

目标

- 条件语句作用
- if语法
- if...else...
- 多重判断
- if嵌套

一. 了解条件语句

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。

二. if 语法

2.1 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     .....
```

2.2 快速体验

```
1 if True:
2     print('条件成立执行的代码1')
3     print('条件成立执行的代码2')
4
5 # 下方的代码没有缩进到if语句块，所以和if条件无关
6 print('我是无论条件是否成立都要执行的代码')
```

执行结果如下：

```
Run: if x
C:\Users\黑马程序员\AppData\Local\Programs\Python\Python37\python3.exe
条件成立执行的代码1
条件成立执行的代码2
我是无论条件是否成立都要执行的代码
Process finished with exit code 0
```

三. 实例：上网

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"。

3.1 简单版

```
1 age = 20
2 if age >= 18:
3     print('已经成年，可以上网')
4
5 print('系统关闭')
```

3.2 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 # input接受用户输入的数据是字符串类型，条件是age和整型18做判断，所以这里要int转换数据类型
2 age = int(input('请输入您的年龄：'))
3
4 if age >= 18:
5     print(f'您的年龄是{age}，已经成年，可以上网')
6
7
8 print('系统关闭')
```

四. if...else...

作用：条件成立执行if下方的代码；条件不成立执行else下方的代码。

思考：网吧上网的实例，如果成年，允许上网，如果未成年呢？是不是应该回复用户不能上网？

4.1 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      .....
5  else:
6      条件不成立执行的代码1
7      条件不成立执行的代码2
8      .....
```

4.2 实用版：网吧上网

```
1  age = int(input('请输入您的年龄: '))
2
3  if age >= 18:
4      print(f'您的年龄是{age},已经成年,可以上网')
5  else:
6      print(f'您的年龄是{age},未成年,请自行回家写作业')
7
8  print('系统关闭')
```

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。

五、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。

5.1 语法

```
1  if 条件1:
2      条件1成立执行的代码1
3      条件1成立执行的代码2
4      .....
5  elif 条件2:
6      条件2成立执行的代码1
7      条件2成立执行的代码2
8      .....
9  .....
10 else:
11     以上条件都不成立执行的代码
```

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，表示以上条件都不成立的时候执行的代码。

5.2 实例：工龄判断

```
1 age = int(input('请输入您的年龄: '))
2 if age < 18:
3     print(f'您的年龄是{age}, 童工一枚')
4 elif age >= 18 and age <= 60:
5     print(f'您的年龄是{age}, 合法工龄')
6 elif age > 60:
7     print(f'您的年龄是{age}, 可以退休')
```

拓展: `age >= 18 and age <= 60` 可以化简为 `18 <= age <= 60`。

六、if嵌套

思考：坐公交：如果有钱可以上车，没钱不能上车；上车后如果有空座，则可以坐下；如果没空座，就要站着。怎么书写程序？

6.1 语法

```
1 if 条件1:
2     条件1成立执行的代码
3     条件1成立执行的代码
4
5     if 条件2:
6         条件2成立执行的代码
7         条件2成立执行的代码
8
```

注意：条件2的if也是出于条件1的缩进关系内部。

6.2 实例：坐公交

6.2.1 判断是否能上车

```

1  """
2  1. 如果有钱，则可以上车
3      2. 上车后，如果有空座，可以坐下
4      上车后，如果没有空座，则站着等空座位
5  如果没钱，不能上车
6  """
7  # 假设用 money = 1 表示有钱，money = 0表示没有钱
8  money = 1
9  if money == 1:
10     print('土豪，不差钱，顺利上车')
11 else:
12     print('没钱，不能上车，追着公交车跑')

```

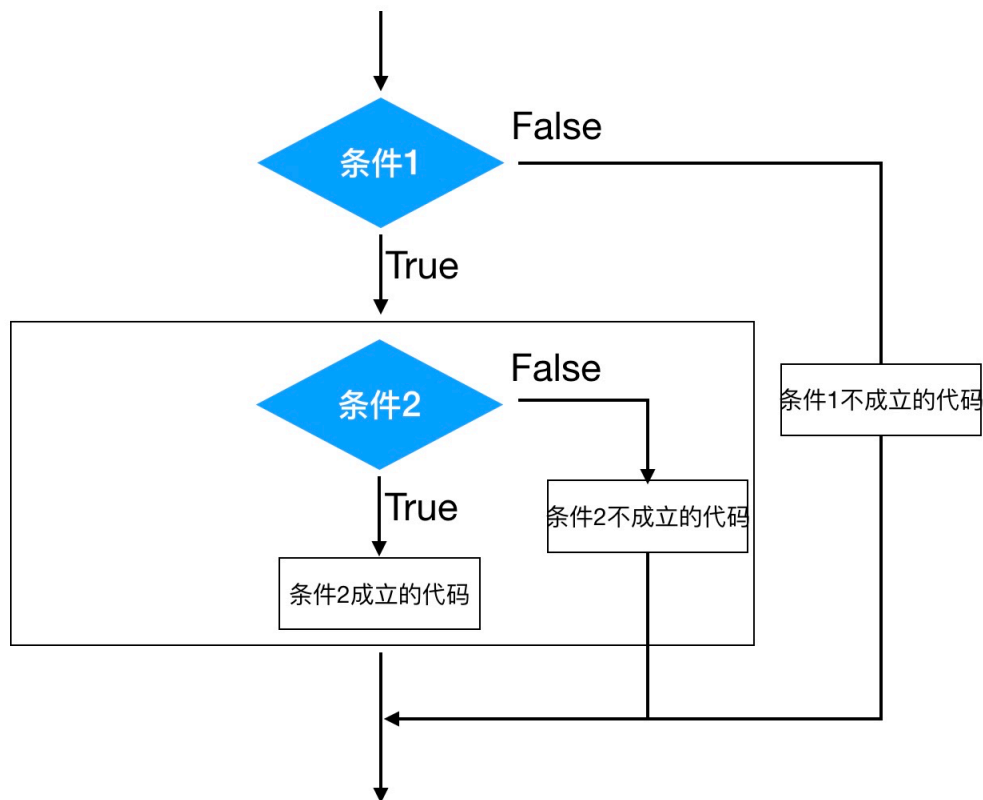
6.2.2 判断是否能坐下

```

1  """
2  1. 如果有钱，则可以上车
3      2. 上车后，如果有空座，可以坐下
4      上车后，如果没有空座，则站着等空座位
5  如果没钱，不能上车
6  """
7  # 假设用 money = 1 表示有钱，money = 0表示没有钱；seat = 1 表示有空座，seat = 0 表示
   没有空座
8  money = 1
9  seat = 0
10 if money == 1:
11     print('土豪，不差钱，顺利上车')
12     if seat == 1:
13         print('有空座，可以坐下')
14     else:
15         print('没有空座，站等')
16 else:
17     print('没钱，不能上车，追着公交车跑')

```

6.3 if嵌套执行流程



七. 应用：猜拳游戏

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

玩家	电脑
石头	剪刀
剪刀	布
布	石头

- 平局
 - 玩家出拳 和 电脑出拳相同
- 电脑获胜

随机做法：

1. 导出random模块

```
1 | import 模块名
```

2. 使用random模块中的随机整数功能

```
1 | random.randint(开始, 结束)
```

```
1  """
2  提示: 0-石头, 1-剪刀, 2-布
3  1. 出拳
4  玩家输入出拳
5  电脑随机出拳
6
7  2. 判断输赢
8  玩家获胜
9  平局
10 电脑获胜
11 """
12
13 # 导入random模块
14 import random
15
16 # 计算电脑出拳的随机数字
17 computer = random.randint(0, 2)
18 print(computer)
19
20 player = int(input('请出拳: 0-石头, 1-剪刀, 2-布: '))
21
22 # 玩家胜利 p0:c1 或 p1:c2 或 p2:c0
23 if ((player == 0) and (computer == 1) or ((player == 1) and (computer == 2) or
24     ((player == 2) and (computer == 0))):
25     print('玩家获胜')
26
27 # 平局: 玩家 == 电脑
28 elif player == computer:
29     print('平局')
30 else:
31     print('电脑获胜')
```

八. 三目运算符

三目运算符也叫三元运算符或三元表达式。

语法如下:

- 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

快速体验：

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

总结

- `if`语句语法

```
1 if 条件:
2     条件成立执行的代码
```

- `if...else...`

```
1 if 条件:
2     条件成立执行的代码
3 else:
4     条件不成立执行的代码
```

- 多重判断

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

- `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
5         ....
```

- 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