

4.2 while循环

- 循环语句可以重复执行部分语句（循环语句/循环体），while语句是一种循环语句，根据一个逻辑条件（循环条件），在条件成立时执行循环语句，不成立时结束循环。
- 格式

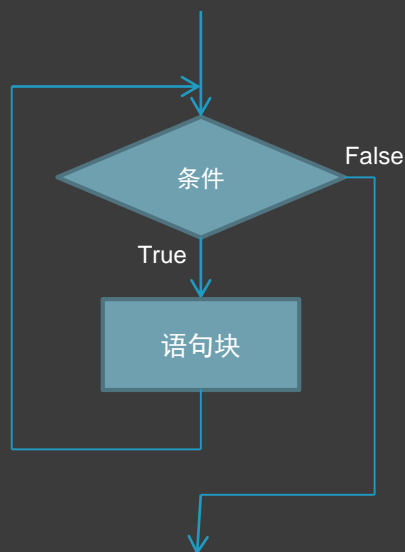
无else子句	有else子句
<pre>while 条件: 语句块1</pre>	<pre>while 条件: 语句块1 else: 语句块2</pre>

while循环

While循环的流程图

while 条件:

语句块1 #书写缩进

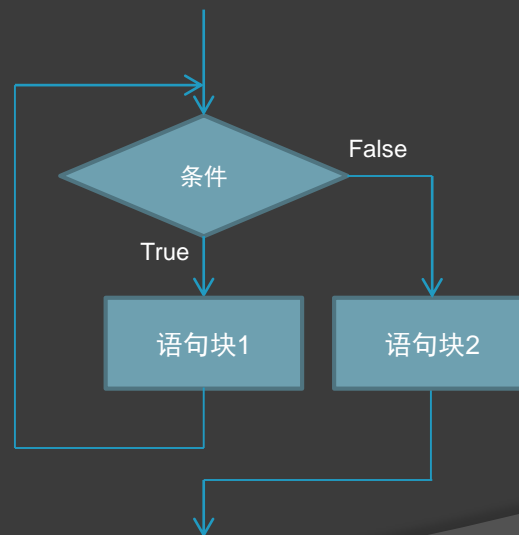


while 条件:

语句块1 #书写缩进

else:

语句块2 #书写缩进



while循环（续）

◎ 要点

- 执行while语句的流程：
 1. 判断条件是否成立
 2. 如果条件成立，则执行语句块1
 3. 回到第1步
 4. 如果条件不成立，若有else子句，执行语句块2
 5. 结束
- 循环语句书写必须缩进
- 在循环体内部，应该有改变循环条件的语句，以控制循环的次数，避免产生无限循环（死循环）。

while循环（续2）

- 例4-2 计算 $\log_2 x$

`x = int(input())`

`count = 0`

`while x > 1:`

`x //= 2`

`count += 1`

`print(count)`

输入：32

输出：5

输入：31

输出：4

缩进

步骤	x	count	说明
1	32	0	进入循环之前
2	16	1	循环的第一轮
3	8	2	
4	4	3	
5	2	4	
6	1	5	

步骤	x	count	说明
1	31	0	进入循环之前
2	15	1	循环的第一轮，31/2->15
3	7	2	
4	3	3	
5	1	4	

练习——二进制转换

```
n=int(input())  
lst=[]  
while n>0:  
    lst.insert(0,n%2)  
    n=n//2  
print("".join(map(str,lst)))
```

n	n%2
---	-----

2	20	0
2	10	0
2	5	1
2	2	0
2	1	1
	0	

while循环（续3）

例4-14 求最大公约数

求两个数的最大公约数可以用辗转相除法。

- 算法：

1. 先用小的一个数除大的一个数，得第一个余数；
2. 再用第一个余数除小的一个数，得第二个余数；
3. 又用第二个余数除第一个余数，得第三个余数；
4. 这样逐次用后一个余数去除前一个余数，直到余数是0为止。那么，最后一个除数就是所求的最大公约数（如果最后的除数是1，那么原来的两个数是互质数）。

while循环（续4）

- 更加形式化的算法描述：
 1. 计算两个数a和b的余数r；
 2. 如果余数r不为0，则以b和r作为新的a和b，回到1重复计算；
 3. 否则b就是余数
- 程序（a>=b）：

```
a, b = map(int, input("请输入两个整数：").split())
```

```
r = a % b
```

```
while r > 0:
```

```
    a, b = b, r
```

```
    r = a % b
```

```
print("最大公约数是{}".format(b))
```

输入：60 21

输出：最大公约数是3

步骤	a	b	r	说明
1	60	21	18	进入循环前
2	21	18	3	循环的第1轮
2	18	3	0	循环的第2轮

循环内的控制

- 跳出循环break

break语句的作用是跳出所在的循环。

例4-15 猜数游戏

```
import random
number = random.randint(0,100)
count = 0
while True: # 循环条件是逻辑常量True，意味着无限循环
    a = int(input('输入你猜的数:'))
    count += 1
    if a == number:
        break # 跳出当前循环，执行while后面的语句。
    elif a > number:
        print('你猜的大了')
    else:
        print('你猜的小了')
print('猜中了！你用了{}次！'.format(count))
```


循环内的控制（续）

● 跳过一轮循环continue

continue语句作用是跳过本次循环，进入到下一次循环。

例4-16 计算偶数的平均数

要求：输入一系列的整数，最后输入-1表示输入结束，然后程序计算出这些数字中的偶数的平均数，输出输入的数字中的偶数的个数和偶数的平均数。

循环内的控制（续2）

```
sum = 0
count = 0
while True:
    number = int(input())
    if number == -1:
        break
    if number % 2 == 1:
        continue # 如果是奇数的话，跳过后面的循环语句，
                 # 进入下一次循环。
    sum += number
    count += 1
average = sum / count
print(average)
```

循环内的控制（续3）

- 例4-17 输入一个大于等于2的正整数，判断是否为素数。

```
num=int(input())
```

```
a=num-1
```

```
while a>1:
```

```
    if num % a == 0:
```

```
        print("不是素数")
```

```
        break    # 跳出当前循环，包括else子句。
```

```
    a=a-1
```

```
else:
```

```
    print("是素数")
```

循环内的控制（续4）

- 说明

- 程序中的循环控制变量a从num-1递减到1，程序在每次循环判断a（从num-1到2）是否是num的因数。
- 如是，则打印“不是素数”，然后break语句跳出while语句，当然也跳过了else子句。
- 如果循环过程中“num % a”始终不为0，即num不能被从2到num-1中的任何一个数整除，说明num是素数，在循环结束后执行else子句，打印“是素数”。

4.3 for循环

- 当有一个序列，需要按照其顺序遍历其中的每一个单元的时候，就可以用for循环。

表 4-3 for语句语法格式

无else子句	带else子句
for 循环变量 in 序列: 语句块1	for 循环变量 in 序列: 语句块1 else: 语句块 2

- for i in [1,2,3,4]:
- print(i,end=" ")
- 结果是： 1 2 3 4

for...in循环

for循环又被叫做for ... in循环，它的一般形式是：

for <变量> in <序列>:

 缩进代码块

非缩进代码块

在循环的每一轮，<变量>会依次取序列中的一个值。对序列中的最后一个值执行完缩进代码块后，程序继续执行非缩进代码块。

两种循环模式

例：针对下列包含十二个月的英文缩写的列表month

```
month = ['JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 'JUL', 'AUG', 'SEP', 'OCT', 'NOV', 'DEC']
```

要遍历这个列表，输出每个月的缩写。

1.计数器循环：

```
for i in range(len(month)):
    print(month[i])
```

2.迭代循环：

```
for name in month:
    print(name)
```

判断素数（1）

输入一个大于等于2的正整数，判断是否为素数。素数是只能被1和自己整除的数，因数的范围是从1到自己。

直接用素数定义求解

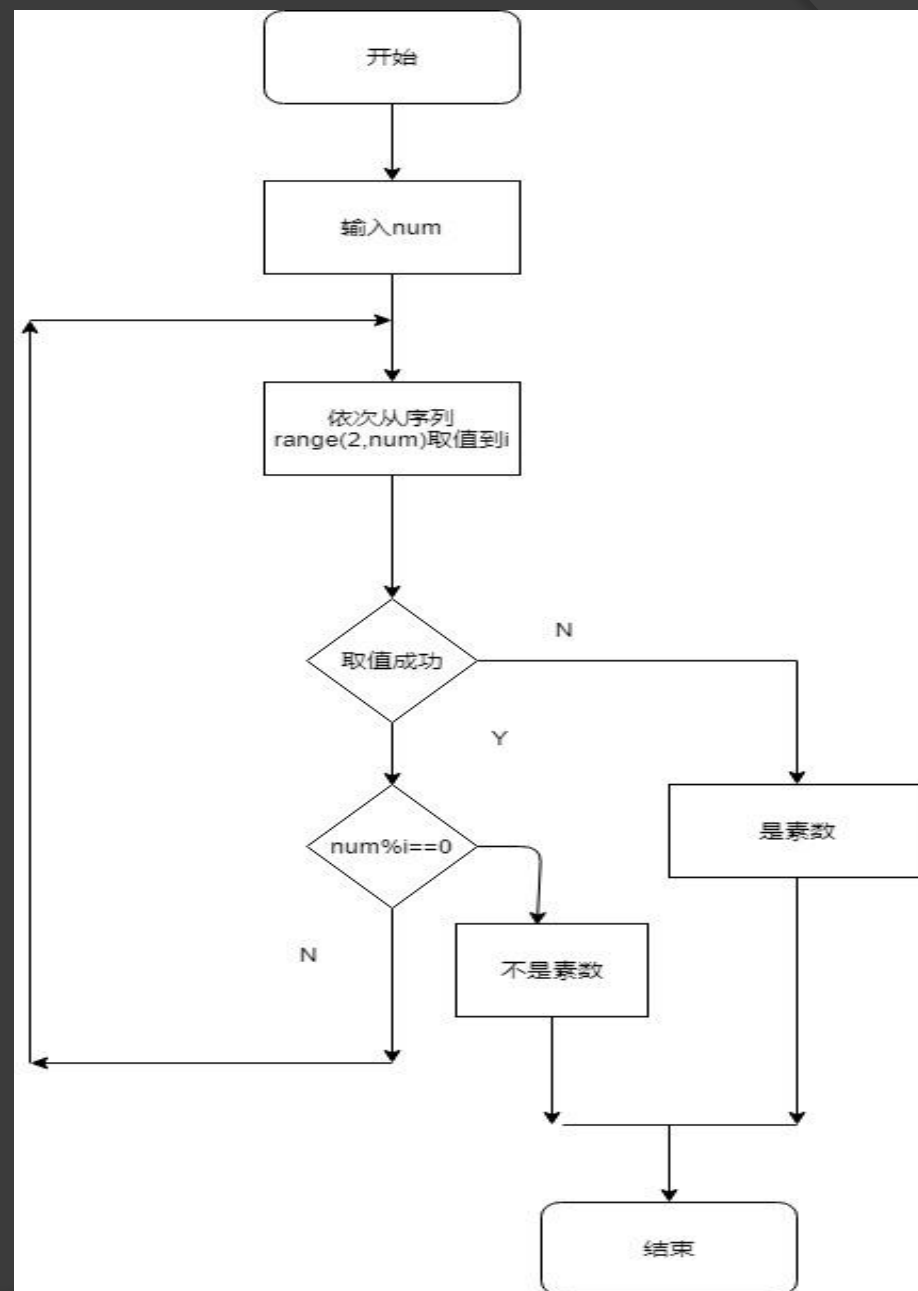
```
num=int(input())
lst=[ factor for factor in range(1,num+1)
      if num % factor==0]
if lst==[1,num]:
    print("是素数")
else:
    print("不是素数")
```


判断素数（2）

例4-18 输入一个大于等于2的正整数，判断是否为素数(for 语句实现)。

```
num=int(input())
for i in range(2,num):
    # i从2到num-1
    if num % i == 0:
        print("不是素数")
        break
else:
    print("是素数")
```

如何优化？



判断素数优化程序

- ◎ 如 $m=a*b$
- ◎ 则必有:
- ◎ $a \leq \sqrt{m}$
- ◎ 或:
- ◎ $b \leq \sqrt{m}$

例4-18 输入一个大于等于2的正整数，判断是否为素数。

```
import math
num=int(input())
for i in range(2,int(math.sqrt(num))+1):
    # i从2到sqrt(num)
    if num % i == 0:
        print("不是素数")
        break
else:
    print("是素数")
```