

## 条件控制-if~else语句(重点)

```
1  格式:
2  if 条件语句:
3      代码块1
4  else:
5      代码块2
6
7  a=10
8  b=20
9
10 if "sss":#0表示假,非0表示真,空字符串为假,非空字符串为真
11     print("a<b")
12 else:
13     print("a>=b")
14
15
16 注意:
17 1) Python没有像其它大多数语言一样使用“{}”表示语句体,
18 而是通过语句缩进来判断语句体, 缩进默认为4个空格。
19 2) 可以使用比较、逻辑表示条件语句
20 3) 可以使用in和not in表示包含关系
21 4) 甚至可以使用布尔类型的判断
22
23 案例:从控制台输入一个数,判断这个数是否是偶数
24 n=int(input("请输入你的数字:"))
25 if n%2:
26     print("不是偶数")
27 else:
28     print("是偶数")
29
30 #案例:从终端输入用户和密码,如果都正确,打印欢迎登入,如果有一个不正确就打印用户名或密码不正确
31 #用户名在str1='makermaker2maker3maker4',密码在str2='0001000200030004'
32
33
```

## if-elif-else(重点)

```
1  格式:
2  if 条件语句1:
3      代码块1
4  elif 条件语句2:
5      代码块2
6  elif 条件语句3:
7      代码块3
8  else:
9      代码块4
10
```

```
11 注意：
12 1、每个条件后面要使用冒号（：），表示接下来是满足条件后要执行的语句块
13 2、使用缩进来划分语句块，相同缩进数的语句在一起组成一个语句块
14 3、在Python中没有switch - case语句
15
16 案例：根据控制台输入的成绩，输出对应的等级
17     90以上：优秀
18     80~90：良好
19     60~80：还行
20     60以下：加油吧，少年
21
22 #案例：从终端输入一个字符串，判断是否为纯数字，是否为纯大写字母，是否为纯小写字母，然后打印对应的字符串
23
```

## if-elif-else的嵌套

```
1 格式：
2  if 条件1：
3     if 条件4：
4         代码块1
5     else：
6         代码块4
7
8  elif 条件2：
9     if 条件5：
10         代码块2
11     else：
12         代码块5
13 else：
14     代码块3
15
16
17 注意：为了代码的维护，建议嵌套不要超过3层
18 案例：用户身份验证：用户名输入admin，密码输入123456，验证成功，用嵌套的方法
19 yzm=3344
20 user="admin"
21 password="123456"
22
23 if yzm==3344：
24     #如果进入到这，表示你输入的验证码正确，然后我才会判断用户名和密码
25     if user=="admin" and password=="123456":
26         print("欢迎登录")
27     else：
28         print("输入的用户名或密码错误")
29 else：
30     print("验证码错误")
31
32 #案例，猜数字，有一个数字，389，猜中第一位才可以继续往下猜，没猜中就现在'很遗憾，没猜中'，猜中了就现在
33 #'恭喜你，猜中了，请继续'，全猜中，打印'太棒了，全中'
34
35
```

## while(重点)

```
1  1.格式:
2  while 条件语句:
3      代码块
4
5  #打印1-100的数字
6  n=1
7  while n<=100:
8      print("n=",n)
9      n+=1 #n=n+1
10
11 #while的应用:猜数字游戏,请用户输入数字,如果用户输入的数字比正确的数字大,那么就显示大了
12 #如果用户输入的数字比正确的数字小,那么打印小了,如果猜中,就打印恭喜你,猜中了
13 while 1:
14     z=66
15     n=int(input("请输入你猜的数字:"))
16     if n<z:
17         print("你猜的小了")
18     elif n>z:
19         print("你猜的大了")
20     else:
21         print("恭喜你,猜对了")
22         break#退出循环
23
24 2.while 循环使用 else 语句
25 while 条件语句:
26     代码块
27 else:
28     代码块
29
30 注意:当while的代码块中有break跳出循环,那么else中的代码块不会执行
31
32 注意:在Python中没有do..while循环
33 #案例:统计1-100之间包含的偶数个数,用while..else
34 案例:使用while嵌套循环打印如下图形
35 *
36 * *
37 * * *
38 * * * *
39 * * * * *
40 案例:编写代码模拟用户登陆。要求:用户名为 python,密码 123456,如果输入正确,打印“欢迎光临”,程序
    结束,如果输入错误,提示用户输入错误并重新输入
41
42
43
```

## for(重点)

```
1 格式:
2  for 元素 in 迭代数据类型:
3      代码块
4
5  解释: 迭代数据类型(字符串, 列表等)
6
7  #案例: 逐个输出"Python"的每个字母
8  #for循环适用于遍历序列
9  for i in "python":
10     print(i)
11
12 #案例: 有字符串 'fdafafa123afd456dasfaf789dafa', 打印出数字
13
```

## range()函数(重点)

```
1  range()函数
2  如果你需要遍历数字序列, 可以使用内置range()函数。它会生成数列
3  你也可以使用range指定区间的值
4  也可以使range以指定数字开始并指定不同的增量(甚至可以是负数, 有时这也叫做'步长')
5  格式:
6  for i in range(开始[, 结束, 步长]):
7      print(i)
8
9  #打印1-100, 生成的数列是1-100, 不包含101
10 for i in range(1, 101):
11     print(i)
12
13 #这里的10表示结束, 生成的数列的0-9
14 for i in range(10):
15     print(i)
16
17 #生成1, 3, 5, 7, 9的数列
18 for i in range(1, 10, 2):
19     print(i)
20
21 #生成10, 8, 6, 4的数列
22 for i in range(10, 2, -2):
23     print(i)
24
25 #案例: 用for加range()函数输出10-20的偶数, 反向打印
26 for i in range(20, 9, -2):
27     print(i)
28
29 #案例: 打印以下图形
30 0123456789
31 101112131415
32 1617181920
33
```

## break,continue

```
1 break:退出循环,嵌套的循环中一个break只能跳出一个循环
2 while 1:
3     print("外循环")
4     while 1:
5         print("内循环")
6         break#只能跳出一个循环
7
8
9 continue:退出本次循环
10 for i in range(11):
11     if i==6:
12         continue#这个语句后面的代码就不执行,回到循环开头
13     print(i)
14
15 #案例:实时监控用户的输入,当用户输入66时,退出
16
17 #案例:猜数字游戏,猜到比确定数字大的显示大了,小的显示小了,重新输入,才中了退出
18
19
```

## for使用else子句(重点)

```
1 格式:
2  for i in range(5):
3     pass
4  else:
5     pass
6
7  for i in range(10):
8     print(i)
9     if i==5:
10         break
11 else:
12     print("for循环结束")
13
14 #案例:输出10-20之间的素数
15 说明:质数又称素数。一个大于1的自然数,除了1和它自身外,不能被其他自然数整除的数叫做质数
16
17 #案例:统计100以内能够被2整除,但是不能被3整除的数的个数
18
19
```

## pass(重点)

```

1 pass语句
2 Python中pass是空语句，是为了保持程序结构的完整性
3 pass 不做任何事情，一般用做占位语句
4 #打印偶数
5 for i in range(10):
6     if i%2:
7         pass
8     else:
9         print(i)
10        print("是偶数")
11
12 '''
13 案例,用for循环打印下图
14 *
15 * *
16 * * *
17 * * * *
18 * * * * *
19 '''
20
21 for i in range(5):#外循环的次数控制行数
22     for j in range(i+1):#内循环的次数控制列数
23         print("*",end=' ')
24     print()#换行

```

## 列表(重点)

```

1 一个变量每次只能存储一个数据，如果一次性存储多个数据，怎么做?答:用列表
2 1.列表是一种数据类型，也是一个容器，里面可以装数据
3 2.列表的创建：
4 格式一：
5 列表名=[数据1,数据2,数据3,.....]
6 比如:mylist=[1,2,3,4,5,6] #在创建列表时就定义好了内存空间的大小,后期通过下标不能越位操作
7 格式二：
8 列表名=[] #这种定义方式，定义的列表是没有空间的，不能通过下标来存储数据
9 比如:mylist2=[]
10 3.列表的数据存储或取出
11 mylist[0]=100
12 print(mylist)#[100,2,3,4,5,6]
13 mylist[6]=200
14 print(mylist)#报错
15 mylist2[0]=100#报错
16
17 注意，
18 1、list写在方括号之间，元素用逗号隔开
19 2、和字符串一样，list可以被索引和切片
20 #切片
21 print(mylist[0:3])
22 3、list可以使用+操作符进行拼接
23 mylist3=['a','b',22]

```

```

24 print(mylist+mylist3)
25 4、list中的元素是可以改变的
26 5.列表中的元素可以是不同的数据类型
27
28 #案例:通过循环的方式打印出list列表中的元素
29 mylist=[1,2,3,4,5]
30 for i in mylist:
31     print(i)
32
33 #案例:有列表my=[1,2,3,4,5],把列表变为[10,20,30,40,50]
34 my=[1,2,3,4,5]
35 for i in my:
36     my[i-1]=i*10
37
38 print(my)
39
40 #案例:有列表my2=[1,2,'a',3,'b',4],把列表中的字母a变为前面的数字之和,把b变为前面数字之和
41 #最后计算整个列表的数字之和.假设字母位置不固定,字母的个数也不固定(有难度)
42

```

## 列表的操作-追加append()(重点)

```

1  append 追加,在列表的末尾添加新的对象
2  当空列表要添加数据时,可以使用append来添加数据
3  mylist=[1,2]
4  mylist2=[]
5
6  mylist.append(100)
7  mylist2.append(200)
8  mylist2.append(300)
9  print(mylist)#[1,2,100]
10 print(mylist2)#[200,300]
11 mylist2.append("hello")
12 print(mylist2)#[200,300,"hello"]
13
14 #案例:定义一个空列表,然后填入1-10

```

## 列表的操作-添加序列extend(重点)

```

1  extend(seq) 把一个序列seq的内容添加到列表中
2  mylist=[1,2]
3  mylist.extend("hello")
4  print(mylist)#[1, 2, 'h', 'e', 'l', 'l', 'o']
5  mylist.extend([10,20,30,40])
6  print(mylist)#[1, 2, 'h', 'e', 'l', 'l', 'o', 10, 20, 30, 40]
7
8  #拆解字符串
9  mystr="world"
10 mylist2=[]

```

```
11 mylist2.extend(mystr)
12 print(mylist2)#[ 'w', 'o', 'r', 'l', 'd']
13
14 #案例:从终端中输入5个数字,以空格分开,然后把这此数字组成一个列表,然后加入到mylist2=[1,2,3,4,5]列表中
15 #注意,要把列表中字符数字变为数字
16 n=input("请输入5位的数字:")
17 mylist=n.split()
18 k=0
19 for i in mylist:
20     mylist[k]=int(i)
21     k+=1
22 print(mylist)
23 mylist2=[1,2,3,4,5]
24 mylist2.extend(mylist)
25 print(mylist2)
```

## 列表的操作-插入insert(重点)

---

```
1 insert(位置,元素) 在指定的索引处插入一个元素
2 mylist=[1,2,3,4]
3 mylist.insert(8,100)#如果位置越位,那么就在最后添加元素
4 print(mylist)#[1, 2, 3, 4, 100]
5 mylist.insert(0,200)
6 print(mylist)#[200, 1, 2, 3, 4, 100]
7
8 #案例:有列表list=[1,2,4,6,8,10],从终端输入一个数字,插入到指定位置
9 #如,你输入的是3,那么插入list列表后[1,2,3,4,6,8,10]
10
11
```