

# Python

张容康  
*2150931*

# 目录

<b>第一部分 写在学前</b>	<b>4</b>
<b>第一章 课程安排</b>	<b>6</b>
1.1 上课时间、地点、座位 . . . . .	6
1.2 成绩组成 . . . . .	6
1.3 教材、作业 . . . . .	6
<b>第二章 资源</b>	<b>7</b>
2.1 课程讲义 . . . . .	7
2.2 GoodNotes . . . . .	7
2.3 B 站 . . . . .	8
<b>第三章 指南、问题、错误、日常</b>	<b>9</b>
3.1 指南 . . . . .	9
3.2 问题 . . . . .	9
3.3 错误 . . . . .	9
3.4 日常 . . . . .	9
<b>第二部分 课程讲义</b>	<b>10</b>
<b>第四章 Python 程序设计基础（上）</b>	<b>12</b>
<b>第三部分 GoodNotes</b>	<b>13</b>
<b>第五章 Python 编程：从入门到实践</b>	<b>15</b>
5.1 变量和简单数据类型 . . . . .	15
5.2 列表简介 . . . . .	16
5.3 操作列表 . . . . .	18
5.4 if 语句 . . . . .	20
<b>第四部分 B 站</b>	<b>22</b>
<b>第六章 花了 2 万多买的 Python 教程全套</b>	<b>24</b>
6.1 Python 自述 . . . . .	24

6.2 基础语法 . . . . . 25

第五部分 杂记 26

## 第一部分

## 写在学前



# 第一章 课程安排

## 1.1 上课时间、地点、座位

时间：每周三晚第 10-12 节课

地点：南楼 413 教室（南楼四楼中央教室）、机房

座位：靠窗组最后一排内座，方便电脑充电

## 1.2 成绩组成

出勤：上课出勤、上机出勤

作业：网上作业、随堂作业

考试：期中考试、期末考试

平时成绩占 20%。

作业提交、测验网址为 <http://jsjjc.tongji.edu.cn>(同济大学计算机基础教研室)。

## 1.3 教材、作业

教材：Python 程序设计基础（上）

作业：实验作业与补充作业

## 第二章 资源

### 2.1 课程讲义

Python 程序设计基础（上）

### 2.2 GoodNotes

贝叶斯思维统计建模的 Python 学习法

从 Excel 到 Python

集体智慧编程

利用 Python 进行数据分析

零起点 Python 机器学习快速入门

流畅的 Python

深度学习入门

数据科学入门

网络爬虫，Python 和数据分析

用 Python 写网络爬虫

征服 Python

A Primer on Scientific Programming with Python

Effective Python

Head First Python

Learning Python

MySQL Cookbook

Natural Language Processing with Python

PyQt5 快速开发与实战

Python Cookbook

Python Machine Learning

Python 编程：从入门到实践

Python 编程（上）

Python 编程（下）

Python 编程快速上手

Python 标准库

Python 参考手册

Python 高级编程  
Python 高级编程 (1)  
Python 核心编程  
Python 灰帽子——黑客与逆向工程师的编程之道  
Python 机器学习及实践  
Python 技术手册  
Python 深度学习  
Python 网络编程  
Python 学习手册  
Python 游戏编程快速上手  
Python 源码剖析  
Python 中文官方文档

## 2.3 B 站

花了 2 万多买的 Python 教程全套



## 第三章 指南、问题、错误、日常

### 3.1 指南

Python 的学习需要每天坚持，坚持是进步的催化剂。每天坚持学习 Python，不但是知识和能力的积累，还是导致质变的必要条件。

### 3.2 问题

使用 `import*` 导入库中的所有函数和类，会不会存在命名重复的情况并产生覆盖。  
如何查看库中含有哪些函数和类。  
PyCharm 如何安装第三方库。

### 3.3 错误

GoodNotes-Python 编程：从入门到实践-P52-squares.py-❸：缩进错误。

### 3.4 日常

2022.3.16 开始学习 Python。

2022.3.16	做第一次上机实验补充内容 听晚课 学习 GoodNotes-Python 编程：从入门到实践 学习 B 站-花了 2 万多买的 Python 教程全套
2022.3.17	学习 GoodNotes-Python 编程：从入门到实践
2022.3.18	学习 GoodNotes-Python 编程：从入门到实践
2022.3.19	学习 GoodNotes-Python 编程：从入门到实践
2022.3.20	学习 GoodNotes-Python 编程：从入门到实践

## 第二部分

## 课程讲义



## 第四章 Python 程序设计基础（上）

## 第三部分

### GoodNotes



## 第五章 Python 编程：从入门到实践

### 5.1 变量和简单数据类型

#### 变量、字符串

为变量赋值的通用格式为：

```
变量 = '语句'
```

变量名只能包含字母、数字、下划线，不能包含空格，开头只能使用字母或者下划线。

修改字符串的大小写：

```
变量.title/upper/lower()
```

分别可使字符串首字母大写、全部大写、全部小写。

字符串的拼接：

```
print('语句 1' + 变量 + '语句 2')
```

```
变量 2='语句 1' + 变量 1 + '语句 2'
```

#### 空白

添加空白：在字符串中使用'\n' 和'\t' 来产生空行或者制表符。换行和制表符本质上是空字符串，所以需要加上引号。在字符串中换行直接在相应位置使用\n 即可。

删除空白的格式为：

```
变量.lstrip/strip/rstrip()
```

即可删除左边、左右、右边的空白。

### Python 之禅

输入 `import this`，运行即输出 Python 之禅。

## 5.2 列表简介

### 创建列表

创建列表的基本格式为：

```
列表名称 = ['元素 1', '元素 2', ..., '元素 n']
```

当列表元素为空时创建为空列表。

### 访问、使用列表元素

访问、使用列表元素的基本格式为：

```
列表名称.[元素位置]
```

其中元素位置可以采取正向也可采取反向，正向第一个元素序号为 0。

### 修改列表元素

修改列表元素的格式为：

```
列表名称.[ ] = '新元素名称'
```

列表末尾添加元素的格式为：

```
列表名称.append('新元素名称')
```

列表中插入元素的格式为：

```
列表名称.insert(元素位置, '新元素名称')
```



根据位置删除列表元素的格式为：

```
del 列表名称.[元素位置]
```

根据值删除列表元素的格式为：

```
列表名称.remove('元素名称'/变量名称)
```

弹出列表元素的格式为：

```
列表名称.pop(元素位置)
```

当括号内为空时默认弹出末尾元素。

### 组织列表

按字母顺序排列：

```
列表名称.sort()
```

括号中使用 `reverse=True` 可按字母顺序反向排列。

反转排列顺序：

```
列表名称.reverse()
```

确定列表长度：

```
len(列表名称)
```

## 5.3 操作列表

### 遍历列表

遍历列表的格式为：

```
for x in 列表名称:
```

其中，x 为任意指定的变量名称，注意 for 语句结束要使用冒号，for 语句之后采用适当缩进的代码行为对列表元素采取的操作。

### range 函数、数字列表

创建数字列表：

```
列表名称 = list(range(左闭, 右开, 步长))
```

遍历 range 函数：

```
for x in range(左闭, 右开, 步长):
```

x 为任意指定的数字列表中的变量名称。

统计数字列表：

```
min/max/sum(列表名称)
```

依次可对数字列表中的元素求最大值、最小值、和。

列表解析：

```
列表名称 =[含数学符号的表达式 for 任意变量名 in range(左闭, 右开, 步长)]
```

### 切片

选取切片：

```
列表名称 [切片左端: 切片右端]
```

省略切片左端，默认从 0 位置开始；省略切片右端，默认到-1 结束；省略切片左端和有段，默认输出整个列表。

遍历切片：

```
for 变量名称 in 列表名称 [起始位置: 结束位置]:
```

复制列表：

```
复制列表名称 = 原列表名称 [:]
```

这是产生的是两个分别独立的列表。

如果采用以下格式：

```
复制列表名称 = 原列表名称
```

此时两列表相互关联，实际上为同一列表。

## 元组

元组与列表的区别是单个元素是否可以改变，其余操作与列表相同。

元组的格式为：

```
元组名称 =(变量, '字符串')
```

访问元组元素：

```
元组名称 [元素位置]
```

元组元素不可单独修改，但可为整个元组赋值：

```
元组名称 =(新元素)
```

## PEP8 格式设置指南

访问地址：<https://peps.python.org/pep-0008/>。

## 5.4 if 语句

### 条件测试

if 后表达式称为条件测试，值为 True 执行下一级，值为 False 忽略下一级。

检查相等/不相等：

if 变量 ==/!= '字符串'/数值:

比较数值：

if 变量 ==/!= >/</>=/<= 数值:

and、or：

and、or 用于条件表达式中判断多个条件:

if '条件表达式 1' and/or '条件表达式 2':

检查元素是否在列表中：

if 变量/'字符串' in/not in 列表名称:

if 语句同 for 函数一样，需要在句末使用冒号。

### if-elif-else 结构

if 条件测试 1:

    执行语句 1

elif 条件测试 2:

    执行语句 2

...

else:

    执行语句 n

### if 语句处理列表

示例:

if 列表名称:

    for 变量名称 in 列表名称:

        执行语句 1

    else:

        执行语句 2

else:

    执行语句 3

第一个 if 语句用于判断列表是否为空，为空执行 else。

## 5.5 字典

**操作字典**

创建字典:

字典名称 = {'键名称': 值字符串/数值,...}

访问字典:

字典名称 ['键名称']

添加键值对:

字典名称 ['键名称'] = 值字符串/数值]

## 第四部分

### B 站



# 第六章 花了 2 万多买的 Python 教程全套

## 6.1 Python 自述

### Python 学习思路

基础课程（自动化运维、多媒体处理、人工智能应用、自动化办公、Web 开发、Python 爬虫）→ 科学计算（数据分析、数据处理、量化交易）→ 机器学习（数据挖掘、搜索算法、机器学习算法、推荐算法）→ 深度学习。

### Python 学习和就业方向

(1) Web 全栈开发方向：

前端开发、数据库管理、后台框架技术 → Web 全栈开发工程师；

(2) 数据科学方向：

数据库管理、数据分析、数据可视化、能够制作数据看板、实现数据指标监控 → 数据产品经理、量化交易、初级 BI 商业分析师。

(3) 人工智能方向-机器学习：

掌握机器学习常用算法和思想、利用 Python 建立机器学习模型、对一些应用场景进行智能化 → 数据分析工程师、机器学习算法工程师、搜索算法工程师、推荐算法工程师。

(4) 人工智能方向-深度学习：

掌握深度学习常用框架、自行搭建图像相关算法、自行搭建 NLP 相关算法、掌握 GAN 网络相关算法 → 人工智能工程师。

### Python 语言特点

(1) 是一种跨平台的计算机程序设计语言；

(2) 是一种解释性的语言，没有编译的环节；

(3) 是一种交互式语言；

(4) 是面向对象语言。

### Python 解释器

解释器是一种计算机程序，可将高级程序语言转换为机器代码。其中：

IDLE 是 Python 自带的简单开发环境；

Python 3.x(32/64-bit) 是交互式命令行程序；



Manuals 是官方技术文档；

Module Docs 是已安装的模块文档。

### Python 编译器

编译器可将高级语言编写的程序转换成机器代码，将人可读的代码转换成计算机可读的代码（0 和 1）。

## 6.2 基础语法

### 输出函数 print

输出函数 print 可以输出到显示器或者文件中。

整数、浮点数、变量、含有运算符的表达式（输出运算结果）可以直接输出，字符串需要加上引号输出。

同一个 print 函数中的内容输出在同一行，不同 print 函数中的内容会换行输出。

输出到指定文件的方式为：

```
fp=open(' 文件地址','a+')
```

```
print(输出内容,file=fp)
```

```
fp.close()
```

如果指定文件位置存在，则在原文件后追加新内容。

如果需要输出多个字符串，字符串之间需要用逗号分割，如：

```
print(' 字符串 1',' 字符串 2')
```

## 第五部分

### 杂记



---

## 输入函数 input

通用格式为：

```
变量 =input('提示语')
```

此时赋予的数据类型均为 string 型。

进行算数运算时，强制转换数据类型的格式为：

```
变量 =int/float(input('提示语'))
```

在提示语中加入变量的方式为：

```
变量 1=input( '提示语 1' +str(变量 2)+ '提示语 2' )
```

同时输入多个变量的格式为：

```
变量 1, 变量 2, ..., 变量 n=input('提示语').split()/('(',')')
```

输入时，可以用用空格分离变量值，也可以用逗号分离变量值。此时返回值为一个列表，不能直接转换类型。

用输入函数为多个变量赋予其他数据类型的方法为：

```
变量 1, 变量 2, ..., 变量 n=map(int/float,input('提示语').split()/('(',')'))
```

## 内置函数 chr 与进制

使用格式为：

```
变量 =chr(进制提示符 相应进制下的 ASCII 码)
```

0b 为二进制，0o 为八进制，0d 为十进制，0x 为十六进制。输入值为相应进制下的 ASCII 码，返回值为对应的字符。

将十进制下的字符转换为其他进制下的 ASCII 码方法为：

```
bin(),oct(),hex()
```

---

可依次将十进制转换为二进制、八进制、十六进制的 ASCII 码。

其他进制转换为十进制的方法：

```
eval('0xASCII 码')
```

## 保留小数位数

(1)

变量 = 浮点数

```
round(变量, 有效数字位数)
```

(2)

```
'%.nf'%a
```

其中，n 为保留小数位数，f 表示数据类型为浮点数。

## 随机数

随机生成一个 [0,1) 的浮点数：

```
random.random()
```

随机产生一个 [a,b) 的浮点数：

```
random.uniform(a,b)
```

随机生成一个 [a,b] 的整数：

```
random.randint(a,b)
```

---

从一个序列中随机选取一个元素：

```
random.choice(sequence)
```

PyCharm 中设置 Python File 开头模板

设置路径为：

File→Settings→Editor→Code Style→File and Code Templates→Python Script