# 入门

## 为什么学Python

1 便宜

开源自由，可以从Internet免费获得，可以随产品分发，甚至可以出售。

可移植的

2 好学好用

使用优美的语法，贴近人类的表达习惯，使你编写程序更容易。

易于使用

简单易学

核心语言相当简单，有经验的程序员通过几天（甚至几个小时）的学习就可以写出不错的python代码。

Python的类库齐全并且使用简洁，如果要实现同样的功能，Python 10行代码可以解决，C可能就需要100行甚至更多.

开发效率高，通常实线同样的功能，python的代码量只有C++或Java代码的五分之一到三分之一，并且可以立即执行，不需要编译链接等步骤，进一步提升开发效率。

3 功能强大

一个大型的标准库支持许多常见编程任务,诸如连接到Web服务器、 正则表达式全文检索、阅读及修改文件。

可以应用于众多领域，如：系统维护，数据库编程，用户图形接口，网络服务，图像处理，快速原型，数值计算和科学计算，人工智能，大数据，机器人等等。目前业内几乎所有大中型互联网企业都在使用Python，如：Youtube、豆瓣、知乎、Google、Yahoo!、Facebook、NASA、百度、腾讯、汽车之家、美团等。互联网公司广泛使用Python来做的事一般有：自动化运维、自动化测试、大数据分析、爬虫、Web 等。

可移植的

可混合的

缺点： Python与C或C++相比在速度上稍显逊色。

不适合使用Python的场景：编写操作系统，通常用C和汇编；编写手机应用；编写3D游戏，最好用C或C++。

## Python安装

第一步：下载Python安装包

在Python的官网 www.python.org 中找到最新版本的Python安装包，点击进行下载，请注意，当你的电脑是32位的机器，请选择32位的安装包，如果是64位的，请选择64位的安装包；

第二步：安装

A.双击下载好的安装包，弹出如下界面：

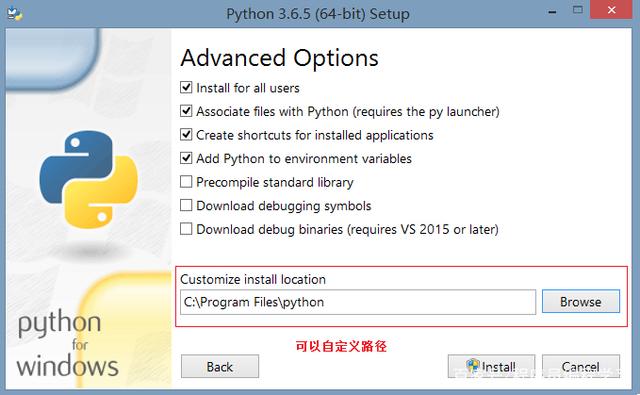


这里要注意的是，将python加入到windows的环境变量中，如果忘记打勾，则需要手工加到环境变量中；在这里我选择的是自定义安装，点击“自定义安装”进行下一步操作；

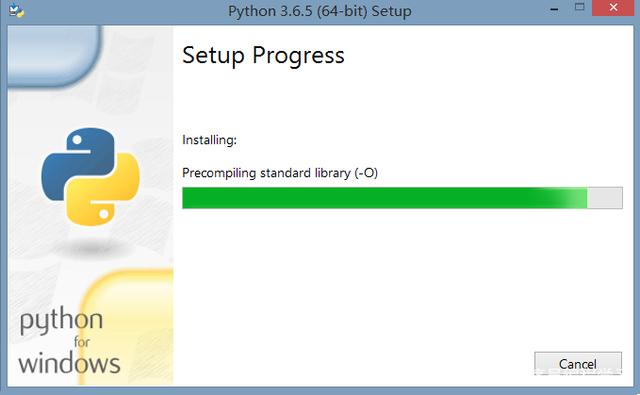
B.进入到下一步后，选择需要安装的组件，然后点击下一步:



C.在这里可以自定义路径选择安装：



D.点击下一步后，就开始真正安装了：



E:安装完成后，会有一个安装成功的提示界面：



第三步：测试

python安装好之后，我们要检测一下是否安装成功，用系统管理员打开命令行工具cmd，输入“python -V”,然后敲回车，如果出现如下界面，则表示我们安装成功了；



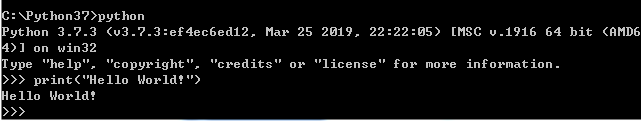
Linux系统通常已经预装了python，也可以通过相应的包管理器安装，比如yum，apt都可以。

## Python程序如何运行

交互命令行

python

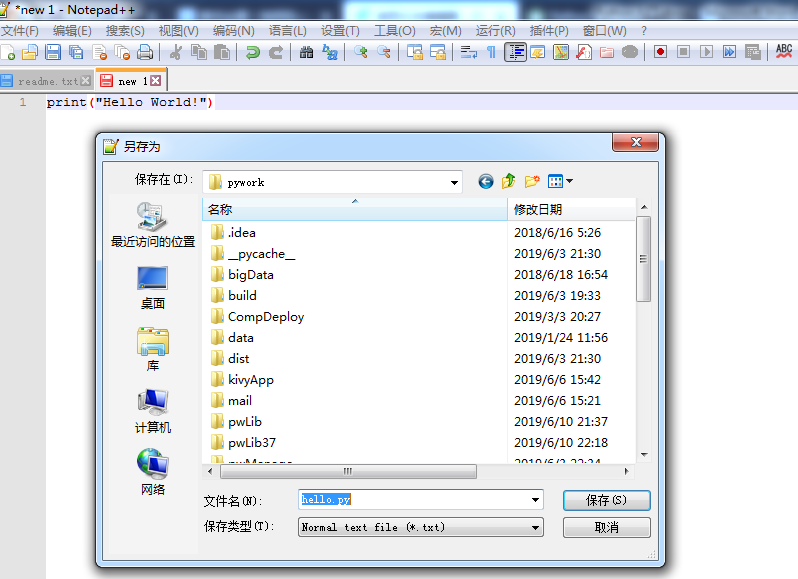
>>>print 'Hello world!'

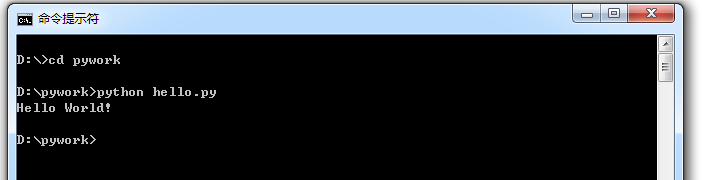


作为Python模块文件

保存为文本文件

python hello.py





作为Unix/Linux文件型脚本

脚本第一行#!解释器路径，赋予可执行权限，即可在命令行直接输入文件名执行

## 基本语法

代码缩进

注释

## 数据类型

常用数据类型：数字(Number)、布尔值(Boolean)、字符串(String)、

元组(Tuple)、列表(List)、字典(Dictionary)、文件(File)

变量定义的规则：

* 变量名只能是 字母、数字或下划线的任意组合
* 变量名的第一个字符不能是数字
* 以下关键字不能声明为变量名  
  ['and', 'as', 'assert', 'break', 'class', 'continue', 'def', 'del', 'elif', 'else', 'except', 'exec', 'finally', 'for', 'from', 'global', 'if', 'import', 'in', 'is', 'lambda', 'not', 'or', 'pass', 'print', 'raise', 'return', 'try', 'while', 'with', 'yield']

**一、作用域**

对于变量的作用域，执行声明并在内存中存在，该变量就可以在下面的代码中使用。

|  |  |
| --- | --- |
| 1  2  3 | if 1==1:      name = 'wupeiqi'  print  name |

**下面的结论对吗？**

外层变量，可以被内层变量使用

内层变量，无法被外层变量使用

基本数据类型

1、数字

int（整型）

　　在32位机器上，整数的位数为32位，取值范围为-2\*\*31～2\*\*31-1，即-2147483648～2147483647  
　　在64位系统上，整数的位数为64位，取值范围为-2\*\*63～2\*\*63-1，即-9223372036854775808～9223372036854775807

long（长整型）　　跟C语言不同，Python的长整数没有指定位宽，即：Python没有限制长整数数值的大小，但实际上由于机器内存有限，我们使用的长整数数值不可能无限大。  
　　注意，自从Python2.2起，如果整数发生溢出，Python会自动将整数数据转换为长整数，所以如今在长整数数据后面不加字母L也不会导致严重后果了。  
float（浮点型）　　浮点数用来处理实数，即带有小数的数字。类似于C语言中的double类型，占8个字节（64位），其中52位表示底，11位表示指数，剩下的一位表示符号。  
complex（复数）  
　　复数由实数部分和虚数部分组成，一般形式为x＋yj，其中的x是复数的实数部分，y是复数的虚数部分，这里的x和y都是实数。

2、布尔值

　　真或假

　　1 或 0

3、字符串

字符串常用功能：

* 移除空白
* 分割
* 长度
* 索引
* 切片

4、列表

创建列表：

|  |  |
| --- | --- |
| 1  2  3 | name\_list = ['alex', 'seven', 'eric']  或  name\_list ＝ list(['alex', 'seven', 'eric']) |

基本操作：

* 索引
* 切片
* 追加
* 删除
* 长度
* 切片
* 循环
* 包含

5、元祖

创建元祖：

|  |  |
| --- | --- |
| 1  2  3 | ages = (11, 22, 33, 44, 55)  或  ages = tuple((11, 22, 33, 44, 55)) |

基本操作：

* 索引
* 切片
* 循环
* 长度
* 包含

6、字典（无序）

创建字典：

|  |  |
| --- | --- |
| 1  2  3 | person = {"name": "mr.wu", 'age': 18}  或  person = dict({"name": "mr.wu", 'age': 18}) |

常用操作：

* 索引
* 新增
* 删除
* 键、值、键值对
* 循环
* 长度

PS：循环，range，continue 和 break

## 数据运算

算数运算 + - \* / % \*\*

比较运算 < > <= >= == !=

赋值运算 =

逻辑运算 and or not

成员运算 in

算数运算：



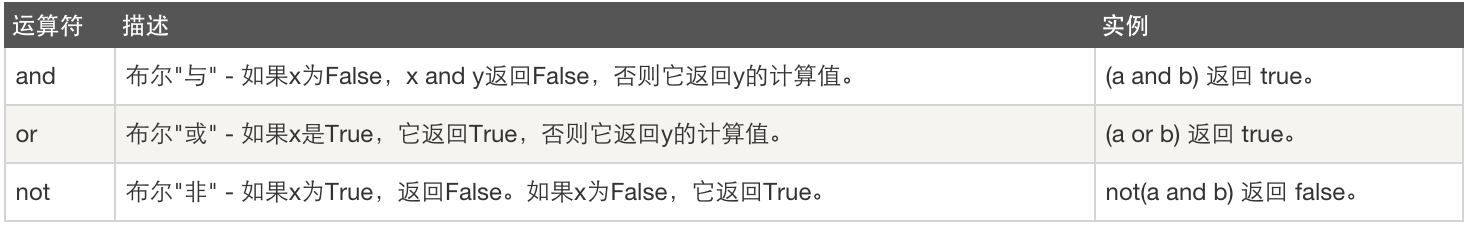
比较运算：



赋值运算：



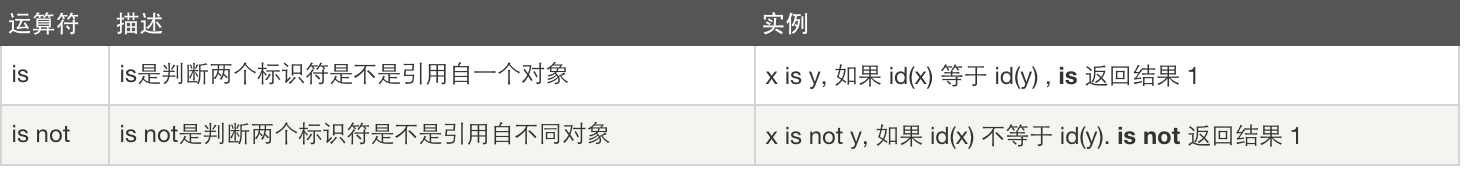
逻辑运算：



成员运算：



身份运算：



位运算：



运算符优先级：



**二、三元运算**

|  |  |
| --- | --- |
| 1 | result = 值1 if 条件 else 值2 |

如果条件为真：result = 值1  
如果条件为假：result = 值2

## 流程控制

分支 if

循环 for

## 函数（模块）设计

## 编程库（包）的使用与设计

## 内置函数与库模块

## 文件操作

excel文件操作，word文件操作

**打开文件：**

　　file\_obj = file("文件路径","模式")

打开文件的模式有：

* r，以只读方式打开文件
* w，打开一个文件只用于写入。如果该文件已存在则将其覆盖。如果该文件不存在，创建新文件。
* a，打开一个文件用于追加。如果该文件已存在，文件指针将会放在文件的结尾。也就是说，新的内容将会被写入到已有内容之后。如果该文件不存在，创建新文件进行写入。
* w+，打开一个文件用于读写。如果该文件已存在则将其覆盖。如果该文件不存在，创建新文件。

**读取文件的内容：**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | # 一次性加载所有内容到内存  obj.read()    # 一次性加载所有内容到内存，并根据行分割成字符串  obj.readlines()    # 每次仅读取一行数据  for line in obj:  　　print line |

**写文件的内容：**

|  |  |
| --- | --- |
| 1 | obj.write('内容') |

**关闭文件句柄：**

|  |  |
| --- | --- |
| 1 | obj.close() |

# 进阶

## Python 面向对象

Python从设计之初就已经是一门面向对象的语言，正因为如此，在Python中创建一个类和对象是很容易的。本章节我们将详细介绍Python的面向对象编程。

如果你以前没有接触过面向对象的编程语言，那你可能需要先了解一些面向对象语言的一些基本特征，在头脑里头形成一个基本的面向对象的概念，这样有助于你更容易的学习Python的面向对象编程。

接下来我们先来简单的了解下面向对象的一些基本特征。

### 面向对象技术简介

* **类(Class):** 用来描述具有相同的属性和方法的对象的集合。它定义了该集合中每个对象所共有的属性和方法。对象是类的实例。
* **类变量：**类变量在整个实例化的对象中是公用的。类变量定义在类中且在函数体之外。类变量通常不作为实例变量使用。
* **数据成员：**类变量或者实例变量, 用于处理类及其实例对象的相关的数据。
* **方法重写：**如果从父类继承的方法不能满足子类的需求，可以对其进行改写，这个过程叫方法的覆盖（override），也称为方法的重写。
* **局部变量：**定义在方法中的变量，只作用于当前实例的类。
* **实例变量：**在类的声明中，属性是用变量来表示的。这种变量就称为实例变量，是在类声明的内部但是在类的其他成员方法之外声明的。
* **继承：**即一个派生类（derived class）继承基类（base class）的字段和方法。继承也允许把一个派生类的对象作为一个基类对象对待。例如，有这样一个设计：一个Dog类型的对象派生自Animal类，这是模拟"是一个（is-a）"关系（例图，Dog是一个Animal）。
* **实例化：**创建一个类的实例，类的具体对象。
* **方法：**类中定义的函数。
* **对象：**通过类定义的数据结构实例。对象包括两个数据成员（类变量和实例变量）和方法。

### 创建类

使用 class 语句来创建一个新类，class 之后为类的名称并以冒号结尾:

class ClassName:

'类的帮助信息' #类文档字符串

class\_suite #类体

类的帮助信息可以通过ClassName.\_\_doc\_\_查看。

class\_suite 由类成员，方法，数据属性组成。

### 实例

## Python 多线程

多线程类似于同时执行多个不同程序，多线程运行有如下优点：

* 使用线程可以把占据长时间的程序中的任务放到后台去处理。
* 用户界面可以更加吸引人，这样比如用户点击了一个按钮去触发某些事件的处理，可以弹出一个进度条来显示处理的进度
* 程序的运行速度可能加快
* 在一些等待的任务实现上如用户输入、文件读写和网络收发数据等，线程就比较有用了。在这种情况下我们可以释放一些珍贵的资源如内存占用等等。

线程在执行过程中与进程还是有区别的。每个独立的进程有一个程序运行的入口、顺序执行序列和程序的出口。但是线程不能够独立执行，必须依存在应用程序中，由应用程序提供多个线程执行控制。

每个线程都有他自己的一组CPU寄存器，称为线程的上下文，该上下文反映了线程上次运行该线程的CPU寄存器的状态。

指令指针和堆栈指针寄存器是线程上下文中两个最重要的寄存器，线程总是在进程得到上下文中运行的，这些地址都用于标志拥有线程的进程地址空间中的内存。

* 线程可以被抢占（中断）。
* 在其他线程正在运行时，线程可以暂时搁置（也称为睡眠） -- 这就是线程的退让。

### 开始学习Python线程

Python中使用线程有两种方式：函数或者用类来包装线程对象。

函数式：调用thread模块中的start\_new\_thread()函数来产生新线程。语法如下:

thread.start\_new\_thread ( function, args[, kwargs] )

参数说明:

* function - 线程函数。
* args - 传递给线程函数的参数,他必须是个tuple类型。
* kwargs - 可选参数。

## 正则表达式

## GUI编程(Tkinter)

Python 提供了多个图形开发界面的库，几个常用 Python GUI 库如下：

* **Tkinter：** Tkinter 模块(Tk 接口)是 Python 的标准 Tk GUI 工具包的接口 .Tk 和 Tkinter 可以在大多数的 Unix 平台下使用,同样可以应用在 Windows 和 Macintosh 系统里。Tk8.0 的后续版本可以实现本地窗口风格,并良好地运行在绝大多数平台中。
* **wxPython：**wxPython 是一款开源软件，是 Python 语言的一套优秀的 GUI 图形库，允许 Python 程序员很方便的创建完整的、功能健全的 GUI 用户界面。
* **Jython：**Jython 程序可以和 Java 无缝集成。除了一些标准模块，Jython 使用 Java 的模块。Jython 几乎拥有标准的Python 中不依赖于 C 语言的全部模块。比如，Jython 的用户界面将使用 Swing，AWT或者 SWT。Jython 可以被动态或静态地编译成 Java 字节码。

## Tkinter 编程

Tkinter 是 Python 的标准 GUI 库。Python 使用 Tkinter 可以快速的创建 GUI 应用程序。

由于 Tkinter 是内置到 python 的安装包中、只要安装好 Python 之后就能 import Tkinter 库、而且 IDLE 也是用 Tkinter 编写而成、对于简单的图形界面 Tkinter 还是能应付自如。

**注意**：Python3.x 版本使用的库名为 tkinter,即首写字母 T 为小写。

import tkinter

创建一个GUI程序

* 1、导入 Tkinter 模块
* 2、创建控件
* 3、指定这个控件的 master， 即这个控件属于哪一个
* 4、告诉 GM(geometry manager) 有一个控件产生了。

### 实例

#!/usr/bin/python # -\*- coding: UTF-8 -\*- import Tkinter top = Tkinter.Tk() # 进入消息循环 top.mainloop()

以上代码执行结果如下图:

## 数据库操作

## Web开发-Flask框架介绍

## 网络爬虫

## 科学计算-numpy, pandas

## 图形pillow

# 附录

## Python IDE

### PyCharm

PyCharm 是由 JetBrains 打造的一款 Python IDE。

PyCharm 具备一般 **Python IDE** 的功能，比如：调试、语法高亮、项目管理、代码跳转、智能提示、自动完成、单元测试、版本控制等。

另外，PyCharm 还提供了一些很好的功能用于 Django 开发，同时支持 Google App Engine，更酷的是，PyCharm 支持 IronPython。

PyCharm 官方下载地址：<http://www.jetbrains.com/pycharm/download/>

### Sublime Text

Sublime Text 具有漂亮的用户界面和强大的功能，例如代码缩略图，Python 的插件，代码段等。还可自定义键绑定，菜单和工具栏。

Sublime Text 的主要功能包括：拼写检查，书签，完整的 Python API ， Goto 功能，即时项目切换，多选择，多窗口等等。

Sublime Text 是一个跨平台的编辑器，同时支持 Windows、Linux、Mac OS X等操作系统。

### Eclipse+Pydev

#### 1、安装Eclipse

Eclipse可以在它的官方网站[Eclipse.org](http://eclipse.org/)找到并下载，通常我们可以选择适合自己的Eclipse版本，比如Eclipse Classic。下载完成后解压到到你想安装的目录中即可。

当然在执行Eclipse之前，你必须确认安装了Java运行环境,即必须安装JRE或JDK，你可以到（<http://www.java.com/en/download/manual.jsp>）找到JRE下载并安装。

#### 2、安装Pydev

运行Eclipse之后，选择help-->Install new Software，如下图所示。

### [vscode](https://code.visualstudio.com/)

安装插件简单方便，可中文，打开速度快，缺点：相对pycharm，提示没那么好

### [atom](https://atom.io/)

安装插件简单方便，不常用