Python学习随记五

类和对象

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。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:

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。使用类的名称 Employee 来实例化。

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。类的方法与普通的函数只有一个特别的区别——它们必须有一个额外的第一个参数名称self

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。

正则表达式

正则表达式是一种用来匹配字符串的强有力的武器。它的设计思想是用一种描述性的语言来给字符串定义

一个规则，凡是符合规则的字符串，我们就认为它“匹配”了，否则，该字符串就是不合法的。

常用正则表达式符号：

|  |  |
| --- | --- |
| ****'.'**** | 默认匹配除\n之外的任意一个字符，若指定flag DOTALL,则匹配任意字符，包括\n |
| ****'^'**** | 匹配字符开头 |
| ****'$'**** | 匹配字符结尾 |
| ****'\*'**** | 匹配\*号前的字符0次或多次 |
| ****'+'**** | 匹配前一个字符1次或多次 |
| ****'?'**** | 匹配前一个字符1次或0次 |
| ****'{m}'**** | 匹配前一个字符m次 |
| ****'{n,m}**** | 匹配前一个字符n到m次 |
| ****'|'**** | 匹配符号两边的任意一个，相当于或 |
| ****'(...)'**** | 分组匹配 |
| ****'\A'**** | 只从字符开头匹配，比如re.search("\Aabc","gggggabc") 是匹配不到的 |
| ****'\Z'**** | 匹配字符结尾，和$一样 |
| ****'\d'**** | 匹配数字0-9 |
| ****'\D'**** | 匹配非数字 |
| ****'\w'**** | 匹配[A-Za-z0-9] |
| ****'\W'**** | 匹配非[A-Za-z0-9] |
| ****'\s'**** | 匹配空白字符、\t、\n、\r |

re模块：用来匹配字符串（动态、模糊的匹配）

Python 自1.5版本起增加了re 模块，它提供 Perl 风格的正则表达式模式。re 模块使 Python 语言拥有全部的正则表达式功能。

常用方法：

re.match(pattern, string, flags=0)：从字符串的起始位置匹配一个模式，匹配成功返回一个匹配的对象，否则返回None。其中，pattern是匹配的正则表达式，string是要匹配的字符串，flags是标志位，用于控制正则表达式的匹配方式。

re.search(pattern, string, flags=0)：扫描整个字符串并返回第一个成功的匹配。

findall(string[, pos[, endpos])：在字符串中找到正则表达式所匹配的所有子串，并返回一个列表，如果没有找到匹配的，则返回空列表.

split(pattern, string[, maxsplit=0, flags=0)：按照能够匹配的子串将字符串分割后返回列表

sub(pattern, repl, string, count=0, flags=0)：替换字符串中的匹配项

group()：匹配对象函数来获取匹配表达式

re.compile(pattern[, flags]):compile 函数用于编译正则表达式，生成一个正则表达式(Pattern)对象，供 match() 和 search() 这两个函数使用。

datetime

datetime模块用于是date和time模块的合集，datetime有两个常量，MAXYEAR和MINYEAR，分别是9999和1.

datetime模块定义了5个类，分别是

1.datetime.date：表示日期的类

2.datetime.datetime：表示日期时间的类

3.datetime.time：表示时间的类

4.datetime.timedelta：表示时间间隔，即两个时间点的间隔

5.datetime.tzinfo：时区的相关信息

http请求

通过request模块，使用get和post方法处理http请求。

requests是一个很实用的[Python](http://lib.csdn.net/base/python" \o "Python知识库" \t "https://www.cnblogs.com/lilinwei340/p/_blank) HTTP客户端库，主要是get请求和post请求。

发送get请求：r = requests.get("http://httpbin.org/get")

如果请求中需要带参数，Requests 提供了 params 关键字参数，允许我们以一个字典来提供这些参数，如：

payload = {'page': '1', 'per\_page': '10'}

r = requests.get("http://httpbin.org/get", params=payload)

发送 POST 请求：r = requests.post("http://httpbin.org/post")

类似地，在发送 POST 请求时需要附上数据，我们可以使用 Requests 提供的 data 参数：

payload = {'page': 1, 'per\_page': 10}

r = requests.post("http://httpbin.org/post", data=payload)

响应方式：

普通响应，使用 r.text 获取

JSON 响应，使用 r.json() 获取

二进制响应，使用 r.content 获取

原始响应，使用 r.raw 获取