

练习总结

practice 1

practice-bmi: 多分支结构

practice-rocket ship: 图案问题

- 两个cone : $2 * height - 1$ 行, 相比上一行, 左右两边增加了一个/ 和 \
- 3个line : + 加上 $2 * height$ 个 =* 加上 +
- 剩下的部分: 可以看到由两种图案合并而成, 只是顺序调换了一下
- 尖头朝上的部分: 总共height行, 利用等差数列, 决定dot的个数
 - | + 多个 dot + /\ + 多个dot + /\ + 多个dot
- 尖头朝下的部分: 总共height行
 - | + 多个 dot + /\ + 多个dot + /\ + 多个dot

practice-nested-loop：图案问题，两层循环

height=3

[illegible]

practice 2

- practice-scores: 模拟打分，累加问题 + 异常处理 + 切片
- primes: 求n以内的素数，一个数如果不是素数，一定是某个数的倍数，一定是某个素数的倍数。
 - 2是素数， $2*2, 2*3, 2*4, \dots$ 等不是素数
 - 3是素数, $3*2, 3*3, 3*4, \dots$ 等不是素数
 - 4不是素数，说明其有更小的素因子，不需要考虑
 - 5是素数, $5*2, 5*3, \dots$ 等不是素数
 - ..., 一直到 $\text{sqrt}(n)$
 - 那些仍然是素数的数就是问题的解

True	True	True	True	True	True	True	True	...	True
1	2	3	4	5	6				n

↓

True	True	True	False	True	False	True	False	...	True
1	2	3	4	5	6				n

↓

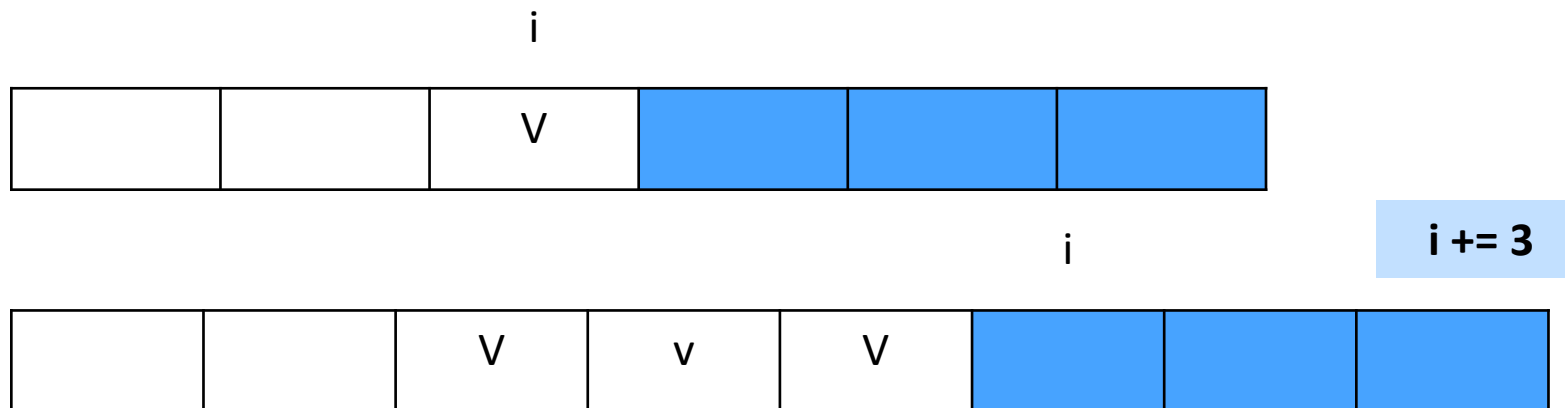
True	True	True	False	True	False	True	False	...	True
1	2	3	4	5	6				n

↓

True	True	True	False	True	False	True	False	...	True
1	2	3	4	5	6				n

practice 2

- triple-value: `triple_value_list([1,2,3,1,1], 1)` , 列表变为[1,1,1,2,3,1,1,1,1,1]
- 自己跟踪下标 , 没有到列表结尾时 , `while i < len(list_)`:

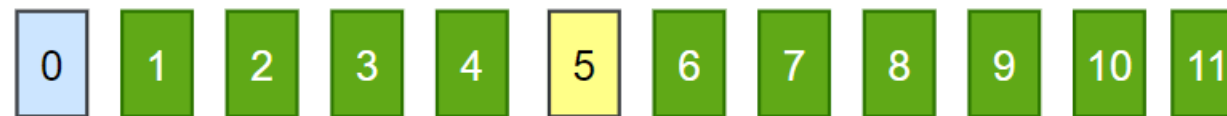


practice 2

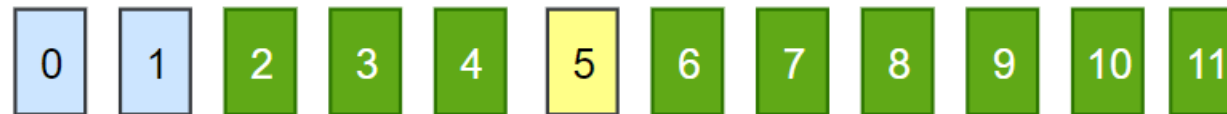
- `is_sorted(seq)`: 判断是否已经按照从小到大的顺序排序好
- `selection_sort(seq)`: 选择排序
 - 首先选择最小的值放在第1个
 - 接下来选择第2小的值放在第2个
 - ...
 - 第*i*轮后：前面的*i*个元素已经排序好，且后面的元素都更大
 - 选择倒数第2小的值放在倒数第2个，排序完毕



最小的



已排序好，且后面都更大



practice 3

practice-list-comprehension:

- 列表推导式 + range , 要产生多少个在某个范围的随机数

practice_prime_factorization : 分解质因数

$$100 = 2 * 2 * 5 * 5$$

$$77 = 7 * 11$$

- 对于正整数 n , 考虑如下可能的质因数 : 2, 3, 5, \sqrt{n}
 - 从 n 开始 , 对于当前可能的因子 , 分解 n 使得其不再含该因子为止
 - 如果分解后剩余的因子为1 , 则不能再继续分解 , 不需要尝试其他可能的因子
- 三种情况 :
 - 分解质因数成功 , 所有的因子都小于等于 \sqrt{n} , 对应 $n = 1$ 。比如 $140 = 2 * 2 * 5 * 7$
 - 分解质因数成功 , 剩余最后的 n 无法进一步分解, $n \neq 1$ 。比如 $123 = 3 * 41$
 - n 为素数 , 无法分解 , 上述可能的质因数都不是 n 的因子 : $\text{factors} == []$, $n \neq 1$ 。比如 $n=127$

```
factors = []
for factor in (2, *range(3,
int(math.sqrt(n)+1), 2)):
    while n % factor == 0:
        n = n // factor
        factors.append(factor)
    if n == 1:
        break

if n != 1:
    factors.append(n)
```

practice 3

chinese_zodiac:

- 十二生肖，通过下标得到对应的农历年属相
- 要知道十二生肖的起始(鼠年)是那一年：已知2000年是龙年

zodiac_sign:

- 已知十二星座的日期范围，(月, 日)
- 利用元组的比较，判断属于哪个星座

practice 4

practice-dict: 字典 : key=最高温 , value = [出现的天数]

- 累加模式 + 字典 : value为可变对象时如何更新
- 排序 + 格式化

practice-letter-counter: 字典, key=字母, value=出现次数

- 累加模式+字典的更新 (value为不可变对象)
- topn : 排序 + 切片
- 如果用列表来统计字符出现的次数?
 - 累加模式+列表的更新 , 字母转换为下标, `ord(ch) - ord('a')`

practice-counter: d: {'a':3, 'b':2, 'c':1}

- `get_elements(d)`: 遍历所有的元素 : 得到key和value , 展开为value个key
- `iter_elements(d)`: 通过生成器函数实现
- `most_common(d, n)`: 排序 (对于items排序 , 排序基准为字典的value) + 切片

practice 5

practice-duplicated-int: 包含了多个整数的列表，其中正好只有一个整数重复出现

- 集合去除重复元素，由于只有一个整数重复，原来的整数求和 - 去除重复元素求和就可找到该整数

practice_format:

- 采用 '{...}'.format方法进行格式化

practice_lotto:

- 机选彩票：产生随机的号码加入到集合中，利用集合的不重复特性去除重复，重复直到得到足够的号码为止
- 兑奖：集合的交集运算，得到在开奖集合中的数字

practice 6

practice anagram: 相似词，包含相同的字母

- 排序后相等
- 统计每个字符出现的次数，两者都一样

practice diamond: 输出菱形

- 图案问题，可以利用居中对齐，将图案看成两个部分

practice insert space:

- 图案问题+切片，观察图形规律，空格所在的位置

practice isdigit:

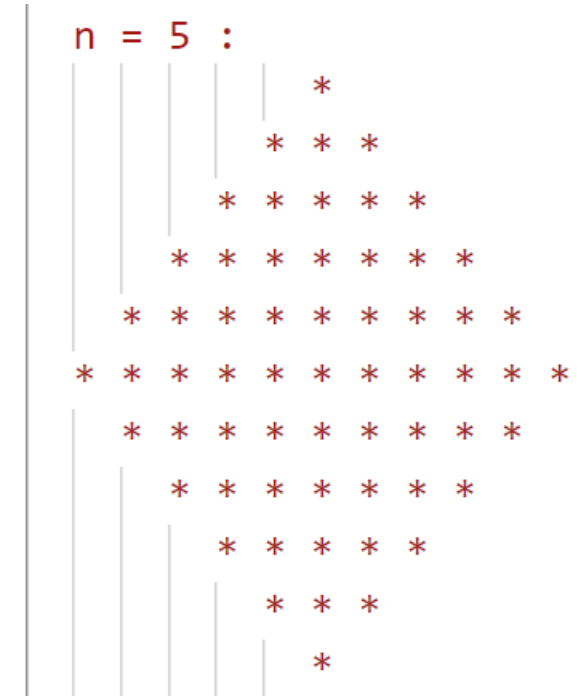
- 自己实现isdigit/islower/isupper等方法，for循环遍历，发现不满足条件时结束循环

practice_title_case:

- 自己实现capitalize和title方法：切片、split和join

Universit y
Universi ty
Univers ity
Univer sity
Unive rsity
Univ ersity
Uni versity
Un iversity
U niversity

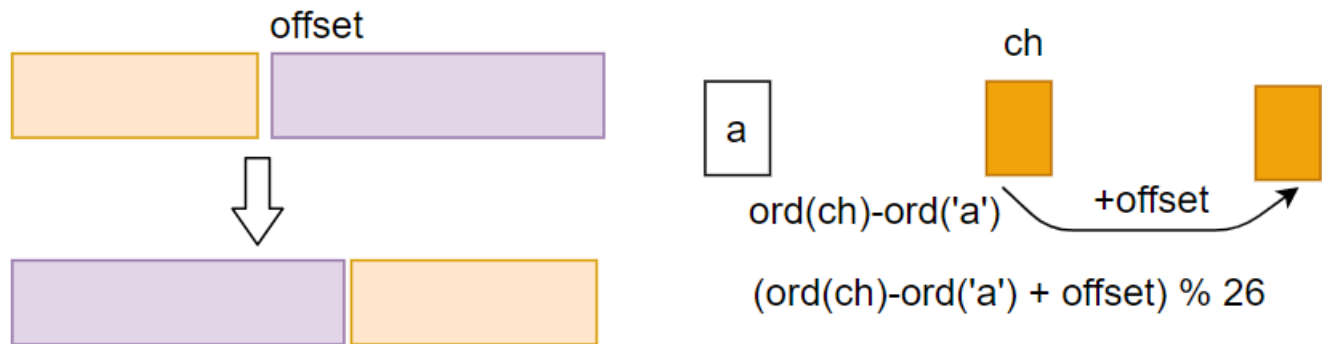
U niversity
Un iversity
Uni versity
Univ ersity
Unive rsity
Univer sity
Univers ity
Universi ty
Universit y



practice 6

practice_caesar : 凯撒密码(caesar cipher)

- 如果为英文字母，则其替换为后面偏移offset个位置的字符。最后一个字符z的下一个位置为从头开始的a
- 使用maketrans构造映射表
 - 切片实现



```
before = string.ascii_letters
after = (string.ascii_lowercase[offset:] + string.ascii_lowercase[:offset] +
        string.ascii_uppercase[offset:] + string.ascii_uppercase[:offset])
```

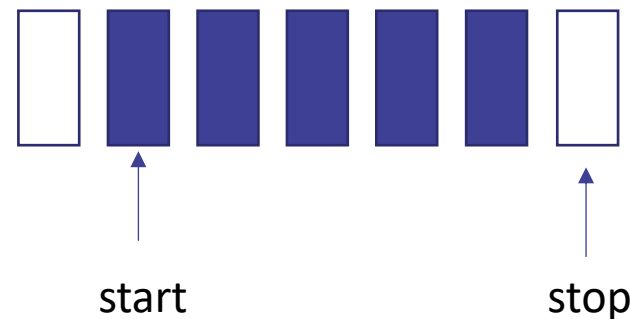
practice 6

practice_substitute: 替代密码

- 产生随机密码本： shuffle + join
- maketrans + translate

practice_word_counter:

- 分析文本，得到文本中的各个单词
- 有限状态自动机：考虑单词和非单词两种状态
 - 对于每个字符：之前记录的状态和当前字符的情况，确定：
 - 单词的开始位置，记录该下标start
 - 单词的结束位置：start到当前位置就是找到的单词
 - 单词的中间位置，继续
 - 尚未找到单词：继续
 - 要考虑单词字符之后就是字符串结束的情形



practice re

practice_email:

- local-part@domain → [name1.name2.name3.name@dn1.dn2.dn3.dn1](#) → (((name1\.){0,}\.name)@
- text 要得到text部分，注意要采用懒惰匹配 +?

practice_ipaddr:

- finditer或findall实现
- 正则表达式：注意.为元字符，需要转义

practice_re_date: 2016-10-31 --> 10/31/2016 或 2016-9-4 --> 09/04/2016

- 子模式，获取日期的同时得到年月日
- 月和日：如果个位数，前面要添加0，自己构造或者格式化（填充0）