

Dictionary

Python 基础入门

回顾

字符串

len() 和 索引

查找 find()

字符串 分片

加法 和 乘法

分割 和 合并

大小写 转换

消息 加密



加密

Caesar Cipher 的问题

- ✓ 一旦 泄漏 密钥，分分钟被 破译
- ✓ 即使密钥 未知，也可以逐一尝试 全部可能
- ✓ 通过 随机替换，能够大大增加 破译难度

$N!$

$$26! \approx 4 \times 10^{26}$$

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	Q	I	T	L	U	G	H	K	E	J	Y	Z	P	X	M	R	C	N	O	D	F	W	S	A	V

加密

如何表示 加密关系

- ✓ A 对应 B, B 对应 Q, 即 密码本
- ✓ 在 Python 中可以使用 字典 表示
Key 和 Value 之间的 对应关系

```
cipher = {}  
cipher = dict()
```

```
cipher['A'] = 'B'  
cipher['B'] = 'Q'  
cipher['C'] = 'I'
```

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	Q	I	T	L	U	G	H	K	E	J	Y	Z	P	X	M	R	C	N	O	D	F	W	S	A	V

对应

举几个 对应 的例子

对应 也称作 映射

例子	Key	Value
字典	字词	解释
学号	字符串	学生
名字	汉字	人

小任务

调查 并 记录
每位同学的 年龄

字典

在 Python 中使用 字典

```
>>> bag = dict()
>>> bag['money'] = 32
>>> bag['candy'] = 3
>>> bag['tissues'] = 20
>>> print(bag)
{'tissues': 20, 'candy': 3, 'money': 32}
>>> print(bag['candy'])
3
>>> bag['candy'] = bag['candy'] + 2
>>> print(bag)
{'tissues': 20, 'candy': 5, 'money': 32}
```



字典

字典 和 列表 的对比

- ✓ 列表用 中括号, 字典用 大括号
- ✓ 列表由一个个 变量 组成
- ✓ 字典由一对对 Key-Value 组成
- ✓ 列表是 有序的, 字典是 无序的
- ✓ 列表用 索引 访问, 字典用 Key 访问

```
>>> ls = list()
>>> ls.append('Tom')
>>> ls.append(15)
>>> print(ls)
['Tom', 15]
>>> ls[1] = 16
>>> print(ls)
['Tom', 16]
```

```
>>> dic = dict()
>>> dic['name'] = 'Tom'
>>> dic['age'] = 15
>>> print(dic)
{'name': 'Tom', 'age': 15}
>>> dic['age'] = 16
>>> print(dic)
{'name': 'Tom', 'age': 16}
```

字典

字典 中的 查找

- ✓ 使用 `in` 判断字典中是否有某个 `Key`
- ✓ 使用 `keys()` 获取全部 `Key`
- ✓ 使用 `values()` 获取全部 `Value`
- ✓ 使用 `items()` 获取全部 `Key-Value`

```
>>> dic = {}
>>> dic['name'] = 'Tom'
>>> dic['age'] = 15
>>> print(dic)
{'name': 'Tom', 'age': 15}
>>> 'name' in dic
True
>>> 'gender' in dic
False
>>> dic.keys()
dict_keys(['name', 'age'])
>>> dic.values()
dict_values(['Tom', 15])
>>> dic.items()
dict_items([('name', 'Tom'), ('age', 15)])
```


字典

访问 不存在 的 Key

- ✓ 访问字典中不存在的 Key 将 报错
- ✓ 使用 `get()` 指定 默认值, 避免报错

```
>>> counts = {'Alice': 2, 'Ben': 1}
>>> x = counts['Claire']
Traceback (most recent call last):
  File "<pyshell#12>", line 1, in <module>
    x = counts['Claire']
KeyError: 'Claire'
>>> x = counts.get('Alice', 0)
>>> print(x)
2
>>> x = counts.get('Claire', 0)
>>> print(x)
0
```

练习

数字 统计

- ✓ 输入一串 很长的 数字，例如 12386609354383173259435
- ✓ 统计 每个数字 出现的 次数，例如 1 出现了 2 次
- ✓ 统计 奇数数字 和 偶数数字 的出现次数



随机替换

如何产生 随机替换

- ✓ 使用 字典 表示 密码本
- ✓ 如何为每个字符, 随机指定 对应的密文

```
{  
  "a": "o",  
  "b": "a",  
  "c": "q",  
  "d": "s",  
  "e": "b",  
  "f": "z",  
  "g": "y",  
  "h": "x",  
  "i": "c",  
  "j": "v",  
  "k": "p",  
  "l": "k",  
  "m": "u",  
  "n": "t",  
  "o": "n",  
  "p": "d",  
  "q": "j",  
  "r": "i",  
  "s": "h",  
  "t": "m",  
  "u": "l",  
  "v": "f",  
  "w": "e",  
  "x": "w",  
  "y": "r",  
  "z": "g"  
}
```

```
{  
  "a": "f",  
  "b": "a",  
  "c": "w",  
  "d": "h",  
  "e": "s",  
  "f": "j",  
  "g": "t",  
  "h": "q",  
  "i": "k",  
  "j": "y",  
  "k": "e",  
  "l": "u",  
  "m": "r",  
  "n": "n",  
  "o": "d",  
  "p": "m",  
  "q": "v",  
  "r": "x",  
  "s": "z",  
  "t": "l",  
  "u": "o",  
  "v": "c",  
  "w": "i",  
  "x": "b",  
  "y": "g",  
  "z": "p"  
}
```

随机替换

打乱 和 拷贝

- ✓ 使用 `shuffle()` 打乱一个列表
- ✓ 使用 `copy()` 复制一个列表
- ✓ 比较 深拷贝 和 浅拷贝 的不同

```
>>> import random
>>> source = ['a', 'b', 'c', 'd']
>>> target = source
>>> random.shuffle(target)
>>> print(source, target)
['b', 'a', 'd', 'c'] ['b', 'a', 'd', 'c']
```

```
>>> import random
>>> source = ['a', 'b', 'c', 'd']
>>> target = source.copy()
>>> random.shuffle(target)
>>> print(source, target)
['a', 'b', 'c', 'd'] ['c', 'b', 'a', 'd']
```

随机加密

随机产生 密码本 并 加密

```
import random

alphabet = 'a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z'
base = alphabet.split(',')
target = base.copy()
random.shuffle(target)
```

```
cipher = dict()
for i in range(len(base)):
    cipher[base[i]] = target[i]

print(cipher)
```

```
msg = input('Please input a message:')
new_msg = ''
for c in msg:
    new_msg = new_msg + cipher.get(c, c)
print('The secret message is:', new_msg)
```

作业

解密 消息

- ✓ 在上一页的基础上，添加 解密 的代码
- ✓ 使用 `new_msg` 得出一开始的明文
- ✓ 在 小写字母 的基础上，增加 大写字母 和 数字

```
# alphabet = 'abcdefghijklmnopqrstuvwxyz'  
alphabet = 'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789'
```

思考

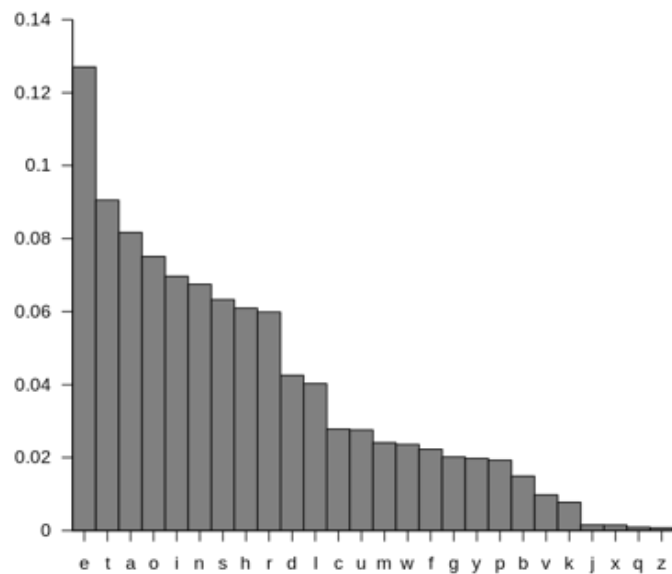
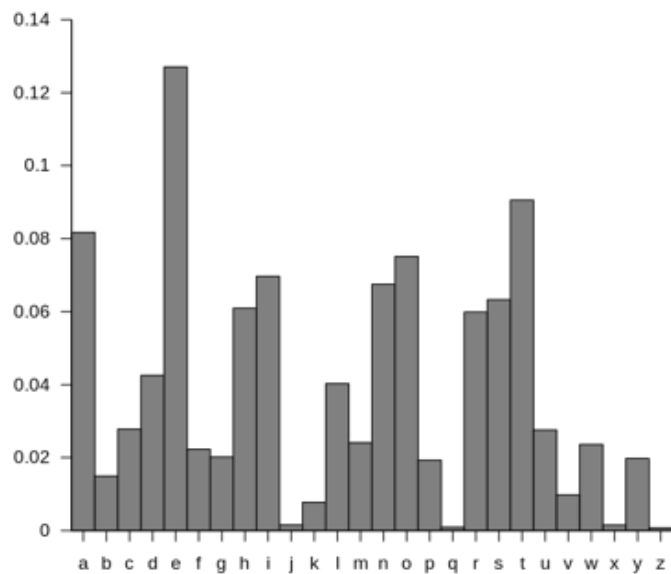
没有 密码本 怎么办

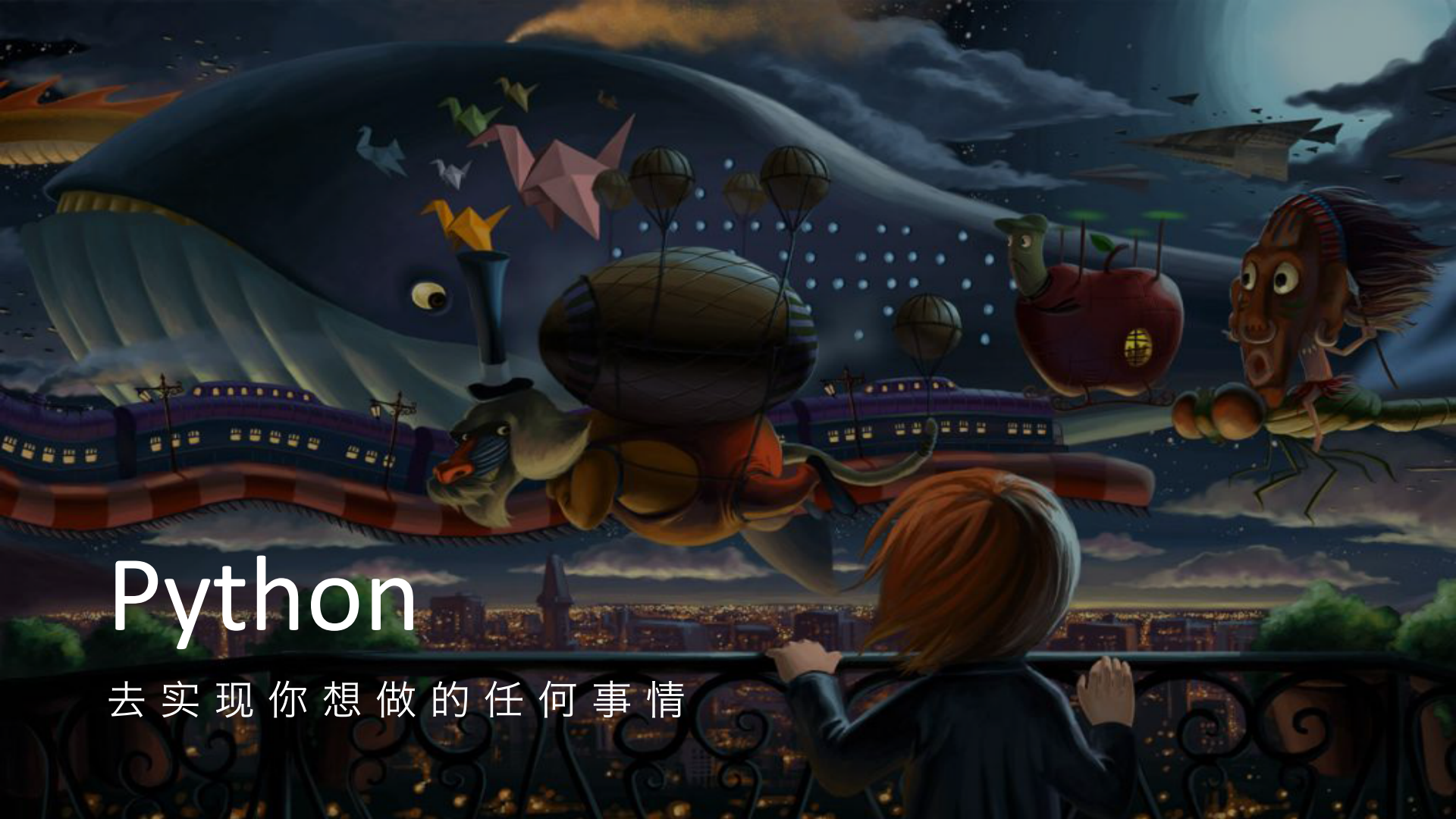
✓ 尝试 所有的 对应可能, 但 工作量 太大

https://en.wikipedia.org/wiki/Letter_frequency

[illegible]

词频 的 不变性





Python

去实现你想做的任何事情