

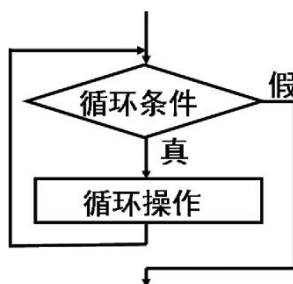
### 3. 循环结构

Python 中的循环结构主要分成 **while** 循环和 **for** 循环两种。循环结构可以简化重复性计算任务，简化程序编码。

循环语句的主要作用是反复执行某一段代码，直到满足一定的条件位置。下面我们分别来了解 **while** 循环和 **for** 循环。

#### (1) while 循环

先了解一下 while 循环的流程图，如下。



#### 【语法】

变量初始化

**while** 循环条件:

    循环体

#### 【语法解释】

- ✓ 关键字 **while** 后的内容是循环条件
- ✓ 循环条件是布尔类型，其结果为 **True** 或 **False**
- ✓ 循环条件后面紧跟冒号 (**:**)

#### 【执行步骤】

1. 先初始化：设置循环的初识状态
2. 再判断条件：当条件满足，进入循环体
3. 第三步：执行循环体
4. 第四步：再次判断条件，反复执行循环体，直至条件为 **False**

#### 小贴士

- **while** 循环，是先判断条件，在执行循环体。如果第一次判断条件就为 **False**，那么，循环一次也不会执行。

示例：计算  $1 \sim 10$ ，所有奇数的和。

```
sum = 0#初始化和
i = 1#初始化变量
while i<=10:
    #判断是否是奇数
    if i%2 != 0:
        sum = sum+i
    i = i+1
print('1~10 所有的奇数和为: ', sum)
```

运行结果如下：



```
Test x
D:\23Python\Study\venv\Scripts\python.exe D:/23P
1~10所有的奇数和为: 25

Process finished with exit code 0
```

示例：100 内奇数均值

需求：求  $1 \sim 100$  中所有奇数的平均值。

提示：正确答案应该为 50。

（点击后，出现如下代码框）

```
sum = 0#初始化和
avg = 0#平均值
count = 0#计数
i = 1#初始化变量
while i<=100:
    #判断是否是奇数
    if i%2 != 0:
        sum = sum+i
        count = count+1
    i = i+1
avg =int(sum/count)
print('1~100 所有的奇数平均值: ', avg)
```

`while` 循环还有一种形式，就是“`while……else`”，在条件语句为 `false` 时执行 `else` 的语句块：

**【语法】**

变量初始化

`while` 循环条件：

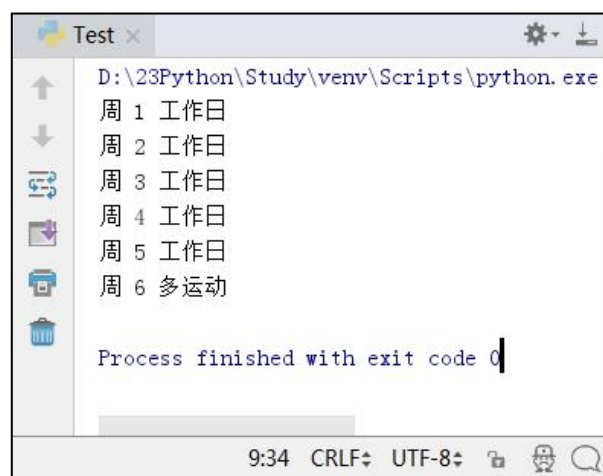
    循环体

`else`

    代码块

如下示例，输出结果为：

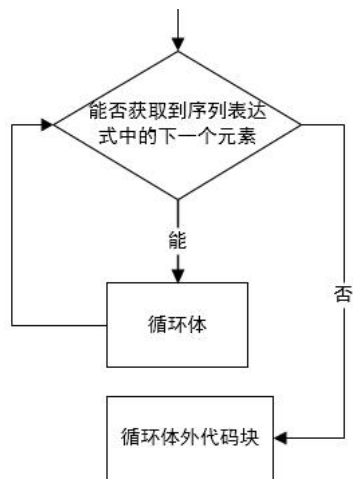
```
count = 1
while count <= 5:
    print ('周', count, "工作日")
    count = count + 1
else:
    print ('周', count, "多运动")
```



## (2) for 循环

Python 中 `for` 循环用来遍历数据集合或迭代器中的元素，如一个列表或者一个字符串。

如下为 `for` 循环的流程图。



### 【语法】

for 循环变量 in 序列表达式:

    循环体

### 【语法解释】

- ✓ for 循环时，序列表达式中的元素会依次赋值给循环变量
- ✓ 在循环体中操作循环变量实现遍历序列表达式

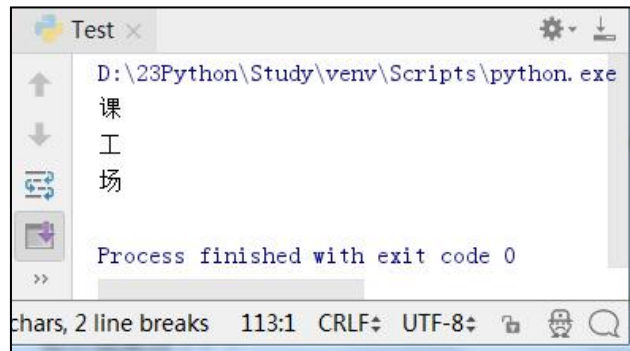
### 【执行步骤】

1. 从序列表达式中获取第一个元素，赋值给循环变量
2. 执行循环体代码
3. 从序列表达式中获取下一个元素，赋值给循环变量
4. 重复执行第二、三步骤
5. 直至无法从序列表达式中获取元素，结束循环

**示例：**实现对字符串“dog”的遍历，输出每一个字符。

```
str = '课工场'
for c in str:
    print(c)
```

运行结果如下：



### 示例：100 内求和

**需求：**利用 `for` 循环求 1~100 中所有数的和。

**提示：**正确答案应该为 5050。

**技术补充：**使用 `for` 循环遍历一个自增的序列，需要结合 `range()` 函数来实现。该函数能够快速构造一个等差序列。`range(start,stop)` 会生成一个左闭右开的数值序列，序列中相邻的两个整数相差 1。例如 `range(1,5)`，生成的序列为：1,2,3,4

查看参考代码：

```
sum = 0#初始化和  
for i in range(0,101):  
    sum = sum+i
```

除了 `for` 循环，再试试使用 `while` 循环实现一样的效果，并对比两段代码。

### (3) 嵌套循环

多重循环，也称为嵌套循环，就是一个循环套着另一个循环，使用多重循环时，要特别注意各个值的变化，仔细思考。例如，如下为一种多重循环的语法样式：

`while` 循环条件 1:

    循环体 1

        for 循环变量 in 序列表达式:

            循环体 2

【语法解释】

- ✓ `while` 循环称为外层循环，`for` 循环称为内层循环
- ✓ 只有两层循环，故称为二重循环

### 【执行过程】

外层 **while** 每循环一次，内层 **for** 循环就从头到尾完整执行一遍。

下面来看一个嵌套循环的示例视频，加深理解嵌套循环的使用。