# 课程介绍

* 1. 随机数模块；
* 2. 时间日期处理模块;
* 3. 散列化模块;

1. 随机数模块

python中的random模块用于生成随机数，当我们需要随机产生测试数据，抽奖，验证码等等，就可以利用这个模块。

* 1. random模块有哪些接口

|  |  |  |
| --- | --- | --- |
| 接口名称 | 类型 | 说明 |
| random.random() | 实数相关 | 用于生成一个0到1的随机浮点数: [0, 1) |
| random.uniform(a,b) | 生成[a,b]或[b,a]之间的均匀分布随机浮点数。 |
| random.randint(a,b) | 整数相关 | 生成[a,b]的随机整数，要求a < b。 |
| random.randrange(a,b) | 生成[a,b)的随机整数，第三个参数可以指定步长。 |
| random.choice(seq) | 序列相关 | 从序列中随机选择一个元素，若序列为空，则抛出异常。 |
| random.shuffle(seqb) | 打乱源序列，源序列必须可写。 |
| random.sample(seq,k) | 从序列中选择k个元素返回，原序列不变。 |
| random.seed(n=none) | 初始化 | 初始化随机熵池。 |

* 1. 注意事项

random.shuffle必须跟可写的序列，同时序列被修改。如果需要保留原序列，那么就是用sample方法，第二个参数使用len来计算取样长度。

random的seed，一旦用定值初始化，则每次运行的随机现象是一致的。

1. datetime模块

时间模块，是开发过程中使用比较频繁的模块，比如记录日志，用户登录时间，特定时间内的数据分析等等。

Python中提供了对时间日期的多种多样的处理方式，主要是在 time 和 datetime 这两个模块里。

* + 1. time模块

在 Python 文档里，time是归类在Generic Operating System Services中，换句话说， 它提供的功能是更加接近于操作系统层面的。通读文档可知，time 模块是围绕着 Unix Timestamp 进行的。

该模块主要包括一个类 struct\_time，另外其他几个函数及相关常量。 需要注意的是在该模块中的大多数函数是调用了所在平台C library的同名函数， 所以要特别注意有些函数是平台相关的，可能会在不同的平台有不同的效果。另外一点是，由于是基于Unix Timestamp，所以其所能表述的日期范围被限定在 1970 - 2038 之间，如果你写的代码需要处理在前面所述范围之外的日期，那可能需要考虑使用datetime模块更好。

|  |  |
| --- | --- |
| 接口名称 | 说明 |
| time.time() | 返回一个距Epoch的秒数，是浮点数。 |
| time.gmtime(seconds) | 将秒数转化为年月日时分秒，以UTC时间为标准。 |
| time.localtime(seconds) | 将秒数转化为年月日时分秒，以当地时间为标准。 |
| time.ctime(seconds) | 返回年月日时分秒的字符串。 |
| time.asctime(tuple) | 从struct\_time返回年月日时分秒字符串。 |
| time.mktime(tuple) | 将struct\_time转换为秒数。 |
| time.strftime(fmt,t) | 按照fmt格式将struct\_time显示成字符串。 |
| time.strptime(str,fmt) | 将年月日时分秒的字符串按照fmt解析成struct\_time结构。 |

* + 1. datetime模块

datetime 比 time 高级了不少，可以理解为 datetime 基于 time 进行了封装，提供了更多实用的函数。在datetime 模块中包含了几个类，具体关系如下:

* timedelta # 主要用于计算时间跨度
* tzinfo # 时区相关
* time # 只关注时间
* date # 只关注日期
* datetime # 同时有时间和日期

在实际实用中，用得比较多的是 datetime.datetime 和 datetime.timedelta。

datetime.datetime围绕一个叫做datetime.datetime对象来进行数据保存。

一般使用datetime.datetime.now()获得这个实例对象。

|  |  |
| --- | --- |
| 实例方法 | 说明 |
| datetime.date() | 返回一个date对象。 |
| datetime.time() | 返回time对象。 |
| datetime.replace(n=v) | 替换字段的值后得到一个新的datetime对象。 |
| datetime.timetuple() | 返回time.struct\_time对象。 |
| datetime.strftime(fmt) | 按照fmt的格式生成字符串。 |

除了实例本身具有的方法,类本身也提供了很多好用的方法：

|  |  |
| --- | --- |
| 类方法 | 说明 |
| datetime.today() | 当前时间，localtime。 |
| datetime.now([tz]) | 当前tz时区的时间。 |
| datetime.utcnow() | 当前UTC时间。 |
| datetime.fromtimestamp() | 返回时间戳的时间，有时区要求。 |
| datetime.strptime(str,fmt) | 按照fmt的格式解析字符串生成datetime。 |

* + 1. timedelta模块

timedelta模块主要针对的是2个日期对象的数学运算，比如当前时间的前或后1个月，2个小时这样的需求。

只要初始化时，指定日期参数就可以设置一个延迟时间。注意这个对象的访问只能是天和秒，需要小时等信息需要进行数学运算。

最大的单位是天。

1. 哈希散列
   1. 什么是哈希算法

哈希算法，又叫散列算法。是一种从任何一种数据中创建小的数字“指纹”的方法。散列函数把消息或数据压缩成摘要，使得数据量变小，将数据的格式固定下来。

* + 1. 哈希算法的特点

如果两个散列值是不相同的（根据同一函数），那么这两个散列值的原始输入也是不相同的。

通过散列值是恢复不出原数据的信息。

* + 1. MD5

MD5：可以产生出一个128位（16字节）的散列值，用于确保信息传输完整一致。

MD5已经广泛使用在为文件传输提供一定的可靠性方面。例如，服务器预先提供一个MD5校验和，用户下载完文件以后，用MD5算法计算下载文件的MD5校验和，然后通过检查这两个校验和是否一致，就能判断下载的文件是否出错。

1996年后被证实存在弱点，可以被加以破解，对于需要高度安全性的数据，专家一般建议改用其他算法，如SHA-2。2004年，证实MD5算法无法防止碰撞（collision），因此不适用于安全性认证，如SSL公开密钥认证或是数字签名等用途。

* + 1. SHA-2

SHA-2，名称来自于安全散列算法2（Secure Hash Algorithm 2）的缩写，一种密码散列函数算法标准，由美国国家安全局研发，由美国国家标准与技术研究院（NIST）在2001年发布。

其中最常用的是：SHA-256、SHA-512

* 1. python中如何使用

所有安全散列算法，都在hashlib标准库里进行了管理，要使用其中的算法，首先导入hashlib库，然后使用里面的函数即可。

* + 1. md5

最简单的使用。

import hashlib

test\_data = b‘abc123’

print(hashlib.md5(test\_data).hexdigest())

正规使用：

* + - 1. 初始化算法库
      2. 使用实例化对象的update方法添加数据
      3. 使用实例化对象的hexdigest方法获得散列值
    1. sha256

import hashlib

test\_data = b‘abc123’

print(hashlib.md5(test\_data).hexdigest())

1. 课程总结
   1. 重点
      * 1. 随机模块的使用
        2. 获取日期时间
        3. 哈希散列模块的使用
   2. 难点
      * 1. 日期差值的模块使用
        2. 散列的特点
   3. 如何掌握？
      * 1. 多用ipython来提示，看帮助；
        2. 把课后练习深入思考，根据参考代码抄写临摹；
2. 课后练习
   * + 1. 获取1个[1,100]的随机数，获取1个[1,100]的随机偶数；
       2. 产生一个包含A-Z0-9字符的4个字符组成的随机码；
       3. 请用面向对象的方法设计一个双色球模拟开奖过程，要求如下：

采用随机模块模拟双色球的抽奖过程；

每次开奖时，把所有结果和抽奖时间保存到一个文件里；每隔3天进行一次抽奖。

* + - 1. 验证pycharm安装包的sha256的值

1. 面试题
   * + 1. 请描述如何产生一个范围的随机数。
       2. 请看下面的描述（Google）

博彩公司要开发一个网络赌博工具，随机摇3个骰子，让用户猜大、小、豹子。要求在用户投注前先摇好骰子，让用户猜，但又不能让用户知道骰子里的数字，请问如何设计？

【说明】：

结果01-10是小，猜中1赔2

结果11-18是大，猜中1赔2

出现3个相同的为豹子 1赔18

如果出现 3 3 3的组合，加在一起是9，即算小，也算豹子。

1. 扩展知识或课外阅读推荐（可选）
   1. 扩展知识

可以根据面试题2的说明，设计一个这样的模拟游戏，看看自己的运气怎么样。