



# 实验5 控制结构1

ANDYWWW

LAST MODIFIED: 7 APR 2019

1.产生一张华氏0 ~ 300度与对应的摄氏温度之间的对照表（每隔30度输出一次）

- 思路：**
- (1) 利用循环，生成0~300度每隔30度的华氏温度  
可以直接用`range(0,301,3)`生成，也可以利用`range(301)`再搭配判断，还可以利用`while F <= 300`
  - (2) 对于给定的F，根据公式计算出对应的摄氏温度C
  - (3) 输出F、C对

1.产生一张华氏0 ~ 300度与对应的摄氏温度之间的对照表（每隔30度输出一次）

答案：

```
for F in range(0,301,30):  
    C = 5 * (F - 32) / 9  
    print("{} {:.6f}".format(F,C))
```

format或%格式化的使用方法见下页

如果我们想对所得到的F、C的一系列数据进行进一步处理，我们需要在循环前建立一个空列表，在循环体中将所得到的数据append到空列表中。

```
for F in range(0,301):  
    if F % 30 == 0:  
        .....#输出部分
```

```
F = 0  
while F <= 300:  
    .....#输出部分  
    F += 30
```

```
F_lst = []; C_lst = []  
for F in range(0,301,30):  
    C = ...  
    F_lst.append(F)  
    C_lst.append(C)
```

也可以用这两个语句进行循环

# 字符串格式化

格式字符串: 'bulabula' % (对象1, 对象2, 对象3...)

格式化模板: 'bulabula'.format(对象1, 对象2, 对象3...)

```
name = "Andywww"
```

```
age = 20
```

```
print("My name is", name, ", and I am", age, "years old.")
```



```
print("My name is %s, and I am %d years old." % (name, age))
```

```
print("my name is {0}, and I am {1} years old.".format(name, age))
```

## 格式字符串

在字符串相应地方用%等一系列代码占位，这一系列代码主要表示该位置数据的原有格式、缩进、精确位数等；在字符串后为一个%和一个元组，依次将元组中的数据调入字符串中

Syntax: “xxx %...xxx %...” % (..., ..., ...) 元组内容与前文中的%依次对应

↓

%[(name)][flags][width][. precision] typecode

flags: 用于对齐、数据表示 [ 可选 ]

width: 数据占有宽度 [ 可选 ]

.precision: 小数点后保留位数 [ 可选 ]

typecode: 对传入数据操作 [ 必选 ]

## typecode [必选] 对传入数据操作

s: 获取返回对象的\_\_str\_\_方法返回值，并将其格式化至指定位置

常见的有\_\_str\_\_方法的数据类型：字符串、浮点型 注意：整型没有！

r: 获取返回对象的\_\_repr\_\_方法返回值，并将其格式化至指定位置

\_\_repr\_\_方法即将表达式转化为字符串，比如[1,2].\_\_repr\_\_()为'[1,2]'; 大部分类型都有

o: 将整数转化为八进制，并将其格式化至指定位置

x: 将整数转化为十六进制，并将其格式化至指定位置

} 只针对整数！

d: 将整数、浮点数(四舍五入至整数)转化为十进制，并将其格式化至指定位置

?: 如果需要输入百分号则需要%%形式，例如"%%" % ()

## typecode [必选] 对传入数据操作

e: 将整数、浮点数转换成科学计数法，并将其格式化至指定位置

E: 将整数、浮点数转换成科学计数法，并将其格式化至指定位置

e指令出现的是诸如1e-12的格式，E指令出现的是诸如1E-12的格式

f: 将整数、浮点数转换成浮点数表示，并将其格式化至指定位置

默认小数点后6位

F: 将整数、浮点数转换成浮点数表示，并将其格式化至指定位置

g: 自动调整将整数、浮点数转换成 浮点型或科学计数法表示（超过6位数用科学计数法），并将其格式化至指定位置

g/G区别与e/E相同

G: 自动调整将整数、浮点数转换成 浮点型或科学计数法表示（超过6位数用科学计数法），并将其格式化至指定位置



## flags [可选] 用于对齐&数据表示

+ 右对齐；正数前加正号，负数前加负号

- 左对齐；正数前无符号，负数前加负号

<sp> (空格) 右对齐；正数前加空格，负数前加负号

0 右对齐；正数前无符号，负数前加负号，空白处用0填充

例："%+8.4f" % (3.1415926)

结果：'+3.1416'

传入的数据为3.1415926，保留到小数点后4位，右对齐，正数前加正号，占8个长度



## 字符串模板

在字符串相应地方用{}等一系列代码占位，这一系列代码主要表示该位置数据的原有格式、缩进、精确位数等；在字符串后为.format(...,...)，依次将元组中的数据调入字符串中

Syntax: “xxx {...} xxx {...}” .format (... , ... , ...)

↓  
[loc:][[fill]align][sign][#][width][,][.precision][type]

fill 空白处填充的字符

align 对齐方式(配合width使用)

sign 数字是否有符号

# 此命令只针对于二进制、八进制、十六进制

, 为数字添加分隔符，例如1,000,000

width 格式化位所占宽度

.precision 小数位保留精度

type 格式化类型

所有参数均  
为可选！

## type 格式化类型

### 字符串类型

s 格式化字符串类型数据

<sp> 未指定类型，默认为None，直接利用str()函数对其强制转化

### 整数类型

b 将整数转换为二进制表示而后格式化

d 十进制整数而后格式化

o 将整数转换为八进制而后格式化

x 将整数转换为十六进制而后格式化 其中x小写

X 将整数转换为十六进制而后格式化 其中X大写

## type 格式化类型

### 浮点类型

- e 转换为科学计数法（小写e）表示，然后格式化；
- E 转换为科学计数法（大写E）表示，然后格式化；
- f 转换为浮点型（默认小数点后保留6位）表示，然后格式化；
- F 转换为浮点型（默认小数点后保留6位）表示，然后格式化；
- g 自动在e和f中切换（科学计数法小写e）
- G 自动在E和F中切换（科学计数法大写E）
- % 显示百分比（默认显示小数点后6位）

## **loc: 位置**

代表此处传入的数据在format后括号的索引，如果只有loc一个参数，冒号可以省略。如果不说明则默认为依次选取

## **align 对齐方式**

- < 左对齐
- > 右对齐（默认）
- = 内容右对齐，符号左对齐，即：符号+填充物+数字
- ^ 内容居中

## sign 设置数字符号

- + 正数加正号，负数加负号
- 正数不加正号，负数加负号
- <sp> 正数加空格，负数加负号

例: "{1:0=+10.3f}".format(1.23456789,3.1415926)

结果: '+00003.142'

传入的数据索引为1，即3.1415926，保留到小数点后3位，正数前加正号，占10个长度，数字右对齐，符号左对齐，其他位置用0填充

## 练习

1. 右对齐加正负号长度为10保留4为小数

`"{:>+10.4f}".format(x)` 或 `"%+10.4f" % (x)`

2. 如果整型year存储着年份、整型month存储着月份、整型day存储着日期，如何利用format生成类似“2019年04月04日”的字符串？

`"{0:d}年{1:0>2d}月{2:0>2d}日".format(year,month,day)`

注意：format是字符串的一个方法，其返回值是一个符合格式化条件的字符串，因此可以对其结果进行赋值！

## 2. 输入一行字符，分别统计出其中英文字母、空格、数字和其它字符的个数。

---

**思路：** 利用循环（字符串是可迭代对象，可以用for循环一个  
一个取）取出每隔字符，判断其是否是字母、空格等

字母：

1. 手动输入一个长度为52的字符串（或列表）s = "ab...z AB...Z"，判断这个字符是否在字符串内（in）
2. 利用字符串的isalpha方法
3. 利用'a'-'z'、'A'-'Z'的ASCII码是连续的特点

数字：

1. 手动输入一个长度为10的字符串（或列表）s = "01...9"，判断这个字符是否在字符串内（in）
2. 利用字符串的isdigit方法
3. 利用'0'-'9'的ASCII码是连续的特点

空格

1. 判断字符是不是" "（这是一个空格）  
！ 注意区分空字符串和空格字符
2. 利用字符串的isspace方法



2. 输入一行字符，分别统计出其中英文字母、空格、数字和其它字符的个数。

答案

```
string = input()
alpha = 0; sp = 0
digit = 0; other = 0
for char in string:
    if char.isalpha():
        alpha += 1
    elif char == " ":
        sp += 1
    elif char.isdigit():
        digit += 1
    else:
        other += 1
print(alpha,sp,digit,other,sep='\n')
```

因为a-z以及A-Z的ASCII码是连续的，因此判断是否为字母时也可以直接利用这个关系

```
if (char >= 'a' and char <= 'z') or (char >= 'A' and char <= 'Z'):
    alpha += 1
```

注意: z和Z的ASCII码不是连着的，逻辑词and和or也要注意，为了避免歧义建议使用小括号

同理'1'到'9'的ASCII码也是连续的，因此也可以利用同样的关系比较

```
if char >= '0' and char <= '9':
    digit += 1
```

3. 输入数组，最大的与第一个元素交换，最小的与最后一个元素交换，输出数组。

**思路：** 首先找到数组中最大的元素，将其与第一个元素交换；然后再找数组中最小的元素，将其与最后一个元素交换

Q1: 如何寻找数组中最大（小）的元素？

1. 可以利用内建函数max、min
2. 遍历整个数组，先假设第一个数是最大（小）值，如果后面的数比记录的最大（小）值还要大（小），则最大（小）值换为此值。直到遍历完整个数组（代码见右侧）

```
lst = [...]  
max_value = lst[0]  
min_value = lst[1]  
for value in lst:  
    if max_value < value:  
        max_value = value  
    if min_value > value:  
        min_value = value
```

初始化一定要在循环前进行！

### 3. 输入数组，最大的与第一个元素交换，最小的与最后一个元素交换，输出数组。

Q2: 如何交换两个指定元素?

1. Python特有的交换元素的语句 `x, y = y, x`
2. 一般情况下，如果要交换x、y两个元素，会引入第三个元素tmp，具体代码见右

```
x = ...; y = ...  
#开始交换  
tmp = x  
x = y  
y = tmp
```

Q3: 为什么不能最大元素交换和最小元素交换同时做?

如果最小的元素就是第一个元素，先把最大的元素与第一个元素交换后再进行交换的就是不是原来的最小的元素了。

3. 输入数组，最大的与第一个元素交换，最小的与最后一个元素交换，输出数组。

---

答案：

```
lst = [int(i) for i in input().split()]
max_index = lst.index(max(lst))
lst[max_index], lst[0] = lst[0], lst[max_index]
min_index = lst.index(min(lst))
lst[min_index], lst[-1] = lst[-1], lst[min_index]
print(lst)
```

max(lst)、min(lst)可以用Q1代码中的max\_value、min\_value代替

列表元素交换的代码也可以用Q2的代码替换

4. 求 $s=a+aa+aaa+aaaa+aa...a$ 的值，其中a（键盘输入）是一个数字

- 思路：**
1. 输入a对应的数字、输入n个数字相加
  2. 利用循环生成n个数字：a、aa、...、aa...a（n个a）
  3. 将n个数字加起来

Q: 如何生成这n个一位到n位的数字

1. 利用数学办法

假设需要生成m个a的数字（ $1 \leq m \leq n$ ）

① 利用m次循环，依次加上这m位，第i位对应乘以 $10^{i-1}$

② 利用递推关系，上一个数 $\times 10 + a$ 就是下一个数

①

```
a = ...  
num = 0  
for i in range(0,m):  
    num += a * 10 ** i
```

②

```
a = ...  
num = 0  
for i in range(0,m):  
    num = num * 10 + a
```

4. 求 $s=a+aa+aaa+aaaa+aa...a$ 的值，其中a（键盘输入）是一个数字

---

Q: 如何生成这n个一位到n位的数字

2. 利用字符串乘法

注意对于字符串而言乘法相当于复制，比如“a” \* 5的结果是“aaaaa”，对于这个题而言可以将a化为字符串对其进行乘法，再转回整型

4. 求 $s=a+aa+aaa+aaaa+aa...a$ 的值，其中a（键盘输入）是一个数字

答案1:

```
a, n = [int(x) for x in input().split()]
s = 0
for i in range(1, n + 1):
    tmp = int(str(a) * i)
    s += tmp
print(s)
```

字符串乘法

答案2:

```
a, n = [int(x) for x in input().split()]
s = 0
for m in range(1, n + 1):
    num = 0
    for i in range(0, m):
        num = num * 10 + a
    s += num
print(s)
```

也可以用Q①  
的代码代替

数学方法



4. 求 $s=a+aa+aaa+aaaa+aa...a$ 的值，其中a（键盘输入）是一个数字

考虑到s的每一项之间有递推关系，可以考虑用列表对数据进行储存，最后利用sum求和

**答案3:**

```
a, n = [int(x) for x in input().split()]
term_lst = [a]
for m in range(1,n):
    term = term_lst[-1] * 10 + a
    term_lst.append(term)
print(sum(term_lst))
```

循环体中，列表的最后一项是目前正在计算项的前一项

5.利用Python中的random模块生成1000个0到100之间的随机整数，并统计每个元素的出现次数。

---

- 思路：
1. 利用提示代码生成1000个整数
  2. 利用for循环遍历0-100这101个数进行count

答案：

```
import random
random.seed(0)
x=[random.randint(0,100) for i in range(1000)]
for i in range(101):
    print("{} {}".format(i,x.count(i)))
```

5.利用Python中的random模块生成1000个0到100之间的随机整数，并统计每个元素的出现次数。

---

如果只考虑出现了的元素，可以利用集合

```
import random
random.seed(0)
x=[random.randint(0,100) for i in range(1000)]
x_set = set(x)
for item in x_set:
    print(item,x.count(item))
```

## 6.输出国际象棋棋盘

这个题主要考察print的用法，见之前的ppt。

默认情况下，print的各个元素之间用一个空格分隔，如果想更改可以利用print(... ,...,sep="xxx")改变；print结尾默认是一个换行符，如果想更改可以用print(..., ..., end = "xxx")改变

Ex.以下print的结果是什么？

- ① print()                      一个换行符
- ② print('\n')                两个换行符
- ③ print(1,1,sep='0')        '101'加换行符
- ④ print(1,1,sep='0',end='2')  
                                  '1012'（无换行符!）

答案：

```
board = ""
flag = 1
for i in range(1,65):
    if i % 2 == flag:
        board += " "
    else:
        board += "*"
    if i % 8 == 0:
        board += "\n"
        if flag == 1:
            flag = 0
        else:
            flag = 1
print(board)
```

## 7. 按递增顺序依次列出所有分母为40，分子小于40的最简分数。

**思路：** 数i从1循环到39，判断40与i是否互质

Q: 如何判断两个数是否互质（求两数的最大公约数）

1. 辗转相除法，如果r是 $a \div b$ 的余数，则a和b的最大公约数也是b和r的最大公约数

```
a = ...; b = ...  
r = a % b  
while r != 0:  
    a = b; b = r  
    r = a % b  
#b是最大公约数
```

2. 穷举法

```
a = ...; b = ...  
max_factor = 1  
for i in range(2, min(a,b) + 1):  
    if a % i == 0 and b % i == 0:  
        max_factor = i  
#max_factor是最大公约数
```

7. 按递增顺序依次列出所有分母为40，分子小于40的最简分数。

答案：

```
true_frac = ""
for i in range(1,40):
    a = 40; b = i
    r = a % b
    while r != 0:
        a = b; b = r
        r = a % b
    if b == 1:
        true_frac += "{}/40,".format(i)
print(true_frac)
```

必须对a、b进行**初始化**，否则a、b的初始值为上个循环结束后的值。在资源充足的前提下，尽可能不要过于省变量！

可以用穷举法代替

也可以直接在这里进行  
`print("{}/40".format(i),end=",")`