

python语言入门与实践

习题课

主讲人：吴陈炜 博士

复旦大学智能机器人研究院 工程博士

北京大学信息科学与技术学院 理学硕士

浙江大学信息与电子工程学系 工学学士

网易（杭州）网络有限公司 项目经理

美国项目管理协会 PMP（项目管理专业人士）

之江实验室 科研主管

2020 年 5 月 19 日

上节课回顾

- 列表基础知识
- 字符串与列表的关系
- for循环遍历
- if条件语句

列表操作CheatSheet

Lists

Also see NumPy Arrays

```
>>> a = 'is'
>>> b = 'nice'
>>> my_list = ['my', 'list', a, b]
>>> my_list2 = [[4,5,6,7], [3,4,5,6]]
```

Selecting List Elements

Index starts at 0

Subset

```
>>> my_list[1]
>>> my_list[-3]
```

Slice

```
>>> my_list[1:3]
>>> my_list[1:]
>>> my_list[:3]
>>> my_list[:]
```

Subset Lists of Lists

```
>>> my_list2[1][0]
>>> my_list2[1][:2]
```

Select item at index 1
Select 3rd last item

Select items at index 1 and 2
Select items after index 0
Select items before index 3
Copy my_list

my_list[list][itemOfList]

List Operations

```
>>> my_list + my_list
['my', 'list', 'is', 'nice', 'my', 'list', 'is', 'nice']
>>> my_list * 2
['my', 'list', 'is', 'nice', 'my', 'list', 'is', 'nice']
>>> my_list2 > 4
True
```

List Methods

```
>>> my_list.index(a)
>>> my_list.count(a)
>>> my_list.append('!')
>>> my_list.remove('!')
>>> del(my_list[0:1])
>>> my_list.reverse()
>>> my_list.extend('!')
>>> my_list.pop(-1)
>>> my_list.insert(0, '!')
>>> my_list.sort()
```

Get the index of an item
Count an item
Append an item at a time
Remove an item
Remove an item
Reverse the list
Append an item
Remove an item
Insert an item
Sort the list

字符串：与列表同根同源的特殊序列

Strings

```
>>> my_string = 'thisStringIsAwesome'
>>> my_string
'thisStringIsAwesome'
```

String Operations

```
>>> my_string * 2
'thisStringIsAwesomethisStringIsAwesome'
>>> my_string + 'Innit'
'thisStringIsAwesomeInnit'
>>> 'm' in my_string
True
```

String Operations

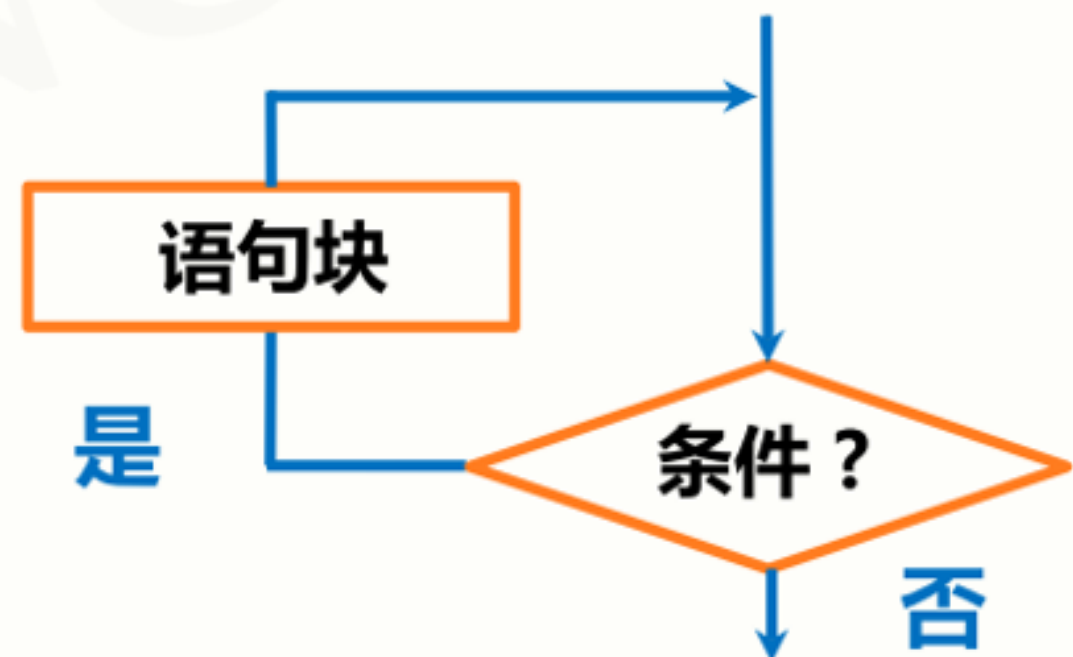
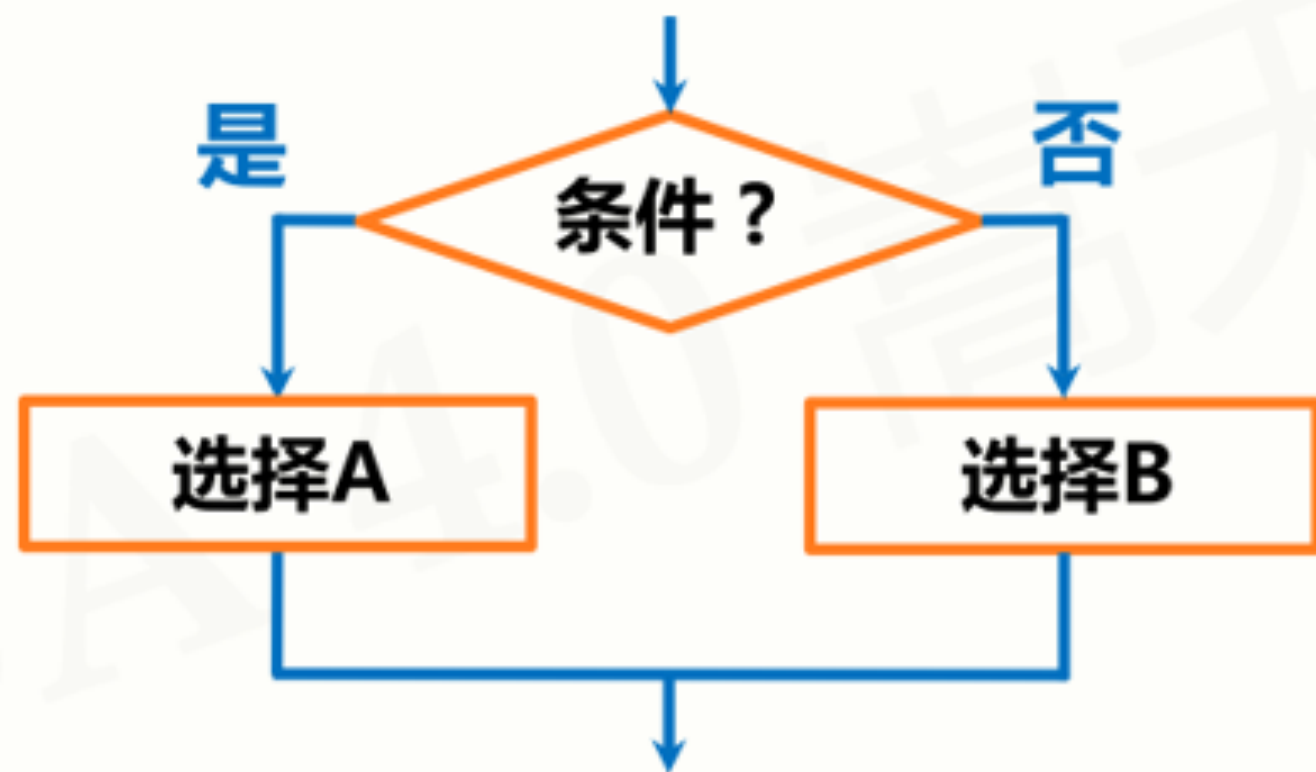
Index starts at 0

```
>>> my_string[3]
>>> my_string[4:9]
```

String Methods

>>> my_string.upper()	String to uppercase
>>> my_string.lower()	String to lowercase
>>> my_string.count('w')	Count String elements
>>> my_string.replace('e', 'i')	Replace String elements
>>> my_string.strip()	Strip whitespaces

if语句和for循环





今日测试习题

- 第一关(C): 打印九九乘法表
- 第二关(B): 寻找所有水仙花数
- 第三关(A): 泰勒级数计算圆周率
- Boss关(S): 哥德巴赫猜想的数值验证

震惊！潘石屹也在学python



潘石屹

11-15 11:43 来自 iPhone 11 P... 已编辑

+关注

我们为什么要学习Python语言？

在农业社会时，我们要学习驾驭马、驴、牛，让它们为我们出力、干活。

在工业社会时，我们要学会驾驭各种机器、火车、轮船、飞机、机床等等。

今天，我们要让机器听我们的指挥，我们就要学习机器能听懂的语言。这类语言也在不断的进化中，越来越接近我们的日常语言。我们选择了进化最好的一种：Python语言。

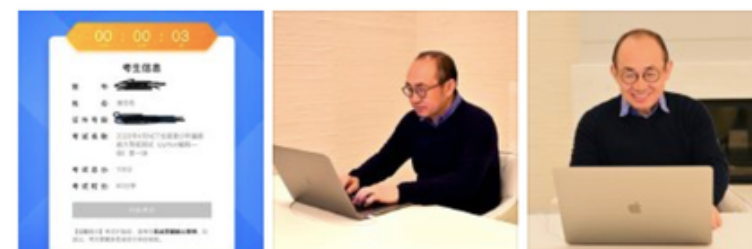
在信息时代，信息量变得越来越大，我们的头脑显得越来越小。如果问我在这个时代什么知识和技能是最有用的、最值得去学习的、最值得让它占据我们的大脑的，Python应该在其中。



潘石屹

4月19日 10:16 来自 iPhone 11 Pro Max

紧张的考试。（张欣 摄）



考生姓名：潘石屹

准考证号：200446000606687471

考试类型：Python编程

考试等级：一级

考试期次：2020年04期

考试成绩：99

合格分数线：60

考试结果：优秀

第一关：打印九九乘法表

按照以下格式打印九九乘法表：

```
1×1=1  1×2=2  1×3=3  1×4=4  1×5=5  1×6=6  1×7=7  1×8=8  1×9=9
2×2=4  2×3=6  2×4=8  2×5=10 2×6=12 2×7=14 2×8=16 2×9=18
3×3=9  3×4=12 3×5=15 3×6=18 3×7=21 3×8=24 3×9=27
4×4=16 4×5=20 4×6=24 4×7=28 4×8=32 4×9=36
5×5=25 5×6=30 5×7=35 5×8=40 5×9=45
6×6=36 6×7=42 6×8=48 6×9=54
7×7=49 7×8=56 7×9=63
8×8=64 8×9=72
9×9=81
```

● 考察知识点：

- for循环的使用；
- range函数的使用；
- print函数的使用；
- 字符串的使用；

第一关参考答案

```
11 ▼ for i in range(1,10):  
12     for j in range(i,10):  
13         print(f'{i}×{j}={i*j}',end='\t')  
14     print('')  
15
```

请思考：有没有其他的更简单的表达方法？

第二关：寻找所有水仙花数

所谓"水仙花数"是指一个三位数，其各位数字立方和等于该数本身。

例如：153是一个"水仙花数"，因为 $153=1^3+5^3+3^3$ 。

- 考察知识点：
 - 使用运算符提取数位
 - if语句的使用
 - for循环的使用

```
1 | [153, 370, 371, 407]
2 | [Finished in 1.2s]
```

第二关参考答案：

```
def is_narcissistic(number):  
    unit = number%10  
    decade = int(number/10)%10  
    hundred = int(number/100)  
    if unit**3+decade**3+hundred**3 == number:  
        return True  
    else:  
        return False  
  
narcissistic_number = []  
for i in range(100,1000):  
    if is_narcissistic(i):  
        narcissistic_number.append(i)  
    else:  
        pass  
print(narcissistic_number)
```



第三关：泰勒级数计算圆周率

- 根据泰勒级数展开，我们可以得到圆周率计算公式如下：

$$\frac{\pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \dots$$

- 可以根据此公式来计算圆周率的近似值

```
3.1415936535907742  
[Finished in 0.3s]
```

第四关：暴力验证哥德巴赫猜想

哥德巴赫猜想：

任何大于2的偶数都能够写成两个质数相加的形式。



验证思路：

编写哥德巴赫分解程序，输入一个数，输出他的两个子质数，如果找不到，返回False。遍历整数，查看是否有False的情况，如果有False的情况，我们认为哥德巴赫猜想不成立，如果没有False，我们认为歌德巴赫猜想是不能证伪的。

质数的判断

质数的概念

在大于1的自然数中，除了1和本身以外不再有其他因数的自然数。

我们可编写函数通过if语句和for循环来判断一个数是否为质数：

```
1  def isprime(num):  
2      for i in range(2,num):  
3          if num%i==0:  
4              return False  
5      return True  
6  print(isprime(97))  
7  print(isprime(99))  
8
```

```
True  
False  
[Finished in 0.1s]
```


歌德巴赫分解

输入一个数，输出他的两个子质数，如果找不到，返回False。

```
9  def goldbachresolve(num):
10     for i in range(2,num):
11         if isprime(i):
12             if isprime(num-i):
13                 print(num,'=',i,'+',num-i)
14                 return True
15     return False
16 print(goldbachresolve(1998))
```

```
1998 = 5 + 1993
True
[Finished in 0.1s]
```


枚举法验证猜想

遍历一定范围的整数，查看是否有False的情况，如果有False的情况，我们认为哥德巴赫猜想不成立，如果没有False，我们认为歌德巴赫猜想没有被证伪的，也即证明了在选定的整数范围内是成立的。

```
def test_goldbach():  
    for i in range(4,1000,2):  
        if goldbachresolve(i):  
            pass  
        else:  
            return('歌德巴赫猜想不成立！')  
    return('歌德巴赫猜想在给定范围内成立！')  
  
print(test_goldbach())
```

```
984 = 7 + 977  
986 = 3 + 983  
988 = 5 + 983  
990 = 7 + 983  
992 = 73 + 919  
994 = 3 + 991  
996 = 5 + 991  
998 = 7 + 991  
歌德巴赫猜想在给定范围内成立！
```

数值计算表明：哥德巴赫猜想在1000以内成立！

作业与思考

- 复习巩固以上知识点，自己动手运行并且通过代码
- 思考如何灵活运用已学过的知识解决约瑟夫环问题：

据说著名犹太历史学家 Josephus 有过以下的故事：在罗马人占领乔塔帕特后，39 个犹太人与 Josephus 及他的朋友躲到一个洞中，39个犹太人决定宁愿死也不要被敌人抓到，于是决定了一个自杀方式，41个人排成一个圆圈，由第1个人开始报数，每报数到第3人该人就必须自杀，然后再由下一个重新报数，直到所有人都自杀身亡为止。

然而Josephus 和他的朋友并不想遵从。首先从一个人开始，越过 $k-2$ 个人（因为第一个人已经被越过），并杀掉第 k 个人。接着，再越过 $k-1$ 个人，并杀掉第 k 个人。这个过程沿着圆圈一直进行，直到最终只剩下一人留下，这个人就可以继续活着。问题是，给定了 n 和 k ，一开始要站在什么地方才能避免被处决？