

Python语法基础

Introduction to Python I V3.0

安德鲁老师

扫描二维码关注微信/小程序，获取最新面试题及权威解答



微信扫一扫，使用小程序



微信扫一扫，关注微信公众号

昵称：安德鲁

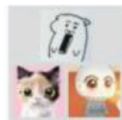
前BAT高级工程师，现FLAG AI工程师。

对深度学习，强化学习，自然语言处理感兴趣。



本期负责课题：Python语法基础-1，朴素贝叶斯，(深度)强化学习

- 加入试听群方便交流
- 对课程有任何问题，可以微信联系Justin
- <http://jiuzhangai.mikecrm.com/biWI5UU>



第三期《人工智能集训营》试听群



该二维码7天内(11月28日前)有效, 重新进入将更新

- 让我们的代码更加“Pythonic”
- Python基础类型之
 - 整型Integer
 - 布尔类型Boolean
 - 字符串String
- Python内置数据结构之List和tuple
- Python的控制流语句IF和Else
- Python的循环结构
- Q&A

```
1      a = 1
```

```
2
```

```
3
```

```
4
```

变量名必须是大小写英文、数字和下划线_的组合，且不能用数字开头

Python中变量赋值的多种“姿势”：

```
1      a, b, c = 1, 1e-5, 'linpz'
```

```
2
```

```
3      print(a, b, c)
```

```
4
```

- -10, 1, 100
- `2 ** 100, 2 ** 1000` # Python的整数可以非常大
- `0x100` # 十六进制用0x前缀和0-9, a-f表示10 - 15

交换两个整数:

```
a, b = 10, 20
```

```
# NP:
```

```
temp = a
```

```
a = b
```

```
b = temp
```

```
# P:
```

```
a, b = b, a
```

- -1.1, -2.0
- 1.2e10, 1.2e-5 # 科学计数法

- True, False
- $3 > 1$, $1 > 3$
- and, or, not操作
- 布尔值经常用在条件判断
- None 空值的表示

Python支持链式比较

```
a, b = 10, 20  
  
# NP:  
a >= 1 and a <= b and b <= 100  
  
# P:  
1 <= a <= b <= 100
```

- 以单引号'或双引号"括起来的任意文本
- 'a', 'abc', "xyz"
- "或""本身只是一种表示方式，不属于字符串
- 如果'本身也是一个字符，那就可以用""括起来，比如“I'm Andrew”包含的字符是I, ' , A, n, d, r, e, w空格
- 提问：如果字符串本身也需要包含单引号和双引号呢？

Python字符串的常见操作

- 连接两个字符串
- 计算字符串的长度
- 遍历字符串中的每一个字符
- 替换，截取，查找
 - 能否修改字符串中的字母？

```
1     str1 = 'Jiuzhang!'  
2     str2 = 'Hello'  
3  
4     print(str1 + ' ' + str2)  
5
```

- + 字符串连接
- * 重复输出字符串
- % 格式字符串

提问：如何给子串连接一个整数，或者一个实数？

str 函数: 其他任何类型的变量转变成为字符串

- **len(string)**

举例：输入密码的时候，长度不够，比如至少**6**位的密码，只输入了**5**位，不合格

len 函数：获取字符串的长度

```
1  str1 = 'Jiuzhang!'  
2  print(str1[0])  
3  print(str1[1])  
4  print(str1[2])  
5
```

通过name[index]的方式访问

```
1  str1 = "jiuzhang"  
2  print(str1[-1])  
3  print(str1[-2])  
4  print(str1[-3])  
5
```


- for循环遍历每一个字符
- 两种for循环方式，取决于是否需要index
 - for c in str:
 - for idx in range(len(str)):

- 能否修改字符串中的字母？
- 替换：

```
1 str1 = "this is string example...wow!!! this is really string"
2 print str1.replace("is", "was")
3 print str1.replace("is", "was", 3)
```

- 截取: 左闭右开

```
1 str1 = "jiuzhang"
2 print(str1[1:4])
```

- 查找

```
1 str1 = "Hello, welcome to my world."
2 x = str1.find("welcome")
3 print(x)
```

<https://www.lintcode.com/problem/reverse-string>

Description

Write a function that takes a string as input and returns the string reversed.

Have you met this question in a real interview?

Example

Given s = "hello"
return "olleh".

```
# NP:
result = ''
for i in range(len(s)-1, -1, -1):
    result += s[i]
return t

# P:
return s[::-1]
```

Python的基本数据结构之一:

```
1 a = [1, 2, 3, 5, 100]
2
```

提问: **list**中的元素必须是同一种类型么?



列表的常见操作:

```
1 persons = ['Bob', 'Alice', 'Linpz']  
2
```

- 索引、切片
- 加、乘
- 元素检测
- 获取list长度
- 求list的最大, 最小值

- 访问第*i*个学生的分数
- 修改第*i*个学生的分数
- 查询是否含有100分（满分）的同学
- 查询班级中的最低分、最高分

```
# NP:
scores = [90, 100, 98, 97, 95, 96]
min = max = scores[0]
for score in scores:
    if score < min:
        min = score
    if score > max:
        max = score
```

```
# P:
scores = [90, 100, 98, 97, 95, 96]
max(scores)
min(scores)
```


- 添加元素
 - append操作
 - extend操作
- 删除元素
 - del 操作
 - pop 操作
 - remove 操作

生成100以内5的倍数的数形成一个List

```
# NP:
result = []
for i in range(101):
    if i % 5 == 0:
        result.append(i)

# P:
result = [i for i in range(101) if i % 5 == 0]
```

- `result = [0] * 100` 可以生成包含100个0的list
- 但是需要在明确元素是常量的时候才使用

```
1 class Student:
2     def __init__(self, val):
3         self.score = val
4
5     def __repr__(self):
6         return str(self.score)
7
8     def __str__(self):
9         return str(self.score)
```

```
11 student = Student(1)
12 group = [student] * 10
13 # 1. 请问group[0]的score是多少?
14 print(group[0])
15 # 2. 请问group[0]的score是多少?
16 group[0].score = 3
17 print(group[0])
18 # 3. 请问group[0]的score是多少?
19 group[1].score = 2
20 print(group[0])
21
22 print(group)
```

如果List中所有元素都是字符串，我们需要把他们拼接起来

```
# NP:
strs = ["linpz", "is", "nice"]
result = ''
for s in strs:
    result += s + ' '
print(result)

# P:
strs = ["linpz", "is", "nice"]
print(' '.join(strs))
```

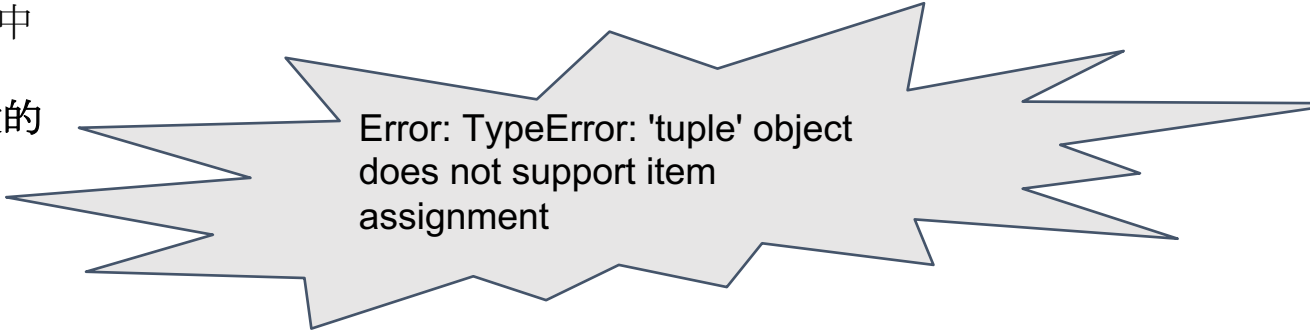
tuple

```
4 tup1 = ('Hello', 'Jiuzhang', 5)
5 tup2 = (1, 2, 3 )
6 tup3 = "a", "b", "c"
7
8 # Create empty tuple
9 tup4 = ()
10
11 tup5 = (10)
12
```

提问：上述tuple的构建中，哪些是错误的？

- 访问第i个元素
- 提取一段sub tuple
- 获取tuple中的元素个数
- 获取tuple中的最大值，最小值
- 判断一个元素是否在tuple中
- **tuple**中的元素是无法修改的
 - **tup2[2] = 4** 出错

```
1 tup1 = ('Hello', 'Jiuzhang', 5)
2 tup2 = (1, 2, 3)
3 tup3 = "a", "b", "c"
4
5 # Create empty tuple
6 tup4 = ()
7
8 tup2[2] = 4
9
10
```



Error: TypeError: 'tuple' object
does not support item
assignment

- 连接操作生成了新的tuple
 - 因为tuple不支持修改，所以只能生成新的
- 使用del语法删除已经存在的tuple
 - del tup2

表达式	结果	说明
<code>len((1,2,3))</code>	3	获取tuple的元素个数
<code>4 in (4,5,6)</code>	True	判断元素是否在tuple中出现
<code>for x in (4,5,6): print x</code>	4 5 6	遍历tuple
<code>(1,2,3)+(4,5,6)</code>	(1,2,3,4,5,6)	连接生成新的tuple

- `tuple([])`
 - `tuple([1,2,3]) => (1,2,3)`
- `list(())`
 - `list((1,2,3)) => [1,2,3]`

String的非空判断

```
name = 'linpz'

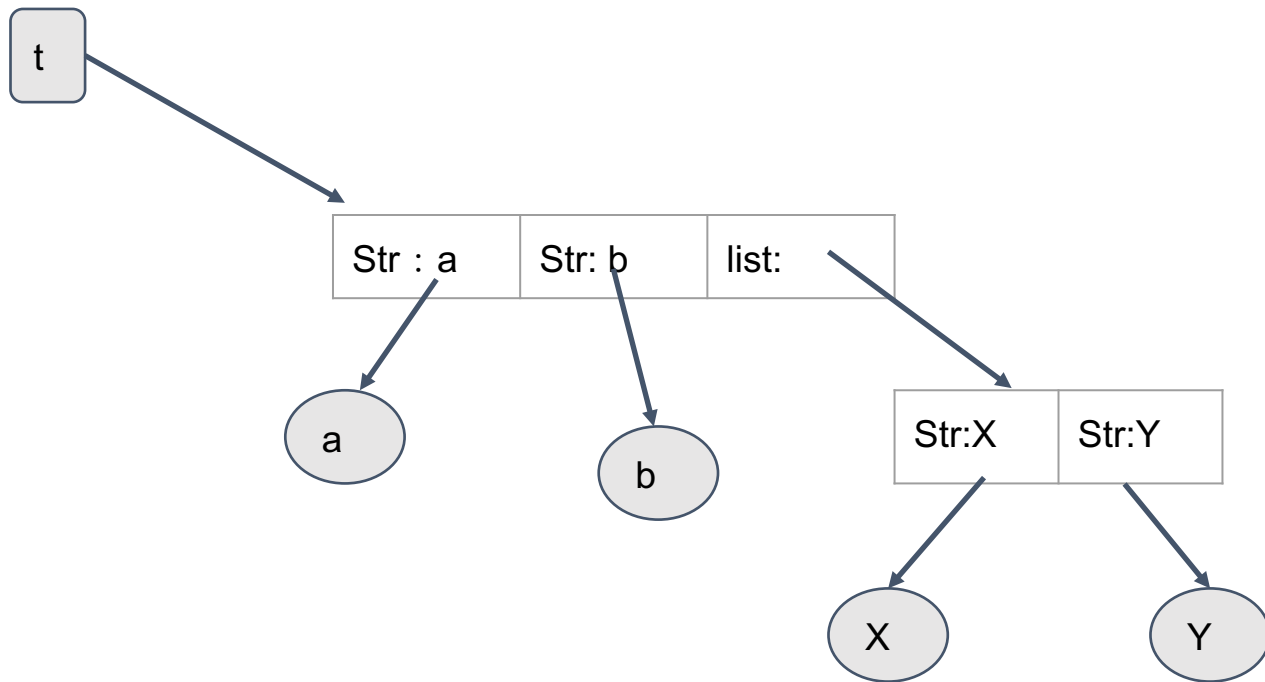
# NP:
if name != '':
    print(name)

# P:
if name:
    print(name)
```

对List/Tuple的非空判断

```
names = ['linpz']  
  
# NP:  
if len(names) > 0:  
    print(names)  
  
# P:  
if names:  
    print(names)
```

```
1 t = ('a', 'b', ['X', 'Z'])
```



- 一个很好的类比例子：
 - 值: “Hello World”, 10, 3.14
 - 引用: <https://www.jiuzhang.com>

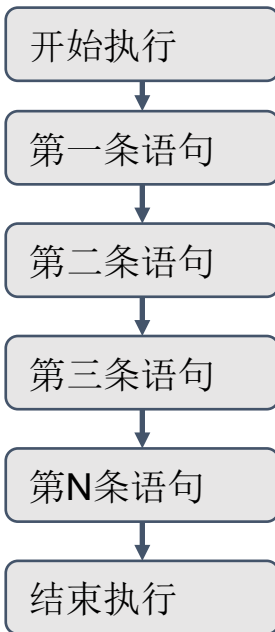
```
1 scores = {"Andrew": 100, "Mina": 89, "Lucy": 96}
2 print(scores["Andrew"])
3 print(scores["Mina"])
4 del scores["Mina"]
5 scores["Gary"] = 88
6 print(scores.keys())
7 print(scores.values())
8 print(scores.items())
9
10 for item in scores.items():
11     print(type(item))
12
13 for key, val in scores.items():
14     print(key, val)
```

- 顺序结构
- 控制结构
- 循环结构


```
1    a = 5
2    b = 2 * a
3    print(a)
4
```

特点：

顺序执行语句，从上到下执行



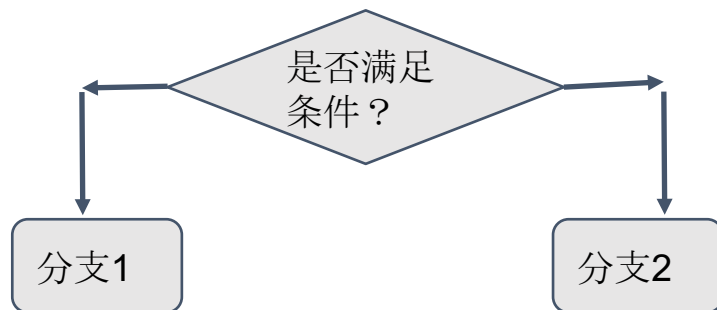
```
1  a = 5
2
3  if a < 5:
4      print(a)
5  else:
6      print(b)
```

```
1  a = 5
2
3  if a >= 5:
4      print(a)
5  else:
6      print(b)
```

特点：

根据条件，选择进入程序执行的分支

问题：同样都是**b**没有定义，那一边的语句块会报错？



多重if语句

```
if <条件判断1>:  
    <执行1>  
elif <条件判断2>:  
    <执行2>  
elif <条件判断3>:  
    <执行3>  
else:  
    <执行4>
```

if语句嵌套



```
if <条件判断1>:  
    if <条件判断2>:  
        <执行2>  
    else <条件判断3>:  
        <执行3>  
else:  
    <执行4>
```

```
# NP:
if a > 10:
    b = 20
else:
    b = 5

# P:
b = 20 if a > 10 else 5
```

如何写好if语句，好的Python if语句的格式是怎么样子的？

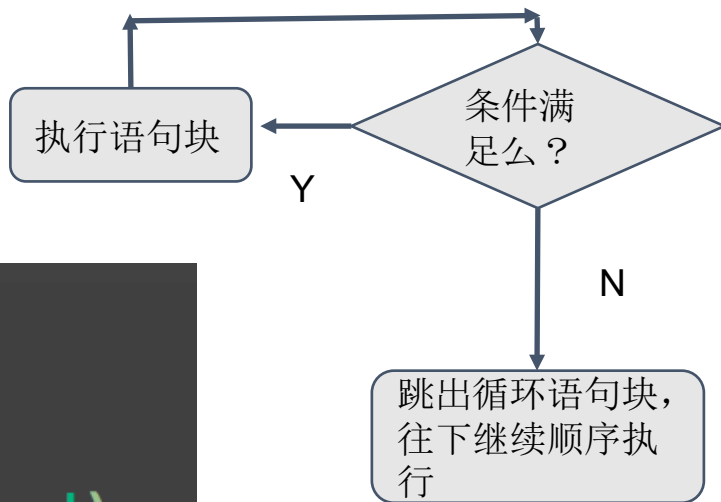
参考资料：

<https://google.github.io/styleguide/pyguide.html?showone=Statements#Statements>

- 缩进的空格数量是**可变的**
- 所有代码块语句必须包含**相同的缩进空格数量**
- 硬性要求我们一律使用**4空格**或者**tab**（最好把**tab**都设置成对应**4空格**）

特点：

不断重复的做一件事情



```
1     nums = [2, 3, 4, 5]
2
3     for num in nums:
4         print(num, end=', ')
5
```

- 1.如何遍历从0~100，或者从0到n
- 2.如何遍历从m到n
- 3.如何遍历从n到m ($n > m$)
- 4.如何遍历从n到m，但是每隔2个（如：100，98，96.....）

range的用法

range(101) 会生成一个0~100的list

range(m, n) 会生成一个m~n-1的list $m < n$

range(n, m, -1) 会生成一个n到m-1的降序的list

range(n, m, -2) 会生成一个n到m-1的降序的每隔一个数的list



有了range, 我们如何实现上一页ppt提到的几个问题?

使用enumerate函数：

- 可以获得下标
- 可以获得数值

```
# NP:
scores = [90, 100, 98, 97, 95, 96]
for i in range(len(scores)):
    print(i, scores[i])

# P:
scores = [90, 100, 98, 97, 95, 96]
for index, score in enumerate(scores):
    print(index, score)
```

While 循环的使用

```
1  while condition:  
2  
3      statements.....  
4  
5  
6
```

循环中的continue语句

```
1     for i in range(5):  
2         if i == 3:  
3             continue  
4         print(i)  
5
```

提问：这个的输出结果是？

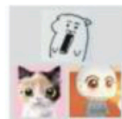
```
1     for i in range(5):  
2         if i == 3:  
3             break  
4         print(i)  
5
```

提问：这个片段的输出结果是？



- `break`语句可以在循环过程中直接退出循环
- `continue`语句可以提前结束本轮循环

- <http://jiuzhangai.mikecrm.com/biWI5UU>
- 加入试听群方便交流
- 对课程有任何问题，可以微信联系Justin



第三期《人工智能集训营》试听群



该二维码7天内(11月28日前)有效，重新进入将更新





谢谢大家