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")
turtle.seth(-40)
for i in range(4):
turtle.circle(40, 80)
turtle.circle(-40, 80)
turtle.fd(40)
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

第4章 程序的控制结构

北京石油化工学院 人工智能研究院

刘 强

第4章 程序的控制结构

控制结构通过**条件语句**实现程序的分支执行，通过**循环语句**实现代码的重复执行，通过**函数**实现代码的模块化和重用。

1、条件分支控制：学习if-elif-else语句实现条件判断，掌握比较运算符和逻辑运算符的使用，学习match语句进行模式匹配，培养逻辑思维能力。

2、循环重复执行：掌握while循环实现条件控制的重复执行，学习break和continue语句控制循环流程，理解循环的设计模式和常见陷阱。

3、随机数生成：学习random模块的使用方法，掌握生成随机整数、随机浮点数、随机选择等功能，了解随机数在程序中的应用场景。

4、函数模块化：掌握函数的定义和调用方法，学习参数传递和返回值的使用，理解函数的作用域，掌握将复杂问题分解为函数模块的能力。

4.1 条件控制语句if

if语句是Python中最基本的条件控制语句，它允许程序根据条件的真假来决定是否执行某段代码。

通过if语句，我们可以让程序具备判断能力，实现诸如"如果成绩大于60分就输出及格"、"如果年龄小于18岁就提示未成年"等逻辑判断。

```
if temperature > 25:
    print("今天很热，建议穿短袖")
    print("记得多喝水")
```

4.1 条件控制语句if

条件控制有三个核心要素：

1. 布尔表达式：能够求值为True或False的表达式，是条件判断的基础

2. 代码分支：根据条件执行不同的代码块，实现程序的分支逻辑

3. 缩进语法：使用缩进来标识代码块的范围，体现Python的简洁优雅

```
if temperature > 25:
    print("今天很热，建议穿短袖")
    print("记得多喝水")
```

4.1.1 布尔类型与布尔值

布尔类型（Boolean）是Python中最简单的数据类型之一，它只有两个值：True和 False。布尔类型在程序的逻辑判断和控制流程中起着关键作用。

```
## 布尔值的基本使用
is_student = True
is_working = False

print(is_student)  # 输出：True
print(is_working)  # 输出：False
print(type(is_student))  # 输出：<class 'bool'>
```

4.1.2 比较运算和逻辑运算

比较运算符：比较运算符用于比较两个值，返回布尔值

运算符	含义	示例
==	等于	5 == 5 → True
!=	不等于	5 != 3 → True
<	小于	3 < 5 → True
<=	小于等于	5 <= 5 → True
>	大于	5 > 3 → True
>=	大于等于	5 >= 5 → True

4.1.2 比较运算和逻辑运算

比较运算符：比较运算符用于比较两个值，返回布尔值

```
## 数值比较
age = 18
print(age >= 18)  # True
print(age == 18)  # True
print(age != 20)  # True
## 字符串比较
name = "Alice"
print(name == "Alice")  # True
print(name != "Bob")    # True
## 字符串按字典序比较
print("apple" < "banana")  # True
print("Python" > "Java")  # True
```

4.1.2 比较运算和逻辑运算

逻辑运算符：逻辑运算符用于组合多个布尔表达式：

运算符	含义	示例
and	逻辑与	True and False → False
or	逻辑或	True or False → True
not	逻辑非	not True → False

4.1.2 比较运算和逻辑运算

逻辑运算符：逻辑运算符用于组合多个布尔表达式：

```
## 逻辑运算示例
age = 25
has_license = True
has_car = False
## and运算：所有条件都为True时结果才为True
can_drive = age >= 18 and has_license
print(f"可以开车：{can_drive}")  # True
## or运算：任一条件为True时结果就为True
needs_transport = not has_car or age < 18
print(f"需要交通工具：{needs_transport}")  # True
## not运算：取反操作
is_minor = not (age >= 18)
print(f"未成年：{is_minor}")  # False
```

4.1.3 if语句的语法和用法

if语句的基本语法

```
if condition:
        # 当条件为True时执行的代码块
        statement1
        statement2
        ...
```

4.1.3 if语句的语法和用法

简单if语句

```
## 基本if语句示例
temperature = 30

if temperature > 25:
    print("今天很热，建议穿短袖")
    print("记得多喝水")

print("程序继续执行")  # 无论条件是否满足都会执行
```

4.1.3 if语句的语法和用法

if-else语句

```
## if-else语句
age = 16
if age >= 18:
    print("您已成年，可以投票")
else:
    print("您还未成年，不能投票")
    years_left = 18 - age
    print(f"还需要等待{years_left}年")
```

4.1.3 if语句的语法和用法

if-elif-else语句

```
## 多条件判断
score = 85
if score >= 90:
    grade = "A"
    print("优秀！")
elif score >= 80:
    grade = "B"
    print("良好！")
elif score >= 70:
    grade = "C"
    print("中等")
elif score >= 60:
    grade = "D"
    print("及格")
else:
    grade = "F"
    print("不及格，需要重修")

print(f"您的成绩等级是{grade}")
```

4.1.3 if语句的语法和用法

嵌套if语句：嵌套if语句是指在一个if语句的代码块中包含另一个if语句。这样可以实现更复杂的条件判断逻辑。

```
## 嵌套if语句示例
age = 20
has_ticket = True

if age >= 18:
    if has_ticket:
        print("您已成年且有票，可以入场观看")
    else:
        print("您已成年，但需要购票才能入场")
else:
    print("未成年人不能入场")
```

- 在实际编程中应避免过深的嵌套（如三层或更多），因为这会降低代码的可读性。
- 如果遇到复杂的条件判断：可以考虑使用逻辑运算符（`and`、`or`）来简化代码，或者将部分逻辑提取为独立的函数。

4.1.3 if语句的语法和用法

条件表达式（三元运算符）：Python提供了简洁的条件表达式语法：

```
    ## 传统if-else写法
    age = 20
    if age >= 18:
        status = "成年人"
    else:
        status = "未成年人"
## 使用条件表达式的简洁写法
status = "成年人" if age >= 18 else "未成年人"
print(status)
```

4.1.3 if语句的语法和用法

```
## 更多条件表达式示例

    max_value = a if a > b else b  # 求两数中的最大值

    abs_value = x if x >= 0 else -x  # 求绝对值

    result = "pass" if score >= 60 else "fail"  # 判断是否及格
```

实践练习

练习 4.1.1：年龄分类器

编写程序根据用户输入的年龄，判断其属于哪个年龄段。年龄段划分标准：

- 儿童：0-12岁
- 青少年：13-17岁
- 成年人：18-59岁
- 老年人：60岁及以上

要求使用if-elif-else语句，并输出友好的提示信息。例如："您今年25岁，属于成年人。"

实践练习

练习 4.1.2：闰年判断

编写程序判断输入的年份是否为闰年。

闰年的判断规则：

- 能被4整除但不能被100整除，或者
- 能被400整除

示例：

- 2020年是闰年（能被4整除但不能被100整除）
- 1900年不是闰年（能被100整除但不能被400整除）
- 2000年是闰年（能被400整除）