



学习目标

1. 掌握在 while 循环语法;
2. 掌握 break 和 continue 的使用;
3. 掌握 for 循环语法;
4. 可以使用 for 循环遍历字符串;

传智播客-黑马程序员



目录

第 1 章 循环结构-----初识循环.....	3
第 2 章 循环结构-----break 和 continue.....	6
第 3 章 循环结构-----while 循环嵌套.....	8
第 4 章 循环结构-----for 循环.....	10

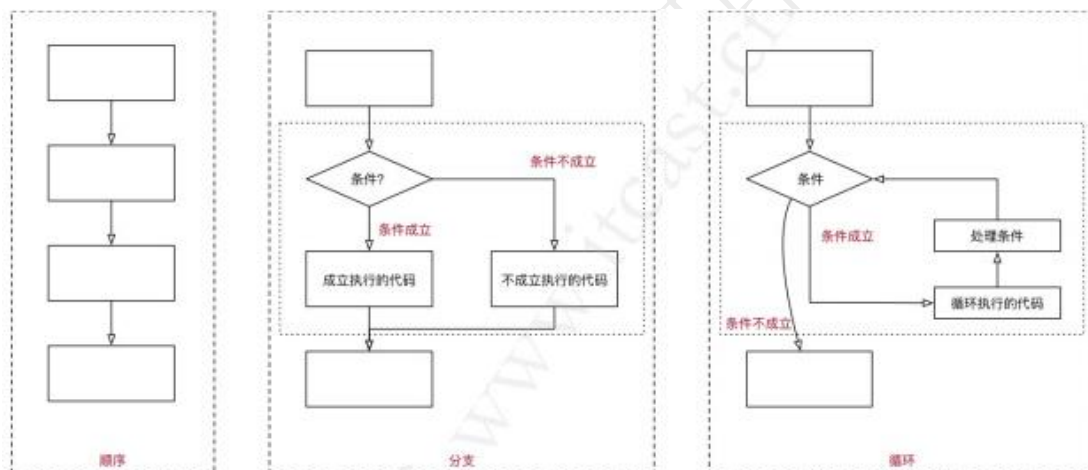
传智播客-黑马程序员

第 1 章 循环结构-----初识循环

一、计算机程序的三大流程

在程序开发中，一共有三种流程方式：

- 顺序 —— 从上向下，顺序执行代码；
- 分支 —— 根据条件判断，决定执行代码的分支；
- 循环 —— 让特定代码重复执行；



二、 while 循环简介

循环的作用就是让指定的代码重复的执行；

while 循环最常用的应用场景就是让执行的代码按照指定的次数重复执行 ；

1. while 语句基本语法

while 条件:

pass



注意：

`while` 语句以及缩进部分是一个完整的代码块

2. 第一个 `while` 循环

- 需求打印 5 遍 Hello Python

```
a = 0
while a < 5:
    print("hello python")
    a += 1
```

3. 死循环

由于程序员的原因，忘记在循环内部修改循环的判断条件，导致循环持续执行，程序无法终止！

```
a = 0
# 注意，这是一个死循环，因为a的值永远为0，所以循环不会结束
while a < 5:
    print("hello python")
```

三、Python 中的计数方法

常见的计数方法有两种，可以分别称为：

- 自然计数法（从 1 开始）—— 更符合人类的习惯；
- 程序计数法（从 0 开始）—— 几乎所有的程序语言都选择从 0 开始计数；

因此，大家在编写程序时，应该尽量养成习惯：除非需求的特殊要求，否则循环 的计数都从 0 开始。



- 循环-输出连续数字-----从 0 开始，截止到 5 的 6 个数字

```
i = 0
while i <= 5:
    print(i)
    i += 1
```

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

```
5
4
3
2
1
0
```

四、循环计算

在程序开发中，通常会遇到利用循环重复计算的需求。

遇到这种需求，可以：

1. 在 while 上方定义一个变量，用于存放最终计算结果；
2. 在循环体内部，每次循环都用最新的计算结果，更新之前定义的变量。

1. 需求----计算 0 ~ 100 之间所有数字的累计求和结果

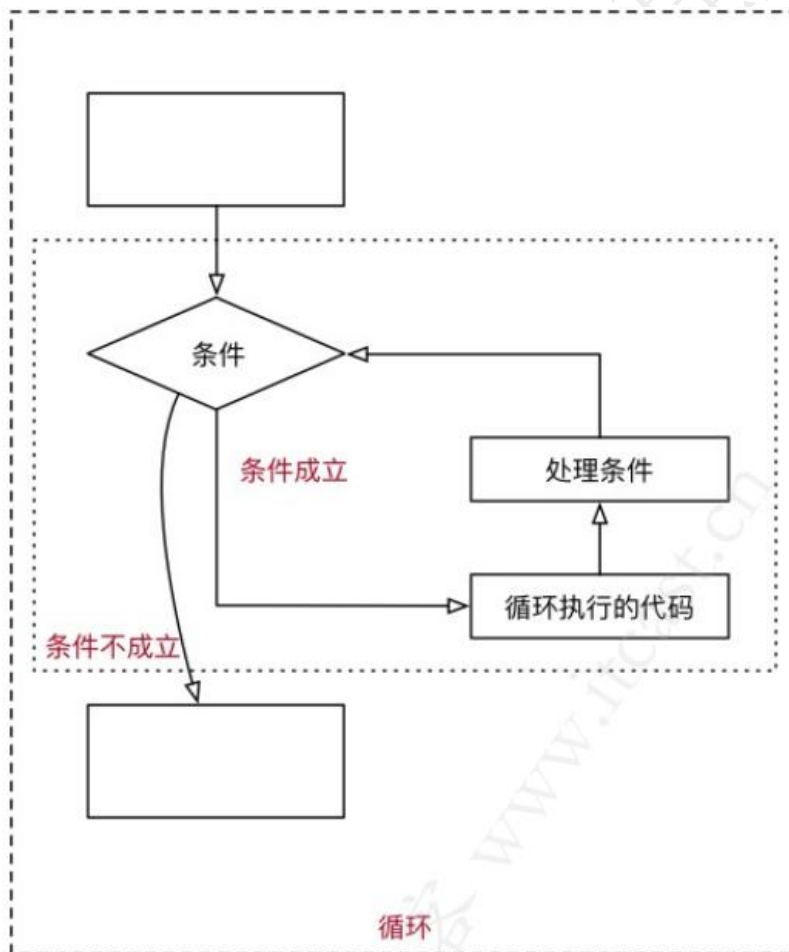
```
# 定义一个变量 num， 存放从 0 到 100 累加的结果
num = 0
# 定义一个变量 a， 决定 while 循环的次数
a = 0
# 只要 a <= 100, 循环就一直会进行，直到 a > 100 循环结束
while a <= 100:
    num += a
    a += 1

# 循环结束后，打印 num 的值
print(num)
```



第2章 循环结构-----break 和 continue

- break 和 continue 是专门在循环中使用的关键字；
- break 某一条件满足时，退出单层循环；
- continue 某一条件满足时，结束本次循环(不执行continue后的循环体语句)，执行下次循环；
- break 和 continue 只针对当前所在循环有效。





一、 break

在循环过程中，如果某一个条件满足后，不再希望循环继续执行，可以使用 `break` 退出循环。

```
a = 0
while a < 10:
    a += 1
    # 如果a 等于5, while 循环退出
    if a == 5:
        break
    print(a)
```

1. 课堂练习---

`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。

二、 continue

在循环过程中，如果 某一个条件满足后，不希望执行循环代码，但是又不希望退出循环，可以使用 `continue`。

也就是：在整个循环中，只有某些条件，不需要执行循环代码，而其他条件都需要执行。

需要注意：使用 `continue` 时，条件处理部分的代码，需要特别注意，不小心会出现死循环

```
a = 0
while a < 10:
    a += 1
    # 如果a 等于5, 跳过下面的print, 直接回到while 的开始继续循环
```



```
if a == 5:  
    continue  
print(a)
```

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

第 3 章 循环结构-----while 循环嵌套

一、循环嵌套

while 嵌套就是： while 里面还有 while ；

1. 循环嵌套演练 1

- 输出以下结果：

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

- 代码实现

```
a = 0  
# 外循环决定输出行数  
while a < 5:  
    # 内循环决定每行输出的*号个数  
    b = 0  
    while b < 5:
```




```
print("*", end="")
b += 1
print("")
a += 1
```

2. 循环嵌套演练 2

- 输出以下结果：

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

- 代码实现

```
a = 0
# 外循环决定输出行数
while a < 5:
    # 内循环决定每行输出的*号个数
    b = 0
    while b <= a:
        print("*", end="")
        b += 1
    print("")
    a += 1
```

3. 课堂练习---

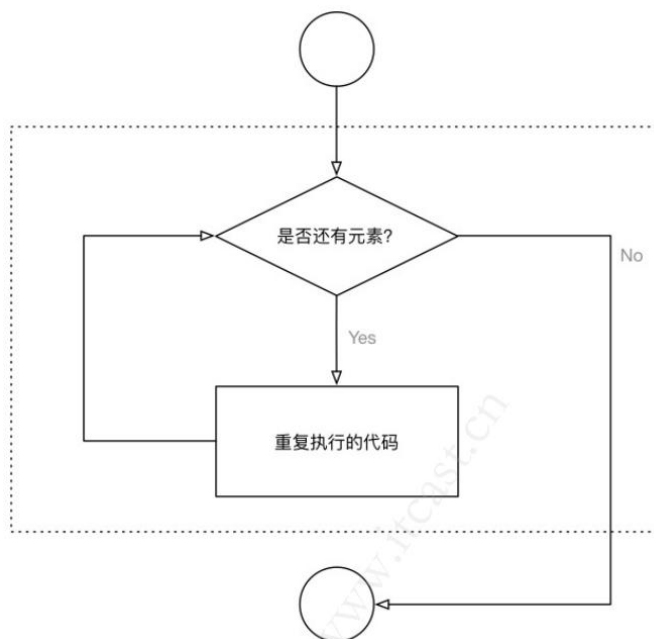
输出以下结果：

```
1
12
123
1234
12345
```



第4章 循环结构-----for 循环

- for 循环主要用在遍历场景；
- 遍历就是从头到尾 依次从列表中获取数据；
- 在循环体内部针对每一个元素，执行相同的操作；



- 语法：

for 变量 in 字符串:

循环里要执行的代码

一、for 演练----遍历字符串

```
str1 = "我今天学习 python"
for n in str1:
    #n 分别代表 str1 字符串里每一个字符
    print(n)
```



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

二、while 与 for 的不同

- while 循环的最大作用是指定次数的循环;
- for 循环最大作用是遍历;
- while 和 for 可以互换。

三、范围函数 range

- range 生成一个指定范围的数据;
- range(start, stop,int);
- start, 范围开始值;
- stop 范围终止值 (不包括 stop) ;
- int,步长, 如果小于 0, 代表生成降序范围, 如果省略 int, 默认值为 1。

```
# 生成从 0 到 9 的范围, 不包括 10
range(0, 10)
# 生成从 2 到 11 的偶数, 不包括 11
range(2, 11, 2)
# 生成从 20 到 11 的降序范围, 不包括 10
range(20, 10, -1)
```

四、for 结合 range 实现指定次数的循环

```
for n in range(0, 10):
    print(n)
```

1. 课堂练习----

for 循环嵌套, 结合 range 实现输出 5 行*号



传智播客-黑马程序员