

Python 基础知识

一. 1-5 章

1. 安装 python;
2. 变量: 使用变量时要注意如下规则
变量名只能包含字母、数字和下划线; 变量名可以字母或下划线打头, 但不能以数字打头; 变量名不能包含空格, 但可使用下划线来分隔其中的单词; 不要将 Python 关键字和函数名用作变量名, 即不要使用 Python 保留用于特殊用途的单词。
3. 字符串。
4. 数字
可用+*/运算
5. 注释: 用#标明
6. 列表:
通常用[]表示, 可用索引调用列表中的元素, 可对列表进行添加、删除、遍历等操作。
7. 元组:
用()表示, 可遍历和修改
8. if 语句
简单举例:

```
for car in cars: if car == 'bmw': print(car.upper())
else: print(car.title())
```

二. 6-10 章

1. 字典:
在 Python 中, 字典是一系列键-值对。每个键都与一个值相关联, 你可以使用键来访问与之相关联的值。与键相关联的值可以是数字、字符串、列表乃至字典。事实上, 可将任何 Python 对象用作字典中的值。在 Python 中, 字典用放在花括号{}中的一系列键-值对表示。可进行删除, 修改, 遍历, 嵌套等操作。
2. 输入
使用 `int()`;
3. while 循环:
举 例 :

```
current_number = 1 while current_number <= 5:
print(current_number) current_number += 1
```
4. 函数
函数是带名字的代码块, 用于完成具体的工作。 关键字 `def`。
5. 类
面向对象编程 是最有效的软件编写方法之一。在面向对象编程中, 你编写表示现实世界中的事物和情景的类, 并基于这些类来创建对象。编写类时, 你定义一大类对象都有的通用行为。基于类创建对象时, 每个对象都自动具备这种通用行为, 然后可根据需要赋予每个对象独特的个性。

使用面向对象编程可模拟现实情景，其逼真 程度达到了令你惊讶的地步。 根据类来创建对象被称为实例化，这让你能够使用类的实例。在本章中，你将编写一些类并创建其实例。你将指定可在实例中存储什么信息，定义可对这些实例执行 哪些操作。你还将编写一些类来扩展既有类的功能，让相似的类能够高效地共享代码。你将把自己编写的类存储在模块中，并在自己的程序文件中导入其他程序员编写的类。

6. 文件和异常