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 1 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

8.2 WEB数据展示：STREAMLIT

北京石油化工学院 人工智能研究院

刘 强

8.2 Web数据展示：Streamlit

Streamlit是一个专为数据科学家和机器学习工程师设计的Python库，它让我们能够用纯Python代码快速构建美观的Web应用，无需掌握HTML、CSS或JavaScript。

本节将学习如何使用Streamlit创建交互式的数据展示应用。

8.2.1 Streamlit简介与环境搭建

Streamlit的设计理念

Streamlit的核心理念是"代码即界面"（Code as UI），这意味着：

- 纯Python开发：只需要Python知识，无需前端技术
- 实时热重载：代码修改后应用自动更新
- 声明式编程：描述想要的结果，而非实现过程
- 组件化设计：提供文本、数据、图表、交互等多类内置组件，易于组合使用

8.2.1 Streamlit简介与环境搭建

环境搭建

首先安装Streamlit和相关依赖：

```
## 安装Streamlit
pip install streamlit

## 安装数据处理库
pip install pandas numpy

## 验证安装
streamlit hello
```

示例 8.2.1：数据展示Web应用

```
## hello_streamlit.py
import streamlit as st
import pandas as pd
import numpy as np

## 页面配置
st.set_page_config(
    page_title="我的第一个Streamlit应用",
    page_icon="🚀",
    layout="wide"
)

## 主标题
st.title("🚀 欢迎使用Streamlit")
st.write("这是一个用Python编写的Web应用！")
```

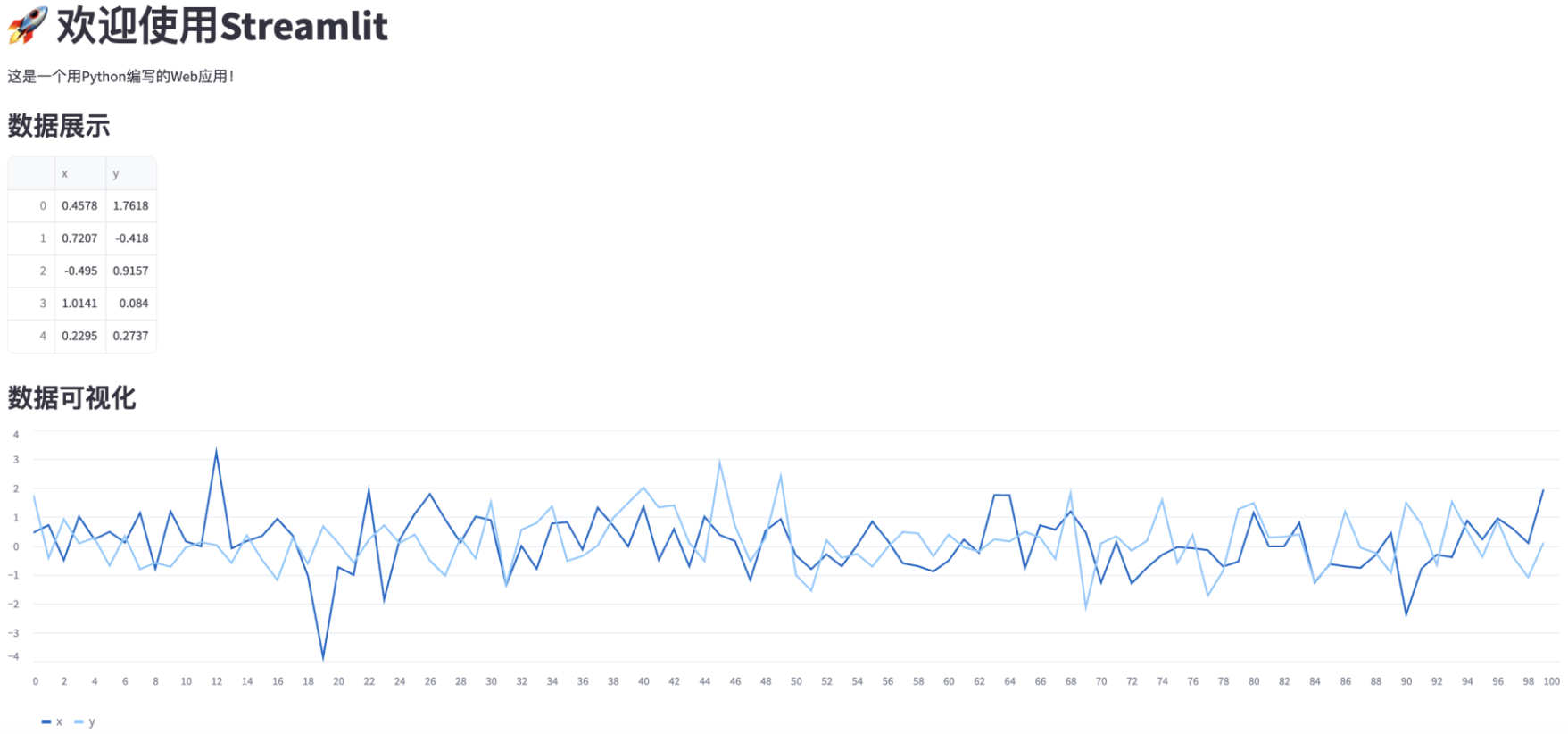
```
## 展示数据
st.subheader(“数据展示”)
data = pd.DataFrame({
        ‘x’ : np.random.randn(100),
        ‘y’ : np.random.randn(100)
})

## 显示数据表格的前5行
st.dataframe(data.head())

## 显示图表
st.subheader("数据可视化")
st.line_chart(data)
```

运行应用：

streamlit run hello_streamlit.py

示例 8.2.1：数据展示Web应用

浏览器会自动打开，显示我们的第一个Streamlit应用。



8.2.1 Streamlit简介与环境搭建

页面配置选项

页面配置是定制Streamlit应用外观的重要功能，通过st.set_page_config()函数可以设置页面标题、图标、布局等属性：

```
st.set_page_config(
    page_title="应用标题",          # 浏览器标签页标题
    page_icon="🎯",               # 浏览器标签页图标
    layout="wide",                # 页面布局：'centered' 或 'wide'
    initial_sidebar_state="expanded" # 侧边栏初始状态
)
```

8.2.2 核心组件与交互功能

文本显示组件

文本是Web应用中最基础的内容展示方式，Streamlit提供了多种文本显示组件：

标题层级组件用于创建不同级别的标题，st.title()创建页面主标题，st.header()和st.subheader()分别创建二级和三级标题，形成清晰的内容层次结构。

文本显示组件包括st.text()用于显示纯文本并保持原始格式，st.write()是万能显示函数可自动识别数据类型，st.markdown()支持Markdown格式的富文本显示。

代码和公式组件中，st.code()可以语法高亮显示代码片段，st.latex()支持显示LaTeX格式的数学公式。

8.2.2 核心组件与交互功能

文本显示组件

```
import streamlit as st
st.title("主标题")
st.header("二级标题")
st.subheader("三级标题")

st.text("这是纯文本")
st.write("这是通用显示函数")
st.markdown("**粗体** 和 *斜体* Markdown文本")

st.code("print('Hello, World!')", language='python')
st.latex(r'\sum_{i=1}^n x_i^2')
```

8.2.2 核心组件与交互功能

数据显示组件

数据展示是Streamlit的核心优势，提供了直观的数据呈现方式：

表格显示组件 st.dataframe()创建交互式数据表格，用户可以对数据进行排序和筛选，非常适合展示结构化数据。

指标显示组件 st.metric()用于展示关键数值指标，可以显示当前值和变化量，常用于仪表板中突出重要数据。

8.2.2 核心组件与交互功能

数据显示组件

```
import pandas as pd
import numpy as np

df = pd.DataFrame({
    '姓名': ['张三', '李四', '王五'],
    '年龄': [25, 30, 35],
    '薪资': [8000, 12000, 15000]
})

st.dataframe(df)
st.metric("平均薪资", f"{df['薪资'].mean():.0f}元")
```

8.2.2 核心组件与交互功能

交互控件

Streamlit提供了多种交互控件，让用户与应用进行实时交互：

按钮控件 st.button()创建可点击的按钮，通过条件判断可以响应用户的点击操作，常用于触发特定的功能或计算。

选择控件 st.selectbox()创建下拉选择框，用户可以从预设的选项中选择一个值，适合提供有限的选择选项。

滑块控件 st.slider()创建数值滑块，用户可以通过拖拽来选择数值范围内的值，直观且易于操作。

输入控件 st.text_input()创建文本输入框，用户可以输入自定义的文本内容，适合收集用户的个性化输入。

8.2.2 核心组件与交互功能

交互控件

```
if st.button("点击我"):
    st.write("按钮被点击了！")

option = st.selectbox(
    "请选择编程语言",
    ["Python", "JavaScript", "Java"]
)
st.write(f"你选择了：{option}")

age = st.slider("选择年龄", min_value=18, max_value=80, value=25)

name = st.text_input("请输入姓名")
if name:
    st.write(f"你好，{name}！")
```

8.2.2 核心组件与交互功能

图表和可视化

数据可视化是数据分析中的重要环节，Streamlit内置了多种图表组件：

折线图组件 st.line_chart()用于显示数据随时间的变化趋势，适合展示连续数据的走势。

柱状图组件 st.bar_chart()用于比较不同类别的数值大小，通过柱子高度直观显示数据差异。

面积图组件 st.area_chart()结合了折线图和柱状图的特点，用填充区域展示数据量的变化。

8.2.2 核心组件与交互功能

图表和可视化

```
import pandas as pd
import numpy as np

chart_data = pd.DataFrame({
    '日期': pd.date_range('2024-01-01', periods=30),
    '销售额': np.random.randint(1000, 5000, 30),
    '访问量': np.random.randint(100, 1000, 30)
})

st.line_chart(chart_data.set_index('日期')['销售额'])
st.bar_chart(chart_data.set_index('日期')['访问量'])
st.area_chart(chart_data.set_index('日期'))
```

8.2.3 运行Streamlit应用

在命令行中运行Streamlit应用：

streamlit run app.py

执行此命令后，Streamlit会启动本地开发服务器，并自动在浏览器中打开应用。默认情况下，应用运行在http://localhost:8501地址。

8.2.4 Ask AI：深入学习Streamlit

掌握了基础功能后，可以向AI助手询问更多高级特性：

- "如何在Streamlit中实现多页面应用？"
- "如何使用st.session_state管理应用状态？"
- "如何实现文件上传和下载功能？"
- "如何自定义Streamlit应用的布局和样式？"
- "如何将Streamlit应用部署到云端？"
- "如何优化Streamlit应用的性能？"

实践练习

练习 8.2.1：学生成绩分析系统

创建一个综合性的学生成绩分析系统，要求：

1. 数据输入：使用st.text_area()让用户输入多个学生的成绩数据（格式：姓名,语文,数学,英语，每行一个学生）

2. 数据展示：使用st.dataframe()展示成绩表，并计算每个学生的总分和平均分

3. 统计指标：使用st.metric()展示全班的平均分、最高分、最低分

4. 交互筛选：使用st.selectbox()选择科目，动态显示该科目的成绩柱状图

5. 及格分析：使用st.slider()设置及格线（0-100），统计并显示及格人数和及格率

实践练习

练习 8.2.2：交互式数据筛选器

开发一个具有筛选功能的数据展示应用，要求：

1. 数据准备：创建或生成包含商品信息的数据集（商品名称、类别、价格、销售日期、销量）

2. 侧边栏筛选：在侧边栏添加多个筛选条件：

- 价格区间（使用st.slider()）

- 类别选择（使用st.multiselect()）

- 日期范围（使用st.date_input()）

3. 动态更新：根据筛选条件实时更新数据表和图表

4. 可视化展示：显示筛选后的销售额趋势图和类别销量柱状图

5. 统计摘要：展示筛选后的总销售额、平均价格等关键指标

实践练习

练习 8.2.3：个人理财计算器

构建一个实用的理财计算工具，功能包括：

1. 收支记录：使用表单组件让用户输入收入和支出项目（金额、类别、日期）
2. 数据存储：使用st.session_state保存多条记录
3. 收支统计：使用st.metric()展示总收入、总支出、结余金额
4. 可视化分析：

- 使用饼图或柱状图展示支出类别占比
- 使用折线图展示收支趋势

5. 交互功能：提供按钮清空所有记录、导出数据等功能