

人工智能程序设计

python



```
import turtle
turtle.setup(650,350,200,200)
turtle.penup()
turtle.fd(-250)
turtle.pendown()
turtle.pensize(25)
turtle.pencolor("purple")
for i in range(4):
    turtle.circle(40, 80)
    turtle.circle(-40, 80)
    turtle.circle(40, 80/2)
    turtle.fd(40)
    turtle.circle(16, 180)
    turtle.fd(40 * 2/3)
```



# 人工智能程序设计

## 2.3 整数与浮点数

北京石油化工学院 人工智能研究院

刘 强

---

## 2.3 整数与浮点数

数字是编程中最基础的数据类型。Python提供了整数（int）和浮点数（float）两种数值类型，用于处理各种数学计算。

### # 正整数

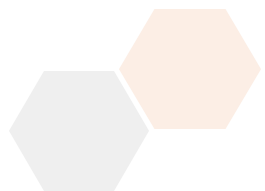
```
age = 25
```

```
population = 1400000000
```

### # 浮点数

```
price = 19.99
```

```
temperature = 36.5
```



## 2.3.1 整数类型

### 整数的表示

Python中的整数可以是正数、负数或零，没有大小限制：

# 正整数

age = 25

population = 1400000000

# 负整数

temperature = -10

debt = -5000

# 零

count = 0

## 2.3.1 整数类型

### 不同进制的整数

Python支持多种进制表示：

# 十进制（默认）

decimal = 100

# 二进制（0b开头）

binary = 0b1100100 # 二进制的100

# 八进制（0o开头）

octal = 0o144 # 八进制的100

# 十六进制（0x开头）

hexadecimal = 0x64 # 十六进制的100

print(decimal, binary, octal, hexadecimal) # 输出: 100 100 100 100

## 2.3.2 浮点数类型

### 浮点数的表示

浮点数用于表示带小数点的数值：

# 常规写法

price = 19.99

temperature = 36.5

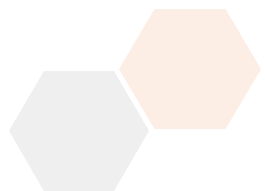
# 科学记数法

large\_number = 1.23e6 #  $1.23 \times 10^6 = 1230000$

small\_number = 4.56e-3 #  $4.56 \times 10^{-3} = 0.00456$

print(large\_number) # 输出: 1230000.0

print(small\_number) # 输出: 0.00456



## 2.3.2 浮点数类型

### 浮点数精度问题

浮点数在计算机中的存储可能存在精度误差：

```
result = 0.1 + 0.2
```

```
print(result)          # 输出: 0.30000000000000004
```

```
# 比较浮点数时要注意精度
```

```
print(0.1 + 0.2 == 0.3) # 输出: False
```

```
# 使用round()函数处理精度
```

```
rounded_result = round(0.1 + 0.2, 1)
```

```
print(rounded_result)   # 输出: 0.3
```

## 2.3.3 数值运算

### 基本算术运算

```
a = 10
```

```
b = 3
```

```
print(a + b) # 加法: 13
```

```
print(a - b) # 减法: 7
```

```
print(a * b) # 乘法: 30
```

```
print(a / b) # 除法: 3.3333333333333335
```

```
print(a // b) # 整除: 3
```

```
print(a % b) # 取余: 1
```

```
print(a ** b) # 幂运算: 1000
```



## 2.3.3 数值运算

### 运算优先级

数学运算遵循标准的优先级规则：

# 先乘除后加减

result1 = 2 + 3 \* 4    # 结果: 14 (不是20)

# 使用括号改变优先级

result2 = (2 + 3) \* 4    # 结果: 20

# 幂运算优先级最高

result3 = 2 \*\* 3 \* 4    # 结果: 32 (2^3 \* 4)

## 2.3.4 类型转换

### 数值类型转换

在程序中经常需要在不同数值类型间转换：

# 整数转浮点数

```
int_num = 42
```

```
float_num = float(int_num)
```

```
print(float_num)      # 输出：42.0
```

# 浮点数转整数（截断小数部分）

```
pi = 3.14159
```

```
int_pi = int(pi)
```

```
print(int_pi)         # 输出：3
```

## 2.3.4 类型转换

### 用户输入的类型转换

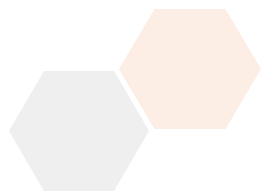
`input()`函数返回的总是字符串，需要转换为数值：

```
# 获取用户输入并转换为整数
```

```
age_str = input("请输入您的年龄：")
```

```
age = int(age_str)
```

```
print(f"明年您将{age + 1}岁")
```



## 2.3.4 类型转换

### 用户输入的类型转换

`input()`函数返回的总是字符串，需要转换为数值：

```
# 获取用户输入并转换为浮点数
```

```
height_str = input("请输入您的身高（米）：")
```

```
height = float(height_str)
```

```
print(f"您的身高是{height}米")
```

```
# 简化写法
```

```
score = int(input("请输入成绩："))
```

```
print(f"您的成绩是{score}分")
```

## 2.3.5 数值处理函数

**Python提供了常用的数值处理函数：**

# 绝对值

```
print(abs(-5))      # 输出： 5
```

# 四舍五入

```
print(round(3.14159)) # 输出： 3
```

```
print(round(3.14159, 2)) # 输出： 3.14
```

# 幂运算

```
print(pow(2, 3))      # 输出： 8 (等同于2**3)
```

## 2.3.6 Ask AI: 探索复数类型

Python还支持复数类型（complex），虽然在日常编程中使用较少，但在科学计算中很重要。当需要处理复数时，可以向AI助手寻求帮助。

### 1. 基础概念：

- "Python中的复数是什么？如何创建复数？"

### 2. 操作方法：

- "Python中如何进行复数运算？"

### 3. 应用场景：

- "复数在什么情况下会用到？"
- "如何用复数解决实际问题？"

# 实践练习

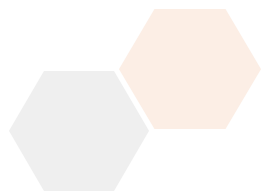
## 练习 2.3.1：体重指数计算

编写一个体重指数BMI计算程序，要求用户输入身高（米）和体重（千克），然后计算并输出BMI值和健康状况评价。(Ask AI：什么是体重指数BMI)

## 练习 2.3.2：类型转换练习

编写程序，要求用户输入一个数字（可能是整数或浮点数），然后：

1. 显示该数字的类型
2. 将其转换为另一种数值类型
3. 显示转换结果



# 实践练习

## 练习 2.3.3：温度转换

编写程序实现摄氏度和华氏度之间的转换：

- 华氏度 = 摄氏度  $\times 9/5 + 32$
- 摄氏度 = (华氏度 - 32)  $\times 5/9$

## 练习 2.3.4：购物金额计算

编写程序模拟购物结账，要求用户输入5个商品的价格，然后计算并输出：

1. 购物总金额
2. 商品平均价格
3. 最贵和最便宜商品的价格

