多项国防科技工程和国家重大基础科技设施中。
- 主持国家自然科学基金3项，发表SCI论文多篇。

计算机语言是一门工具，也是认识世界改造世界的武器。

能满足要求的趁手武器，就是好武器。

先耍起来！





一 零基础编程入门



课程目标

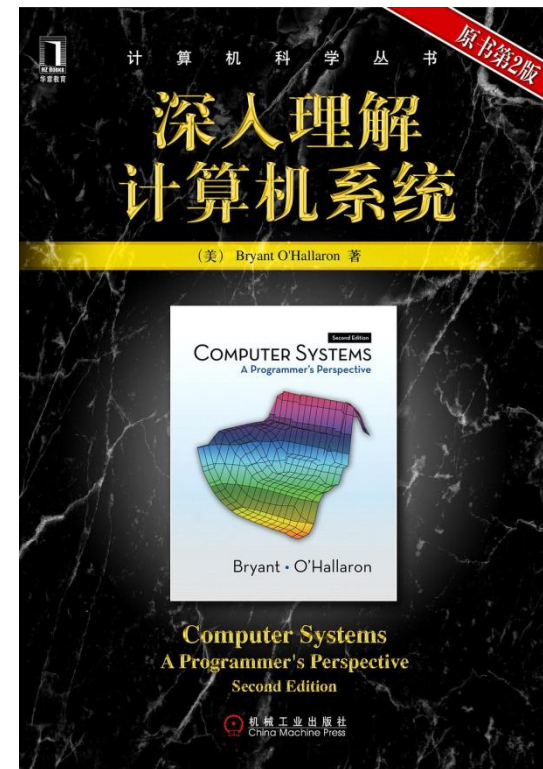
通过8天，16个课时的课程，令大家基本掌握Python的基础知识，能利用Python语言做一些小型的应用。
在实践中感受编程的乐趣和魅力。

课程内容

课程阶段	课时主题	实践课内容
基础入门班	1.理论：Python与编程导论	
	2.实践：乌龟爬图与调库	利用python绘图
	3.理论：数值与函数	
	4.实践：编程求解数学问题	python解复杂方程
	5.理论：三种基本控制流程结构	
	6.实践：编程算法的实现	几种基本算法的python实现
	7.理论：四种基本容器（数据结构）	
	8.实践：统计学应用	对托福高频词汇的分析

计算机语言，程序和编程

1. 计算机是什么？
2. 计算机程序是什么？
3. 计算机程序被拿来做什么？
4. 计算机语言又是什么？
5. 编程？



《深入理解计算机系统》
机械工业出版社
(美)布赖恩特 (Bryant,R.E.) 2016

计算机语言，程序和编程

1. 计算机是什么？ 🤔

按照程序运行，自动、高速处理海量数据的电子设备。由软硬件组成，可以进行数值计算和逻辑计算，还具有存储记忆功能。但未必以PC的形式呈现。



零基础编程入门

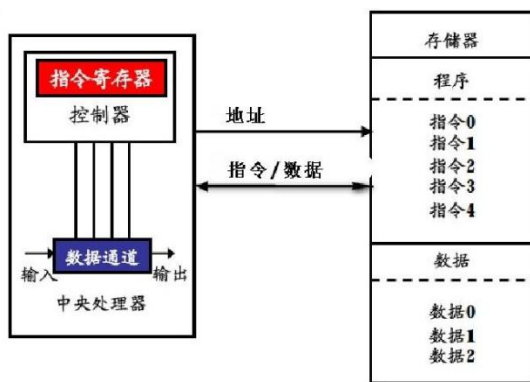
计算机语言，程序和编程

1. 计算机是什么？

基本构架：**冯·诺依曼架构**和**哈佛架构**

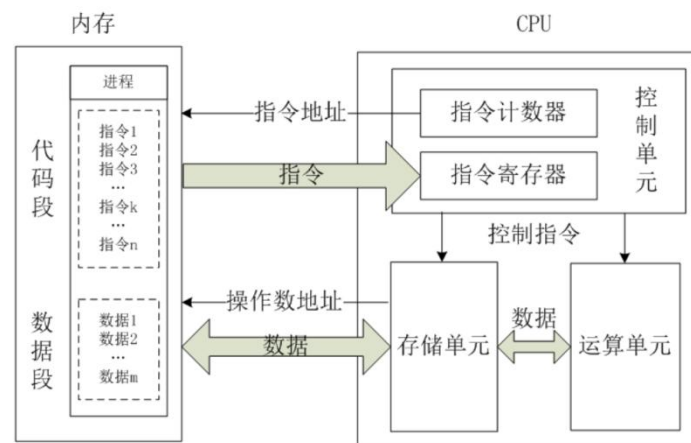
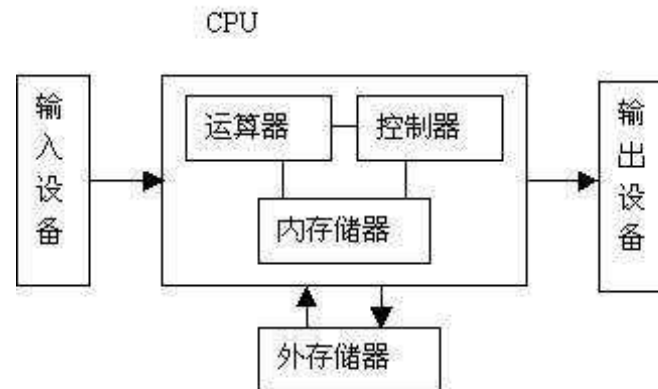
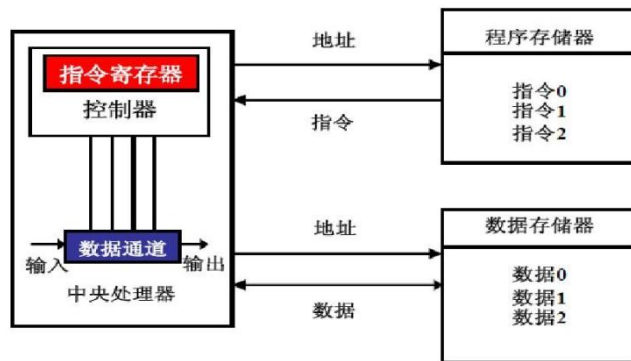
冯诺依曼结构

单一存储、统一编址、分时复用



哈佛结构

分开存储、独立编址、两倍带宽、执行效率更高

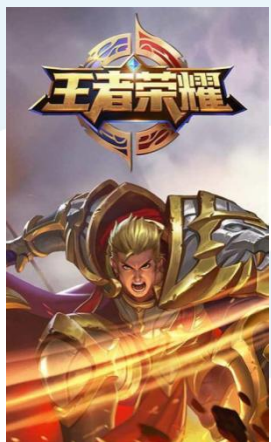


计算机语言，程序和编程

2. 计算机程序是什么？

俗称“软件”，代码化的指令序列，可以被硬件执行，通常是为了实现特定的目标。

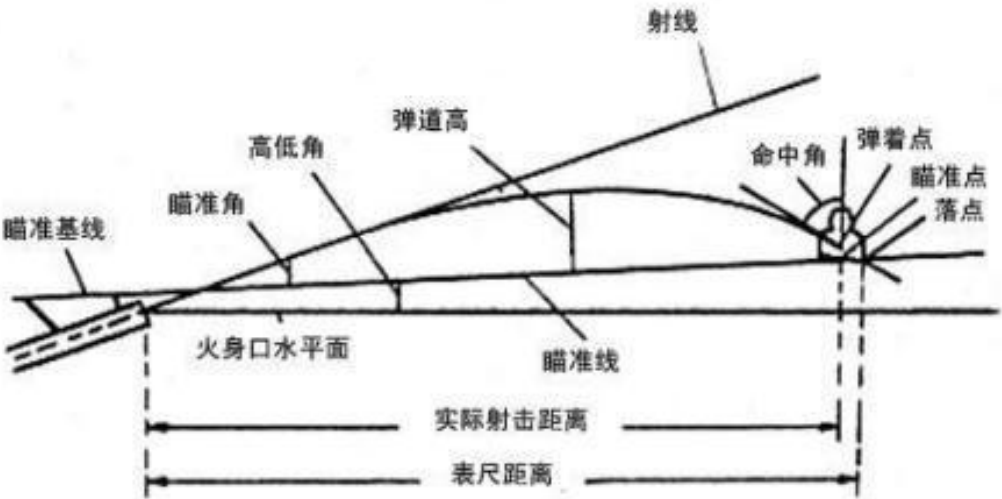
一串逻辑指令 + 一堆数据 = 程序



计算机语言，程序和编程

3. 计算机程序被拿来做什么？

最开始是用于配合最初的计算机ENIAC来做数值计算（求解大规模微分方程组，用来计算弹道射表）。1946



诸元	计算值	参考值	相对误差/%
X/m	30 088	30 166	0.26
T/s	99.2	99.4	0.20
Y/m	11 980	12 030	0.41
$\theta_c/(\text{°})$	66.3	66.4	0.15
$v_c/(\text{m} \cdot \text{s}^{-1})$	381	383	0.52

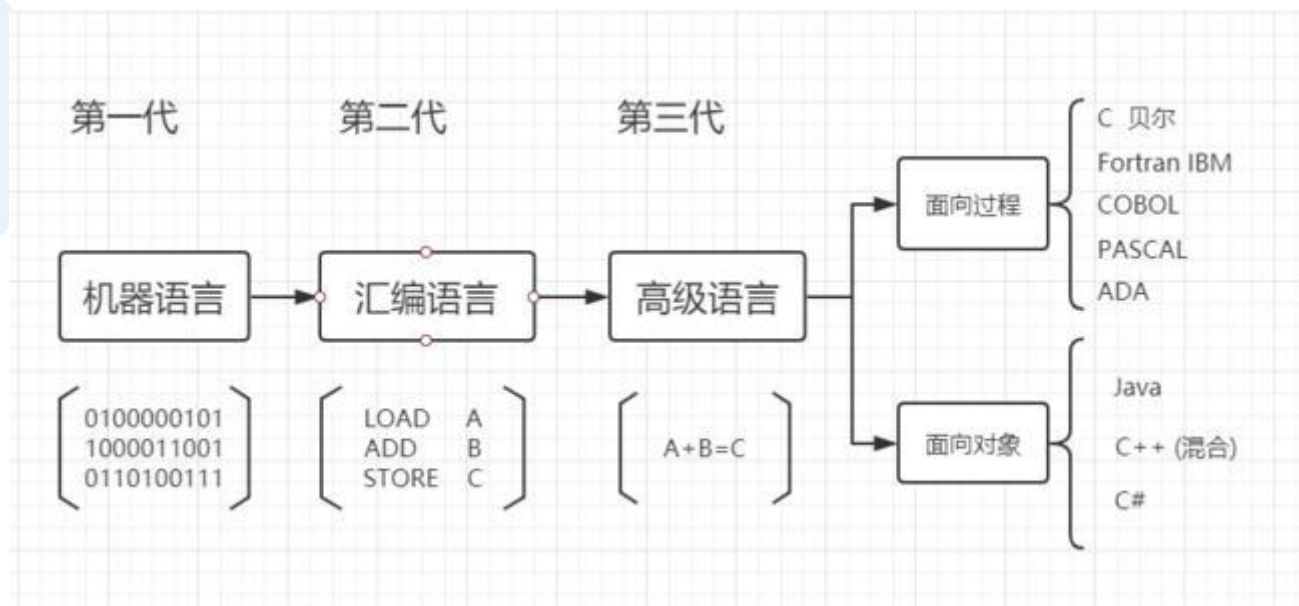
现在几乎无处不在，所有学科都离不开程序的辅助。最具代表性的为科学计算/数据分析，多媒体应用，工业自动化生产，互联网等。



计算机语言，程序和编程

4. 计算机语言是什么？

所有程序都基于机器语言运行，也就是“0”和“1”的电路信号。但是在编写的时候，通常由编程人员使用高级语言（C，C++，JAVA，Python等）编写。



计算机语言，程序和编程

4. 计算机语言是什么？

计算机语言和自然语言的区别在于**逻辑性**和**精确性**。表达追求单意，拒绝多意。

老婆给当程序员的老公打电话：“下班顺路买二斤包子带回来，如果看到卖西瓜的，买一个。”当晚，程序员老公手捧一个包子进了家门.....老婆怒道：“你怎么就买了一个包子？！”老公答曰：“因为看到了卖西瓜的。”

1. **buy** one kilogram steamed stuffed bun
2. **if** you see someone selling watermelons
3. **buy** one

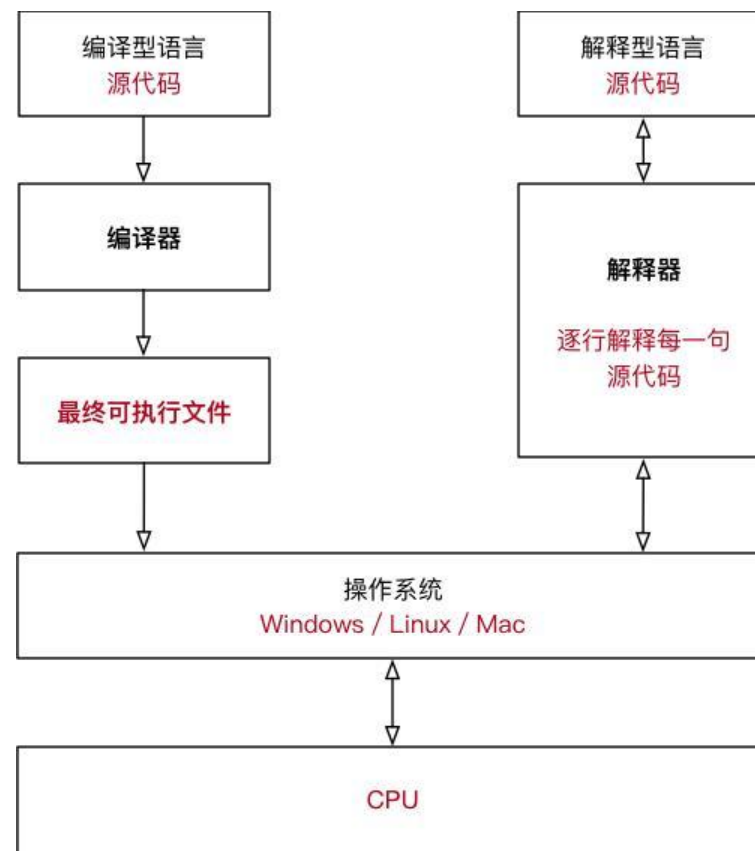
if，启动了一个条件判断语句

计算机语言，程序和编程

4. 计算机语言是什么？

由普通编程人员编写的高级语言代码，一般需要经过编译和解释，被**编译器/解释器**转译为机器语言，才能得以执行。因此高级语言也分为编译型和解释型。

而面向对象和面向过程则更多的是一种编程思想，现代语言多兼顾两者的特点，均可以支持，可供程序员自由选择。



计算机语言，程序和编程



根据执行方式不同，编程语言分为两类

- 静态语言：使用编译执行的编程语言

C/C++语言、Java语言，编译器一次性生成目标代码，优化更充分，程序运行速度更快

- 脚本语言：使用解释执行的编程语言

Python语言、JavaScript语言、PHP语言。执行程序时需要源代码，维护更灵活

计算机语言，程序和编程

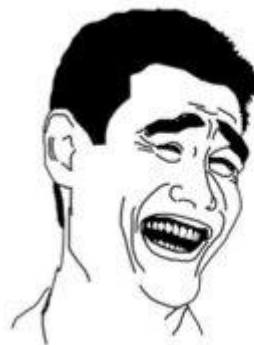
4. 计算机语言是什么？

编译型语言的特点是翻译和执行是分开的。我们常见的程序类型“.exe”可执行文件，和“app”程序，实际上就是一系列代码和数据被编译以后的对外交互窗口。编译的过程往往是不可逆和加密的，也就是说，我们使用商业程序的时候通常看不到代码。效率高，但是死板，兼容性差。(C,C++, Fortran)

解释型语言的特点是从上到下，翻译一句，执行一句。效率低，但灵活，兼容性好。(Python, Matlab, PHP, JAVA)

那Python开发的程序有没有保密问题呢？

OPEN SOURCE CODE



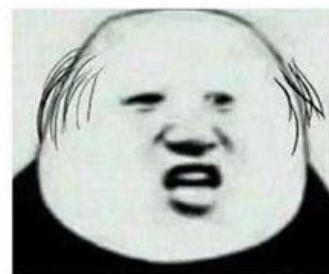
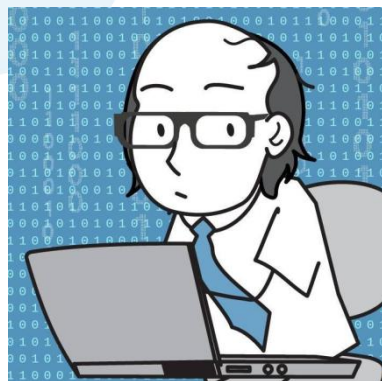
大家都那么忙，
哪有闲功夫破解你的烂代码

5. 编程？

编程就是制作程序的过程，利用数据和代码制作程序，和用水泥砖头盖房子本质上是一回事。

需要将人类的思路、知识、意图通过编程语言，下达给计算机进行执行。因此掌握编程知识只是一方面，增加自己其它领域知识的积累也很重要。就好像并不是英语学得很好，就能成为莎士比亚，水泥抹的匀就能成为贝聿铭一样。

计算机程序开发是周而复始的，需要经历：**编写新代码、测试、分析**，从事此类工作的人都可以称为**程序员**。



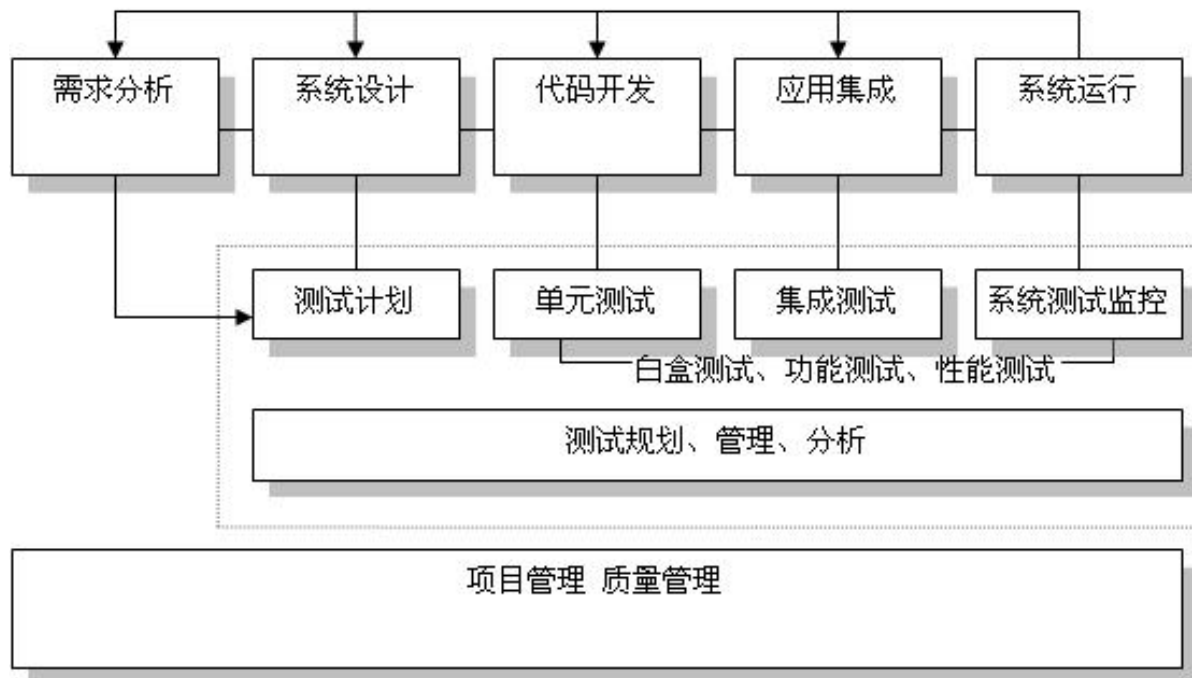
计算机，学着挺有意思的，就是头冷

编程人员级别等级判定



正式编程的过程

1. 定义问题（Problem Formulation）
2. 分析问题，如何实现（Design & Plan）
3. 写代码并测试（Coding & Debugging）
4. 评估（Evaluation）
5. 优化（Improving）



如何学好一门编程语言？

1. 掌握基本语法与数据结构

2. 读范例代码

3. 亲自改写代码并反复debug

1.4. 结合需求，自己编写小型应用程序

5. 参与大型程序项目

一些意义

编程体现了一种抽象交互关系、自动化执行的思维模式
计算思维：区别逻辑思维和实证思维的第三种思维模式
能够促进人类思考，增进观察力和深化对交互关系的理解

编程不单纯是求解计算问题
不仅要思考解决方法，还要思考用户体验、执行效率等
能够帮助程序员加深用户行为以及社会和文化认识

编程能够提供展示自身思想和能力的舞台
让世界增加新的颜色、让自己变得更酷、提升心理满足感
在信息空间里思考创新、将创新变为现实

能够更好地利用计算机解决问题
显著提高工作、生活和学习效率
为个人理想实现提供一种借助计算机的高效手段

程序员是信息时代最重要的工作岗位之一
国内外对程序员岗位的缺口都在百万以上规模



二 Python背景知识

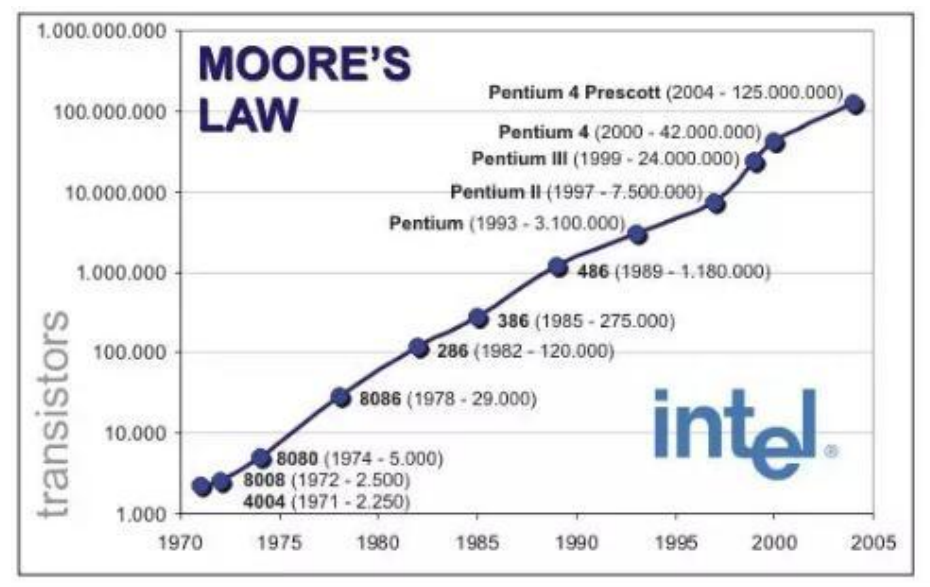


Python背景知识

Python由荷兰数学和计算机科学研究学会的Guido van Rossum 于20世纪90年代初设计。设计的初衷便是为了方便非计算机专业从业者进行编程工作。

早期的计算机语言受限于硬件性能的孱弱，程序员往往是一种特化的职业，他们必须尽力迁就硬件性能，压榨每一寸硬件潜能，因此计算机语言往往拗口难懂。

但随着摩尔定律不断的驱动，硬件瓶颈不再是主要矛盾，因此简洁易用对人友好的Python开始逐渐流行起来了。



Python的发展历程

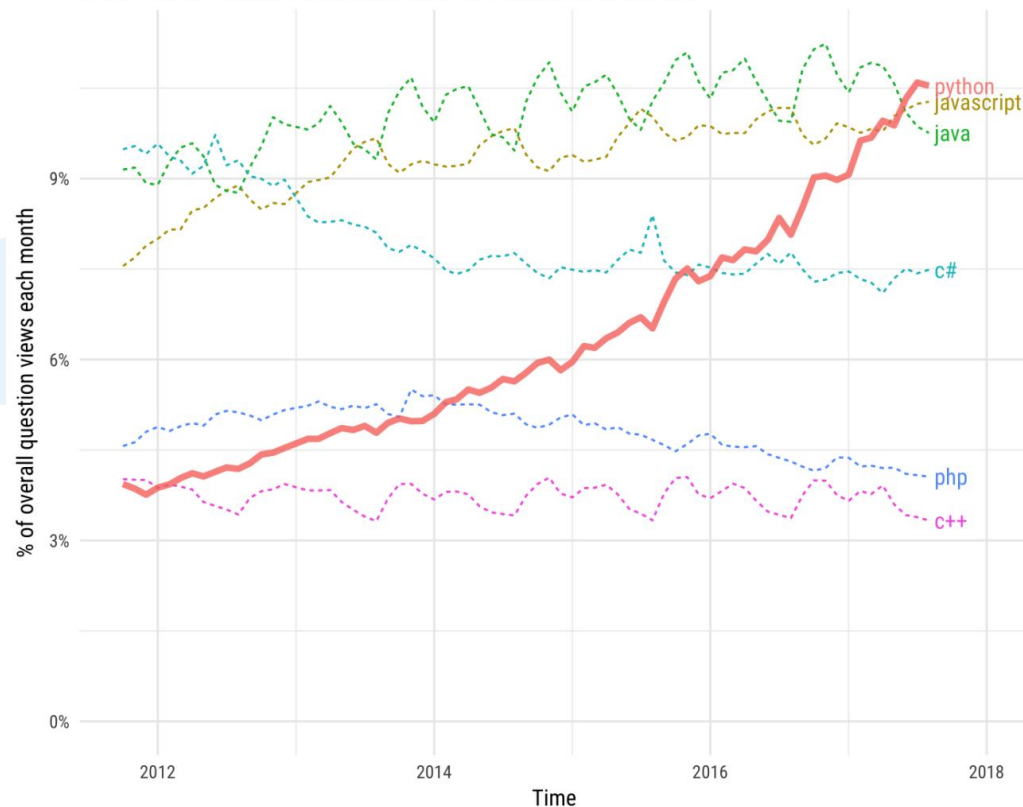
TIOBE 编程社区指数
(The TIOBE Programming
Community index)

Feb 2021	Feb 2020	Change	Programming Language	Ratings	Change
1	2	▲	C	16.34%	-0.43%
2	1	▼	Java	11.29%	-6.07%
3	3		Python	10.86%	+1.52%
4	4		C++	6.88%	+0.71%
5	5		C#	4.44%	-1.48%
6	6		Visual Basic	4.33%	-1.53%
7	7		JavaScript	2.27%	+0.21%
8	8		PHP	1.75%	-0.27%
9	9		SQL	1.72%	+0.20%
10	12	▲	Assembly language	1.65%	+0.54%
11	13	▲	R	1.56%	+0.55%
12	26	▲▲	Groovy	1.50%	+1.08%
13	11	▼	Go	1.28%	+0.15%
14	15	▲	Ruby	1.23%	+0.39%
15	10	▼▼	Swift	1.13%	-0.33%
16	16		MATLAB	1.06%	+0.27%
17	18	▲	Delphi/Object Pascal	1.02%	+0.27%
18	22	▲▲	Classic Visual Basic	1.01%	+0.40%
19	19		Perl	0.93%	+0.23%
20	20		Objective-C	0.89%	+0.20%

Python的发展历程

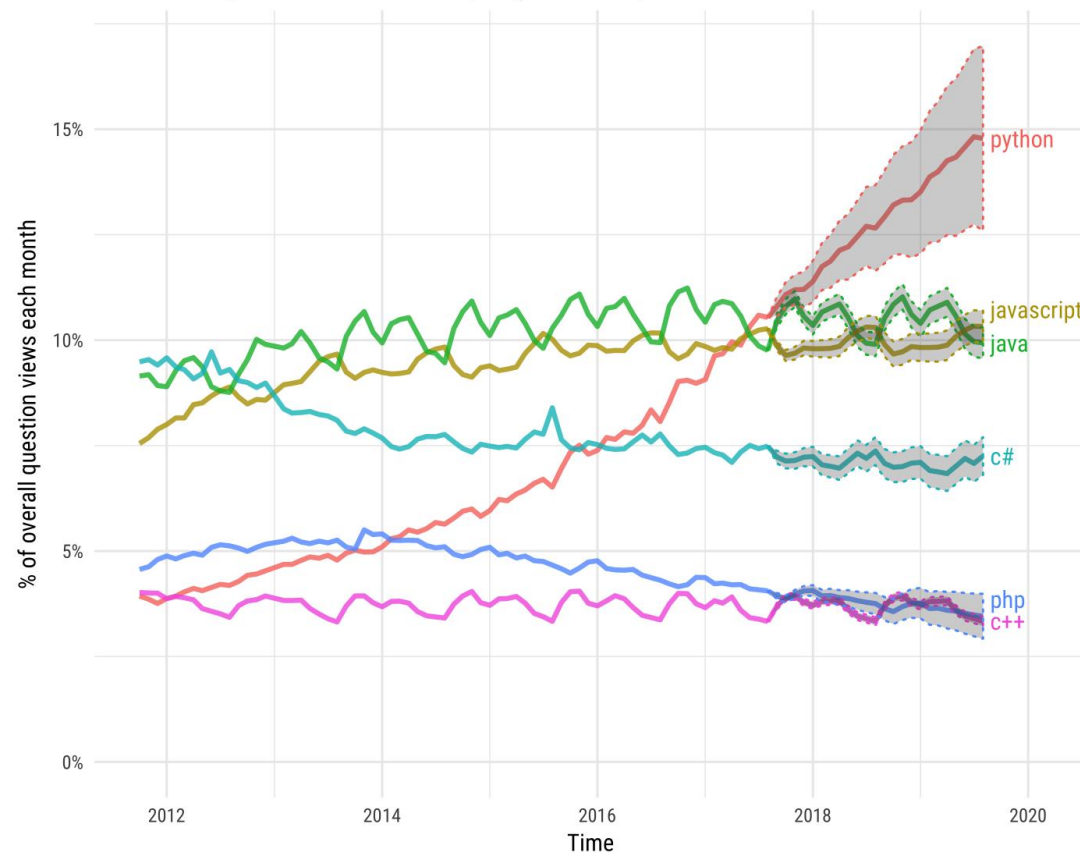
Growth of major programming languages

Based on Stack Overflow question views in World Bank high-income countries



Projections of future traffic for major programming languages

Future traffic is predicted with an STL model, along with an 80% prediction interval.

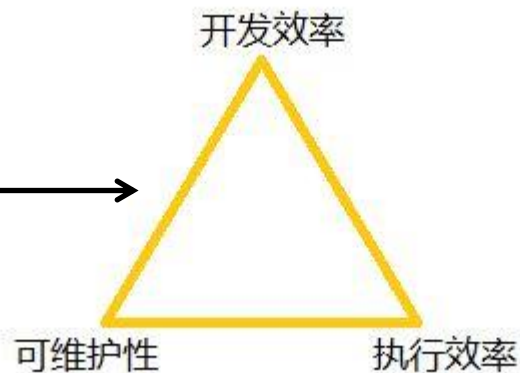


Python的特点

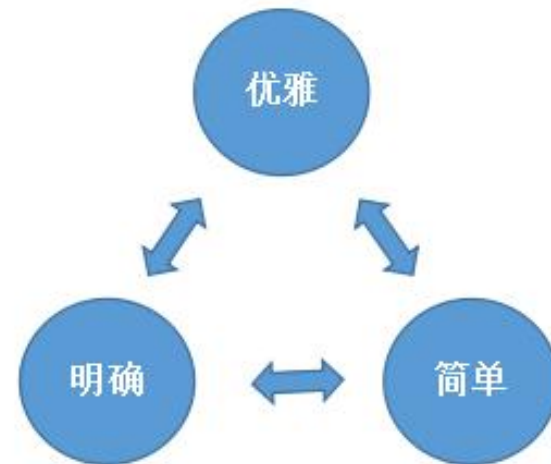
1. 简单易学，可读性强
2. 免费开源
3. 强大的功能
4. 可移植性好，可扩展性强，具有丰富的扩展库。

1. 效率较低，计算速度慢
2. 难以加密

大概在这！



编程语言的
不可能三角



Python的应用场合

1. 人工智能（神经网络，机器学习领域）
2. 数据分析
3. 科学计算（数值计算与云计算）
4. 爬虫开发
5. 网页WEB开发（douban, zhihu）
6. 轻量级游戏开发（yys）

几乎所有的程序都可以利用python开发，
当然，如果不在乎效率的话。



Python的版本

早期的python教材里python版本多为python2，但是2008年 Guido van Rossum为了解决2里的若干设计问题重新开发了python3。

3的问题在于对下不兼容，因此有一些历史遗留问题。

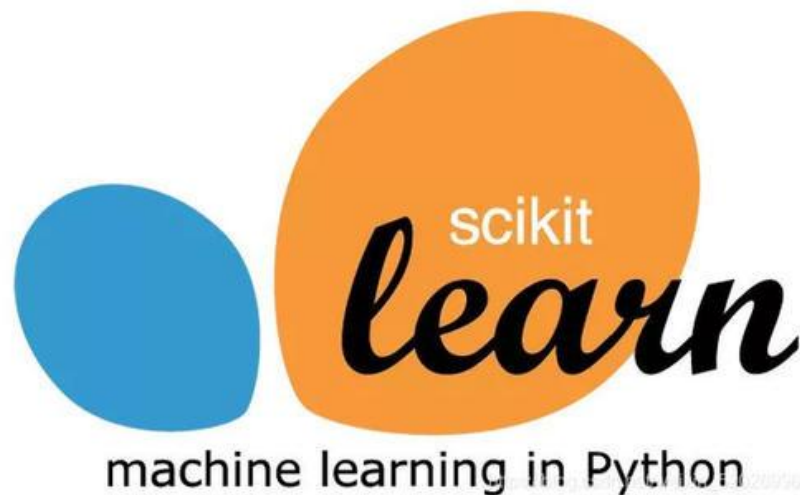
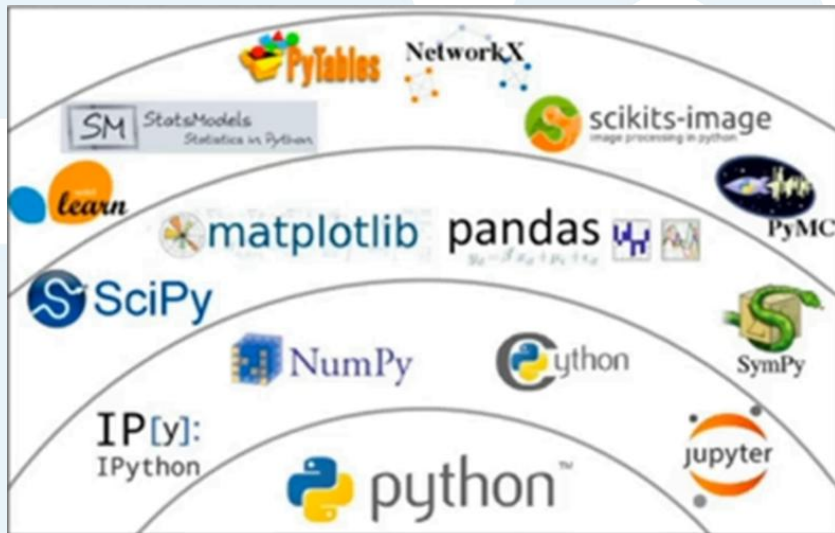
建议直接python3，绕过2，目前2里的库几乎都有3的版本了。



Python的库

库，包，功能模块...类似的概念，本质上是一回事。就是别人写好的python的程序片段，可以直接引用其功能，无需重复造轮子。

可分为标准库和第三方库。标准库需要熟悉内置函数，第三方库则五花八门，但往往也是对标准库的再封装和丰富完善。



Python的库

灵活使用各种成熟的库可以极大的提高开发效率，节约人时。

但任何事物都有两面性，如果想要进一步磨砺自己的技术水平，提高自己程序的运行效率，就必须深入库的内部，搞清楚它们运行的原理，甚至亲自写这一段功能代码。





三 开发环境设置





开发环境设置

Editor vs IDE vs REPL

Editor

代码编辑器(用来敲代码的图形界面)

Sublime Text, notepad++,甚至是word...

IDE

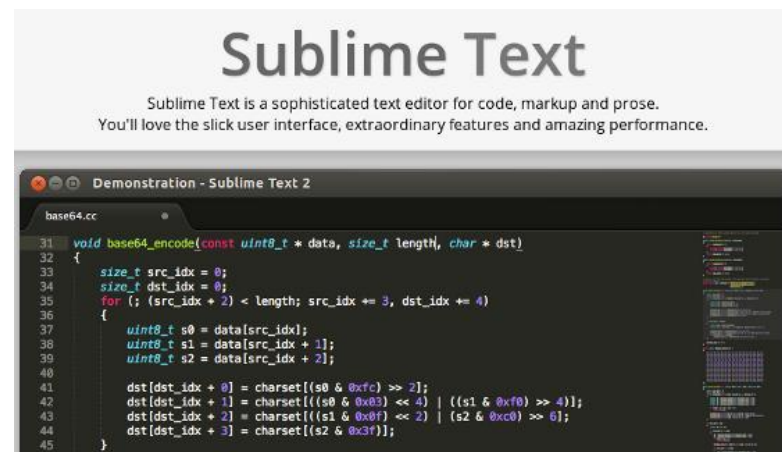
集成式开发环境(通常内置了Editor)

PyCharm,vs code,atom,Spyder

REPL

Read-Eval-Print-Loop(交互式解释器)

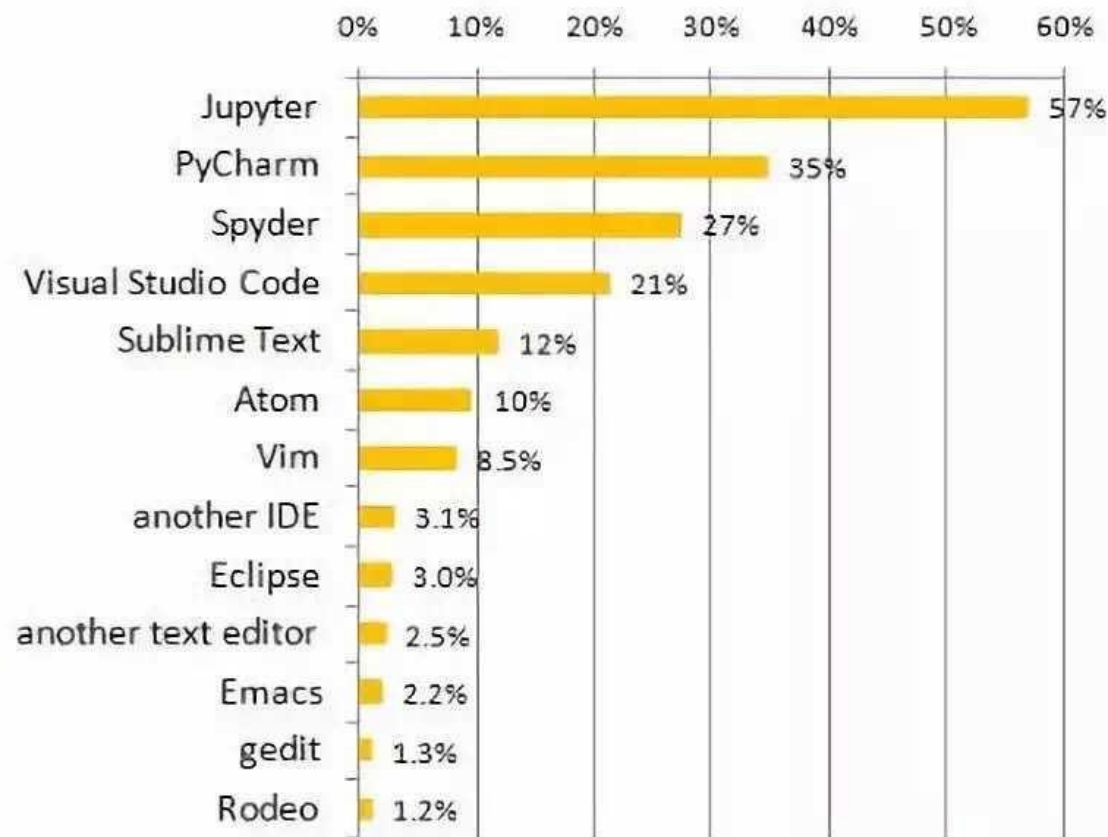
Jupyter



Editor vs IDE vs REPL

- Jupyter: 轻量级的浏览器风格IDE，文学化，草稿化的设计思想，适合做数据分析。
- PyCharm: 功能强大的老牌IDE，略臃肿。
- Spyder: 适合做科学计算的IDE，类似Matlab的界面，易于上手。
- Vs code: 简洁开源。

Most Popular Python IDE, Editors



开发环境设置

Python开发环境搭建：传统方法

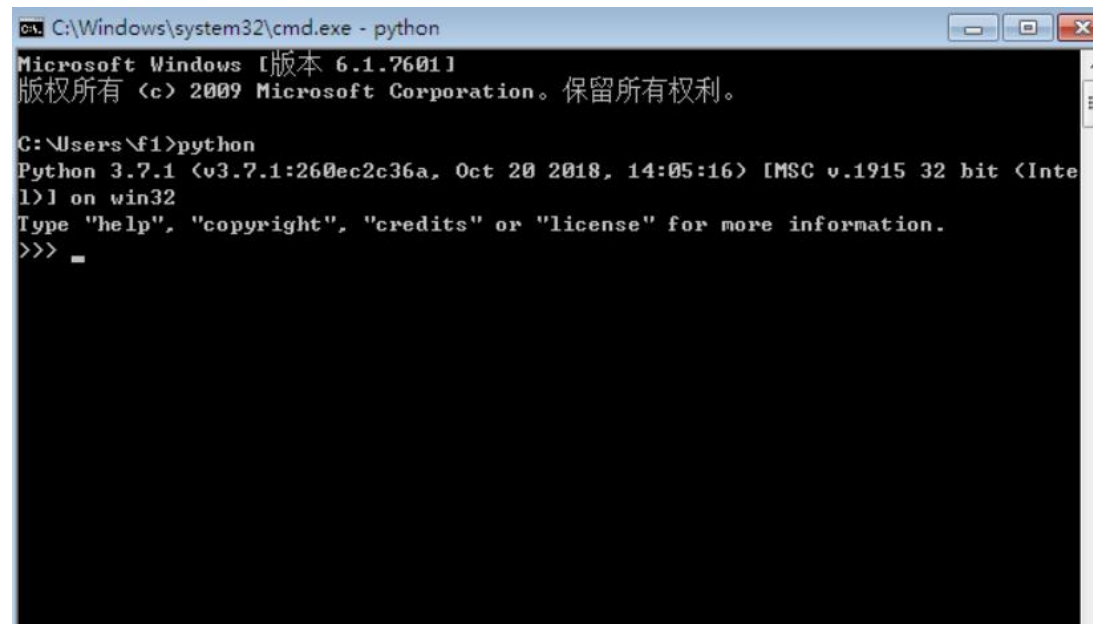
<https://www.python.org/>

<https://www.python.org/downloads/windows/>

1. 下载Python安装包
2. 配置环境变量
3. 安装IDE
4. 运行Python

特点：

- 纯净原生Python，体积很小，可以自由安装库。
- 但是功能有限，安装使用繁琐，IDE需要自己安装。
- 适合老手



```
C:\Windows\system32\cmd.exe - python
Microsoft Windows [版本 6.1.7601]
版权所有 (c) 2009 Microsoft Corporation。保留所有权利。

C:\Users\f1>python
Python 3.7.1 (v3.7.1:260ec2c36a, Oct 20 2018, 14:05:16) [MSC v.1915 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> _
```

开发环境设置

Python开发环境搭建：一站式方法

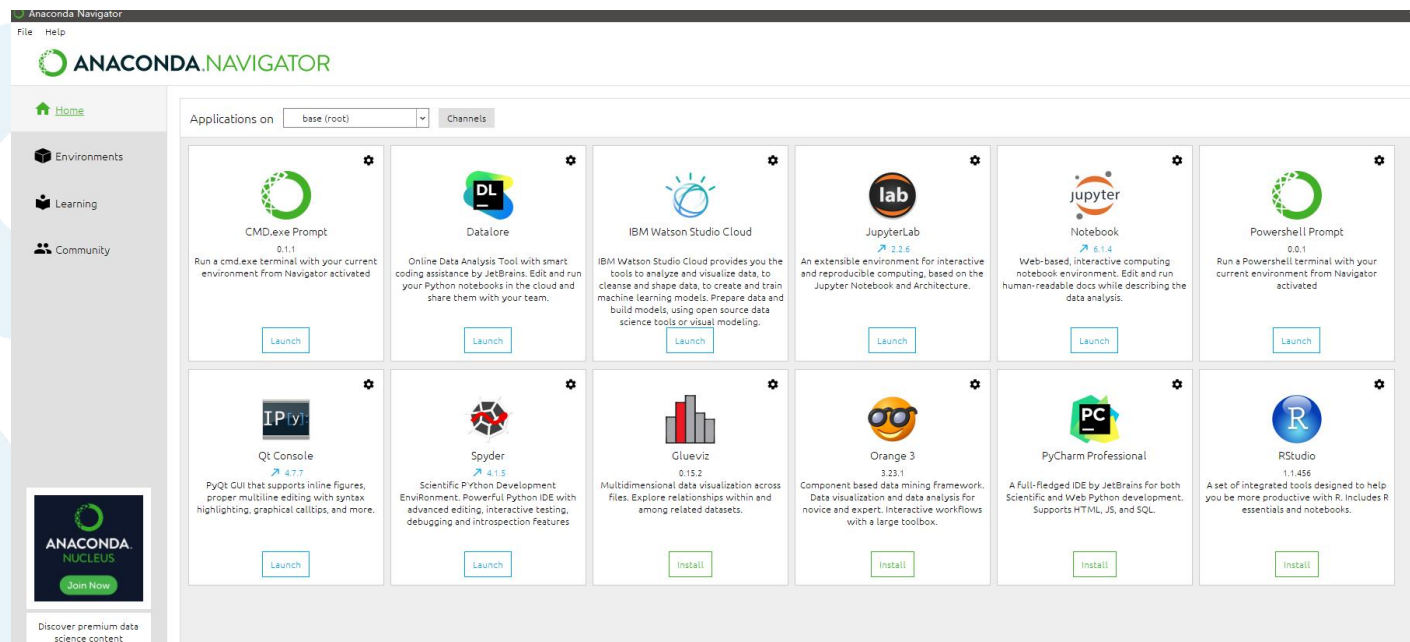
<https://www.anaconda.com/products/individual>

Anaconda是一个开源的Python发行版本，其内置了conda、Python、Numpy等数百个科学计算相关的库及其依赖项。conda是一个开源的包、环境管理器。装了这个就不需要再单独去装Python了，而且自带了IDE。



特点：

- 懒人包，体积较大。
- 适合数据分析和科学计算领域，几乎是默认的环境。
- 适合新手。



开发环境设置

安装完成Anaconda后可以在Navigator里面查看已安装库的版本号。

版本号是比较重要的，不同版本下的代码有时候可能会不兼容。

当然，简单的基本代码往往是通用的，使用了越高级功能的代码，越有可能产生兼容性问题。

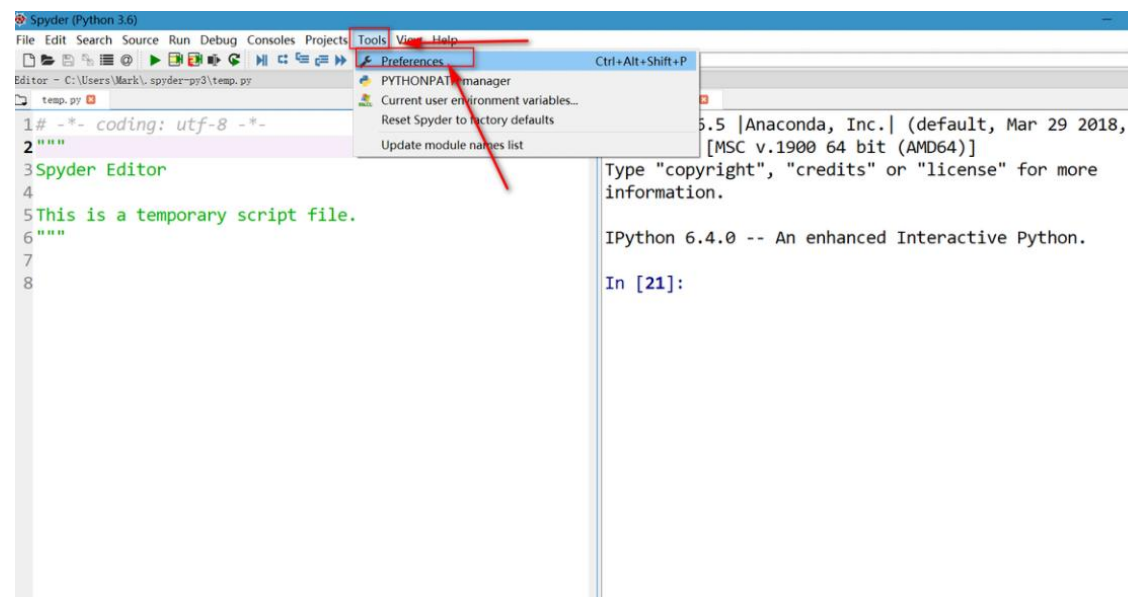
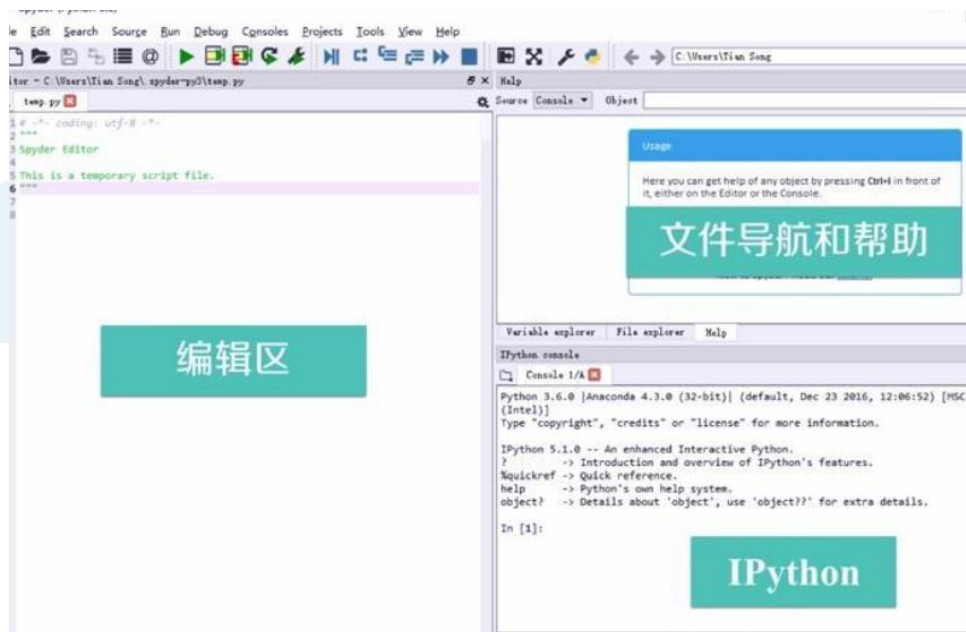
python	3.8.5
--------	-------

可以对这些库完成更新，安装，卸载等操作。

Installed	Channels	Update
Name	Description	
nbclient		
nbconvert		

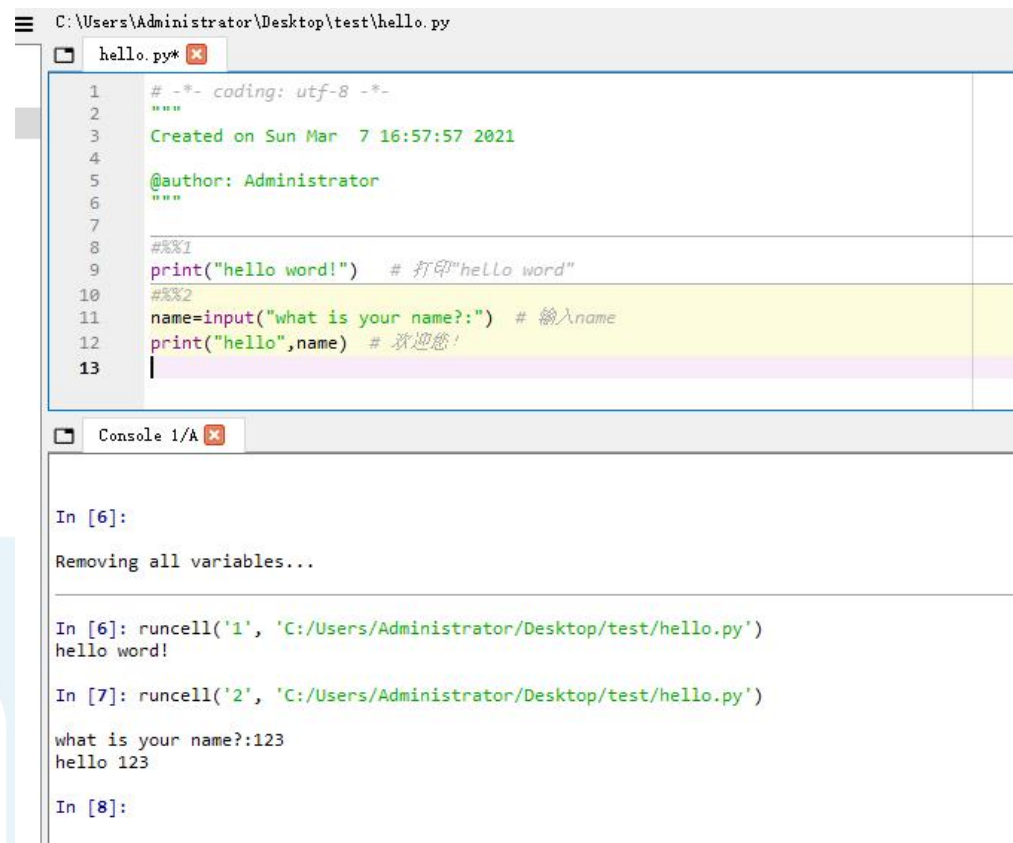
开发环境设置

后续课程我们选择Spyder作为主要的IDE，也就是开发环境，不必纠结于具体的IDE，实际上只要学会了一种，其它IDE最多半小时熟悉一下风格即可随意上手。



开发环境设置

后续课程我们选择Spyder作为主要的IDE，也就是开发环境，不必纠结于具体的IDE，实际上只要学会了一种，其它IDE最多半小时熟悉一下风格即可随意上手。



```
C:\Users\Administrator\Desktop\test\hello.py
hello.py*
1  # -*- coding: utf-8 -*-
2  """
3  Created on Sun Mar  7 16:57:57 2021
4
5  @author: Administrator
6  """
7
8  #%%1
9  print("hello word!") # 打印"hello word"
10 #%%2
11 name=input("what is your name?:") # 输入name
12 print("hello",name) # 欢迎您!
13

Console 1/A
In [6]:
Removing all variables...

In [6]: runcell('1', 'C:/Users/Administrator/Desktop/test/hello.py')
hello word!

In [7]: runcell('2', 'C:/Users/Administrator/Desktop/test/hello.py')

what is your name?:123
hello 123

In [8]:
```



四 课后作业



- 下载，安装完成**Anaconda**
 - 注意自己的操作系统，选择合适的版本
 - 安装完成以后，打开**Spyder**，熟悉界面，将界面风格调整成自己喜欢的样式
-



感谢参与 下堂课见



