人工智能程序设计

python

人工智能程序设计

字符串

北京石油化工学院 人工智能研究院

刘 强

字符串

字符串是Python中最常用的数据类型之一，用于存储和处理文本信息。无论是用户输入、文件内容还是程序输出，字符串都扮演着重要角色。

```
title = "Python程序设计”
greeting = "欢迎学习Python"
```

2.2.1 字符串的创建

Python中可以使用单引号、双引号或三引号来创建字符串：

```
# 单引号
name = 'Python'
message = 'Hello World’
# 双引号
title = "Python程序设计"
greeting = "欢迎学习Python”
# 三引号（多行字符串）
description = """
Python是一种简洁、优雅的编程语言，
广泛应用于各个领域。
"""
```

2.2.1 字符串的创建

引号的选择原则

- 包含单引号时使用双引号：`text = "I'm learning Python"`
- 包含双引号时使用单引号：`text = 'He said "Hello"'`
- 多行文本使用三引号：适合长段落或文档字符串

在线练习

https://www.pythonguru.cn/python_ai_course/practice-2026-Fall/practice-PAP2.2.html

2.2.2 字符串的基本操作

字符串连接

使用加号（+）连接字符串：

```
first_name = "张"
last_name = "三"
full_name = first_name + last_name
print(full_name)  # 输出：张三

# 连接字符串和空格
greeting = "Hello" + " " + "Python"
print(greeting)   # 输出：Hello Python
```

2.2.2 字符串的基本操作

字符串重复

使用乘号（*）重复字符串：

```
separator = "-" * 20
print(separator)  # 输出：--------------------

emphasis = "重要！" * 3
print(emphasis)  # 输出：重要！重要！重要！
```

2.2.2 字符串的基本操作

字符串长度

使用 `len()`函数获取字符串长度：

```
text = "Python编程"
length = len(text)
print(f"字符串长度：{length}")  # 输出：字符串长度：8
```

2.2.3 字符串索引和切片

字符串索引

通过索引访问字符串中的单个字符：

```
text = "Python"
print(text[0])   # 输出：P（第一个字符）
print(text[1])   # 输出：y（第二个字符）
print(text[-1])  # 输出：n（最后一个字符）
print(text[-2])  # 输出：o（倒数第二个字符）
```

2.2.3 字符串索引和切片

字符串切片

使用切片获取字符串的子串：

```
text = "Python编程"
# 基本切片
print(text[0:3])    # 输出：Pyt
print(text[2:6])    # 输出：thon
print(text[:3])     # 输出：Pyt（从开始到索引3）
print(text[3:])     # 输出：hon编程（从索引3到结束）
# 带步长的切片
print(text[::2])    # 输出：Pto编（每隔一个字符）
print(text[::-1])   # 输出：程编nohtyP（反向）
```

2.2.4 字符串格式化

f-string格式化（推荐）

下面演示如何使用f-string将变量值嵌入到字符串中：

```
name = "张三"
age = 20
score = 95.5

# f-string格式化
message = f"姓名：{name}，年龄：{age}，成绩：{score}"
print(message)
```

2.2.4 字符串格式化

f-string格式化（推荐）

下面演示如何使用f-string将变量值嵌入到字符串中：

```
# 格式控制
price = 123.456
formatted_price = f"价格：{price:.2f}元"
print(formatted_price)  # 输出：价格：123.46元
```

2.2.5 字符串查找和替换

在文本处理中，经常需要查找特定内容或进行替换操作。

```
text = "Python是一种编程语言，Python很强大"

# 查找
print(text.find("Python"))      # 输出：0（第一次出现的位置）
print(text.count("Python"))     # 输出：2（出现次数）

# 替换
new_text = text.replace("Python", "Java")
print(new_text)  # 输出：Java是一种编程语言，Java很强大
```

2.2.6 字符串分割和连接

处理结构化文本数据时，分割和连接是常用操作。以下示例展示了如何处理以逗号分隔的数据：

```
# 分割字符串
text = "apple,banana,orange"
fruits = text.split(",")
print(fruits)  # 输出：['apple', 'banana', 'orange']

# 连接字符串
words = ["Python", "is", "awesome"]
sentence = " ".join(words)
print(sentence)  # 输出：Python is awesome
```

2.2.7 Ask AI：探索更多字符串方法

1. 功能导向的提问：

- "Python中如何将字符串转换为大写？"
- "怎样去除字符串两端的空格？"
- "如何判断字符串是否只包含数字？”

2. 问题导向的提问：

- "我想把'apple,banana,orange'分割成列表，应该用什么方法？"
- "如何检查字符串是否以特定内容开头？"
- "怎样替换字符串中的某个词？"

实践练习

练习 2.2.1：字符串基本操作

编写程序，要求用户输入姓名和年龄，然后输出格式化的欢迎信息。

练习 2.2.2：字符串处理

给定字符串"Hello, Python World!"，要求：

1. 去除首尾空格
2. 将所有字母转为小写
3. 替换"Python"为"Java"
4. 统计字符串长度

练习 2.2.3：文本分析

编写程序分析一段文本中单词的数量和最长单词的长度。