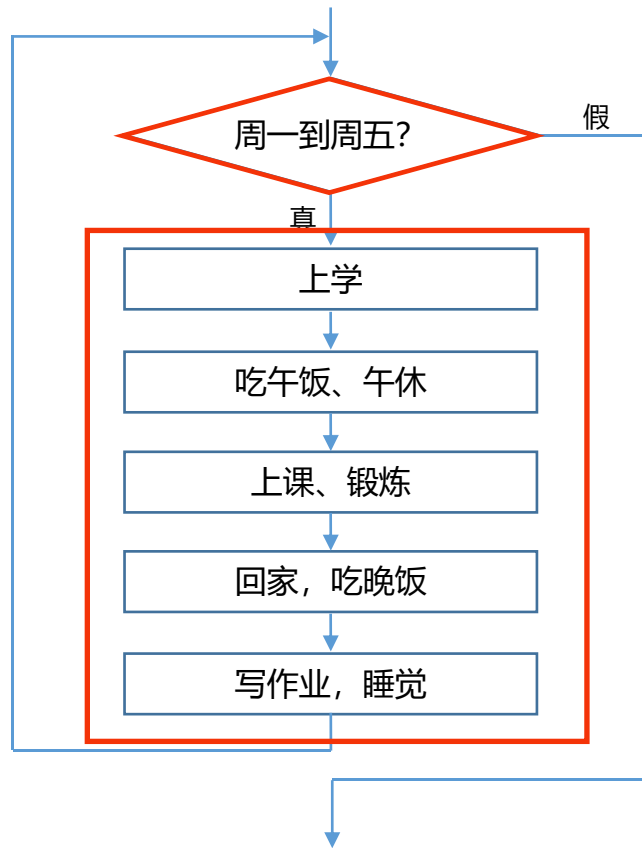




3.5.2 循环语句

■ 循环结构

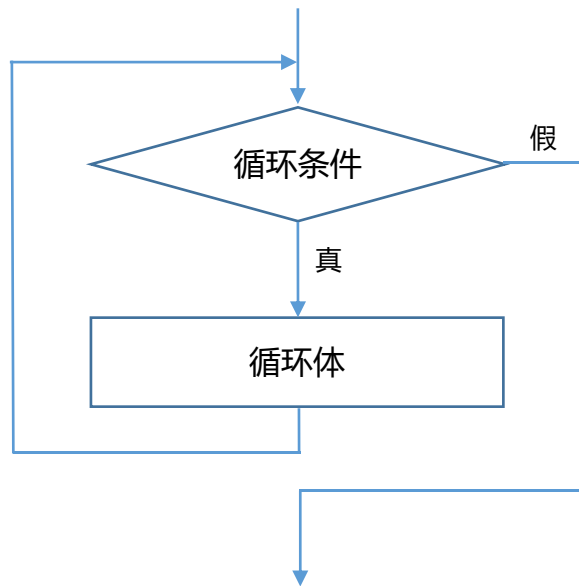
在**一定条件下反复执行**某段程序，
直到这个**条件不成立**为止。



■ while 语句

while 循环条件：
循环体

死循环：循环条件始终为真，
一直无法满足退出循环的条件。



3.5.2 循环语句

例：输出 $i=0,1,2$ 的值

```
1 i = 0
2 while i < 3:
3     print("i = %d" % i)
4     i += 1
5 print("continue....")
```

运行结果：

```
i = 0
i = 1
i = 2
continue.....
```



3.5.2 循环语句

例：实现1-100的累加和

```
1  #计算1--100的和
2  i=1
3  sum=0
4  while i<101:
5      sum += i
6      i+=1
7  print(sum)
```

#设置计数器初值
#设置累加和初值
#循环条件
#累加
#计数器+1
#输出累加和



运行结果：

5050



■ for语句

for 标识符 in 可迭代对象:
 循环体

接受**可迭代对象** (Iterable) 作为参数,
并对这个可迭代对象进行**遍历**。

```
>>>for c in "Python":  
    print(c)
```

P
y
t
h
o
n

```
>>>for c in "Python":  
    print(c, end="")
```

Python

```
>>>for c in "Python":  
    print(c, end=" ")
```

P y t h o n



□ range()函数

range (起始数字, 结束数字, 步长)

- **前闭后开**: 整数序列中不包括**结束数字**
- **起始数字**省略时, 默认从**0**开始
- **步长**省略时, 默认为**1**

range(5) [0,1,2,3,4]

range(0,5) [0,1,2,3,4]

range(1,10,2) [1,3,5,7,9]

[1,3,5,7,9]

```
>>>for i in range(1,10,2):  
    print(i)
```

1
3
5
7
9

```
>>>for i in range(1,10,2):  
    print(i, end=";")  
1;3;5;7;9;
```



3.5.2 循环语句

例：实现1-100的累加和

```
1  #计算1到100的累加和
2  sum = 0                    #设置初值
3  for i in range(101):      #循环条件
4      sum+=i                #累加
5  print("sum=",sum)         #输出累加结果
```

运行结果：

```
sum= 5050
```



■ **continue语句**：终止**本次循环**，开始下一次循环

例：计算1-100之间的所有奇数的和。

```
#计算1到100的奇数和
sum = 0
for i in range(1,101):
    if i % 2 == 0:
        continue
    sum += i
print("sum=",sum)
```

#设置初值
#循环条件
#如果是偶数，则跳过本次循环，不进行累加

#累加
#输出累加结果

运行结果：

```
sum= 2500
```



3.5.2 循环语句

例：计算1-100之间的所有**奇数**的和。

```
#计算1到100的奇数和
sum = 0
for i in range(1,101,2):
    sum += i
print("sum=",sum)
```

#设置初值
#循环条件
#累加
#输出累加结果

运行结果：

```
sum= 2500
```



■ break语句：跳出循环体，结束循环

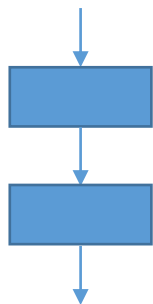
例：计算1-100的累加和。如果在计算过程中，累计和**超过3000**，则**终止计算**，并输出当前结果。

```
i=1                #设置初值
sum=0
while i<101:        #循环条件
    if sum<=3000:    #附加条件
        sum += i
        i+=1
    else:             #累积和>3000,跳出循环,终止计算
        break
print(sum)
```

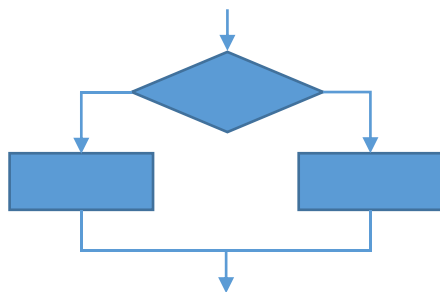


■ 程序流程控制

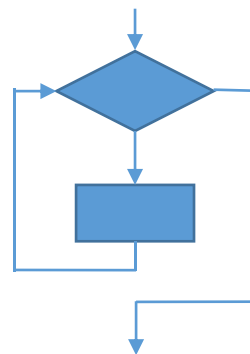
- 顺序结构：从上到下依次顺序执行。
- 选择结构：根据**判断条件**，来**选择**执行不同的任务。
- 循环结构：根据**循环条件**，来**重复**执行某段代码。



顺序结构



选择结构
条件分支语句



循环结构
for语句, while语句

