

python基础语法总结

• 标识符

- 在python中凡是我们自己取得名字，都是标识符，
- 例如：变量名、函数名、类名、模块名、包名
- 规范：标识符由字母、数字、下划线组成,但不能以数字开头，注意点：不能用python的关键字来命名标识符。

一、数据类型

1、int：整数

2、float:浮点数

3、bool：布尔类型

- 只有True和False两个值

4、str：字符串

- 1、字符串的表示：单引号、双引号、三引号
- 2、字符串的下标取值和切片操作
- 3、字符串转义
 - `\t \n r`
- 4、字符串拼接
 - `".join([a1,s2,s3]),`
 - `"■"`
- 5、字符型格式化输出
 - `format`方法
 - 传统方式`%`
- 6、字符串的常用方法：
 - `format`:
 - `join`:
 - `find`:

- count:
- replace:
- split:
- upper:
- lower:

5、list: 列表

- 1、列表的表示方式（定义）：中括号
- 2、列表的下标取值和切片操作
- 3、列表常用的方法
 - 增：append、insert、extend
 - 删：pop、remove、clear
 - 查：index、count
 - 改：通过下标修改值
 - 其他：copy、reverse、sort

6、tuple : 元组

- 1、元组的表示方式（定义）：小括号
- 2、元组的下标取值和切片操作
- 3、元组的方法：
 - count:
 - index:

7、dict:字典

- 1、字典的表示:花括号 {}, 字典中每一个元素都是 由{key:value} 键值对组成。
 - 注意点：字典中的键必须是唯一的，键必须是不可变类型的（一般情况下都是用字符串），值可以是任意类型的数据（字典，元组、列表等等都可以）
- 2、字典的增删查改
 - 添加元素
 - 通过指定键去添加对应的值

```
dic = {"name":999}
dic["name"] = "小明"
print(dic["name"])
print(dic.get("name"))
```

- update方法:

- 修改元素

- 通过指定键去修改对应的值

- 查找元素

- 通过键去查找对应的值

- get:

- keys:

- values:

- itmes:

- 删除元素

- pop

- popitem:

8、set：集合

- 1、集合的表示：花括号{}
- 2、注意点：空集合使用 set()来定义
- 3、特性一：集合中不可存在重复的元素
 - 利用集合来对列表去重
- 4、特性二：集合中只能存放不可变类型的数据
 - 利用集合来区分可变还是不可变类型

9、其他知识

- 数据根据是否可变分类：
 - 可变类型：列表、字典、集合
 - 不可变类型：数值 (int,float,bool) ,字符串, 元组
- 随机数模块：random

- `random.random()`:生成0-1之间的浮点数。
- `random.randint()`:生成指定范围的整数。

二、运算符

1、算术运算符

- `+` `-` `*` `/` `%` `//` `**`

2、赋值运算符

- `=`、`+=`、`-=`、`/=`、`%=`、`**=`、

3、比较运算符

- `==`、`!=`、`<=`、`>=`、`<`、`>`

4、逻辑运算符

- `and`：一假为假
- `or`：一真为真
- `not`：取反

5、成员运算符

- `in`
- `not in`

6、身份运算符

- `is`
- `is not`

二、控制流程

1、条件语句

- `if`
- `if - else`
- `if - elif`
- `if - elif - else`

2、while条件循环

- 条件循环
- break:跳出循环体，终止当前循环
- continue:中止当前本轮循环，开启下一轮循环
- else: while条件不成立的时候执行

3、for条件循环

- break:跳出循环体，终止当前循环
- continue:中止当前本轮循环，开启下一轮循环
- else:遍历完所有的数据时执行

三、函数

1、函数的定义和调用

- 1、函数定义的关键字：def
 - def 函数名 () :
- 2、函数的命名规范：
 - 遵循标识符的命名规范
 - 推荐使用下划线命名法（单词小写，单词和单词直接使用下划线）
- 3、函数的调用：函数名 ()

2、函数的参数

- 1、形参：定义的参数
 - 必需参数：
 - 默认参数：
 - 不定长参数：
 - *args
 - **kwargs
- 2、实参：调用时实际的传递的参数
 - 位置传参：
 - 关键字传参：

3、函数的返回值

- return 返回的内容

4、函数的作用域

- 1、局部变量
- 2、全局变量
- 3、global

5、内置函数

- 1、基本内置函数
 - print : 输出
 - input : 输入(输入的不管是什么类型数据, 都会当成字符串处理)
 - type: 查看数据类型
 - id : 获取数据内存地址
 - range : 生成数据
 - len : 获取数据的长度 (元素总数)
 - int、float、bool、str、list、tuple、dict、set : 代表对应的数据类型
 - min 求最小值
 - max 求最大值
 - sum 求和
- 2、高级内置函数
 - enumerate
 - eval
 - filter
 - zip:

四、模块&文件

1、模块和包导入

- 1、什么是模块：以 .py 结尾的文件，就是一个模块
- 2、什么是包：python中的包就是一个包含一个 `__init__.py` 文件的目录（文件夹）
- 3、模块导入
 - `import 模块名`
 - `import 模块名 as 别名`
 - `from 模块名 import 函数（变量、类）`
 - `from 模块名 import *`
- 4、包导入
 - `from 包名 import 模块名`
 - `from 包名.模块名 import 函数（变量、类）`

2、文件操作

- 1、打开文件：`open（'文件名', 'r',encoding="utf8"）`
- 2、文件打开的模式：
 - `r w a r b a b w b`
- 3、文件读取
 - `read`
 - `readline`
 - `readlines`
- 4、文件写入
- 5、文件关闭
- 6、上下文管理器：with

```
with open(filename,'r',encoding='utf8') as f:  
    pass
```

3、路径处理

- 1、魔法变量：

- `__name__`：
- `__file__`：

- 2、os模块

- `os.path.dirname`方法：
- `os.path.join()` 方法：
- 其他的方法：

方法	描述
<code>os.getcwd()</code>	显示当前的工作路径
<code>os.chdir ()</code>	切换工作路径
<code>os.mkdir()</code>	在某个目录下创建一个新目录
<code>os.rmdir()</code>	删掉一个目录
<code>os.listdir()</code>	获取当前路径下的目录列表，返回列表格式数据
<code>os.path.isdir ()</code>	判断当前文件是否是目录,返回布尔值
<code>os.path.isfile ()</code>	判断当前文件是否是文件,返回布尔值

五、异常处理

1、异常分析

- 如何通过异常找到错误代码

2、异常捕获

- 1、异常捕获语句

- `try`:
 - 有可能出现异常的代码
- `except` :
 - 捕获到异常之后的处理方案、
- `else`:

- 没有捕获到异常

- finally:

- 不管是否发生异常都会执行

- 2、指定捕获多个异常

#####

- 3、捕获所有类型常见的异常

3、断言：assert

4、主动抛出异常：raise

六、面向对象

1、类的定义&创建对象

- 1、类定义

- 方式一：class 类名：
- 方式二：class 类名 (object) :

- 2、创建对象

- obj = 类名 ()

2、属性

- 1、类属性

- 类属性的定义：直接定义在类里面的变量,类属性可以通过类去访问，也可以通过对象访问
- 类的公有属性：不管在类里面还是类外面可以访问。
- 类的私有属性：双下划线开头的，只能在类里面使用，在类外部是无法使用的

- 2、实例属性

- 实例属性的定义：对象.属性名 = 属性值
- 实例属性只能通过对象去访问。

3、方法

- 1、关于self的理解

- self: 代表的是对象本身，那个对象去调用方法，那么方法中的self代表的就是那个对象

- 2、实例方法

- 定义在类中的普通函数，第一个参数为self
 - 只能够通过对象调用

- 3、类方法

- 使用@classmethod装饰的方法叫类方法，类方法的第一个参数为cls
 - 调用：实例对象和类对象都可以调用。

- 4、静态方法

- 使用装饰器@staticmethod装饰的方法。没有“self”和“cls”参数，
 - 调用：实例对象和类对象都可以调用。

4、初始化方法 `__init__`

- 创建对象的时候，会自动调用，
- 可以通过该方法初始化设置实例属性

5、继承

- 如何继承？
 - 定义类时，类名后面的括号中写上继承的父类
- 继承的作用
 - 子类通过继承可以获得父类的属性和方法，提高开发的效率及代码的复用率。
- 注意点：
 - 私有属性不能够继承

6、方法重写和调用

- 重写父类方法：

- 在子类中定义和父类同名的方法，叫重写父类方法
- 在子类中再调用父类中被重写的方法
 - 方式一：父类名.方法名 (self)
 - 方式二：super().方法名 ()

7、属性动态设置

- setattr:设置属性
 - 参数1：对象
 - 参数2：属性名
 - 参数3：属性值
- getattr:获取属性
 - 参数1：对象
 - 参数2：属性名
- delattr删除属性
 - 参数1：对象
 - 参数2：属性名