# PYTHON程序设计

计算机学院 王纯

课程概述

# 你对于Python和编程的熟悉程度怎么样?

- A 小白,用其他语言编写过代码
- B 略懂, 掌握基本语法, 能写简单程序
- 高手,掌握第三方库,能写复杂程序
- D 大神, 言语已经不能形容

序号	教学内容	学时分配
1	课程概述	1
2	数据类型和表达式	2
3	程序控制结构	2
4	组合数据类型	4
5	函数	2
6	文件	2
7	面向对象程序设计	2
8	异常处理	2
9	数据获取	4
10	数据预处理	4
11	数据计算和分析	2
12	数据可视化	4
13	应用实例	2
14	课程小结、大作业	1

- ■雨课堂、腾讯会议
- ■0J在线系统 HUSTOJ
- 师生交流QQ群---点击链接加入群聊【Python程序设计2022秋季】:

https://jq.qq.com/?\_wv=1027&k=oS1z6pwu



群名称: Python程序设计2022秋季

群号: 851323719

- 程序和编程语言
- Python的发展和应用
- 学习准备: 开发环境、方法
- 第一个Python程序
- Python的编码规范

课程概述

#### 程序是什么:基本概念



#### 通俗定义

程序是为完成某项任务而编写的计算机可以理解和执行的指令集合。

### 经典定义

程序=算法+数据结构。尼古拉斯·沃斯 (Niklaus Wirth, 1934.2.15—)。

备注: Niklaus Wirth, 号称凭借一句话获得图灵奖的人, 同时也是Pascal之父。这个公式揭示了程序的本质, 对计算机科学的影响很大。

#### 算法

是解决某类问题的明确和有限的步骤描述

#### 数据结构

是计算机存储、组织数据的方式,是带有结构特性的数据元素的集合。<br/>

#### 程序设计

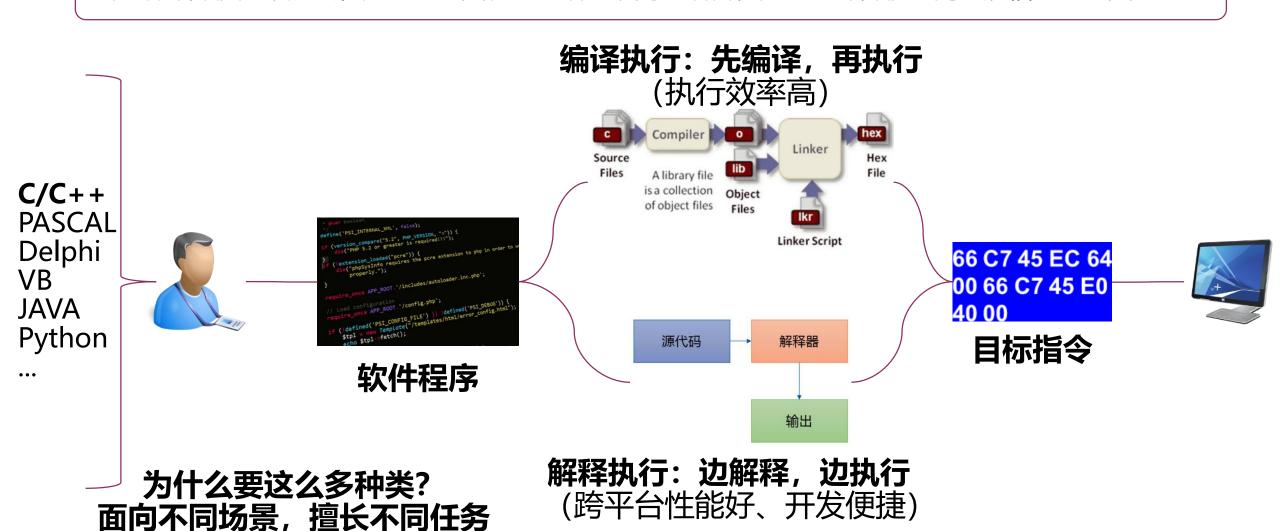
程序=算法+数据结构

编程人员用编程语言编写程序和软件的活动

软件=程序+文档

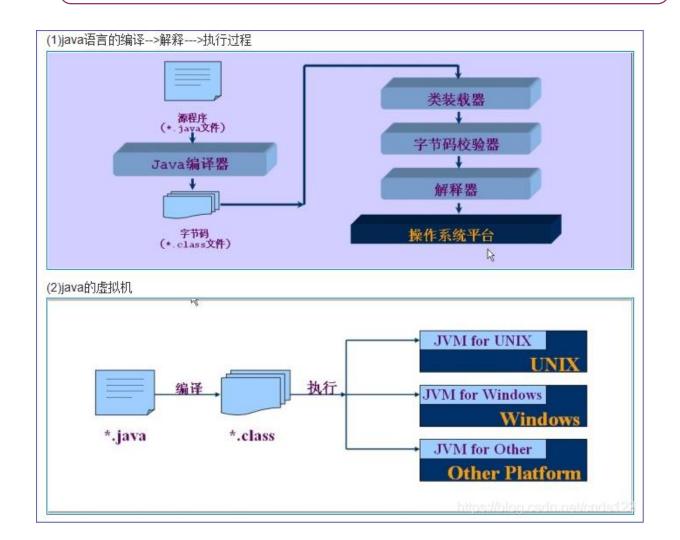
#### 程序是什么: 人与机器眼中的编程语言

是指计算机能够接受和处理的、具有一定语法规则的语言,是人与计算机之间交换信息的工具

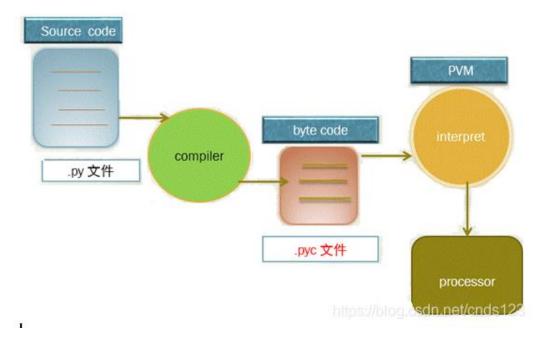


#### 程序是什么:解释器的不同实现

#### 强大的解释型编程语言通常不是纯粹的解释执行



#### **JAVA VS PYTHON**



# 编程语言:语言的要素与分类

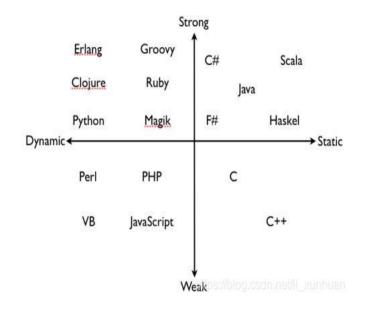
The acts of the mind, wherein it exerts its power over simple ideas, are chiefly these three: I. Combining several simple ideas into one compound one, and thus all complex ideas are made. 2. The second is bringing two ideas, whether simple or complex, together, and setting them by one another so as to take a view of them at once, without uniting them into one, by which it gets all its ideas of relations. 3. The third is separating them from all other ideas that accompany them in their real existence: this is called abstraction, and thus all its general ideas are made.

- John Locke, An Essay Concerning Human Understanding (1690)

# 语言要素

基本的表达 形式 primitive expressions

组合方法 means of combination 抽象方法 means of abstraction



#### 编程范式分类

- 命令式编程语言
  - 面向过程
  - 面向对象
- 声明式编程语言
  - 函数式 Lisp
  - 逻辑式 Prolog

Python支持多种编程范式

数据类型分类: 动态、静态

编程语言:编程语言分类

机器语言

汇编语言

高级语言

是用二进制代码表示的计算机能直接识别和执行的一种机器指令的集合。机器语言具有灵活、直接执行和速度快等特点。按着一种计算机的机器指令编制的程序,不能在另一种计算机上执行。

一条指令一般包括:操作码、操作数读取地址、操作结果存储地址、下条指令地址(顺序执行可省略)

假定: 0000: 加载指令 (LOAD); 0000: 暂存器A; 000000010000: 代表地址为16的存储器

指令示例: 0000,0000,000000010000 代表: LOAD,A,16 (把第16号存储器的值加载到暂存器A)

优点 执行速度快

缺点 看不懂、难编写、易出错、难调试

### 编程语言:编程语言分类

机器语言

#### 汇编语言

高级语言

为了解决机器语言的缺点,改进为用助记符和符号地址来表示二进制数,形成的助记符语言称为汇编语言。特定的汇编语言和特定的机器语言指令集是一一对应的,不同平台之间不可直接移植。

计算机不能直接识别和执行汇编语言程序,需要一种起到翻译作用的程序将其翻译为机器语言程序。这个过程称为汇编,这个翻译程序称为汇编程序。

前面的 "LOAD,A,16" 就是汇编语言程序

优点: 执行速度快、占用空间小

缺点:编程难度大、效率低;需要了解硬件、指令、寻址等知识;不具备通用性和可移植性。

```
data segment ;数据段
 string db 'Hello,World!$'; 定义一个名为string的字符串
data ends
code segment ;代码段
assume cs:code,ds:data;指定code段与CS寄存器关联,data段与DS寄存器关联
start:
 mov ax,data ;获取段基址
 mov ds,ax;将段基址送入寄存器
 mov dx,offset string; 获取string的偏移地址
 mov ah,9;调用9号DOS功能(显示字符串)
 int 21h
 mov ah,4ch; 调用程序结束功能
 int 21h
code ends
end start
                        输出"Hello,World!
```

#### 编程语言:编程语言分类

机器语言

汇编语言

高级语言

屏蔽了机器的细节,基本脱离了机器的硬件系统,提高了语言的抽象层次。是较接近自然语言和数学公式的编程语言,用人们更易理解的方式编写程序。 输出"Hello,World!">>>print('Hello World!')

计算机不能直接识别和执行高级语言程序, 所以需要编译器编译执行, 或者解释器解释执行。

优点:可读性强,编程效率高,可移植性好。

缺点: 执行效率低、占用空间大。

面向过程编程语言,自顶向下、逐步求精。采用模块分解与功能抽象,分而治之。如Fortran、C

面向对象编程语言,采用抽象建模的思想,将数据及操作封装在一起,形成相互依存、不可分离的对象整体。如C++、Java



Phthon里面一切皆对象 以解决问题为主,方法为辅;先有全局思考、后局部实现

# 编程语言:编程语言一览

第一个世界推广的高级语言—Fortran(数值计算)

首批诞生影响深远的高级语言—ALGOL (科学计算; 算

法描述;影响Pascal、Ada、C等)

最简单的语言——BASIC

第一个结构化编程语言——Pascal(语法严谨、层次分)

明;编程语言里一个重要的里程碑)

现代程序语言革命的起点——C语言

简单易学功能强大的Windows语言——VB

生态丰富的主流大型跨平台语言——Java

主流的WEB页面开发脚本语言——JavaScript

简单易用的主流大型跨平台语言——C#

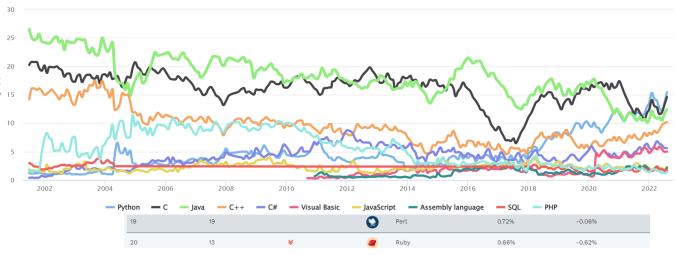
人人可学、无处不在的应用——Python

#### 世界编程语言排行榜

Sep 2022	Sep 2021	Change	Programming Language	Ratings	Change
1	2	^	Python	15.74%	+4.07%
2	1	•	<b>G</b> c	13.96%	+2.13%
3	3		Java	11.72%	+0.60%
4	4		G C++	9.76%	+2.63%
5	5		<b>©</b> C#	4.88%	-0.89%
6	6		VB Visual Basic	4.39%	-0.22%
_	_		<u> </u>		

TIOBE Programming Community Index

Source: www.tiobe.com



### PYTHON的发展: 起源



GUIDO VAN ROSSUM (吉多•范•罗苏姆)

#### Macintosh 128K

ANNOUNCED: October 1983

RELEASED: Jan. 24, 1984

FEATURES: Mouse, keyboard and graphical user interface (System 1.0)

DDICE. \$2.40



1956, 出生于荷兰 1982, 阿姆斯特丹大学数学与 计算机科学硕士学位

1982-95,荷兰数学和计算机科学研究学会 (CWI)程序员

2005-12, 谷歌资深主任工程师

2013-19, Dropbox首席工程师

2019, 退休

2020,复出加入微软

8MHzCPU主频 128KB内存

效率和空间要 发挥到极致;

好不好学 好不好用都不重要 **ABC** 

(一种教学语言, CWI开发,

目标"让用户感觉更好";

太超前了,"机器感觉不好"

Guido参与其中)

Guido希望有一种语言, 这种语言能够像C语言 那样,能够全面调用计 算机的功能接口,又可 以像shell那样,可以轻 松的编程。

C. SHELL



1989,为打发圣诞假期, Guido开始写Python语言 的编译/解释器。 1991,第一个Python编译 器/解释器诞生。

Python来自电视剧Monty Python's Flying Circus (BBC1960-70年代播放的 室内情景幽默剧)。

#### PYTHON的发展: PYTHON的哲学

Python是一种动态的、解释型的脚本语言,它的语法优美易读,非常适合初学者学习使用。

Python的设计哲学是简单、优雅、明确。

̄简单:专注于解决问题而非语言

易学: 语法简单, 容易上手

免费:免费开源

高级:高级语言,无需关注内存等细节

优点 移植:解释执行,可移植性强

对象: 完全面向对象, 也支持面向过程

扩展: 便捷调用其他语言编写扩展功能

生态:丰富的库,丰富的资料

- 规范:强制缩进,可读性强

一执行效率不高;源代码无法加密; 全局解释器锁GIL限制多线程并发 (CPython解释器下,为了兼容非线程安全的 \_ C扩展,避免内存泄漏的互斥锁机制)

#### **PYTHON**

C

print( "Hello world!" )

#include <stdio.h>
int main()
{
 printf( "Hello world!" );
}

(1 <= n <= 10)

(n>=1 && n<=10)

Bruce Eckel (1957-) (Thinking in C++/Java)



LIFE IS SHORT, YOU NEED PYTHON!



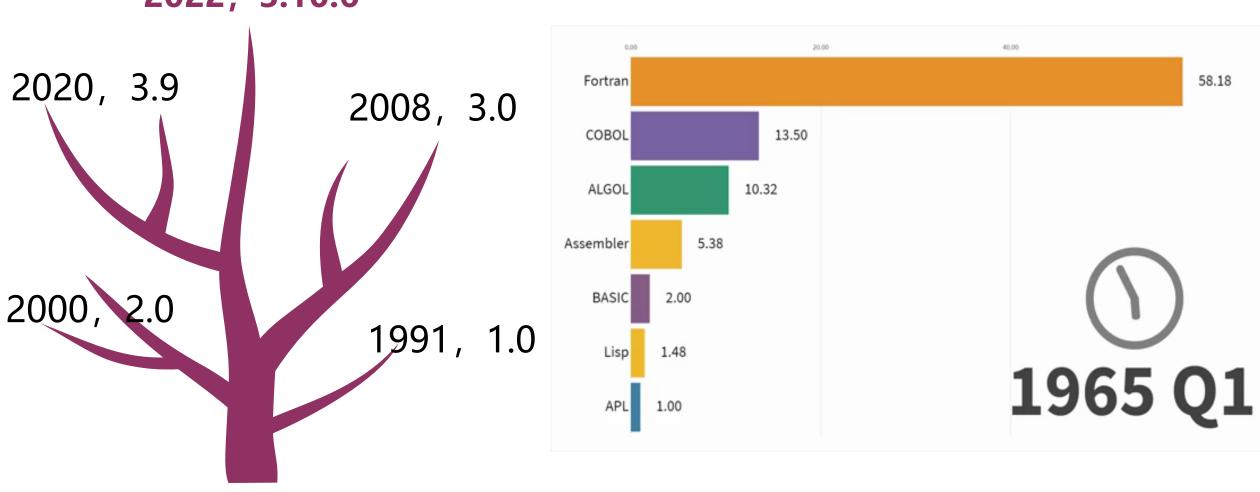
人生苦短,我用PYTHON!



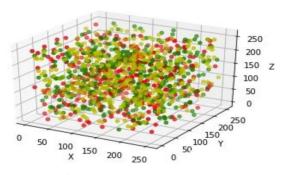
# PYTHON的发展

2022, 3.10.6

**Python** 



#### PYTHON的应用:应用领域及示例



数据可视化



烟花



数据可视化



游戏



云图



人脸识别

人工智能 云计算 大数据 web开发 科学计算 传统软件开发等

得益于Python简单、优雅、明确的设计哲学,和功能强大且丰富的库,

几十行、几百行代码就可以实现上述效果!

# 学习的准备:下载安装

- 下载适当版本:本课程选择 Windows平台上的 python 3.8版本。
- 以管理员身份安装程序。
- 重新启动Windows系统。 环境变量 等都已设置好。

http://www.python.org/downloads

- **交互模式**: 在命令行中执 行Python
- 集成开发环境: Python 自带的IDLE集成开发环境
- PyCharm (推荐),
   Spyder, Eclipse +
   PyDev, Atom,
   Jupyter Notebook,
   Vscode

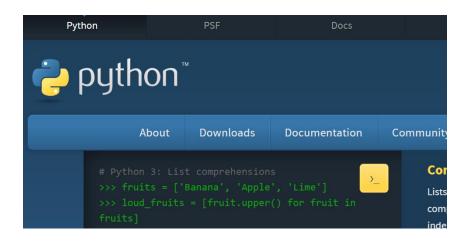


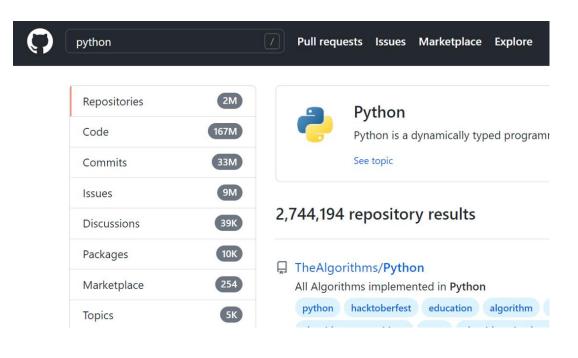




# 学习的准备: 有效的方法

- 编程日常: 读程序、写程序、查手册、找资料
- 简单的方法:带任务学习、用到时查阅、有问题解决
- 开源社区, python文档3.9.6 Documentation (python.org)以及其他丰富的网络资源等等





以上只能保证会用,要想成为高手,还要持续的钻研学习,并经过大项目的锻炼

# 第一个PYTHON程序

- Python 程序执行主要有两种方式:命令行、脚本。
- 编写并执行我们的第一个Python程序 "Hello World!"

>>>print("Hello World!")
Hello World!

print(): Python内置函数名称,作用是输出括号中的内容;

"Hello World":字符串类型的数据,作为参数传递给

print函数

input()函数是Python语言中值的最基本输入方法,通过用户输入,接受一个标准输入数据,默认为字符串类型,基本用法:

>>>object = input('提示信息')

把刚才的Hello World加上输入:

>>>print ("Hello", input("请输入一种 语言"))

请输入一种语言>? Python

**Hello Python** 

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!

The Zen of Python, by Tim Peters

- Python语言规范 Python语言编写程序的一些注意事项
- Python风格规范 Python程序的格式

PEP8 (Python Enhancement Proposal #8)

一行一个语句

大写字母定义常量

一行不要超过80个字符

小写字母定义变量

4个空格缩进

少使用全局变量

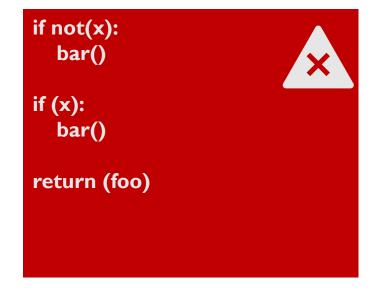
适当使用空行和注释,增加可读性

常量、变量名字多个单词间用""分割

一行的长度不要超过80字符

括号宁缺毋滥

```
if foo:
    bar()
while x:
    x = bar()
if x and y:
    bar()
if not x:
    bar()
return foo
for (x, y) in dict.items(): ...
```



包 使用全路径名来导入以避免同名冲突

异常 谨慎使用以避免令人困惑的控制流

全局变量 尽量避免使用全局(模块级别)的变量

嵌套/本地/内部类或函数

鼓励使用: 类定义在方法、函数或者类中, 函数定义在方法或函数中

避免过时语言特性

使用当前python版本中更好的替代品

推荐静态Scoping

内层函数可访问外层的变量,但不可修改

避免过于紧凑的结构 奇技淫巧导致可读性差,结果难以理解和调试

#### 使用原则

#### 编码规范很繁杂

- ✓ 不同人使用的规范常 略有不同
- ✓ 难以在各种情况坚持 使用规范

#### 编码规范的利弊权衡

- ✓ 有利:程序可读性或 者正确性增加
- ✓ 不利: 繁琐增加成本

- 程序和编程语言
- Python的发展和应用
- 学习准备: 开发环境、方法
- 第一个Python程序
- Python的编码规范

课程概述

# 谢谢