



四川大學
SICHUAN UNIVERSITY

脚本语言程序设计

Python基础教程

方智阳

QQ: 43471492

微信/手机: 13550345266

个人主页: <https://fangzhiyang.wjlpt.com>





- 编写C程序就像一群人拿着剃刀在刚打过蜡的舞场内跳快舞。
 - » **Waldi Ravens**
- C++：难学更难用，因为它天生如此。
 - » **匿名**
- 在很多方面，Java就是C++。
 - » **Michael Feldman**
- 现在请看一种前所未有的表演。。。
 - » **Monty Python的《飞行马戏团》**






Python的创始人荷兰人吉多·范罗苏姆[3] (Guido van Rossum)。1989年圣诞节期间，在阿姆斯特丹，Guido为了打发圣诞节的无趣，决心开发一个新的脚本解释程序，之所以选中Python（大蟒蛇的意思）作为该编程语言的名字，是取自英国20世纪70年代首播的电视喜剧《蒙提·派森的飞行马戏团》(Monty Python's Flying Circus)。



python

英 ['paɪθən]  

美 ['paɪθɑ:n]  

n. 蟒蛇; 蟒; 蚺蛇





四川大學
SICHUAN UNIVERSITY



抖音

抖音号: 1927910042

没有谁可以欺负我的女儿
——程序员版



Python : 在座的都是弟弟

应用软件

系统软件

引言

虚拟机

实际机器





- 主要数据类型
- 函数
- 控制语句
- 类和面向对象
- 异常处理
- 文件处理
- 正则表达式





- 签到： **10%**（超星课堂签到）
- 课堂小测验： **40%**（现场答题）
- 期末考试： **50%**（项目实现）

期末项目实现评分标准	
评分项	分值
项目难度	20
项目工作量及实现效果	40
文档书写	30
代码规范性	10





1. 什么是Python

Python 是初学者的语言：**Python** 对初级程序员而言，是一种伟大的语言，它支持广泛的应用程序开发，从简单的文字处理到 **WWW** 浏览器再到游戏。

Python 是一种面向对象、解释型、动态数据类型的计算机（高级）程序设计语言，由 **Guido van Rossum** 于 **1989** 年底发明，第一个公开发行人版发行于 **1991** 年。

Python 语法简洁而清晰，具有丰富和强大的类库，常被昵称为胶水语言，能够把用其他语言制作的各種模块（尤其是 **C/C++**）很轻松地联结在一起。





1. 什么是Python

Python是一门优雅而健壮的编程语言，它继承了传统编译语言的强大性和通用性，同时也借鉴了简单脚本和解释语言的易用性。

就像**Python**主页<https://www.python.org/>上面介绍的一样 “**Python is a programming language that lets you work quickly and integrate systems more effectively.**”





1. 什么是Python

Python 2.0 于 **2000** 年 **10** 月 **16** 日发布，增加了实现完整的垃圾回收，并且支持 **Unicode**。

Python 3.0 于 **2008** 年 **12** 月 **3** 日发布，此版不完全兼容之前的 **Python** 源代码。不过，很多新特性后来也被移植到旧的**Python 2.6/2.7**版本。

Python 3.0 版本，常被称为 **Python 3000**，或简称 **Py3k**。相对于 **Python** 的早期版本，这是一个较大的升级。

Python 2.7 被确定为最后一个 **Python 2.x** 版本，它除了支持 **Python 2.x** 语法外，还支持部分 **Python 3.1** 语法。**Python**核心团队在**2020**年停止支持**Python 2**。





Python 应用

Youtube - 视频社交网站

Dropbox - 文件分享服务

豆瓣网 - 图书、唱片、电影等文化产品的资料数据库网站

知乎 - 一个问答网站

果壳 - 一个泛科技主题网站

Bottle - **Python**微**Web**框架

EVE - 网络游戏**EVE**大量使用**Python**进行开发

Blender - 使用**Python**作为建模工具与**GUI**语言的开源**3D**绘图软件

.....





2. 为什么使用Python

TIOBE公布的2018年8月、2019年8月、2020年8月、2021年5月编程语言指数排行榜：

Aug 2018	Jul 2019	Aug 2020	Aug 2021	Aug 2020	Change	Programming Language	Ratings	Change
1	1	1	1	1		C	12.57%	-4.41%
2	2	2	2	3	▲	Python	11.86%	+2.17%
3	3	3	3	2	▼	Java	10.43%	-4.00%
4	4	4	4	4		C++	7.36%	+0.52%
5	5	5	5	5		C#	5.14%	+0.46%
6	6	6	6	6		Visual Basic	4.67%	+0.01%
7	7	7	7	7		JavaScript	2.95%	+0.07%
8	8	8	8	9	▲	PHP	2.19%	-0.05%
9	9	9	9	14	▲▲	Assembly language	2.03%	+0.99%
10	10	10	10	10		SQL	1.47%	+0.02%
11	11	11	11	18	▲▲	Groovy	1.36%	+0.59%
12	12	12	12	17	▲▲	Classic Visual Basic	1.23%	+0.41%
13	13	13	13	42	▲▲	Fortran	1.14%	+0.83%
14	14	14	14	8	▼▼	R	1.05%	-1.75%
15	15	15	15	15		Ruby	1.01%	-0.03%
16	16	16	16					
17	17	17	17					
18	18	18	18					
19	19	19	19					
20	20	20	20					



2. 为什么使用Python

- 简单易学
- 开源/免费
- 可移植
- 可扩展
- 易维护
- 健壮性
- 解释性和编译性





1) 简单易学

Python是一种代表简单主义思想的语言。

阅读一个良好的**Python**程序就感觉像是在读英语一样，它使读者能够专注于解决问题而不是去搞明白语言本身。

Python关键字少、结构简单、语法清晰。

学习者可以在很短的时间内轻松上手。





2. 为什么使用Python

2) 开源/免费

Python是**FLOSS**（自由/开放源码软件）之一，**Python** 源代码同样遵循 **GPL**（**GNU General Public License**）协议。

使用者可以自由地发布这个软件的拷贝、阅读它的源代码、对它做改动、把它的一部分用于新的自由软件中。





3) 可移植

Python本身也是使用**C**写的，由于**C**的可移植性，使得**Python**可以运行在任何带有**ANSI C**编译器的平台上。

Linux

Windows

FreeBSD

Google基于Linux开发的Android平台

◦ ◦ ◦





4) 可扩展

对于所有的模块，不管是**Python**标准库的还是自己创建的，**Python**都能使用相同的语法进行访问，可以根据需要对**Python**进行“扩展”。

Python的标准实现是使用**C**语言完成的（**CPython**），**Python**的**Java**实现被称作“**Jython**”，还有针对**.NET**或**Mono**平台的**C#**实现**IronPython**，可以使用对应的不同的语言编写其扩展。





5) 易维护

源代码维护是软件开发生命周期的组成部分。只要不被其他软件取代或者放弃使用，开发者的软件通常会保持继续的再开发。





6) 健壮性

Python提供了“安全合理”的退出机制，使得在程序发生错误时程序员可以掌控局面，根据出错条件提供处理机制。

一旦**Python**程序由于错误崩溃，解释程序就会转出一个“堆栈跟踪”，那里有可用到的全部信息，包括程序崩溃的原因，以及是哪段代码（文件名、行数、函数调用等）出错了。这些错误称为“异常”，**Python**可以监控这些错误并进行处理。





7) 解释性和编译性

Python是一种解释型语言，即在开发过程中没有编译这个环节。一般来说由于不是以本地机器码运行，纯粹的解释型语言通常比编译型语言运行的慢。

然而类似**Java**，**Python**实际上是字节编译的，其结果就是可以生成类似机器语言的中间形式。这一方面改善了**Python**的性能，又保持了解释型语言的优点。





7) 解释性和编译性

Python源文件通常会以**.py**作为扩展名，当源文件被解释器加载或者显式地进行字节码编译时，会被编译成字节码，生成以**.pyc**或者**.pyo**作为扩展名的文件。





◆ Python与其他语言的对比

◆ Perl

拥有字符串模式匹配能力：**Python**的正则表达式引擎基于**Perl**，用于过滤、识别和抽取字符串文本。

Perl语言对符号语法过度使用，解读困难，耗费时间。

◆ Java

尽管**Java**语法比**C++**简单，但针对小任务时，**Python**提供了更加快速的开发环境。

◆ Ruby

Python是多种编程范式的混合，而**Ruby**是完全面向对象，相对更易阅读。





◆ Python与其他语言的对比

◆ JavaScript

类似Python的面向对象脚本语言，基于原型系统的（克隆对象），而Python遵循传统的面向对象系统。

◆ Python具有函数化编程结构，利于列表解析的特性

lambda

map

filter

reduce





缺点:

(1) 它用缩进来区分语句关系的方式还是给很多初学者带来了困惑，即便是有经验的Python开发者也可能会因为TAB键和空格键混用，使用不同IDLE先后编辑同一文件（如先使用IDLE再使用NotePad++编辑）导致语法错误，肉眼却无法区分。

(2) 在目前现有的实现方式下，与C和C++相比，Python的执行速度还不够快。





◆ 系统编程

Python提供对操作系统服务的内置接口，其标准库绑定了**POSIX**以及其他的常规操作系统的工具：像环境变量，文件，套接字，管道，进程等。

◆ Internet脚本

Python提供了标准**Internet**模块，它使得**Python**可以通过套接字广泛地和网络进行通信。也可以通过高级的库如**httplib**、**urllib**、**urllib2**等进行网络编程。

◆ 数据库编程

Python提供了对所有主流关系数据库系统的接口，如：**Sybase**，**Oracle**，**Informix**，**ODBC**，**MySQL**，**PostgreSQL**，**SQLite**（**Android**）等。





◆ 用户图形接口

Python内置了**TKinter**标准的面向对象接口**Tk GUI API**，使**Python**可以生成可移植的本地观感的**GUI**。通过一些适当的库，可以在**Python**中使用其他的**GUI**工具包，如：通过**PyQt**使用**Qt**、通过**PyGTK**使用**GTK**，通过**PyWin32**使用**MFC**、通过**IronPython**使用**.NET**以及通过**Jython**使用**Swing**等。

◆ 数学计算和科学计算编程

通过和**NumPy**的结合，**Python**可以做很多有关数值计算的工作，如：矩阵计算，和标准数学库结合执行更多的数学建模等。

◆ 其他

还有游戏，图像，人工智能，**XML**，机器人等方面的应用。





①二进制安装

对于Windows平台，可以到<https://www.python.org/>网站上下载安装包，
如：python-3.9.6进行安装。

②源码安装

对于*NIX系列系统，一般系统自带Python，但是版本可能比较低，可以手动下载源码包编译安装：

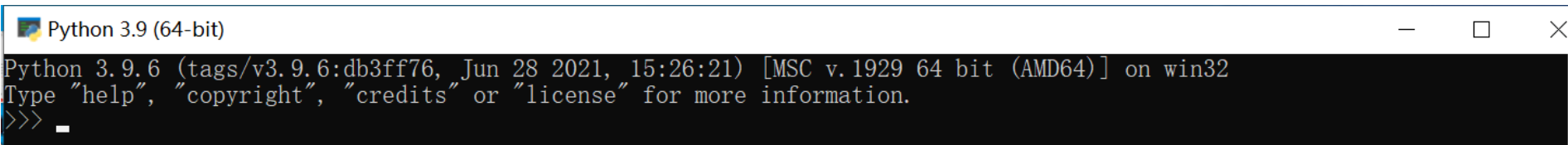
```
./configure  
make  
make install
```



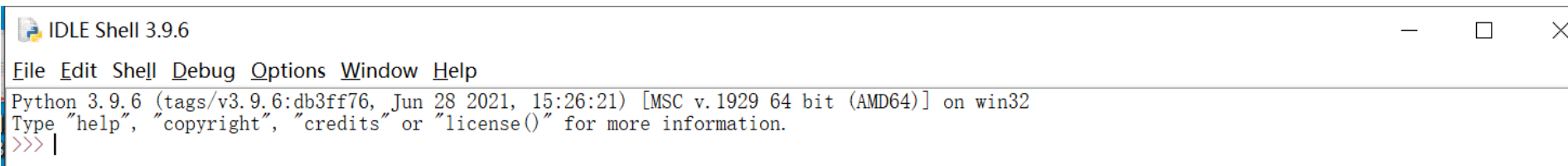


交互式解释器

在**Unix**，**DOS**或任何其他提供了命令行或者**Shell**的系统进行**Python**编码。



IDE运行





命令行脚本

A screenshot of a Windows Command Prompt window. The title bar reads "C:\ 命令提示符". The window content shows the following text:

```
Microsoft Windows [版本 6.1.7601]  
版权所有 (c) 2009 Microsoft Corporation。保留所有权利。  
  
C:\Users\Lib>python C:\Users\Lib\Desktop\test.py  
Hello,World!  
  
C:\Users\Lib>_
```



目前，计算机可以做很多事情，但仍然不能完全自主思考，程序员就像给小孩喂饭一样告诉它具体的细节，并且使用计算机能够理解的方式——算法。

算法包括一些对象（数字、类），以及指令（语句）。

算法---程序执行过程的抽象！

数学模型。。。

◆ 顺序

◆ 条件

◆ 选择





数字:

整数 >>> 2 + 2

浮点数 >>> 1 / 2

十六进制 >>> 0xAF





代表（或者引用）某值的名字。

```
>>> x = 3
```

请注意：在使用变量之前，需要对其赋值！

变量名可以包含字母、数字和下划线（ ）。变量不能以数字开头，如**Plan9**，**Xman5**就是合法的变量名，而**9Plan**、**5Xman**不合法。





表达式与语句的区别？

表达式---某件事情

语句---做某件事情

```
>>> 2*2
```

```
4
```

```
>>> print (2*2)
```

```
4
```





```
>>> 2**3
```

```
8
```

```
>>> pow(2, 3)
```

```
8
```

函数：

用来实现特定的功能，就像小型程序一样。

Python有很多函数，使用者可以自定义函数，通常把pow等标准函数称为内建函数（built-in）。





input函数

```
>>> s = input('-->')  
-->Monty Python's Flying Circus  
>>> s  
"Monty Python's Flying Circus"  
>>>
```





模块是一种组织形式，将相互间有关系的Python代码组织到一个一个独立文件当中。

- 执行代码
- 函数
- 类
- 组合

当创建了一个Python源文件，模块的名字就是不带.py后缀的文件名。

一个模块创建之后，可以从另外一个模块中使用import语句导入使用。





注意：

- 每一个脚本文件都是一个模块
- 模块以磁盘文件的形式存在
- 一个模块不应有太多功能
- 一段直接执行的脚本
- 一堆类似库函数的代码





模块:

可以把模块想象成导入到Python以增强其功能的扩展。

需要使用特殊的命令import来导入模块。

```
>>> import math
```

```
>>> math.floor(32.9)
```

```
32
```

```
>>> int(32.9)
```

```
32
```





思考

- (1) 为什么要使用Python?
- (2) 如何安装和执行Python?
- (3) 尝试打印'Hello World!'

