

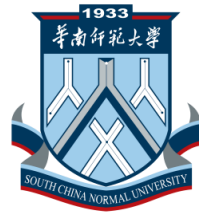


人工智能程序设计

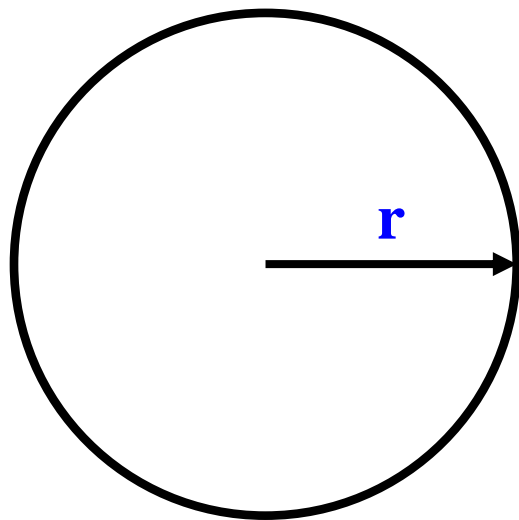
计算机学院

李晶晶

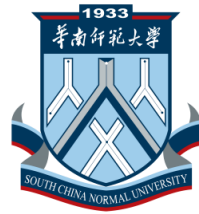
问题描述



请设计一个小程序，输入半径，输出周长。（取pi为3.14）



问题分析



- 分析问题的计算部分：采用周长公式解决计算问题。
- 确定功能：
 - 输入：半径
 - 处理：计算公式
 - 输出：周长
- 设计算法：
 - 周长公式为： $C=2\pi r$



涉及的基础知识一

- Python编码规范
- Python的输入和输出
- Python变量
- Python常用操作运算符

● 缩进

- 每一行代码开始前的空白区域，用来表示代码之间的包含和层次关系。
- 1个缩进 = 4个空格
 - ✓ 用以在Python中表明代码的层次关系
 - ✓ 缩进是Python语言中表明程序框架的唯一手段

Python编码规范



```
def toTxtFile(fn):  
    with open(fn, 'w') as fp:  
        for i in range(10):  
            if i%3==0 or i%7==0:  
                fp.write(str(i)+'\n')  
            else:  
                fp.write('ignored\n')  
        fp.write('finished\n')  
    print('all jobs done')  
toTxtFile('text.txt')
```

- # 函数定义
- # 函数体开始, 相对def缩进4个空格
- # with块开始, 相对with缩进4个空格
- # 选择结构开始, 再缩进4个空格
- # 语句块, 再缩进4个空格
- # 选择结构的第else分支, 与if对齐
- # for循环结构结束
- # with块结束
- # 函数定义结束, 调用函数

- 空行与空格

- 在每个类、函数定义或一段完整的功能代码之后增加一个空行
- 在运算符两侧各增加一个空格，逗号后面增加一个空格，让代码适当松散一点，不要过于密集

Python编码规范

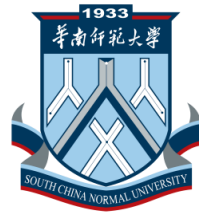


```
7      # 美元兑换人民币实时汇率
8      USD_VS_CNY = 6.6799
9
10     # 外币（美元）的输入
11     usd_str = input('请输入美元(USD)金额: ')
12
13     # 将字符串转换为数字
14     usd_value = eval(usd_str)
15
16     # 计算兑换人民币金额
17     cny_value = usd_value * USD_VS_CNY
18
19     # 输出人民币金额
20     print('人民币(CNY)金额是: ', cny_value)
21
```


● 标识符命名

- 必须以英文字母、汉字或下划线开头
- 名字中可以包含汉字、英文字母、数字和下划线，不能有空格或任何标点符号
- 不能使用保留字（关键字）
- 不建议使用系统内置的模块名、类型名或函数名以及已导入的模块名及其成员名作变量名或者自定义函数名
- 标识符对大小写敏感，python和Python是两个不同的名字

Python编码规范



- Python 3.x保留字列表 (33个)

- 查看方式

```
import keyword  
print(keyword.kwlist)
```

and	elif	import	raise
as	else	in	return
assert	except	is	try
break	finally	lambda	while
class	for	nonlocal	with
continue	from	not	yield
def	global	or	True
del	if	pass	False
			None

● 续行

- 尽量不要写过长的语句，应尽量保证一行代码不超过屏幕宽度
- 如果语句确实太长而超过屏幕宽度，最好在行尾使用续行符“\”表示下一行代码仍属于本条语句，或者使用圆括号把多行代码括起来表示是一条语句

Python编码规范



```
expression1 = 1 + 2 + 3\  
              + 4 + 5
```

#使用\作为

```
expression2 = (1 + 2 + 3  
              + 4 + 5)
```

**#把多行表达式放在
#圆括号中表示是一条语句**

- 注释

- 单行注释以#开头

- #Here are the comments

- 多行注释以'''或'''开头和结尾
'''

- This is a multiline comment
used in Python

- '''

Python输入与输出



● 输入函数

- 获得用户输入之前，input()函数可以包含一些提示性文字
- `<变量> = input(<提示性文字>)`

```
>>>input("请输入: ")
请输入: python
'python'
>>> input("请输入: ")
请输入: 1024.256
'1024.256'
```

Python输入与输出



- 输出函数

```
print(<expressions>)
```

```
print(<expression>, <expression>,.....<expression>)
```

```
>>>print('hello world!')
```

```
hello world!
```

```
>>> print("hello", "world!")
```

```
hello world!
```

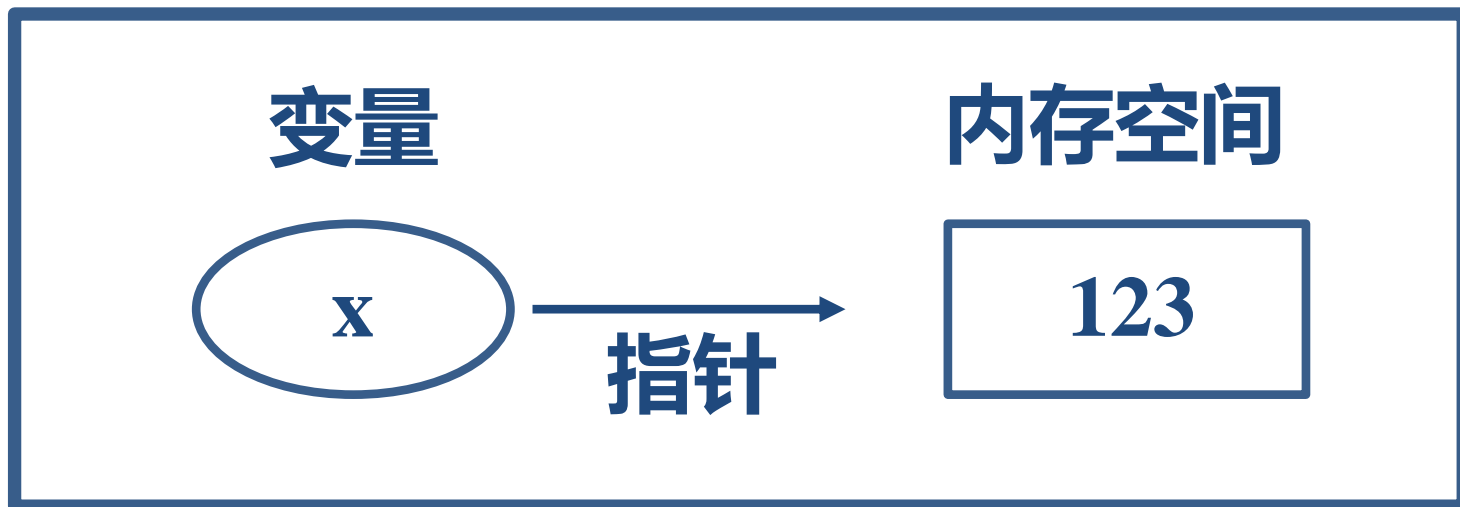
● 输出函数

- print()函数用来输出字符信息，或以字符形式输出变量
- print()函数可以输出各种类型变量的值
- print()函数通过%来选择要输出的变量

```
num1 = input("The first number is")
num2 = input("The second number is")
avg_num = (float(num1) + float(num2)) / 2
print("The average number is %f" % avg_num)
```


● 变量

- 变量不需要提前声明
- 一旦创建一个变量，就需要给该变量赋值



- 变量赋值 (=)

- 一个变量值赋给一个变量名
- 一个值同时赋给多个变量
- 几个变量同时赋值
- 变量交换

- 相互转化数值型变量

- int, float, bool, complex(real+J/j后缀)
- type() 显示数据类型
- 内置函数实现数值型数据类型的转换

Python变量



```
print(int(3.5))           # 获取实数的整数部分
print(int('119'))        # 把整数字符串转换为整数
print(int('1111', 2))    # 把1111看作二进制数，转换为十进制数
print(int(' 9\n'))       # 自动忽略字符串两个的空白字符
print(float('3.1415926')) # 把字符串转换为实数
print(complex(3, 4))      # 复数
print(complex(6j))
bool(1+23j)               # 把复数转换为bool类型
```

Python常用操作运算符



● 算数运算符

- 运算符用于在表达式中对一个或多个操作数进行计算并返回结果值

运算符	功能说明
+	算术加法，列表、元组、字符串合并与连接，正号
-	算术减法，集合差集，相反数
*	算术乘法，序列重复
/	真除法
//	求整商
%	求余数
**	幂运算

Python常用操作运算符



● +运算符

➤ +运算符除了用于算术加法以外，还可以用于列表、元组、字符串的连接。

```
print(3 + 5)
```

```
print(3.4 + 4.5)
```

```
print((3+4j) + (5+6j))
```

```
print('abc' + 'def')
```

```
print([1,2] + [3,4])
```

```
print((1,2) + (3,))
```

Python常用操作运算符



<例2.7: 加法运算>

a = 99999992222222222222999999999999999999922222222999999222299

b = 1234567890643337899876568997668866543345678997543356788

sum = a + b

print(sum)

a = 123.456789123456789111111

b = 123456789123456789123456789.0

print(a)

print(b)

int(a)

print(a)

Python常用操作运算符



● -运算符

- -运算符除了用于整数、实数、复数之间的算术减法和相反数之外，还可以计算集合的差集。
- 需要注意的是，在进行实数之间的运算时，有可能会出现误差。

```
print(7.9 - 4.5)
```

注意，结果有误差

```
print(5 - 3)
```

```
num = 3
```

```
print(-num)
```

```
print(--num)
```

注意，这里的--是两个负号，负负得正

```
print(-(-num))
```

与上一行代码含义相同

```
print({1,2,3} - {3,4,5})
```

计算差集

```
print({3,4,5} - {1,2,3})
```


Python常用操作运算符



● *运算符

➤ *运算符除了表示整数、实数、复数之间的算术乘法，还可用于列表、元组、字符串这几个类型的对象与整数的乘法，表示序列元素的重复，生成新的列表、元组或字符串。

```
print(33333 * 55555)
```

```
print((3+4j) * (5+6j))
```

```
print('重要的事情说三遍！' * 3)
```

```
print([0] * 5)
```

```
print((0,) * 3)
```

Python常用操作运算符



- 运算符/和//

- 在Python中分别表示真除法和求整商。在使用时，要特别注意整除运算符//“向下取整”的特点
- 例如， $-17 / 4$ 的结果是-4.25，在数轴上小于-4.25的最大整数是-5，所以 $-17 // 4$ 的结果是-5。

```
print(17 / 4)
```

```
print(17 // 4)
```

```
print((-17) / 4)
```

```
print((-17) // 4)
```

Python常用操作运算符



- %运算符

➤ %运算符可以用于求余数运算，还可以用于字符串格式化。在计算余数时，结果与%右侧的运算数符号一致。

```
print(365 % 7)
```

```
print(365 % 2)
```

```
print('%c, %c, %c' % (65, 97, 48)) #把65、97、48格式化为字符
```

Python常用操作运算符



● **运算符

- **运算符表示幂运算。
- 使用时应注意，该运算符具有右结合性，也就是说，如果有两个连续的**运算符，那么先计算右边的再计算左边的，除非使用圆括号明确修改表达式的计算顺序。

```
print(2 ** 4)
```

```
print(3 ** 3 ** 3)
```

```
print(3 ** (3**3))
```

```
print((3**3) ** 3)
```

```
print(9 ** 0.5)
```

```
print((-1) ** 0.5)
```

与上一行代码含义相同

使用圆括号修改计算顺序

计算9的平方根

对复数计算平方根得到复数

Python常用操作运算符



● 比较运算符

- 数值比较，字符比较
- 比较结果返回True，False

运算符	功能说明
<、<=、>、>=、==、!=	（值）大小比较，集合的包含关系比较

Python常用操作运算符



```
print(3+2 < 7+8)           # 关系运算符优先级低于算术运算符
print(3 < 5 > 2)           # 等价于3<5 and 5>2
print(3 == 3 < 5)         # 等价于3==3 and 3<5
print('12345' > '23456')  # 第一个字符'1'<'2', 直接得出结论
print('abcd' > 'Abcd')    # 第一个字符'a'>'A', 直接得出结论
print([85, 92, 73, 84] < [91, 82, 73])
# 第一个数字85<91, 直接得出结论
print([180, 90, 101] > [180, 90, 99])
# 前两个数字相等, 第三个数字101>99
print({1, 2, 3, 4} <= {3, 4, 5})
# 第一个集合不是第二个集合的子集
print([1, 2, 3, 4] > [1, 2, 3])
# 前三个元素相等, 并且第一个列表有多余的元素
```

Python常用操作运算符



- 赋值运算符

- 用于对变量的赋值和更新

运算符	功能说明
<code>+=</code> 、 <code>-=</code> 、 <code>*=</code> 、 <code>/=</code> 、 <code>%=</code> 、 <code>**=</code> 、 <code>//=</code>	<code>a+=b</code> 等效于 <code>a=a+b</code> 、 <code>a%=b</code> 等效于 <code>a=a%b...</code>

Python常用操作运算符

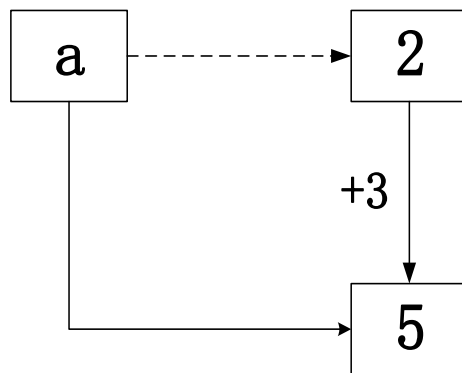


● 赋值运算符

➤ $a = a + 3$



(a) $a = 2$



(b) $a = a + 3$

Python常用操作运算符



- 逻辑运算符

运算符	功能说明
and、or、not	逻辑与、逻辑或、逻辑非

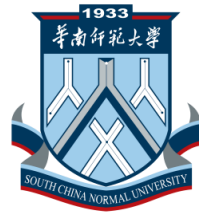
```
print(11 and 22)
```

```
print(11 or 22)
```

```
print(3-3 or 5-2)
```

```
print(not (11 and 22))
```

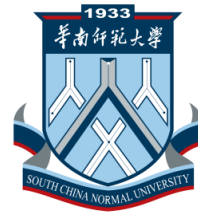
Python常用操作运算符



- 成员运算符
- 身份运算符
- 按位运算符

运算符	功能说明
in、 not in	成员测试
is、 is not	测试两个对象是否为同一个对象的引用
、 ^、 &、 <<、 >>、 ~	位或、位异或、位与、左移位、右移位、位求反

问题1



- 问题描述
- 问题分析
- 涉及基础知识：基本语法元素
- 程序实现
- 上机实践

问题1



- 演示代码

- 输入：半径
- 处理：计算公式
- 输出：周长

- 调试程序

```
# <问题1程序：求圆的面积>  
r = input("请输入半径:")  
pi = 3.14  
c = 2 * r * pi  
print("半径为", r, "的圆形", "其周长等于", c)
```



● 圆的计算

➤ 输入：圆的面积s

➤ 处理：圆的半径r和周长c

转化公式为： $r = \sqrt{s/\pi}$

其中， π 是常量，为3.14

round函数制定保留小数的位数

例如： $\text{round}(a, 2)$ 对a保留2位小数

➤ 输出：圆的半径和周长

上机实践(2)



通过表达式计算给定3个数值的均值、方差、标准差

均值、方差、标准差的计算公式如下所示。

$$\bar{x} = \frac{x_1 + x_2 + \cdots + x_n}{n}$$
$$s_N^2 = \frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2$$
$$s_N = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2}$$

\bar{x} 表示均值, s_N^2 表示方差, s_N 表示标准差

输入3个数值, 这里指定为11、2、5, 计算他们的均值、方差、标准差, 并打印输出。