

# 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.color("purple")
for i in range(4):
    turtle.circle(40, 80)
    turtle.circle(-40, 80)
    turtle.circle(40, 80/2)
    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

## 12.2 深度学习框架PYTORCH

北京石油化工学院 人工智能研究院

刘 强

# PyTorch简介

在深度学习的实践中，选择合适的框架至关重要

- PyTorch是当前最受欢迎的深度学习框架之一
- 以其直观的设计理念和强大功能赢得了学术界和工业界的广泛认可



## 12.2.1 PyTorch简介与环境配置

### 学习内容：

- PyTorch的动态计算图和Python原生设计
- 安装配置方法 (CPU版本和GPU版本)
- 推荐的开发环境和工具



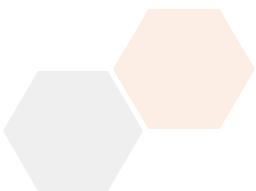
# PyTorch的独特优势

## 动态计算图

- 在运行时构建计算图
- 网络结构可以根据需要动态改变
- 调试过程就像调试普通Python代码一样简单

## Python原生设计

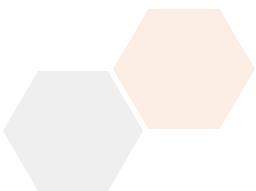
- API设计符合Python编程习惯
- 可以无缝集成NumPy、Pandas等常用库
- 学习成本相对较低



# PyTorch的社区支持

## 社区支持强大：

- 学术界的主流选择
- 大量研究论文提供PyTorch实现
- 在工业界也有广泛应用
- 保证了框架的持续发展和稳定性



# 环境配置与安装

安装PyTorch的过程相对简单，需要根据系统配置选择合适的版本：

```
## CPU版本（适合学习和小规模实验）
pip install torch torchvision torchaudio
```

```
## GPU版本（需要NVIDIA显卡和CUDA支持）
pip install torch torchvision torchaudio --index-url https://download.pytorch.org/whl/cu118
```

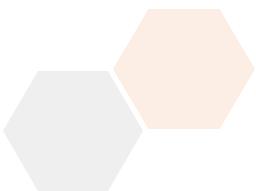


# 开发环境建议

## 推荐安装的开发工具：

- **Jupyter Notebook**: 交互式开发环境
- **matplotlib**: 数据可视化
- **tqdm**: 显示训练进度

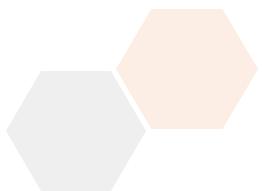
```
pip install jupyter matplotlib tqdm pandas scikit-learn
```



## 12.2.2 Ask AI：深度学习框架选择

### 学习内容：

- 主流框架对比：PyTorch、TensorFlow、JAX等
- PyTorch与TensorFlow的优劣势分析
- 不同场景下的框架选择建议



# 深度学习框架的选择考量

除了PyTorch之外，还有多个优秀框架可供选择：

框架	特点
PyTorch	动态计算图，Python原生设计，学术界主流
TensorFlow	成熟的生产环境支持，完整生态系统
JAX	函数式编程，高性能计算
PaddlePaddle	国产框架，中文社区支持好



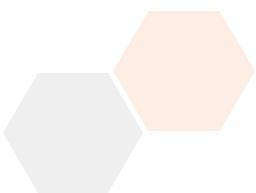
# PyTorch vs TensorFlow

## PyTorch的优势：

- 动态计算图和Python原生设计
- 模型开发和调试更加直观
- 适合研究和原型开发
- 学习曲线相对平缓

## TensorFlow的特点：

- 成熟的生产环境支持
- 完整的生态系统
- TensorFlow Lite支持移动设备部署
- TensorFlow.js支持Web端部署



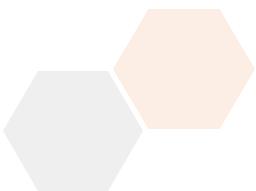
# 框架选择建议

**初学者和研究人员**: 推荐PyTorch

- API简洁直观
- 调试方便
- 学术论文代码多用PyTorch实现

**工业部署场景**: 考虑TensorFlow

- 成熟的部署工具链
- 多平台支持
- 大规模分布式训练支持



# 实践练习

## 练习 12.2.1：环境配置

1. 安装PyTorch (CPU或GPU版本)
2. 验证安装是否成功 (运行`import torch; print(torch.__version__)`)
3. 安装配套工具 (Jupyter Notebook、matplotlib等)



# 实践练习

## 练习 12.2.2：框架对比分析

分析以下场景应该选择PyTorch还是TensorFlow，并说明理由：

1. 学术研究中快速验证新算法
2. 将模型部署到手机App
3. 团队成员都是Python初学者
4. 需要在浏览器中运行深度学习模型

