

# day02

## day02

- 一、输入输出
- 二、判断语句 (if...elif...else)
  - 综合案例——石头剪刀布
- 三、循环 (while)
  - 翻译插件安装
  - 快捷键
- 四、列表 (list)
- 五、for循环

## 一、输入输出

## 1. 普通输出 `print()`

```
1 a = 10
2 b = 20
3 print(a, b)
4 # 结果:
5 10 20
```

## 2. 格式化输出

```
1 name = "UR机器人"
2 a = 10
3 # 不同数据类型不能直接相加
4 print("机器人是: %s载重: %d" % (name, a))
5 # 结果:
6 机器人是: UR机器人载重: 10
7 ""=====2=====
8 name = "UR机器人"
9 a = 10
10 print("机器人名字: %s" % name)
11 # 结果:
12 机器人名字: UR机器人
```

### 3. 练习

```
1  """----- 练习题一 -----"""
2  """
3  1. 定义字符串变量 name，输出 我的名字叫 小明，请多多关照！
4  2. 定义整数变量 student_no，输出 我的学号是 000001
5  3. 定义小数 price、weight、money，输出 苹果单价 9.00 元 / 斤，购买了 5.00 斤，需
   要支付 45.00 元
6  4. 定义一个小数 scale，输出 数据比例是 10.00%
```

```

7  """
8  name = "小明"
9  student_no = 1
10 price = 9.00
11 weight = 5.00
12 money = price * weight
13 scale = 10.00
14 print("我的名字叫 %s" % name)
15 print("我的学号是 %06d" % student_no)
16 print("苹果单价 %.2f 元 / 斤, 购买了 %.2f 斤, 需要支付 %.2f 元" % (price,
    weight, price * weight))
17 print("数据比例是 %.2f%" % scale)
18 # 结果:
19 我的名字叫 小明
20 我的学号是 000001
21 苹果单价 9.00 元 / 斤, 购买了 5.00 斤, 需要支付 45.00 元
22 数据比例是 10.00%

```

#### 4. 占位输出 {}

```

1  name = "UR机器人"
2  load = 5
3  print("机器人是: {}负载是: {}".format(name, load))
4  # 结果:
5  机器人是: UR机器人负载是: 5

```

#### 5. 输入函数 input()

任何数据输入都会是一个字符串, 想要将数据按次别的类型就需要进行转换

```

1  a = int(input("用户输入a:"))
2  b = int(input("用户输入b:"))
3  print("a+b={}".format(a+b))
4  # 结果:
5  用户输入a:10
6  用户输入b:20
7  a+b=30

```

#### 6. 练习:

```

1  """----- 练习2 -----"""
2  # 超市买苹果计算金额
3  """
4  收银员输入苹果的价格,单位:元/斤
5  收银员输入用户购买苹果的重量,单位:斤
6  计算并输出付款金额
7  """
8  # price = int(input("收银员输入苹果的价格,单位:元/斤"))
9  # weight = int(input("收银员输入用户购买苹果的重量,单位:斤"))
10 # print("应付: {}元".format(price * weight))
11 # 个人名片
12 """
13 在控制台依次提示用户输入: 姓名、公司、职位、电话、邮箱

```

```

14  """
15  name = input("请输入姓名: ")
16  company = input("请输入公司名字: ")
17  manager = input("请输入职位: ")
18  number = input("请输入电话号码: ")
19  mail = input("请输入邮箱: ")
20
21  # print("*****")
22  print("*"*50)
23  print("公司名称")
24  print(company)
25  print("姓名:" + name + "职位:" + manager)
26  print("")
27  print("电话:" + number)
28  print("邮箱:" + mail)
29  # print("*****")
30  print("*"*50)
31  # 结果:
32  收银员输入苹果的价格,单位:元/斤10
33  收银员输入用户购买苹果的重量,单位:斤10
34  应付: 100元
35  请输入姓名: 刘小刚
36  请输入公司名字: 电子协会
37  请输入职位: 划水队大队长
38  请输入电话号码: 1534535634
39  请输入邮箱: 153435434@qq.com
40  *****
41  公司名称
42  电子协会
43  姓名:刘小刚职位:划水队大队长
44
45  电话:1534535634
46  邮箱:153435434@qq.com
47  *****

```

## 7. 常用运算符

- 赋值运算符

运算符	描述
<code>=</code>	赋值
<code>+=</code>	加等于( <code>a += 10 == a = a + 10</code> )
<code>-=</code>	减等于( <code>a -= 10 == a = a - 10</code> )
<code>*=</code>	乘等于( <code>a *= 10 == a = a * 10</code> )
<code>/=</code>	除等于( <code>a /= 10 == a = a / 10</code> )
<code>//=</code>	取整等于( <code>a //= 10 == a = a // 10</code> )
<code>%=</code>	取余等于( <code>a %= 10 == a = a % 10</code> )
<code>**=</code>	幂等于( <code>a **= 10 == a = a ** 10</code> )

- 比较运算符

运算符	描述
<code>==</code>	等于
<code>!=</code>	不等于
<code>&gt;</code>	大于
<code>&lt;</code>	小于
<code>&gt;=</code>	大于等于
<code>&lt;=</code>	小于等于

- 逻辑运算符：主要用于逻辑运算

运算符	描述
<code>and</code>	逻辑与
<code>or</code>	逻辑或
<code>not</code>	逻辑非

## 二、判断语句 (if...elif...else)

### 1. `if` 语句格式

if 条件:  
    条件成立时, 要做的事情  
    .....

## 2. 练习:

```
1 # 输入你的年龄
2 age = int(input("请输入你的年龄: "))
3 # 判断是否满足岁
4 if age >= 18:
5     print("允许进网吧")
6 # 结果:
7 请输入你的年龄: 20
8 允许进网吧
```

## 3. if...else 语句格式

if 要判断的条件:  
    条件成立时, 要做的事情  
    .....  
else:  
    条件不成立时, 要做的事情  
    .....

## 4. 练习:

```
1 # 输入你的年龄
2 age = int(input("请输入你的年龄: "))
3 # 判断是否满足岁
4 if age >= 18:
5     print("允许进网吧")
6 else:
7     print("回家写作业")
8 # 结果:
9 请输入你的年龄: 12
10 回家写作业
```

## 5. if 的条件语句中可以加逻辑运算符 and or not

## 6. 练习:

```
1 # 1. 练习1: 定义一个整数变量age, 编写代码判断年龄是否正确
2 #     要求人的年龄在 0-120 之间
3 age = int(input("输入年龄: "))
4 if age > 0 and age < 120:
5     print("年龄合法")
6 else:
7     print("年龄不合法")
8
9 # 2. 练习2: 定义两个整数变量python_score、c_score, 编写代码判断成绩
10 #     要求只要有一门成绩 > 60 分就算合格
11 python_score = int(input("输入python成绩: "))
```

```

12 c_score = int(input("输入C语言成绩:"))
13 if python_score > 60 or c_score > 60:
14     print("合格")
15 else:
16     print("不合格")
17 # 结果:
18 输入年龄: 20
19 年龄合法
20 输入python成绩:60
21 输入C语言成绩:80
22 合格

```

## 7. elif 格式

```

if 条件1:
    条件1满足执行的代码
    .....
elif 条件2:
    条件2满足时，执行的代码
    .....
elif 条件3:
    条件3满足时，执行的代码
    .....
else:
    以上条件都不满足时，执行的代码
    .....

```

## 8. 练习:

```

1 # 1. 定义 holiday_name 字符串变量记录节日名称
2 holiday_name = input("输入节日: ")
3 # 2. 如果是 情人节 应该 买玫瑰 / 看电影
4 if holiday_name == "情人节":
5     print("买玫瑰 / 看电影")
6 # 3. 如果是 平安夜 应该 买苹果 / 吃大餐
7 elif holiday_name == "平安夜":
8     print("买苹果 / 吃大餐")
9 # 4. 如果是 生日 应该 买蛋糕
10 elif holiday_name == "生日":
11     print("买蛋糕")
12 # 5. 其他的日子每天都是节日啊.....
13 else:
14     print("其他的日子每天都是节日啊.....")
15 # 结果:
16 输入节日: 啊哈
17 其他的日子每天都是节日啊.....

```

## 9. if 的嵌套

- if 条件 1:
- 条件 1 满足执行的代码
- .....
- if 条件 1 基础上的条件 2:
- 条件 2 满足时，执行的代码
- .....
- # 条件 2 不满足的处理
- else:
- 条件 2 不满足时，执行的代码
- .....
- # 条件 1 不满足的处理
- else:
- 条件1 不满足时，执行的代码
- .....

## 10. 练习:

```
1  # 1. 定义布尔型变量 has_ticket 表示是否有车票
2  has_ticket = bool(input("输入是否有票（0表示没有，1表示有票）:"))
3  # 2. 定义整型变量 knife_length 表示刀的长度，单位：厘米
4  knife_length = int(input("输入刀的长度:"))
5  # 3. 首先检查是否有车票，如果有，才允许进行 安检
6  if has_ticket:
7      # 4. 安检时，需要检查刀的长度，判断是否超过 20 厘米
8      # 如果超过 20 厘米，提示刀的长度，不允许上车
9      if knife_length > 20:
10         print("不允许上车")
11     # 如果不超过 20 厘米，安检通过
12     else:
13         print("允许上车")
14 # 5. 如果没有车票，不允许进门
15 else:
16     print("不允许上车")
17 # 结果:
18 输入是否有票（0表示没有，1表示有票）:1
19 输入刀的长度:12
20 允许上车
```

## 综合案例——石头剪刀布

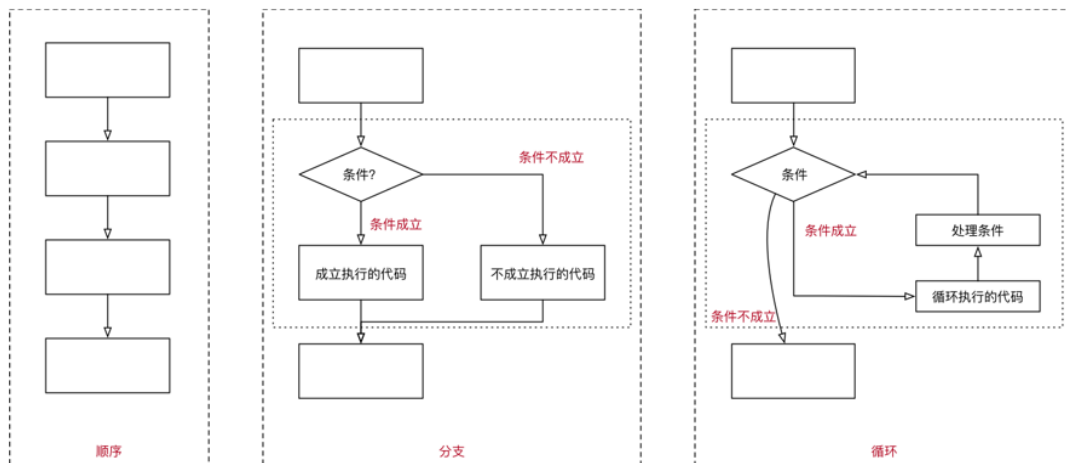
导入随机数的包random

参数的提示快捷键Ctrl+P

```
1 import random
2
3 computer_i = random.randint(1, 3)
4 computer_dict = {1: "石头", 2: "剪刀", 3: "布"}
5 user = input("用户请出拳")
6 if computer_dict[computer_i] == user:
7     print("平局")
8 else:
9     if (user == "石头" and computer_dict[computer_i] == "剪刀") or
10        (user == "剪刀" and computer_dict[computer_i] == "布") or
11        (user == "布" and computer_dict[computer_i] == "石头"):
12         print("玩家获胜")
13     else:
14         print("电脑获胜")
```

## 三、循环 (while)

### 1. 程序的三种执行顺序



### 2. 循环的作用：重复的执行

### 3. while 循环的基本语法



## 变量设置初始值

**while** 条件(判断 变量是否满足条件):

    条件满足时, 做的事情1

    条件满足时, 做的事情2

    条件满足时, 做的事情3

    ...(省略)...

修改变量

### 4. 循环变量和死循环

- 循环变量主要作用是控制循环什么时候停止下来
- 死循环的作用是保证程序持续执行,不会终止

#### 快速创建死循环

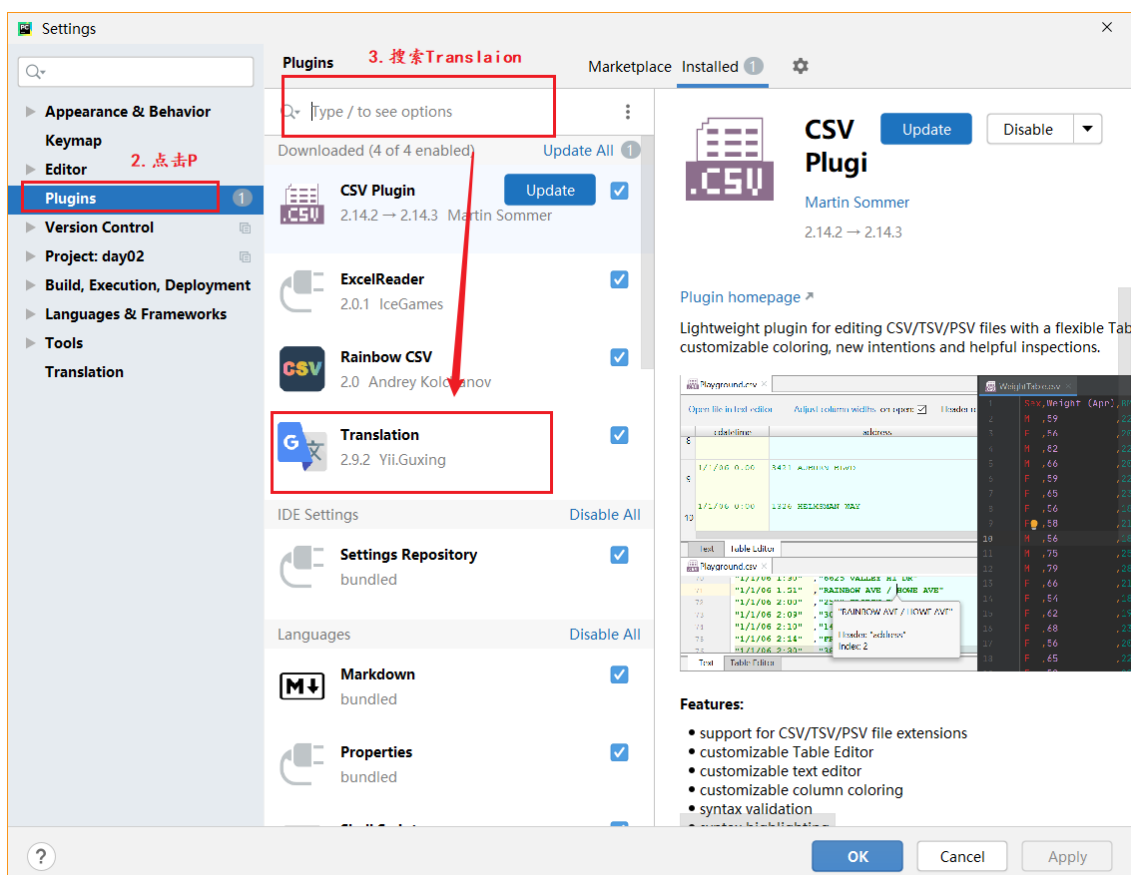
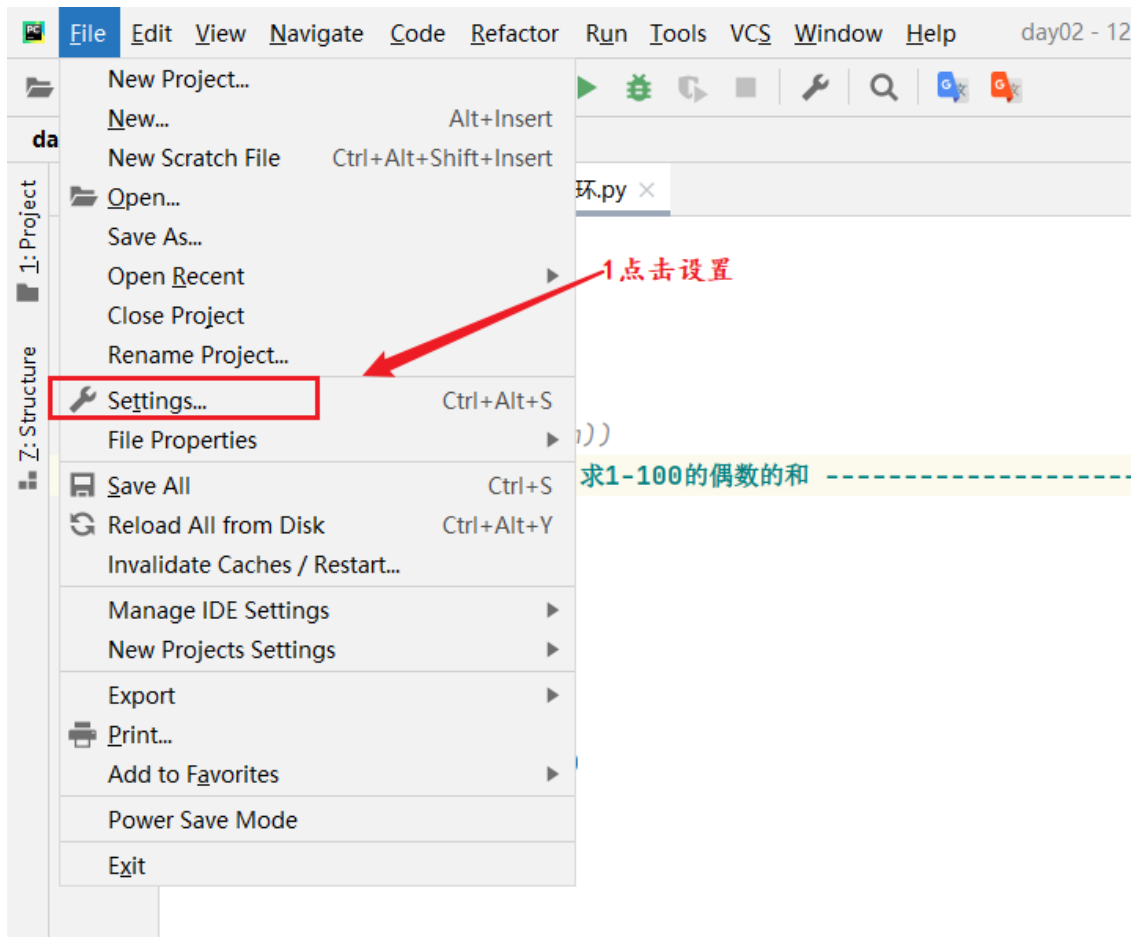
```
1 while True:
2     pass
```

**pass** 占位

### 5. 求和练习

```
1  """----- 求1-100的和 -----"""
2  # 循环变量
3  i = 1
4  # 求和变量
5  Sum = 0
6  # 循环
7  while i <= 100:
8      Sum += i
9      i += 1
10 print("和为{}".format(Sum))
11 # 结果:
12 和为5050
13 """----- 求1-100的偶数的和 -----"""
14 i = 1
15 Sum = 0
16 while i <= 100:
17     if i % 2 == 0:
18         Sum += i
19     i += 1
20 print("和为{}".format(Sum))
21 # 结果:
22 和为2550
```

## 翻译插件安装



## 6. 回文数练习

```

3  Palindrome = 100
4  print("回文数是", end=":")
5  while Palindrome < 1000:
6      if Palindrome // 100 == Palindrome % 10:
7          print("{}".format(Palindrome), end=",")
8      Palindrome += 1
9  # 结果:
10 回文数是:101,111,121,131,141,...
11  """----- 打印三角形 -----"""
12  i = 1
13  while i < 4:
14      print("*"*i)
15      i += 1
16  # 结果:
17  *
18  **
19  ***
20  """----- 打印倒三角形 -----"""
21  i = 3
22  while i > 0:
23      print("*"*i)
24      i -= 1
25  # 结果:
26  ***
27  **
28  *
29  """----- 打印倒三角形 -----"""
30  i = 0
31  j = 3
32  while i < j:
33      print(" "*i,end= "")
34      print("*"*(j-i))
35      i += 1
36  ***
37  **
38  ■

```

## 7. 循环嵌套格式

```
while 条件 1:
    条件满足时, 做的事情1
    条件满足时, 做的事情2
    条件满足时, 做的事情3
    ... (省略) ...

while 条件 2:
    条件满足时, 做的事情1
    条件满足时, 做的事情2
    条件满足时, 做的事情3
    ... (省略) ...

    处理条件 2

处理条件 1
```

## 8. 九九乘法表

```
1  """----- 九九乘法表 -----"""
2  i = 1
3  while i <= 9:
4      j = 1
5      while j <= i:
6          print("{}*{}={}".format(j,i,i*j),end="\t")
7          j += 1
8      print()
9      i += 1
10 # 结果:
11 1*1=1
12 1*2=2  2*2=4
13 .....
```

## 9. 转义字符

转义字符	描述
\	反斜杠
"	单引号
""	双引号
\n	换行
\t	制表符
\r	回车

快捷键

- 1. `Ctrl+R`：选中变量，快速替换



四、列表 (list)

- 1. 列表的定义

列表用 `[]` 定义、元素之间用 `,` 分割

列表中可以有不同的数据类型

- 2. 访问列表

- 查找元素只按照索引进行查找的
- 索引：元素在列表中位置的编号，有被称为下标、索引值是从0开始的

- 3. 列表的操作：增、删、改、查

- 增加

方法	说明
<code>append(数据)</code>	在末尾追加数据
<code>insert(索引, 数据)</code>	在指定位置插入数据（位置前有空位置，会补位）

1

- 删除

方法	说明
<code>del 列表[索引]</code>	删除指定索引的数据
<code>remove(数据)</code>	删除第一个出现的指定数据
<code>clear()</code>	清空列表

1

- 修改

方法	说明
<code>列表[索引] = 数据</code>	修改指定索引的数据，数据不存在会报错

1

- 查找

方法	说明
<code>index(数据)</code>	根据值查询索引, 返回 <b>首次</b> 出现时的索引, 没有查到会报错

1

## 五、for循环

1. `in` 和 `not in`

`in` 操作符用于判断元素是否存在于容器中, 存在, 返回 `true`, 否则返回 `false`

2. for格式

**for** 临时变量 **in** 列表或者字符串等可迭代对象:  
循环满足条件时执行的代码

```
1 nume = ['螺丝刀1', '螺丝刀2', '螺丝刀3']
2 for ele in nume:
3     print(ele)
4 # 结果:
5 螺丝刀1
6 螺丝刀2
7 螺丝刀3
```