

分支结构

王肖燕



➔ 内容回忆

- ① 学习哪些数据类型？为什么会分类型？
- ② 算术运算符*的含义？
- ③ 关系运算符？逻辑运算符？
- ④ 什么是字符串？
- ⑤ 用索引如何访问字符串中的字符及子串？
- ⑥ Input函数如何使用？
- ⑦ `>>> 3 * "star"`
- ⑧ `int()`, `float()`, `str()`
- ⑨ 变量的命名规则是？

➡ 内容回忆

- 与其它程序设计语言不同
- Python关系运算符最大的特点是可以连用
- `>>> 1<3<5`
- `>>> True`
- `>>> 3<5>2`
- `>>> True`
- `>>> 1>6<8`
- `>>> False`
- `>>> 'Hello' > 'world'`
- `>>> False`
- 操作数之间必须可比较大小，即必须有意义



内容介绍

一．引例

二．分支结构

- ✓ 单分支结构、双分支结构、多分支结构

三．分支嵌套

- ✓ 结构嵌套结构的情况

四．应用举例

➡ 引例1

1. 输入三个数，判断能否组成三角形？能够组成，打印“可以组成三角形”。

① 输入使用input函数

② a, b, c能够组成三角形，满足：

- $a+b>c$ ，并且
- $a+c>b$ ，并且
- $b+c>a$

➔ 单分支结构

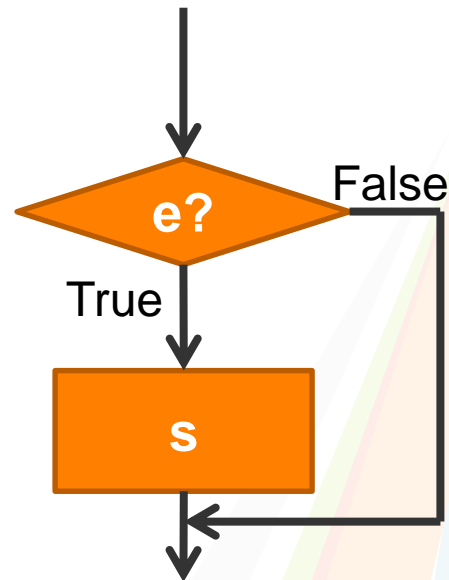
• 两种形式

✓ if 表达式 **e**:

语句段 **s**

• 说明

- 表达式**e**：算术表达式，关系表达式或逻辑表达式
- 表达式值：按非零为True，零为False进行判断
- **s**可以是一条语句，也可以是多条语句
- 格式：语句段**s**相比上一句要缩进一个Tab键



➡ 缩进

- Python中代码的缩进非常重要
 - 按一次Tab键表示一次缩进（我们的规定）
- 它是体现代码逻辑关系的重要方式
- 同一个代码块必须保证**相同的缩进量**
- 如果代码量少可以一行写多个语句，每两句之间使用**分号**隔开
 - `>>> if True : print(3) ; print(5)`

➡ 引例1实现

```
a = float(input("请输入三角形的第一条边:"))
b = float(input("请输入三角形的第二条边:"))
c = float(input("请输入三角形的第三条边:"))

if a+b>c and a+c>b and b+c>a:
    print("可以组成三角形")
```

```
请输入三角形的第一条边:1
请输入三角形的第二条边:2
请输入三角形的第三条边:3
>>>
===== 1
请输入三角形的第一条边:2
请输入三角形的第二条边:2
请输入三角形的第三条边:3
可以组成三角形
>>>
```


引例2

2. 输入三个数，判断能否组成三角形？能够组成，打印“可以组成三角形”，否则打印“此三边不能组成三角形”。

- 对有对的处理，错有错的去处**
- 使用双分支来解决**

➔ 双分支结构

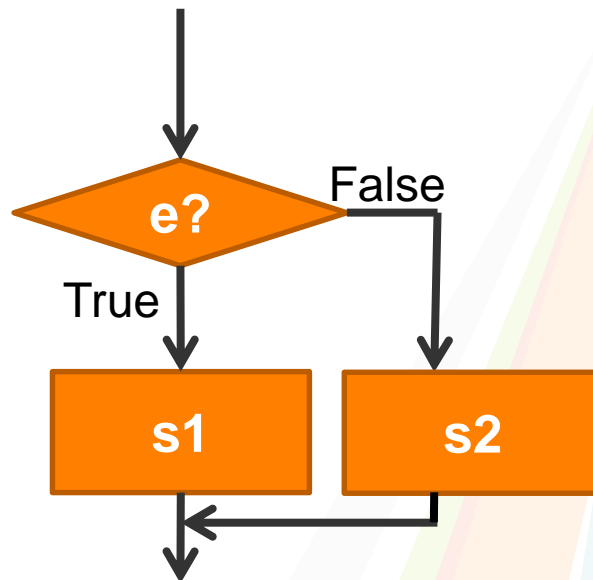
- 形式：

if 表达式 e:
 语句段 s1

else:
 语句段 s2

- s1和s2可以分别多含多条语句

- 写法：包含关键字if或else的语句要加：作为结束



➡ 引例2实现

```
a = float(input("请输入三角形的第一条边:"))
b = float(input("请输入三角形的第二条边:"))
c = float(input("请输入三角形的第三条边:"))

if a+b>c and a+c>b and b+c>a:
    print("可以组成三角形")
else:
    print("此三边不能组成三角形")
```

```
请输入三角形的第一条边:1
请输入三角形的第二条边:2
请输入三角形的第三条边:3
此三边不能组成三角形
>>>
```

```
===== 1
请输入三角形的第一条边:2
请输入三角形的第二条边:2
请输入三角形的第三条边:3
可以组成三角形
>>>
```

引例

3. 输入三角形三条边 a, b, c ，判断是何种类型的三角形？（不合理边长，等边，等腰，任意三角形）

- 多种情况对应多种分支**
- 只使用单分支或双分支都无法处理**

➡ 多分支结构

• 形式：

if 表达式 e_1 :

语句段 S_1

elif 表达式 e_2 :

语句段 S_2

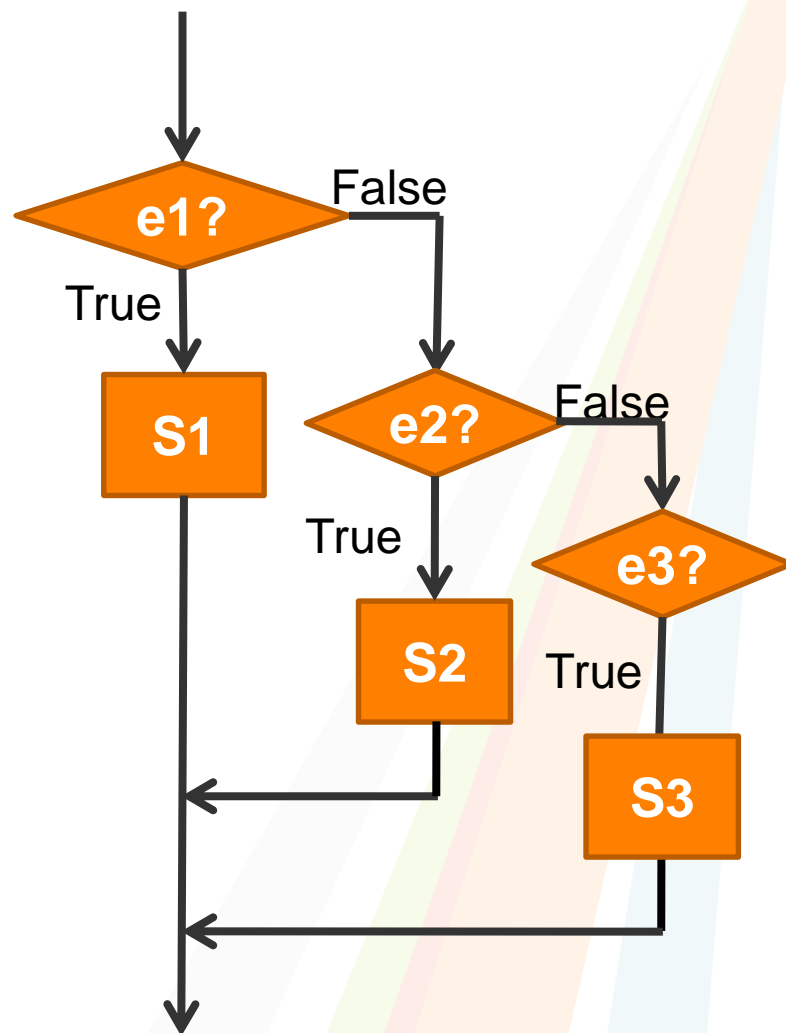
.....

elif 表达式 e_n :

语句段 S_n

[else:

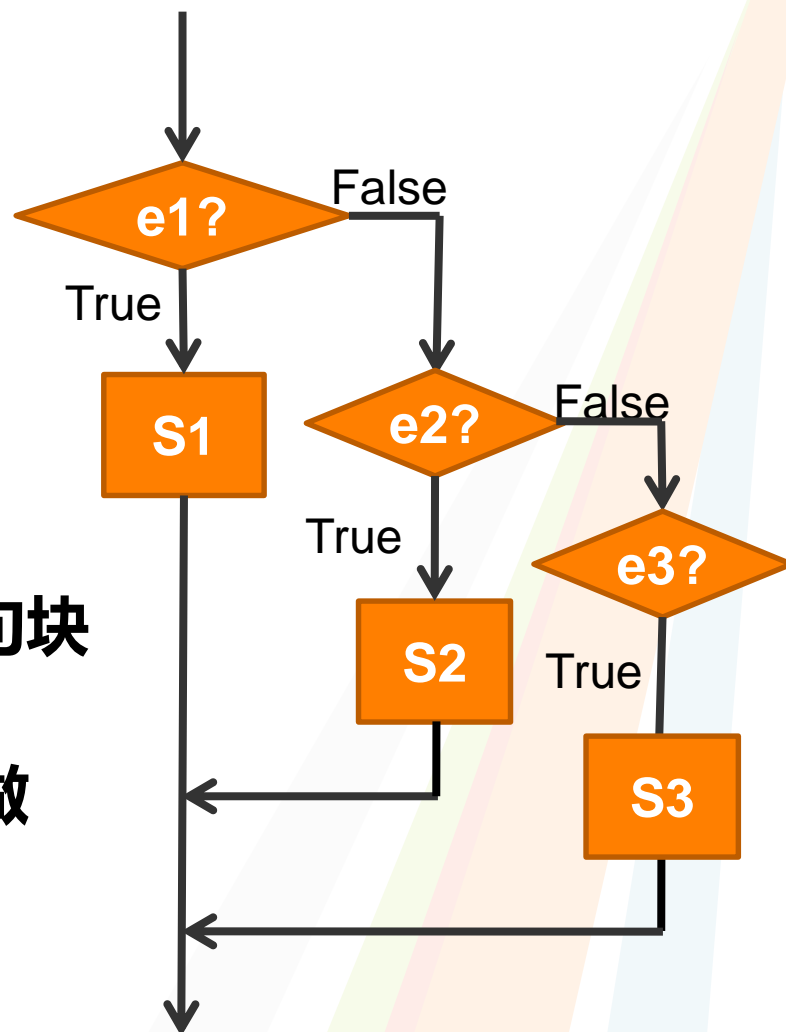
语句段 $S_{n+1}]$



➡ 多分支结构

• 注意：

- 找到一个满足的分支，退出
- 都不满足
 - 有else分支，执行else语句块
 - 没有else分支，什么也不做
- elif表示 else if



➡ 引例3实现

```
a = float(input("请输入三角形的第一条边:"))
b = float(input("请输入三角形的第二条边:"))
c = float(input("请输入三角形的第三条边:"))

if not(a+b>c and a+c>b and b+c>a):
    print("此三边不能组成三角形")
elif a==b and b==c:
    print("等边三角形")
elif a==b or a==c or b==c:
    print("等腰三角形")
else:
    print("任意三角形")
```

```
请输入三角形的第一条边:3
请输入三角形的第二条边:3
请输入三角形的第三条边:3
等边三角形
>>>
```

```
请输入三角形的第一条边:2
请输入三角形的第二条边:2
请输入三角形的第三条边:3
等腰三角形
>>>
```

```
请输入三角形的第一条边:3
请输入三角形的第二条边:4
请输入三角形的第三条边:5
任意三角形
>>>
```



If嵌套

#另一种解法，问题划分成两个集合：能和不能

If $a+b>c$ **and** $a+c>b$ **and** $b+c>a$:

对情况的进一步判断

else:

print(“此三边不能组成三角形”)

➔ 嵌套实现

```
a = float(input("请输入三角形的第一条边:"))
b = float(input("请输入三角形的第二条边:"))
c = float(input("请输入三角形的第三条边:"))

if a+b>c and a+c>b and b+c>a:
    if a==b and b==c:
        print("等边三角形")
    elif a==b or a==c or b==c:
        print("等腰三角形")
    else:
        print("任意三角形")
else:
    print("此三边不能组成三角形")
```

➡ 应用举例1

1. 判断你输入的字符串中第3个字符是不是's'。
如果是，隔一行输出“是”，否则输出“不是”。

✓输入：一个长度大于3的字符串

✓处理：

① 如果字符串中第3个字符是's'，打印“是”

② 如果字符串中第3个字符不是's'，打印“不是”

✓输出：打印判断结果

➡ 应用举例2

2. PM2.5空气质量提醒

✓ 输入：PM2.5的值

✓ 处理：

- ① 如果PM2.5值大于等于75，打印“空气污染警告”
- ② 如果PM2.5值小于75，并且大于等于35，打印“空气质量良，建议适度户外运动”
- ③ 如果PM2.5值小于35，打印“空气质量优，建议户外运动”

✓ 输出：打印空气质量提醒

➡ 应用举例3

3. 根据身体质量指数BMI进行指标分析

✓ 输入：某人的体重（Kg） w 和身高（m） h

✓ 处理：

- ① 计算身体质量指数 $BMI = w \div h^2$
- ② 如果BMI小于18.5，分类为“偏瘦”
- ③ 如果BMI值大于等于18.5，并且小于24，分类为“正常”
- ④ 如果BMI值大于等于24，并且小于28，分类为“偏胖”
- ⑤ 如果BMI值大于等于28，分类为“肥胖”

✓ 输出：打印此人指标分类信息

➡ 练习1

- 将输入的学生成绩（百分制），按如下规则转换为相应的等级：
 - 90 ~ 100分：A级
 - 80 ~ 90分(不包括90)：B级
 - 70 ~ 80分(不包括80)：C级
 - 60 ~ 70分(不包括70)：D级
 - 0 ~ 60分(不包括60)：E级

➡ 练习2

- 编写程序，实现分段函数计算，如下所示：

$$y = \begin{cases} 0 & x < 0 \\ x & 0 \leq x < 5 \\ 3x - 5 & 5 \leq x < 10 \\ 0.5x - 2 & 10 \leq x < 20 \\ 0 & x \geq 20 \end{cases}$$

➡ 练习3

- 编写程序计算货物运费T。设货物运费每吨单价p(元)与运输距离s(公里)之间有如下关系：

P=	{	30	$S < 100$
		27.5	$100 \leq S < 200$
		25	$200 \leq S < 300$
		22.5	$300 \leq S < 400$
		20	$S \geq 400$

- 根据要托运的货物W吨和距离S公里，计算运费
 $T = P * W * S.$

Thank You!

