

一、Python 入门、安装



1-Python 简介

Python 是一种面向对象的解释型计算机程序设计语言。

Python 的作者，Guido van Rossum（龟叔），荷兰人。



1982 年，Guido 从阿姆斯特丹大学获得了数学和计算机硕士学位。

1989 年圣诞节期间，在阿姆斯特丹，Guido 为了打发圣诞节的无聊，决心开发一个新的脚本解释程序——Python（n. 巨蟒；大蟒）。

1991 年，第一个 Python 编译器诞生。它是用 C 语言实现的，并能够调用 C 语言的库文件。

Python 优、缺点

优点：简单易学、胶水语言、丰富的库

缺点：执行效率(解决办法：可用 C/C++ 改写代码的关键部分)

Python 应用领域

web 应用开发：Django、Flask

爬虫数据采集：Scrapy、pyspider

服务器运维：Tornado、Twisted

自动化测试：Selenium

科学计算：Numpy、Pandas、Scipy、Matplotlib

机器学习：Scikit-learn

深度学习：Tensorflow、Caffe

Python 版本选择

Python2

Python3（课程推荐）

Anaconda

. . .

May 2018	May 2017	Change	Programming Language	Ratings	Change
1	1		Java	16.380%	+1.74%
2	2		C	14.000%	+7.00%
3	3		C++	7.668%	+2.92%
4	4		Python	5.192%	+1.64%
5	5		C#	4.402%	+0.95%
6	6		Visual Basic .NET	4.124%	+0.73%
7	9	▲	PHP	3.321%	+0.63%
8	7	▼	JavaScript	2.923%	-0.15%
9	-	▲	SQL	1.987%	+1.99%
10	11	▲	Ruby	1.182%	-1.25%

Python 相关资料

Python 官网：<https://www.python.org/>

Python 下载地址：<https://www.python.org/downloads/>

《Python 基础》课程第 1 次教学活动讲义

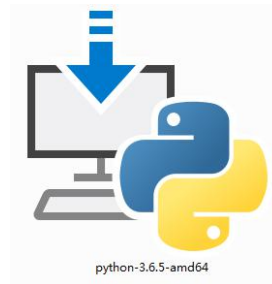
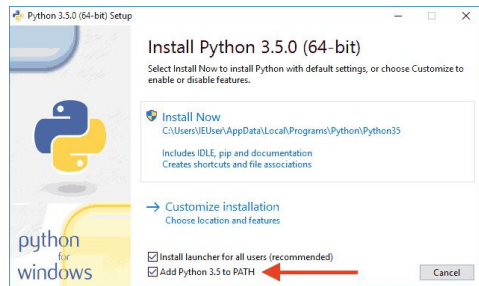
TIOBE 排行榜: <https://www.tiobe.com/tiobe-index/>

2-Python 安装

1、Python 下载地址: <https://www.python.org/downloads/> (按系统进行选择下载)

2、下载完成之后, 直接双击运行即可

注意:



(老师带学生走一遍流程)

3-运行 Python 程序的三种方法

交互式解释器 (无法保存打好的程序, 不推荐)

命令行输入: python, 即可进入 Python 的开发环境

```
C:\Users\Administrator>python
Python 3.6.4 (v3.6.4:d48ecef, Dec 19 2017, 06:54:40) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
```

命令行脚本 (开发界面不友好, 效率太慢, 不推荐)

命令行输入: python xxx.py

```
C:\Users\Administrator\Desktop>python first.py
Here is my first program.
```

集成开发环境 (推荐使用 Pycharm)

Pycharm 官网: <https://www.jetbrains.com/pycharm/>

Pycharm 安装教程: (一路 Next 即可, 太简单了, 略)

Pycharm 破解教程: (随版本不同而有所变动)

<https://blog.csdn.net/u014044812/article/details/78727496>



Pycharm 快捷键

- 1、基础快捷键
- 2、Ctrl + Alt + S: 设置
- 3、Ctrl + Shift + F10: 运行当前程序

- 4、Ctrl + /：快速注释
- 5、Alt + 1：调出侧边栏

4-第一个 Python 程序

```
print("Hello world!")
```

5-随堂练习

- 1、请编写程序输出“我爱 Python”；
- 2、练习使用 3 种方式运行 Python 程序；

二、Python-基础概念

1-注释

问题引入-请说明下面程序做了什么事情？

```
from sklearn.datasets import load_boston
from sklearn.model_selection import train_test_split
x, y = load_boston(return_X_y=True)
x1, x2, y1, y2 = train_test_split(x, y, test_size=0.3, random_state=1)
```

这样呢？

```
# 线性回归-评估
# 加载数据
from sklearn.datasets import load_boston
from sklearn.model_selection import train_test_split
x, y = load_boston(return_X_y=True)
x1, x2, y1, y2 = train_test_split(x, y, test_size=0.3, random_state=1)
```

注释：在程序中，起到辅助说明的提示文字，增加程序的可读性。

```
# 我是单行注释，只需要在前面加上一个#号

"""
我是多行注释，只需要在前后都加上三个双引号
"""

'''
我也是多行注释，只需要在前后都加上三个单引号
'''
```

2-变量

```
a = 1
b = 'Tony'
```

类似 a, b 就是变量

3-标识符

在开发过程中，常常需要开发人员自定义一些名称，这样的名称就称为标识符，如变量名、函数名等。

命名规则：标识符由字母、下划线和数字组成，且数字不能开头
变量命名规范：见名知意、驼峰命名法

```
xxx = 18
age = 18

studentname = 'Peter'
StudentName = 'Peter'
```

随堂练习：哪些命名是错误的？

```
mike
ae86
_exo
_@163
l1p82ndlf
1234
```

4-关键字

关键字：Python 一些具有特殊功能的标识符

```
import keyword
print(keyword.kwlist)
```

5-变量类型

查看变量类型使用：type()

1、数字(number)

- 2、布尔(bool)
- 3、字符串(str)
- 4、列表(list)
- 5、元组(tuple)
- 6、集合(set)
- 7、字典(dict)

6-Python 数字

- 整型(int)
- 浮点型(float)
- 复数(complex)

```
# 整型(int)
a = 1
print(a)
print(type(a))

# 浮点型(float)
b = 2.0
print(b)
print(type(b))

# 科学计数法，也是浮点型
c = 3e4
print(c)
print(type(c))

# 复数(complex)，不是很常用
e = 5 + 6j
print(e)
print(type(e))
```

7-运算符

赋值运算符

运算符	描述
=	赋值

- 特殊用法
- a = b = 1
- a, b = b, a

《Python 基础》课程第 1 次教学活动讲义

算术运算符

运算符	描述
+	加
-	减
*	乘
/	除
//	取整
%	取余
**	求幂

复合赋值运算符

运算符	描述	实例
+=	加法赋值运算符	<code>c += a</code> 等效于 <code>c = c + a</code>
-=	减法赋值运算符	<code>c -= a</code> 等效于 <code>c = c - a</code>
*=	乘法赋值运算符	<code>c *= a</code> 等效于 <code>c = c * a</code>
/=	除法赋值运算符	<code>c /= a</code> 等效于 <code>c = c / a</code>
//=	取整除赋值运算符	<code>c //= a</code> 等效于 <code>c = c // a</code>
%=	取模赋值运算符	<code>c %= a</code> 等效于 <code>c = c % a</code>
**=	求幂赋值运算符	<code>c **= a</code> 等效于 <code>c = c ** a</code>

8-随堂练习

- 1、定义 5 个数字
- 2、熟练运用数字变量和运算符编写代码

三、Python-字符串、输出与输入

1-字符串定义(str)

```
# 单行字符串
a = '字符串'
b = "字符串"
# 多行字符串
c = """字符串"""
d = """字符串"""
```

2-字符串的运算

```
# 字符串加法
```

《Python 基础》课程第 1 次教学活动讲义

```
a = 'Tony' + 'Stark'
# 字符串乘法
b = '6' * 3
```

3-数据类型强转

```
# age 的数据类型是数字（整型）
a = 18

# b 的数据类型是字符串
b = str(a)

print(type(b))
```

eval()的用法比较特殊，这里需要单独强调一下

```
# eval()可以把字符串转化为 Python 表达式
a = eval('6+3')

print(a)
print(type(a))
```

4-格式化输出

普通输出

print()

常用格式化输出

格式化输出	输出类型
%s	字符串
%d	整型
%f	浮点型

```
# 重点掌握以下三个，剩下的作为了解
name = 'Gary'
print('My name is %s' % name)

age = 18
print("I'm %d year's old" % age)

height = 176.5
print("I'm %f cm tall" % height)
```

《Python 基础》课程第 1 次教学活动讲义

```
# #####3
# %c 字符 http://ascii.911cha.com/
print('%c' % 64)

# %i 有符号十进制整数
print('%i' % -50)
# %u 无符号十进制整数
print('%u' % -50)

# %o 八进制整数
print('%o' % 50)

# %x 十六进制整数(小写字母)
# %X 十六进制整数(大写字母)
print('%x' % 60)
print('%X' % 60)

# %e 科学计数法(小写'e')
# %E 科学计数法(大写'E')
print('%e' % 50)
print('%E' % 50)

# %g %f 和 %e 的简写
# %G %f 和 %E 的简写
print('%g' % 50)
print('%G' % 50)
```

5-格式化输出（进阶）

```
height = 176.5
# 初始情况
print("I'm %f cm tall" % height)
print("***40)

# %X.Yf
print("I'm %8.2f cm tall" % height)
print("***40)

# +代表保留正负号
print("I'm %+8.2f cm tall" % height)

# -代表左对齐
```


《Python 基础》课程第 1 次教学活动讲义

```
print("I'm %-8.2f cm tall" % height)
# 空格代表用空格填满多余位置
print("I'm % 8.2f cm tall" % height)
# 0 代表用 0 填满多余位置
print("I'm %08.2f cm tall" % height)
```

格式化输出(共 6 位)	意义
%	格式化输出的符号，必须写
+	格式化输出的格式，可选(+、-、0、空格)
0	输出的总宽度，不满足则忽略
.	小数点，不写就是不精确，但这样没有意义
2	小数点后，精确到第几位
f	输出的格式

6-输入

`input()`: 把用户输入的全部数据作为一个字符串

```
name = input('请输入您的姓名:')

print(name)
```

7-随堂练习

```
# 1、输出个人名片
# ***** (这里有 20 个 “*” 号)
# 姓名: (这里填写自己的姓名)
# 手机号: (这里填写自己的手机号)
# QQ: (这里填写自己的 QQ)
# 身高: 1.81m (用格式化输出身高，精确到两位小数)
# 体重: 75.50kg (用格式化输出体重，精确到两位小数)
# ***** (这里有 20 个 “*” 号)

# 2、提示用户进行输入数据
# 第一条提示: 请输入苹果的价格: (元/斤)
# 第二条提示: 请输入苹果的斤数: (斤)
# 最后输出: 请支付 元
```