



链滴

Python 学习之第二章 程序基本结构

作者: [xiaoqian](#)

原文链接: <https://ld246.com/article/1642166554819>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

<h2 id="第二章-程序基本结构">第二章 程序基本结构</h2>

<h3 id="2-1-顺序结构">2.1 顺序结构</h3>

<h4 id="2-1-1-程序的-IPO-模型">2.1.1 程序的 IPO 模型</h4>

<p>I: 输入

P: 处理

O: 输出</p>

<blockquote>

<p>例 1:输入两个数求平均值</p>

</blockquote>

```
<pre> <code class="highlight-chroma">num1 = input("输入第一个数:")
```

```
num2 = input("输入第二个数:")
```

```
avg_num = (float(num1)+float(num2))/2
```

```
print(avg_num)
```

```
</code></pre>
```

<h4 id="2-1-2-输入和输出">2.1.2 输入和输出</h4>

<blockquote>

<p>例 2.2 输入同学姓名和成绩，输出姓名和成绩的值及其值的类型</p>

</blockquote>

```
<pre> <code class="highlight-chroma">name = input("请输入姓名: ")
```

```
score = input("请输入成绩: ")
```

```
print(name, score)
```

```
print(type(name), type(score))
```

```
</code></pre>
```

<p>知识要点: </p>

<p>变量：变量是一个标识符，要遵循两个规则 1.只能由字母、数字和下划线组成 2.不能由数字开头</p>

<p>输入：input()函数，里面可以填写输入提示</p>

<p>数据类型：Python3 中支持 6 中数据类型:数字、str(字符串)、list(列表)、tuple(元组)、set(集合)、dict(字典)</p>

<p>数字类型：int(整型)、float(浮点型)、bool(布尔型)、complex(复数型)</p>

<p>输出：print()函数，</p>

<blockquote>

<p>例 2.3 假设标准体重(KG)的计算公式为:体重=[身高(厘米)-100]x0.9,请根据输入的身高计算一个的体重</p>

</blockquote>

```
<pre> <code class="highlight-chroma">length = int(input("请输入身高(厘米): "))
```

```
weight = (length-100)*0.9
```

```
print("身高为:%d厘米,标准体重为:%fkg" % (length, weight))
```

```
</code></pre>
```

<p>知识点: </p>

<p>类型转换：int(),float(),str()三种之间的互相转换，其中 int()转换是向下取整</p>

<p>格式化输出：和 C 语言中的格式化输入一样，保留小数点，占用列宽都一样


```
print("格式串"%(输出对象 1, 输出对象 2))</pre>
```

<p>巩固与拓展: </p>

<p> (1) 从键盘输入 3 名同学的数学成绩，输出平均分。要求平均分用 3 种形式输出:保留 2 位小的结果、四舍五入的结果（例如 3.62 得 4）和向下取整（例如 3.62 得 3）的结果</p>

```
<pre> <code class="highlight-chroma">score1 = int(input("第一位同学的成绩: "))
```

```
score2 = int(input("第二位同学的成绩: "))
```

```
score3 = int(input("第三位同学的成绩: "))
```

```
avg_score = (float(score1+score2+score3))/3
```

```
print("平均成绩:%.2f" % (avg_score)) #保留两位小数
```

```
print("平均成绩:%d" % (round(avg_score))) #四舍五入
```

```
print("平均成绩:%d" % (int(avg_score))) #向下取整
```

```
</code></pre>
```

<p> (2) 从键盘输入两个正整数 a 和 b, 输出 a 除以 b 的商和余数。 </p>

```
<pre><code class="highlight-chroma">a = int(input("请输入两个正整数:"))
b = int(input("请输入两个正整数:"))
print("商为:%d" % (divmod(a, b)[0]))
print("余数为:%d" % (divmod(a, b)[1]))
</code></pre>
```

<p> (3) 从键盘输入一个三位数的正整数, 输出其各位数字反转后得到的三位数 </p>

```
<pre><code class="highlight-chroma">num1 = input("请输入一个三位数:")
num2 = num1[::-1]
print(num2)
</code></pre>
```

<p> (4) 将 a 和 b 的值交换过来。例如:交换前:a = 2, b = 3; 交换后:a = 3, b = 2 </p>

```
<pre><code class="highlight-chroma">num1 = int(input("请输入一个整数:"))
num2 = int(input("请输入一个整数:"))
print("交换前:(%d,%d)" % (num1, num2))
t = num1
num1 = num2
num2 = t
print("交换后:(%d,%d)" % (num1, num2))
</code></pre>
```

<p> (5) 小明练习游泳, 有一天他从 a 时 b 分一直游泳到当天的 c 时 d 分 (时间按 24 小时制) 请算小明这天游泳的时长是多少小时多少分钟? </p>

```
<pre><code class="highlight-chroma">a, b = eval(input("开始游泳的时间: "))
c, d = eval(input("结束游泳的时间: "))
minutes = (c-a-1)*60+(60-b)+d
print("小明游了%d小时%d分钟" % (divmod(minutes, 60)[0], divmod(minutes, 60)[1]))
</code></pre>
```

2.2 分支结构 </h3>

2.2.1 单分支结构 </h4>

<blockquote>

<p>从键盘输入两个整数, 按从大到小的顺序输出 </p>

</blockquote>

```
<pre><code class="highlight-chroma">num1 = int(input("请输入一个整数:"))
num2 = int(input("请输入一个整数:"))
if num1 < num2:
    t = num1
    num1 = num2
    num2 = t
print("从大到小为:%d,%d" % (num1, num2))
</code></pre>
```

<p>知识点:</p>

<p>1.单分支——if 语句 </p>

<p>单分支结构代码块执行次数为 1 次或 0 次 </p>

<p>2.关系表达式 </p>

<p>表达式的值一定是布尔值(True 或 False) </p>

2.2.2 多分支结构 </h4>

<blockquote>

<p>例 2.5 如果年份能被 100 整除但不能被 4 整除或者能被 400 整除则是闰年, 否则, 不是闰年, 键盘输入一个年份, 判断其是不是闰年。 </p>

</blockquote>

```
<pre><code class="highlight-chroma">year = int(input("请输入一个年份:"))
if (year % 4 == 0 and year % 100 != 0) or year % 400 == 0:
    print("%d年是闰年" % (year))
else:
```

```
    print("%d年不是闰年" % (year))
```

```
</code></pre>
```

<p>知识点: </p>

<p>双分支——if...else 语句 两个都必须加:</p>

<p>逻辑运算符及表达式</p>

<p>Python 中逻辑运算符有三个(not>&and>&or)</p>

<blockquote> <p>例 2.6 从键盘输入一个字符串(全部由字母字符组成), 如果串的长度为 1, 则输出字符的 ASCLL , 如果串长度为 2, 则将字符重复 5 次并输出, 如果串长度为 3, 则将字符串中的所有字符大写并输 , 如果串长度大于 3, 统计子串'ab'在字符串中出现的次数</p> </blockquote> ``` <pre><code class="highlight-chroma">s = input("请输入一个字符串:") ``` ``` length = len(s) ``` ``` if length == 1: ``` ``` print(ord(s)) ``` ``` elif length == 2: ``` ``` print(s*5) ``` ``` elif length == 3: ``` ``` print(s.upper()) ``` ``` else: ``` ``` print(s.count('ab')) ``` ``` </code></pre> ``` <p>知识点: </p> order()函数: 返回字符 ASCLL 码 输出 5 次可以写成 s*5,其他次数也一样 将所有字母大写是 s.upper() 统计子串出现的次数用 s.count() <p>巩固与拓展</p> <p> (1) 编写程序, 计算长方形、圆和梯形的面积。程序运行时, 显示如下功能列表:
 a、计算长方形的面积
 b、计算圆的面积
 c、计算梯形面积
 例如, 输入 a, 计算长方形的面积。输入 b, 计算圆的面积。输入 c, 计算梯形面积。输入其他值则错。计算面积前, 根据需要输入图形的尺寸。比如计算长方形面积, 需要输入长和宽的值。计算圆的积, 要输入半径值。 </p> ``` <pre><code class="highlight-chroma">import math ``` ``` print("a、计算长方形的面积\nb、计算圆的面积\nc、计算梯形面积\n") ``` ``` s = input("请选择要使用的功能:") ``` ``` if s == 'a': ``` ``` x = float(input("请输入长方形的长:")) ``` ``` y = float(input("请输入长方形的宽:")) ``` ``` print("长方形的面积为:%.2lf" % (x*y)) ``` ``` elif s == 'b': ``` ``` r = float(input("请输入半径:")) ``` 原文链接: [Python 学习之第二章 程序基本结构](#)

```
print("半径为:%lf的圆的面积为:%.2lf" % (r, math.pi*r*r))
```

else:

```
x = float(input("请输入梯形的上底:"))
```

```
y = float(input("请输入梯形的下底:"))
```

```
z = float(input("请输入梯形的高:"))
```

```
print("该梯形的面积为:%.2lf" % ((x+y)*z/2))
```

```
</code></pre>
```

<p> (2) 输入三角形的三条边, 判断是否能构成三角形。 </p>

```
<pre><code class="highlight-chroma">a = float(input("请输入三角形的一个边:"))
```

```
b = float(input("请输入三角形的一个边:"))
```

```
c = float(input("请输入三角形的一个边:"))
```

```
if a+b > c and a+c > b and b+c > a:
```

```
    print("这三个边可以构成三角形")
```

else:

```
    print("这三个边不能构成三角形")
```

```
</code></pre>
```

<p> (3) 已知 2020 年 9 月 1 日是周二, 计算经过 n 天后是星期几, n 从键盘输入。 </p>

```
<pre><code class="highlight-chroma">print("2020 年 9 月 1 日是周二")
```

```
n = int(input("请输入n:"))
```

```
if n % 7 == 0:
```

```
    print("%d天后是周二" % (n))
```

```
elif n % 7 == 1:
```

```
    print("%d天后是周三" % (n))
```

```
elif n % 7 == 2:
```

```
    print("%d天后是周四" % (n))
```

```
elif n % 7 == 3:
```

```
    print("%d天后是周五" % (n))
```

```
elif n % 7 == 4:
```

```
    print("%d天后是周六" % (n))
```

```
elif n % 7 == 5:
```

```
    print("%d天后是周日" % (n))
```

else:

```
    print("%d天后是周一" % (n))
```

```
</code></pre>
```

<p> (4) 从键盘输入 2 个字符串, 将其拼接在一起, 去除串中的空格字符并输出。 </p>

```
<pre><code class="highlight-chroma">s1 = input("请输入一个字符串:")
```

```
s2 = input("请输入一个字符串:")
```

```
s3 = s1+s2
```

```
print(s3.replace(" ", ""))
```

```
</code></pre>
```

<p> (5) 从键盘输入一个字符串, 如果串中只有数字字符, 则将字符串转成整数输出, 如果串中只字母字符, 则将字符串中的首字母大写输出, 否则, 输出串的长度。 </p>

```
<pre><code class="highlight-chroma">s = input("请输入一个字符串:")
```

```
if s.isdecimal() is True:
```

```
    print(int(s))
```

```
elif s.isalpha() is True:
```

```
    print(s.capitalize())
```

else:

```
    print(len(s))
```

```
</code></pre>
```

2.3 循环结构</h3>

<p>在 python 中是没有 do while 的</p>

2.3.1 while 语句</h4>

```
<blockquote>
```

<p>例 2.7 假设程序运行时，需要输入密码，如果密码正确，显示“欢迎使用本系统”，否则，显示“c”，知道密码输入正确，结束程序。(密码设为:admin2021)</p>

</blockquote>

```
<pre><code class="highlight-chroma">password = "admin2021"
password1 = input("请输入密码:")
while password1 != password:
    password1 = input("密码错误，请重新输入!\n")
print("欢迎使用本系统")
</code></pre>
```

<p>知识点:</p>

<p>while 语句:</p>

<blockquote>

<p>while 条件: </p>

<p>代码块</p>

</blockquote>

<p>循环次数必须是有限的，不能死循环</p>

<p>break 和 continue 语句</p>

<p>break:结束所在层的所有循环</p>

<p>continue:结束所在层的一次循环，转而执行下一次循环</p>

<h4 id="2-3-2-for-循环语句">2.3.2 for 循环语句</h4>

<blockquote>

<p>例 2.8 假设登录一个系统需要输入密码，但只有 5 次机会。如果第 1 次输入不正确，显示“密码错误！还剩 4 次机会！”和“请重新输入:”，第 2 次还不对，显示“密码错误！还剩 3 次机会！”和“请重新输入:”。如果密码正确，显示“欢迎使用本系统！”，结束程序。如果 5 次输入都错误，则显示“密码错误，次数用完，请下次再试！”结束程序。(假设密码为 1234xyz) </p>

</blockquote>

```
<pre><code class="highlight-chroma">key = "1234xyz"
password = input("请输入密码:")
for i in range(1, 6):
    if password == key:
        print("欢迎使用本系统!")
        break
    else:
        if i &lt; 5:
            print("密码错误！还剩%d次机会！" % (5-i))
            password = input("请重新输入:")
        else:
            print("密码错误，次数用完，请下次再试！")
</code></pre>
```

</pre>

<p>知识点:</p>

<p>range()函数</p>

<p>语法格式:range(start=0, end, step=1)</p>

<p>for 语句</p>

<p>格式:for 变量 in 序列:</p>

<pre> <code class="highlight-chroma">代码块

</code> </pre>

<h4 id="2-3-3-循环的嵌套">2.3.3 循环的嵌套</h4>

<blockquote>

<p>例 2.9 利用 1、2、3.....、6 共六个数字组成一个 2 位数，个位只能取 1~3 共 3 个数字，能组成多少个无重复数字的 2 位数？输出这些 2 位数，每五个数一行。 </p>

</blockquote>

<pre> <code class="highlight-chroma">count = 0

for i in range(1, 7):

for j in range(1, 4):

if i != j:

print(i*10+j, end=" ")

count += 1

if count % 5 == 0:

print("")

print("总共有:%d个" % (count))

</code> </pre>

<p>巩固与拓展</p>

<p> (1) 输出 1 ~ 100 (包括) 之间能被 3 但不能被 7 整除的所有整数。 </p>

<pre> <code class="highlight-chroma">for i in range(1, 101):

if i % 3 == 0 and i % 7 != 0:

print(i, end=" ")

</code> </pre>

<p> (2) 编写程序，输出 100 ~ 2000 之间最大的 10 个素数。 </p>

<pre> <code class="highlight-chroma">sushu = []

for i in range(100, 2001):

for j in range(2, i):

if i % j == 0:

break

if i == j+1:

sushu.append(i)

print(sushu[-10:])

</code> </pre>

<p> (3) 求出两个整数 mn 之间所有整数各个位上的奇数数字之和。例如 m = 7, n = 12 则 mn 之间的所有整数是:7、8、9、10、11、12, 各个位上的奇数数字之和 7 + 9 + 1 + 1 + 1 + 1 = 20</p>

<pre> <code class="highlight-chroma">m = int(input("请输入一个整数:"))

n = int(input("请输入一个整数:"))

sum = 0

for i in range(m, n+1):

if i < 10:

if i % 2 != 0:

sum += i

else:

if i % 2 != 0:

sum += i % 10

i = i//10

if i % 2 != 0:

sum += i

else:

```

    i = i//10
    if i % 2 != 0:
        sum += i
print(sum)

```

(4) 韩信点兵。如果从 1 到 5 报数，最末一个兵报数为 1，从 1 到 6 报数，最末一个兵报数为 1，从 1 到 7 报数，最末一个士兵报数为 4，从 1 到 11 报数，最末一个兵报数为 10，请帮韩信计算他少有多少兵。

```

<code class="highlight-chroma">i = 0
while(1):
    if i % 5 == 1 and i % 6 == 5 and i % 7 == 4 and i % 11 == 10:
        print(i)
        break
    i += 1

```

算术游戏:

```

<code class="highlight-chroma">import sys
import random
for i in range(3):
    psd = input("请输入密码")
    if psd.strip(" ") == "123":
        break
    else:
        if(i == 2):
            print("密码错误3次,程序停止运行!")
            sys.exit(1)
        else:
            print("密码错误,请重新输入!")
signs = ['+', '-', '*', '//']
right = 0
error = 0
for i in range(10):
    op1 = random.randint(0, 100)
    op2 = random.randint(0, 100)
    sign = random.choice(signs)
    if sign == '+':
        print(str(op1)+sign+str(op2)+"=", end=' ')
        answer = int(input())
        if answer == op1+op2:
            print("恭喜你答对了!")
            right += 1
        else:
            print("很遗憾答错了!")
            print("正确答案是:%d" % (op1+op2))
            error += 1
    elif sign == '-':
        print(str(op1)+sign+str(op2)+"=", end=' ')
        answer = int(input())
        if answer == op1-op2:
            print("恭喜你答对了!")
            right += 1
        else:
            print("很遗憾答错了!")
            print("正确答案是:%d" % (op1-op2))

```



```
        error += 1
elif sign == '*':
    print(str(op1)+sign+str(op2)+"=", end=' ')
    answer = int(input())
    if answer == op1*op2:
        print("恭喜你答对了!")
        right += 1
    else:
        print("很遗憾答错了!")
        print("正确答案是:%d" % (op1*op2))
        error += 1
if sign == '/':
    print(str(op1)+sign+str(op2)+"=", end=' ')
    answer = int(input())
    if answer == op1//op2:
        print("恭喜你答对了!")
        right += 1
    else:
        print("很遗憾答错了!")
        print("正确答案是:%d" % (op1//op2))
        error += 1
print("答对的次数: ", right, "答错的次数: ", error)
</code></pre>
```