

## 二、运算符

运算符的分类：

- 算数运算符
- 赋值运算符
- 复合赋值运算符
- 比较运算符
- 逻辑运算符

### 2.1 算术运算符

```
1 +  
2 -  
3 *  
4 /  
5 // # 整除  
6 % # 取余  
7 ** # 指数  
8 () # 小括号
```

混合运算优先级顺序： `()` 高于 `**` 高于 `*` `/` `//` `%` 高于 `+` `-`

### 2.2 赋值运算符

- 单个变量赋值

```
1 num = 1  
2 print(num)
```

- 多个变量赋值

```
1 num1, float1, str1 = 10, 0.5, 'hello world'  
2 print(num1)  
3 print(float1)  
4 print(str1)
```

- 多变量赋相同值

```
1 a = b = 10  
2 print(a)  
3 print(b)
```

## 2.3 复合赋值运算符

```
1 +=
2 -=
3 *=
4 /=
5 //= # 整除赋值运算符  c//=a -> c =c//a
6 %= # 取余赋值运算符  c%=a -> c = c%a
7 **= # 幂赋值运算符    c**=a -> c = c**a
```

```
1 a = 100
2 a += 1
3 # 输出101 a = a+1 , 最终a = 100+1
4 print(a) # 101
5
6 b = 10
7 b -= 1 # b = b-1
8 print(b) # 9
9
10 #注意: 先算复合运算符右边的表达式; 算复合赋值运算
11 c = 10
12 # c +=3 -- c=c+3
13 c += 1+2 # c= c+1+2
14 print(c) # 13
15
16 d = 10
17 d*=1+2
18 print(d) # 30
19
```

## 2.4 比较运算符

比较运算符也叫关系运算符，通常用来判断。

运算符	描述	实例
==	判断相等。如果两个操作数的结果相等，则条件结果为真(True)，否则条件结果为假(False)	如a=3,b=3, 则 (a == b) 为 True
!=	不等于。如果两个操作数的结果不相等，则条件为真(True)，否则条件结果为假(False)	如a=3,b=3, 则 (a == b) 为 True 如a=1,b=3, 则(a != b) 为 True
>	运算符左侧操作数结果是否大于右侧操作数结果，如果大于，则条件为真，否则为假	如a=7,b=3, 则(a > b) 为 True
<	运算符左侧操作数结果是否小于右侧操作数结果，如果小于，则条件为真，否则为假	如a=7,b=3, 则(a < b) 为 False
>=	运算符左侧操作数结果是否大于等于右侧操作数结果，如果大于，则条件为真，否则为假	如a=7,b=3, 则(a < b) 为 False 如a=3,b=3, 则(a >= b) 为 True
<=	运算符左侧操作数结果是否小于等于右侧操作数结果，如果小于，则条件为真，否则为假	如a=3,b=3, 则(a <= b) 为 True

## 2.5 逻辑运算符

运算符	逻辑表达式	描述	实例
and	x and y	布尔"与": 如果 x 为 False, x and y 返回 False, 否则它返回 y 的值。	True and False, 返回 False。
or	x or y	布尔"或": 如果 x 是 True, 它返回 True, 否则它返回 y 的值。	False or True, 返回 True。
not	not x	布尔"非": 如果 x 为 True, 返回 False。如果 x 为 False, 它返回 True。	not True 返回 False, not False 返回 True

```
1 a = 1
2 b = 2
3 c = 3
4 # 与
5 print((a<b) and (b<c)) # True
6 print((a>b) and (b<c)) # False
7 # 或
8 print((a>b) or (b<c)) # True
9 # 非: 取反
10 print(not False) # True
11 print(not (a>b)) # True
```

### 拓展：数字逻辑运算符

```
1 a = 0
2 b = 1
```

```

3  c = 2
4
5  # and运算符，只要有一个值为0，则结果为0，否则结果为最好一个非0数字
6  print(a and b) # 0
7  print(b and a) # 0
8  print(a and c) # 0
9  print(c and a) # 0
10 print(b and c) # 2
11 print(c and b) # 1
12
13 #or运算符，只有所有值为0结果才为0，否则结果为第一个非0数字
14 print(a or b) # 1
15 print(a or c) # 2
16 print(b or c) # 1

```

## 2.6 运算符总结

- 算符运算的优先级

混合运算优先级顺序： `()` 高于 `**` 高于 `*` `/` `//` `%` 高于 `+` `-`

- 赋值运算符

=

- 复合运算符

+=

-=

优先级

- 1、先算复合赋值运算符右侧的表达式
- 2、再算复合赋值运算的算数运算
- 3、最后算赋值运算

- 比较运算符

判断相等：==

大于等于：>=

小于等于：<=

不等于：!=

- 逻辑运算符

与：and

或：or

非：not

## 三、条件语句

假设一个场景：

- 同学们这个年龄去过网吧吗？
- 去网吧进门想要上网必须做的一件事是做什么？（考虑重点）
- 为什么要把身份证给工作人员？
- 是不是就是为了判断是否成年？

- 是不是如果成年可以上网？如果不成年则不允许上网？

其实这里所谓的判断就是条件语句，即条件成立执行某些代码，条件不成立则不执行这些代码。

## 3.1 语法

```
1 if 条件:
2     条件成立执行的代码1
3     条件成立执行的代码2
4     ...
```

**快速体验：**

```
1 if True:
2     print("条件成立执行的代码1")
3 #注意：在这个下方的没有加缩进的代码，不属于if语句块，即和条件成立与否无关
4 print("这个代码成立嘛？")
```

## 3.2 实例

需求分析：如果用户年龄大于等于18岁即成年，输出已经成年，可以上网。

```
1 age = 20
2 if age >= 18:
3     print("可以上网")
4     print("系统关闭")
```

**进阶版：**新增需求：用户可以输出自己的年龄，然后系统进行判断是否成年，成年则输出您的年龄是用户输入的年龄，已经成年，可以上网。

```
1 #1、用户输入
2 #2、保存用户输入的年龄
3 #3、if
4 age = input("请输入年龄：")
5 age = int(age)
6 if age >= 18:
7     print(f'您输入的年龄是{age},已经成年，可以上网')
8
```

**if...else...**

```
1 #1、用户输入
2 #2、保存用户输入的年龄
3 #3、if
4 age = input("请输入年龄：")
5 age = int(age)
6 if age >= 18:
7     print(f'您输入的年龄是{age},已经成年，可以上网')
8 else:
9     print(f'您输入的年龄是{age},未成年，不可以上网')
```

注意：如果某些条件成立执行了相关的代码，那么其他的情况的代码解释器根本不会执行。

## 3.3 多重判断

思考：中国合法工作年龄为18-60岁，即如果年龄小于18的情况为童工，不合法；如果年龄在18 -60岁之间为合法工龄；大于60岁为法定退休年龄。

```
1  if 条件1:
2      条件1成立执行的代码
3  elif 条件2:
4      条件成立执行的代码
5      ...
6  else:
7      以上条件都不成立执行的代码
8
```

多重判断也可以和else配合使用。一般else放到整个if语句的最后，表示以条件都不成立的时候执行的代码。

```
1  """
2  1、用户输入自己的年龄
3  2、做判断
4  3、输出提示信息：您输入的年龄：，合法与否
5  """
6  age= input('请您输入年龄：')
7  age = int(age)
8  if age<18:
9      print(f'您输入的年龄是{age},童工')
10 elif (age>=18) and (age<=60):
11     print(f'您输入的年龄是{age},合法')
12 elif age>60:
13     print(f'您输入的年龄是{age},退休年龄')
14
```

**if嵌套：**

```
1  if 条件1:
2      条件1成立执行的代码
3      if 条件2:
4          条件2成立执行的代码
```

条件2的if也是出于条件1的缩进关系内部

**实例：**

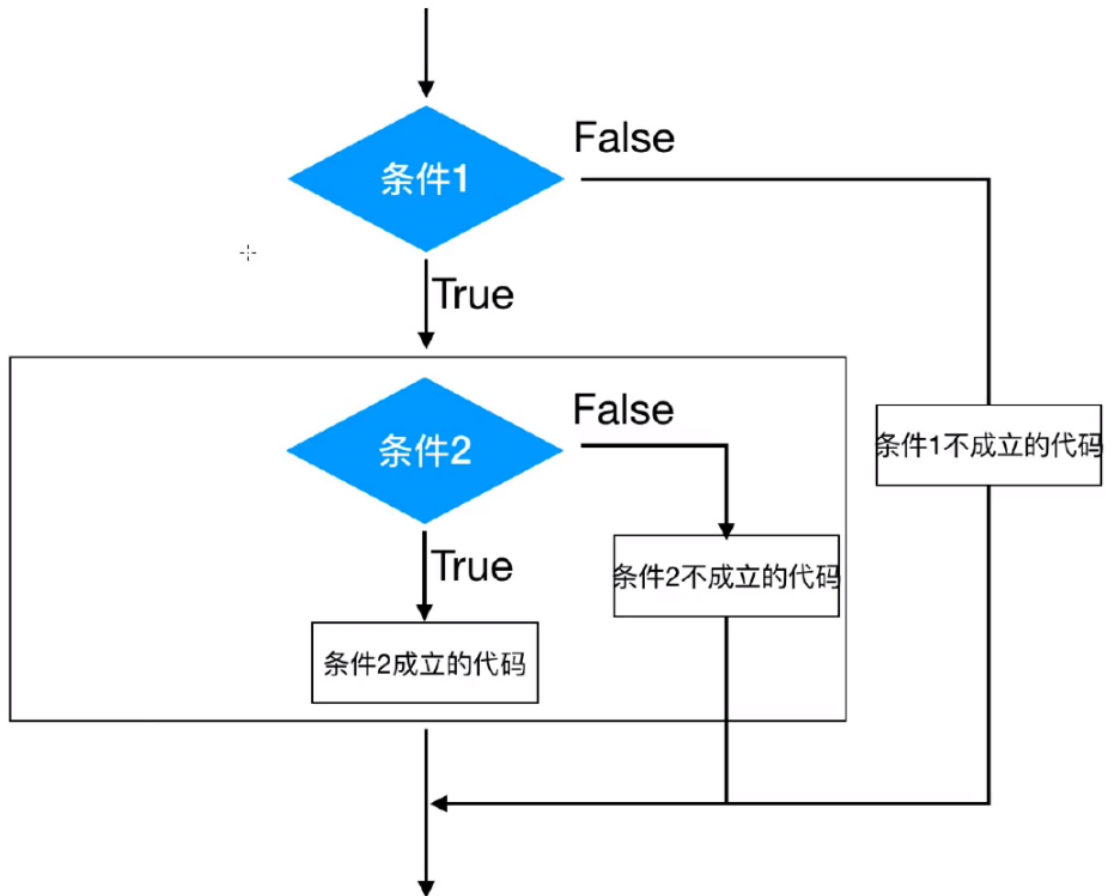
```
1  """
2  1、准备将来要做判断的数据：钱和空座位
3  2、判断是否有钱：上车 和 不能坐上车
4  3、上车了：判断是否能坐下：有空座位 和 无空座位
5  """
6  money = 1
7  seat = 1
```

```

8  if money == 1:
9      print('土豪，请上车')
10     #判断能否坐车
11     if seat ==1:
12         print('有空座，坐下了')
13     else:
14         print('没有空座，站着吧你')
15 else:
16     print('土豪，跑去去')

```

### if嵌套执行流程



## 3.4 综合应用

需求分析：

- 参与游戏的角色
  - 玩家
  - 手动出拳
  - 电脑
  - 随机出拳
- 判断输赢
  - 玩家获胜
  - 平局
  - 电脑获胜

```

1  """
2  1、出拳
3      玩家：手动输入
4      电脑：1.固定 ：剪刀；2. 随机
5
6  2、判断输赢
7      玩家获胜
8      平局
9      电脑获胜
10
11 """
12 import random
13 player = int(input("请出拳：0--石头；1--剪刀；2--布"))
14 #电脑
15 #computer = 1
16 computer = random.randint(0,2)
17 print(computer)
18
19 # 2、判断输赢
20 if ((player==0) and (computer==1) and (player==1) and (computer==2) and
21     (player==2) and (computer==0)):
22     print('玩家赢')
23 elif player == computer:
24     print('平局')
25 else:
26     print('电脑赢')

```

随机数做法：

```

1  import 模块名
2  random.randint(开始,结束)

```

```

1  import random
2  num = random.ranint(0,2)
3  print(num) # 一个随机整数0,1,2
4

```

## 四、三目运算符

1 | 条件成立执行的表达式 if 条件 else 条件不成立执行的表达式

```

1  a = 1
2  b = 2
3  c = a if a>b else b
4  print(c) # 2
5
6  # 需求：有两个变量，比较大小，变量1 大于 变量2 执行 变量1 - 变量2； 否则 变量2 - 变量1
7  aa = 10
8  bb = 6
9  c = aa-bb if aa>bb else bb--aa
10 print(c) # 4
11

```



# 五、循环语法

---