



Python语言基础与应用

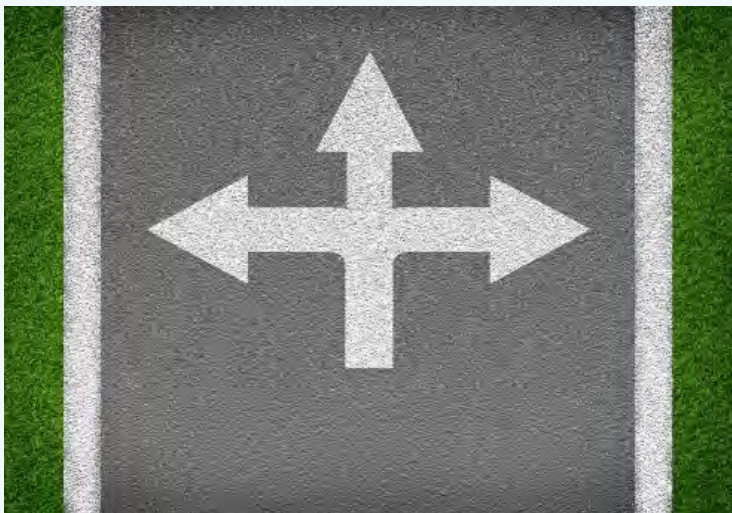
计算和控制流 / 控制流：条件分支语句 (if)

控制流：条件分支语句 (if)

- › 十字路口的选择
- › 条件分支的要素
- › 条件语句
- › 多种情况的条件分支

十字路口的选择

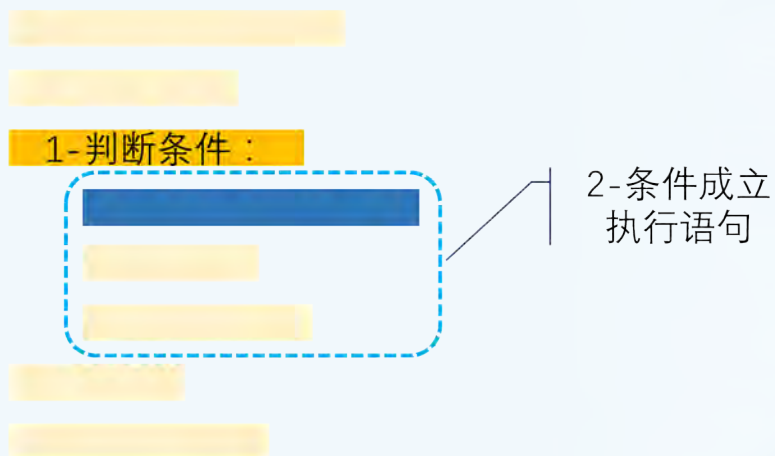
- › **条件分支让程序有了判断力**
 - 根据计算机内部情况(如变量值)，来决定下一步做什么，这样的控制流程，就称为条件分支
- › **根据预设条件来控制下一步该运行哪段语句**



条件分支的要素

› 基本要素

- 1: 预设的判断条件
- 2: 达成条件后执行的语句

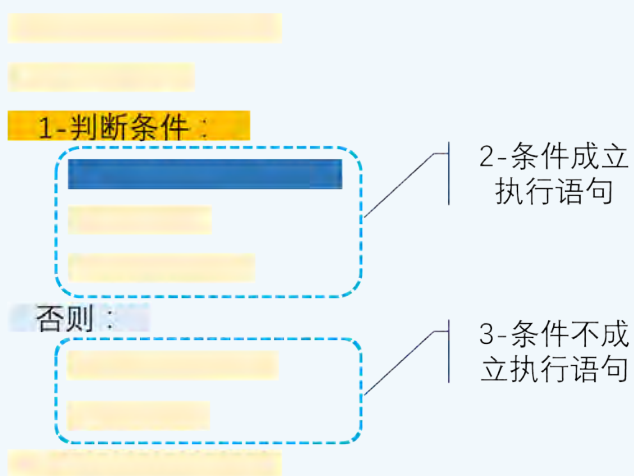


条件分支结构两个基本要素

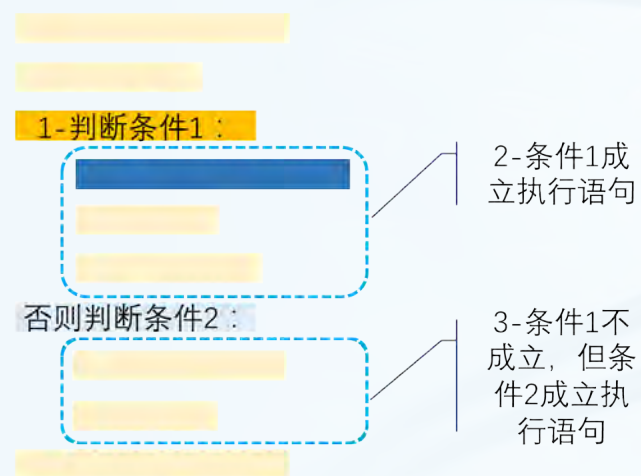
条件分支的要素

› 扩展要素

- 1: 当条件不满足时执行的语句
- 2: 多条件时哪个满足执行哪个的条件



条件分支结构扩展要素1



条件分支结构扩展要素2

条件语句

› Python中条件分支的实现

```
if <逻辑表达式>:  
    <语句块1>
```

.....

```
else:  
    <语句块2>
```

- if和else都是“保留字”
- “逻辑表达式”是指所有运算的结果为逻辑类型（True或False）的表达式
- “语句块”就是条件满足后执行的一组语句
- 冒号表示语句层次
- 语句块缩进

条件语句

- › 各种类型中某些值会自动被转换为False, 其它值则是True:

None, 0, 0.0, '', [], (), {}, set()

- › 实例：求距离的程序

```
m = 327
n = 829
d = m - n
if d < 0:
    d = -d
else:
    d = d
print("m:", m)
print("n:", n)
print('distance:', d)
```

```
>>>
===== RESTART: /Users/chenbin/Documents/教学项目/Python教材/code/distance.py
m: 327
n: 829
distance: 502
```

多种情况的条件语句

› 使用多条if语句进行判定

有时需区分多种情况，对应每种情况执行相应的语句



```
age = 20
print("年龄: ", age)
if 0 <= age <= 6:
    print("童年")
if 7 <= age <= 17:
    print("少年")
if 18 <= age <= 40:
    print("青年")
if 41 <= age <= 65:
    print("中年")
if age >= 66:
    print("老年")
```


多种情况的条件语句

› 使用else子句进行判定

```
age = 20
print("年龄: ", age)
if 0 <= age <= 6:
    print("童年")
else:
    if 7 <= age <= 17:
        print("少年")
    else:
        if 18 <= age <= 40:
            print("青年")
        else:
            if 41 <= age <= 65:
                print("中年")
            else:
                print("老年")
```

```
===== RESTART: /Users/chenbin/Documents/教学项目/Python教材/code/age.py =====
年龄:  20
青年
```

多种情况的条件语句

› 使用elif语句进行判定

if <逻辑表达式1>:

 <语句块1>

elif <逻辑表达式2>:

 <语句块2>

elif <逻辑表达式3>:

 <语句块3>

... ..

else:

 <语句块n>

```
>>> a = 12
>>> if a > 10:
        print ("Great!")
elif a > 6:
        print ("Middle!")
else:
        print ("Low!")
```

Great!