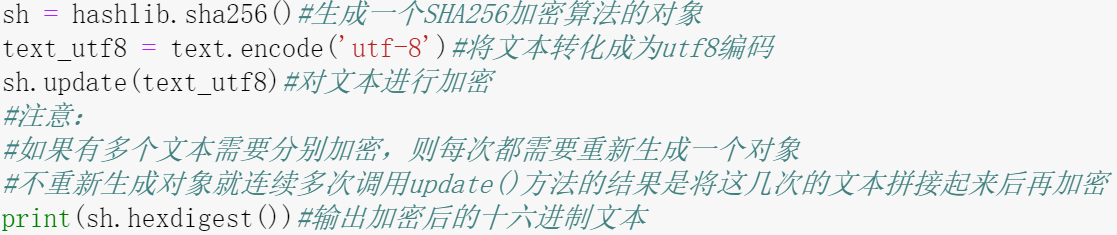
**实验任务1：密文破解**

现有三段密文，密文1使用了SHA256不对称加密的方式来进行加密，密文2、3使用对称加密的方式进行加密。对密文1破解后即可获得密文2、3对称加密的密钥。密文1对应的明文1的格式为“卡号位移个数为x，卡密位移个数为y”（不包含双引号），其中x、y均为1到10的某个数字（可能相同也可能不同）。可使用hashlib包中的SHA256不对称加密算法暴力破解密文，该包的使用范例如下图所示。



密文2、3分别为卡号和卡密，其加密方式如下。

1）将所有的字符转换成ASCII码的十进制形式（用ord(字符)函数）

2）从第一个字符开始每个字符的ASCII码值加i，其中i从1到N递增并循环。例如N=3，则第一个字符+1、第二个字符+2、第三个字符+3、第四个字符+1、第五个字符+2、第六个字符+3……以此类推。其中N就是明文1中所说的位移个数。

3）将最终的ASCII码转换为字符（用chr(ASCII码值)函数）

密文1为：

1a73cd563ab2d3ec7c0b6a6601440353b85c46927863b0f0bc621896bbd86c90

密文2为：KFY6<7334573;=512346

密文3为：D6D92?>9D

**实验任务2：自动化编写小说**

使用Python模拟武侠小说中两派人的一场遭遇战。

每个人都有名字、血量（HP）、一系列技能及技能对应的攻击输出。每个人的攻击方式均一致，即随机从自己的技能包中抽取一个技能进行攻击。两派人（自己命名）均继承自武林人士这个父类，但有各个门派都有各自的技能包及对应的输出值。门派下每个人拥有的技能包都是在实例化对象的时候随机赋予的。

遭遇战的时候，每个帮派的人数不确定。战斗采用回合制的方式进行，即帮派A中的一个人随机选择帮派B中的一个人进行攻击。B受到的伤害值为A随机选择的技能的输出值加上20%的标准正态分布生成的随机数*向下取整*，即伤害值=输出值\*(1+0.2\*N(0,1))。技能被选择的概率和其对应的输出值成反比，如某人拥有两个技能，输出值分别为10、30，则技能1的被选择概率为。其中某一门派参与战斗的所有人都则阵亡（即HP降为0）则战斗结束。

几个随机数生成函数：首先引入random包，import random；random.random()生成一个0~1的均匀分布随机数；random.randint(a,b)生成取值范围为[a,b]的一个整数随机数；random.normalvariate(0,1)生成一个标准正态分布的随机数。

一个输出的范例：

