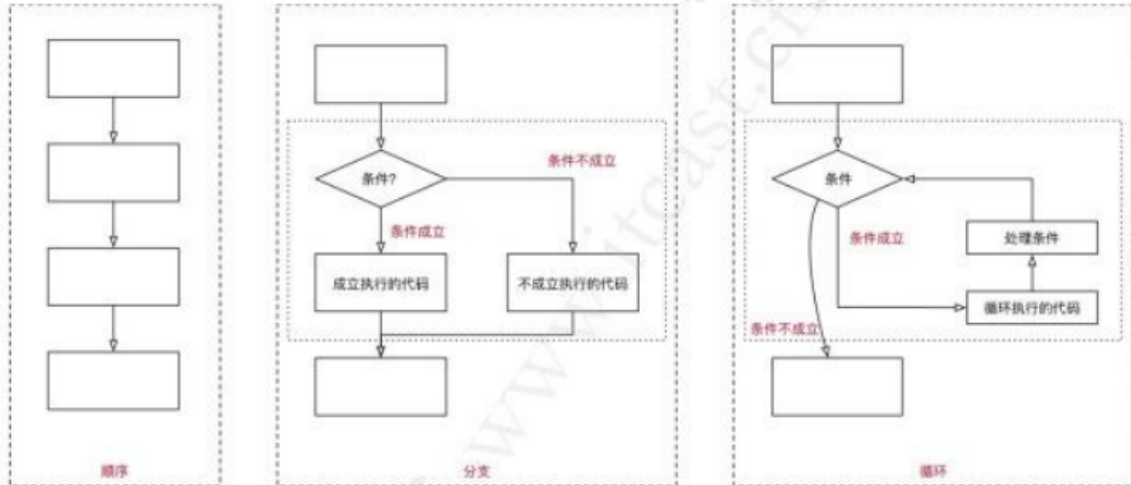


# 程序的三种结构

- 顺序,代码自上而下执行
- 分支,根据条件有选择的执行特定代码
- 循环,根据条件循环执行特定代码



```
1 # print("hello world")
2 # print("hello world")
3 # print("hello world")
4 a = 0
5 while a < 3:
6     print("hello world")
7     a += 1
```

The screenshot shows a Python IDE with a file named "01-第一个while.py". The code is as follows:

```
1 a = 0
2 while a < 3:
3     print("hello world")
4     a += 1
```

Red arrows and text annotations explain the execution flow:

- Line 1:** "定义了一个变量a,值为0" (Defined a variable a, value is 0).
- Line 2:** "条件成立,所以while可以循环" (Condition is true, so while can loop).
- Line 3:** "开始第二次循环,a的值为1,条件成立,继续循环" (Start second loop, a's value is 1, condition is true, continue loop).
- Line 4:** "执行完这句话后a的值为1" (After executing this sentence, a's value is 1).
- Line 5:** "a的值为2" (a's value is 2).
- Line 6:** "开始第三次循环,a值为2,条件成立,继续循环" (Start third loop, a's value is 2, condition is true, continue loop).
- Line 7:** "a的值为3" (a's value is 3).
- Line 8:** "开始第四次循环,a的值为3,条件不成立,循环终止" (Start fourth loop, a's value is 3, condition is false, loop ends).

The output window shows the result of the program:

```
"C:\Program Files\Python36\python.exe" C:/file/teach/05python/day03/3-code/01-第一个while.py
hello world
hello world
hello world
```

如果要输出的是100个hello world,该怎么做?

```
1 a = 0
2 while a < 100:
3     print("hello world")
4     a += 1
```

# while基本语法

- 语法

```
1 while 条件:
2     要循环执行的代码
3 # 如果条件成立,那么循环执行,如果条件不成立,那么循环终止
```

## 死循环

- 程序应该避免出现死循环

```
1 a = 0
2 while a < 3:
3     print("hello world")
4 # a的值永远不会变,所以循环条件永远成立
```

## 循环的计数习惯

- 人类计数习惯从1开始
- 程序计数习惯从0开始,除非有明确的要求,可以不从0开始,否则一定要从0开始计数

- 循环-输出连续数字-----从 0 开始，截止到 5 的 6 个数字

```
1 a = 0
2 while a < 6:
3     print(a)
4     a += 1
5
```

```
1 a = 0
2 while a < 6:
3     print(a)
4     a += 1
```

第一次a的值为0,条件成立,循环开始

第二次,a的值为1,条件成立,继续循环

输出0

a的值为1

输出1

第七次,a的值为6,条件不成立,循环终止

a的值为2

到后来,a的值为6

03-输出从0开始的6个数字

"C:\Program Files\Python36\python.exe" C:/file/teach/05python/day03/3-code/03-输出从0开始的6个数字.py

0

1

2

3

4

5

print和a+=1都执行了6次

while后面条件判断了7次,最后一次判断条件,条件不成立,所以循环终止了

## 课堂练习

### 1. 课堂练习---输出连续数字---从 5 开始, 截止到 0 的 6 个数字

```
1 a = 5
2 while a >= 0:
3     print(a)
4     a -= 1
```

a的值为5,条件成立,循环开始

输出5

a的值为4,条件成立,循环继续

输出4

a的值为3

a的值为-1,条件不成立,循环终止

a的值为-1

04-课堂练习-循环输出从5到0

"C:\Program Files\Python36\python.exe" C:/file/teach/05python/day03/3-code/04-课堂练习-循环输出从5到0.py

5

4

3

2

1

0

## 循环计数

- 循环在程序中经常用到的功能是循环计数
- 先在循环开始之前定义一个变量,用来存放计算结果

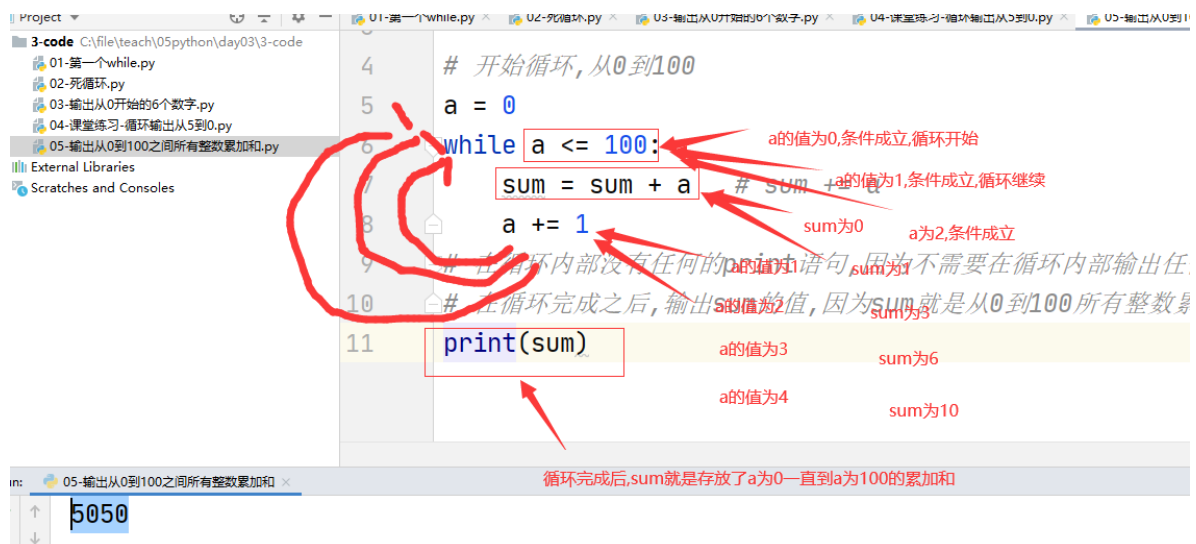
- 在循环内部对这个变量进行计算
- 循环完成后,输出这个变量的值

## 1. 需求----计算 0 ~ 100 之间所有数字的累计求和结果

```

1  # 定义一个变量,来存放累加的和
2  sum = 0
3
4  # 开始循环,从0到100
5  a = 0
6  while a <= 100:
7      sum = sum + a    # sum += a
8      a += 1
9  # 在循环内部没有任何的print语句,因为不需要在循环内部输出任何数字
10 # 在循环完成之后,输出sum的值,因为sum就是从0到100所有整数累加的和
11 print(sum)

```



## 循环计算从300到415之间所有数据累加和

```

1  sum = 0
2
3  a = 300
4  while a <= 415:
5      sum += a
6      a += 1
7
8  print(sum)

```

## break终止循环

- break必须放到循环内部
- 如果循环遇到break,立刻终止
- 一般在循环内部通过if的条件,来执行break,某一个条件成立的时候来break

```
1 a = 0
2 while a < 10:
3     print(a)
4     if a == 3:
5         break # break执行,循环不在执行了 # break由于是在if语句里面,所以要缩进两次
6     a += 1
7 # break之后,代码到这里了
```

break 退出循环: 在 while 循环内部, 通过 input 接收用户输入, 当用户输入 exit, while 循环终止。

```
1 a = 0
2 while a < 100000000:
3     str1 = input("请输入一个字符串")
4     if str1 == "exit":
5         break
6     print(str1)
7     a += 1
```

## 人造死循环

- 代码中循环开始的时候不知道循环次数,可以使用人造死循环
- 在人造死循环内部,一定要有if结合break来退出循环的机制

```
1 # 如果while后面的条件为True,这个while就是一个人造的循环
2 # 循环内部一定要有if结合break来让循环退出的机制
3 while True:
4     str1 = input("请输入一个字符串")
5     if str1 == "exit":
6         break
7     print(str1)
```

## continue

- 循环一旦遇到continue就不在执行continue下面的循环代码了,而是直接返回到循环开始判断条件
- continue一定是放到if后面,

```

1 a = 0
2 while a < 5:
3     a += 1
4     if a == 3:
5         continue
6     print(a)

```

- 使用continue一定要注意,因为使用不当很容易成死循环

```

1 a = 0
2 while a < 5:
3     if a == 3:
4         continue # a等于3的时候,就再也没机会执行a += 1这个代码了
5     print(a)
6     a += 1

```

## 1. 课堂练习---

通过 `input` 输入任意一个整数

如果输入的是 5, 那么就打印 5 行\*号, 如果输入 10, 就打印 10 行\*号

如果输入大于 20, 最多只打印 20 行\*号

```

1 # a还是一个循环计数器
2 # b决定了循环的最大次数
3 a = 0
4 b = int(input("请输入b的值"))
5
6 while a < b:
7     print("*")
8     a += 1
9     if a >= 20: # 不管b的值多大,一旦a到了20,循环就退出
10        break

```

- 再讲一次

```
# 如果输出3行星号
a = 0
b = 3
while a < b:
    print("*")
    a += 1
```

a是0, b是3,条件成立,循环执行

a是1,b是3,条件成立

a的值为1

```
# 如未b的值不是代码写死的,是通过input输入的,是超过input输入的
a = 0
b = int(input("请输入b的值")) # 假设用户输入了数字3
while a < b:
    print("*")
    a += 1
```

a的值0,b的值3,条件成立

a的值为1,b的值3

a的值为1

python.exe" C:/file/teach/05python/day03/3-code/12-课堂练习-输出

```
1 # 如果输出3行星号
2 # a = 0
3 # b = 3
4 # while a < b:
5 #     print("")
6 #     a += 1
7 # 如果b的值不是代码写死的,是通过input输入的
8 a = 0
9 b = int(input("请输入b的值")) # 假设用户输入了数字10
10 while a < b:
11     print("")
12     a += 1 # 在循环内部,a的值一直在变化,但b的值从来不变
13     if a >= 20:
14         break
```

## 循环嵌套

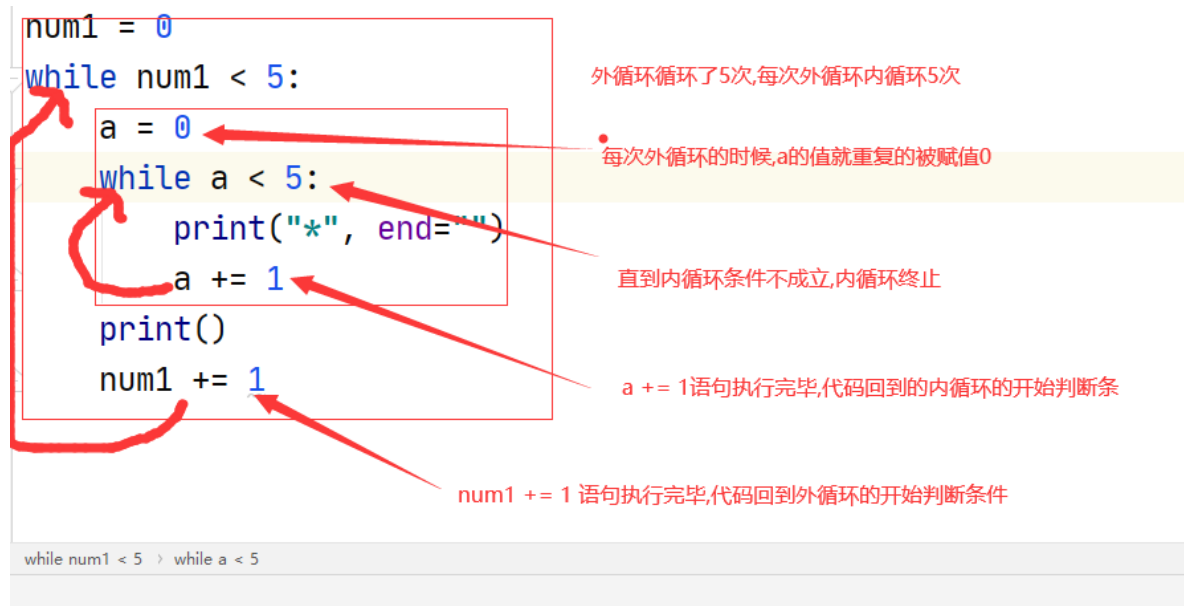
- 一个循环语句里面还有其他的循环语句
- 语法

```
1 while 条件:
2     while 条件:
3         代码
4
```

```

1  # ***
2  # ***
3
4  # 一个print只输出一个星号，如何输出连续的三个星号
5  num1 = 0
6  while num1 < 2: # 外循环一共循环2次，每次外循环的时候，内循环循环3次
7      a = 0
8      while a < 3:
9          print("*", end="")
10         a += 1
11     print() # 输出一个空的回车换行
12     num1 += 1
13 print("end")
14
15

```



36\python.exe" C:/file/teach/05python/day03/3-code/14-循环嵌套输出

## 循环嵌套输出三角形

```

1  # 一个print只能输出一个星号
2  # *
3  # **
4  # ***
5  num1 = 0
6  while num1 < 3:
7      a = 0
8      while a <= num1:
9          print("*", end="")
10         a += 1
11     print()
12     num1 += 1

```



- 输出以下结果：

```
*  
**  
***  
****  
*****
```

```
1  
2 num1 = 0  
3 while num1 < 5:  
4     a = 0  
5     while a <= num1:  
6         print("*", end="")  
7         a += 1  
8     print()  
9     num1 += 1
```

## 输出数字三角形

### 3. 课堂练习---

输出以下结果：

```
1  
12  
123  
1234  
12345
```

```
1  
2 num1 = 0  
3 while num1 < 5:  
4     a = 0  
5     while a <= num1:  
6         print(a + 1, end="")  
7         a += 1  
8     print()  
9     num1 += 1
```

# for循环

- for主要是用于遍历字符串
- 字符串中有几个字符,for就会循环几次
- 语法

```
1  for 变量名 in 字符串:
2      循环代码
```

```
1  # for主要用于遍历,字符串中有几个字符,for就会循环几次
2  # 这个for循环了5次,因为str1这个字符串中有5个字符
3  str1 = "hello"
4  for n in str1:    # n是自己定义的一个变量
5      print("*")
6      print(n)
7  # 第一次循环的时候n的值是h
8  # 第二次循环的时候n的值是e
9  # 第三次循环的时候n的值是l
10 # 第四次循环的时候n的值是l
11 # 第五次循环的时候n的值是o
```

## 4. 课堂练习---循环遍历计算字符串"hello itcast", 计算字符串中有多少字符

```
1  str1 = "hello itcast"
2  sum = 0 # 存放计算结果
3  for n in str1:
4      sum += 1 # 每循环一次sum的值就加1,结果是for循环几次,sum的值就是几
5
6  print(sum)
```

# while和for的不同

- while用在指定次数的循环
- for主要用在遍历
- while和for可以互换

# range函数

- 生成指定范围的一个数字序列
- 语法

```
1  range(start, stop, step)
2  start代表范围的开始值
3  stop代表范围的结束值,但不包含stop
4  step步长,可以为负数,正数代表升序范围,负数代表降序范围
```

```
1 range(1, 5, 1)    # 生成1, 2, 3, 4
2 range(1, 5)       # 生成1, 2, 3, 4
3 range(1, 6, 2)     # 生成1, 3, 5
4 range(6, 2, -1)    # 6, 5, 4, 3
5 range(6, 2, -2)    # 6, 4
```

## 用for实现指定次数的循环(用for替换while)

```
1 for 变量名 in range(开始值, 结束值, 步长):
2     执行代码
```

```
1 # a = 0
2 # while a < 10:
3 #     print(a)
4 #     a += 1
5 for a in range(0, 10): # for用于遍历,所以这里遍历的是一个范围,a就是范围中每个数字
6     print(a)
```

for 循环嵌套, 结合 range 实现输出 5 行\*号

```
1 # 外循环循环了5次
2 # 内循环循环了5次
3 # 内循环的print("*", end="")
4 # 每次内循环完成外循环都有个print()
5 for a in range(0, 5):
6     for b in range(0, 5):
7         print("*", end="")
8     print()
```

## 课堂练习

```
1 用for循环
2 计算从50到124之间所有偶数的累加和
3 包含50也包含124
```

```
1 # 在循环外,先定义一个变量存放累加和
2 # 在循环内,计算
3 # 循环完成后,输出这个变量的计算结果
4 sum = 0
5 for n in range(50, 125, 2):
6     sum += n
7 print(sum)
```

