# 24.如何用Python破解凯撒加密

百家号不支持代码格式，文章里的代码排版都是乱的。

如果需要拷贝代码，可以去同名微信公众号。

上篇说到，每个字母都对应了一个数字。（ASCII码）。



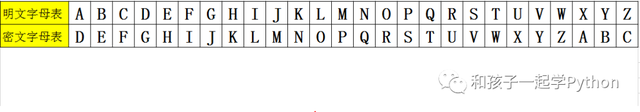
所以，将字母向后平移多少位，就相当于将字母对应的数字加上平移的位数。

我们称向后平移的位数为秘钥。

假设秘钥 key = 3

对字母 A 加密就相当于 65+3 = 68，68对应的字母是 D。

但对字母 Y 加密， 89+3 =92，92对应的字符是“\”，而不是正确的字母 B。



如何处理这种情况呢？来看下面这张表格。

Y 对应的数字是 65 + 24，将它向后平移 3 位，也即秘钥 key = 3

得到 65 +24 +3

我们只需要关心 24 +3 除以 26的余数，也即 (24+3) % 26 的结果是 1

65 + 1 是不是正好就是字母 B 呢？

为什么是除以 26 的余数？ 因为是 26 个英文字母。

无论秘钥是多少，比如 k = 1000

字母 A 加密之后 = 65 + 0 + 1000

其中 0 + 1000这一部分始终要落在 [0,25]之间。



完整的程序，不再需要字母映射表，只需要知道秘钥key就可以了

source = 'I miss you'

target = ''

key = 3

for i in range(len(source)):

if source[i].isalpha():

if ord(source[i]) < 97:

target += chr( 65 + ( ord(source[i]) - 65 + key) % 26 )

else:

target += chr( 97 + ( ord(source[i])- 97 + key ) % 26 )

else:

target += source[i]

如何解密

加密是将字母对应的数字加上秘钥。

解密则是将字母对应的数字减去秘钥。

不过，因为秘钥可是是任意整数，相减的差可能是负数。

比如秘钥 key = 1000，对字母 C (67) 加密

65 + (2+1000) % 26 = 65 + 14 = 79，对应字母 O 。

反之，对字母 O 解密

65 + （14 - 1000）% 26 = 65 + 2 = 67， 对应字母 C。

解密的结果是正确的，所以我们不需要担心字母减去秘钥为负数。

原因在于Python里，计算负数除以整数的余数，得到的一定也是正数。

比如 -10 % 3

-10 = （-3） \* 3 + （-1）

-10 =（-4） \* 3 +（2）

这2个等式都成立，但Python的 % 运算符用的是第2个等式的逻辑。

加密一封完整的信：Python文件读写

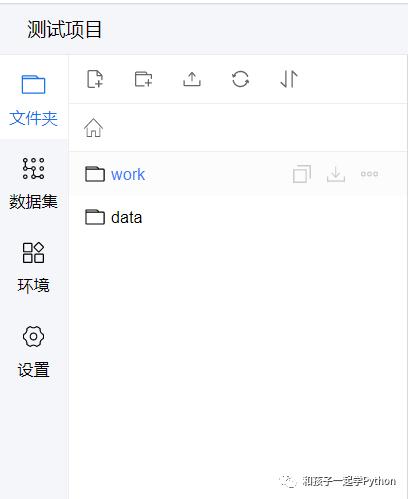
前面我们只加密了信里的一句话。

实际生活中，如果用电脑写信。我们一般会把它保存成word文本文件，或者txt文本文件。

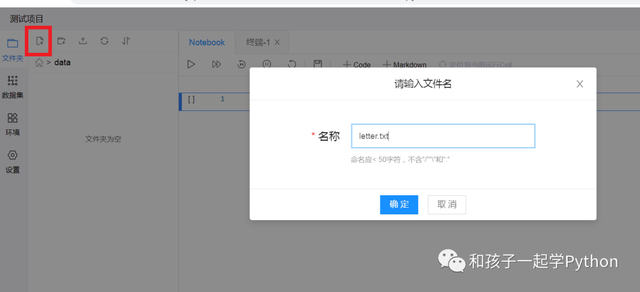
Python能不能读取文本文件里的内容，加密之后再写到文本文件里呢？

在百度的在线开发环境里，也提供了保存文件的地方。

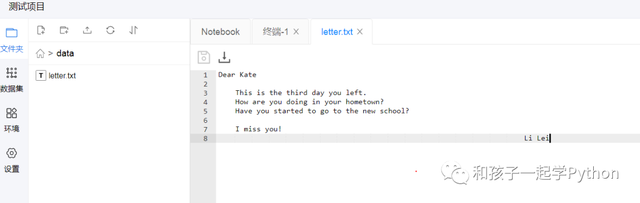
我们习惯把文件保存在如下图的 data文件夹里。



进入data文件夹，点击左上角的“创建文件”图标，创建一个txt的文本文件 letter.txt



然后打开letter.txt，输入信的内容，并保存。



接下来我们用Python来读取这封信的内容。

f\_read = open('data/letter.txt')

content = f\_read.read()

print(content)

f\_read.close()

Python的文件读写非常简单。

内置函数 open()，用来打开一个文件。

open()函数最重要的参数有2个。

第1个是文件的路径，比如上面的'data/letter.txt'

第2个是打开的方式，是“只读”方式打开，还是“读写”方式，这个参数如果不写的话，默认是“只读”方式。

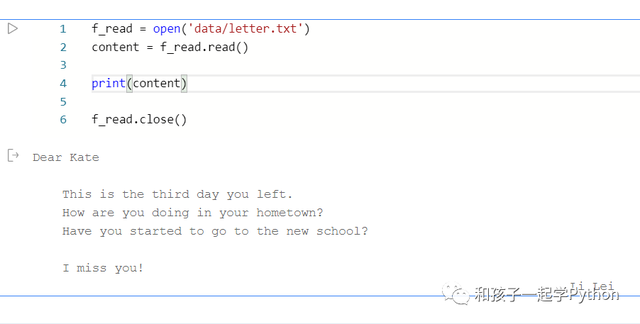
open()执行成功后，可以认为它返回了一个文件的对象。

我们用变量 f\_read 来指向这个文件。

这个文件类的对象，有一个 read() 函数，可以读取文件里的内容。

content = f\_read.read()

我们把读取到的内容打印出来。果然是文件里的内容。



最后，千万别忘记了，打开的文件要关掉！

f\_read.close()

我们先看完整的程序

f\_read = open('data/letter.txt')

content = f\_read.read()

target = ''

key = 3

for i in range(len(content)):

if content[i].isalpha():

if ord(content[i]) < 97:

target += chr( 65 + ( ord(content[i]) - 65 + key) % 26 )

else:

target += chr( 97 + ( ord(content[i])- 97 + key ) % 26 )

else:

target += content[i]

f\_read.close()

f\_writer = open('data/result.txt',mode='w')

f\_writer.write(target)

f\_writer.close()

中间部分是对信的内容进行加密。target是加密后的结果。

后面三行代码是，打开一个用于保存加密结果的文件（先要创建）， mode = 'w'就是前面说的第2个参数，’w'表示用“写入”模式。

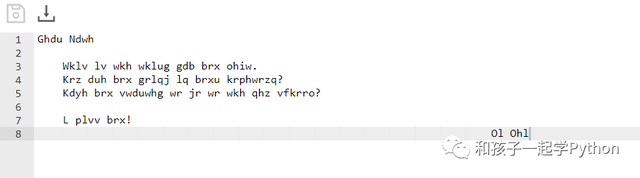
然后用 write() 函数把加密后的内容写入到文件里。

最后，千万别忘记，把文件关闭！

如何破解“凯撒加密”

李雷和Kate于是开始使用“凯撒加密”进行通信。

加密后信件就如下图一样。



但这样就保险了吗？

他们的信件再次落入父母的手中，李雷爸爸看到信的内容哈哈大笑。

“小儿岂敢班门弄斧哉！”



为什么李雷爸爸会对李雷的加密不屑一顾呢？

因为“凯撒加密”是非常容易破解的。

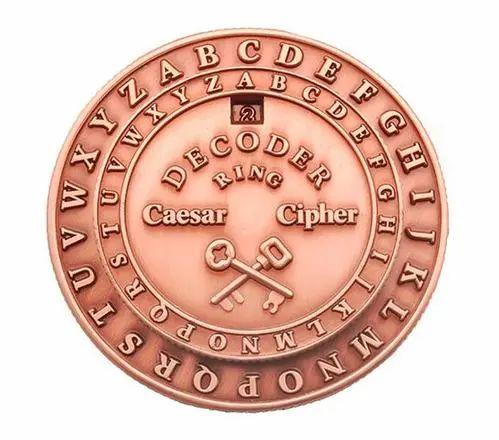
所谓破解，就是在不知道秘钥的情况下，把密文还原成明文。

为什么“凯撒加密”非常容易破解？

“凯撒加密盘”就可以给出直观的答案。

因为“凯撒加密”的有效秘钥只有 25个。

所以，只需要把 key 从 1到 25 都尝试解密一遍。就能找到正确的明文。



注： 当然，首先李雷爸爸要能猜到李雷用的是“凯撒加密”。这就是李雷爸爸的经验了。

帮李雷想一种新的加密方式

聪明的李雷在使用“凯撒加密”通信一段时间后，也敏感地察觉到好像爸爸妈妈还是知道信里的内容。他用类似的方法进行了试探，果然证实了他的猜测。

怎么办呢？有没有更难破解的加密方法，同时也方便描述，操作简单呢？