亿图图示

控制流语句 Contents

if判断

基础的if判断

```
if 条件:
    条件为真（True）执行操作
```

英文冒号后回车时会自动换行缩进（1Tab=4个空格）

if-else语句

```
if 条件:
    条件为真（True）执行操作
else:
    条件为假（False）执行操作
```

英文冒号后回车时会自动换行缩进（1Tab=4个空格）

else冒号后不需要写条件

input函数

input函数输出的结果是字符串

字符串类型之间可以判断是否相等，但不能比较大小，需要赋值

需要使用判断语句时，需要转换为数值类型 int(input('你的生日是:')

if-elif-else语句

```
if <条件判断1>:
    <执行1>
elif <条件判断2>:
    <执行2>
elif <条件判断3>:
    <执行3>
else:
    <执行4>
```

从上至下判断，较难判断条件写在上面：大于最大，小于最小）

对同一个条件内容进行判断

三元运算符

将if和else写到一行

```
print('偶数') if x%2==0 else  print('奇数')
```

if嵌套

多条件判断，先判断条件一

从外向内判断，不满足时后面不会继续判断

```
if x=='2':
    if y=="2":
        if z=='2':
            print('都是2')
        else:
            print('z输入错误')
    else:
        print('y输入错误')
else:
    print('x输入错误')
```

循环语句

while循环

while语句用于循环执行程序,即在某条件下,循环执行某段程序,以处理需要重复处理的相同任务

```
while 判断条件:
    执行语句...
```

判断连续数据的奇偶

```
i=0
while i<=5:
    if i%2==0:
        print(i,'偶数')
    else:
        print(i,'奇数')
    i+=1
```

1到100的累计求和

```
i=1
s=0
while i<101:
    s+=i
    i+=1
print(s)
```

为了避免死循环，需要自增长赋值：i+=1
但要注意缩进位置与else对齐

while循环嵌套

九九乘法表

```
i=1
while i<10:
    j=1
    while j<=i:
        print('{}*{}={}'.format(j,i,i*j),end=' ')
        j+=1
    print()      #强制换行，每一行以空格结尾换行
    i+=1
```

for循环

迭代：一次从容器中取一个元素

遍历：将所有元素依次一一取出

语法：for 临时变量 in 列表或者字符串等容器中:

- 字符串在迭代过程中会产生一个字符
- 列表在一次迭代过程中产生一项

for循环的嵌套

```
for i in a:
    if type(i)==list:
        for j in i:
            print(j)
    else:
        print(i)
```

break和continue

break 结束整个循环

```
for i in strl:
    if i==',':
        break
    print(i)
```

continue 终止本次循环,继续下一次循环，不输出本次循环结果

```
for i in strl:
    if i=='你':
        continue
    print(i)
```

pass语句

空语句，保证语法结构的完整性

作用是一个占位符

```
if x>3:
    pass
else:
    print('错')
```

range函数

生成range对象

语法：range（start，stop，step） stop取不到，值减一

设置参数时必须都要写完整

1到100的累计求和

```
s=0
for i in range(1,101):
    s+=i
print(s)
```

1到10的奇数

```
for i in range(1,11,2):
    print(i)
```