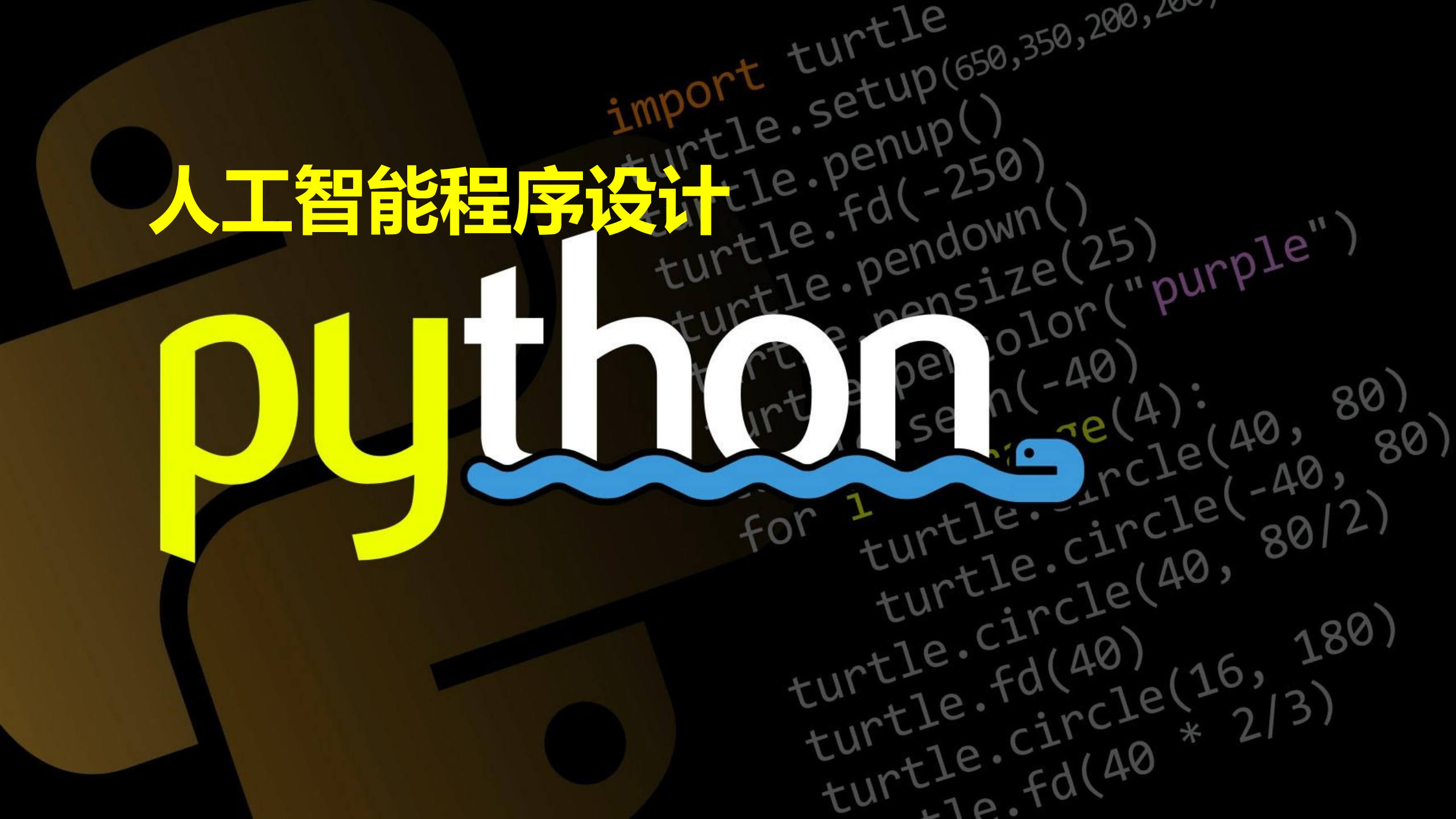
人工智能程序设计


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

第3章 列表与元组

北京石油化工学院 人工智能研究院

刘 强

第3章 列表与元组

单一变量与数据集合的差异

使用单一变量存储数据时，每个数据项都需要一个独立的变量名

列表和元组提供了集合式的数据管理能力，可以在一个变量中存储多个数据项，

通过索引访问任意元素。

列表支持动态增删改，元组保证数据不可变。

配合循环结构，可以高效处理批量数据

3.1 创建与访问列表

列表（List）是Python中最重要的数据结构之一，它允许我们在一个变量中存储多个数据项。列表可以存储任意数量的数据，并且这些数据可以是不同类型的。

```
numbers = [1, 2, 3, 4, 5]
names = ["张三", "李四", "王五"]
mixed = [1, "Python", 3.14, True]
```

3.1.1 列表的创建

使用方括号创建列表

最常见的创建列表的方法是使用方括号，并用逗号分隔各个元素

```
numbers = [1, 2, 3, 4, 5]
names = ["张三", "李四", "王五"]
mixed = [1, "Python", 3.14, True]
empty = []  # 空列表

print(numbers)  # 输出：[1, 2, 3, 4, 5]
print(names)    # 输出：['张三', '李四', '王五']
print(mixed)    # 输出：[1, 'Python', 3.14, True]
```

3.1.1 列表的创建

使用list()函数创建列表

从其他可迭代对象创建列表

```
text_list = list("Python")
print(text_list)  # 输出：['P', 'y', 't', 'h', 'o', 'n']
```

从range对象创建列表

```
number_list = list(range(1, 6))
print(number_list)  # 输出：[1, 2, 3, 4, 5]
```

3.1.2 访问列表元素

使用索引访问

列表中的每个元素都有一个位置索引，从0开始计数：

```
fruits = [“苹果”，“香蕉”，“橙子”，“葡萄”]

## 正向索引
print(fruits[0])  # 输出：苹果（索引0）
print(fruits[1])  # 输出：香蕉（索引1）

## 负向索引
print(fruits[-1])  # 输出：葡萄（索引-1，最后一个元素）
print(fruits[-2])  # 输出：橙子（索引-2，倒数第二个元素）
```

3.1.3 列表切片

切片允许我们获取列表的一部分：

```
numbers = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
## 基本切片
print(numbers[2:5])    # 输出：[2, 3, 4]
print(numbers[:3])     # 输出：[0, 1, 2]（从开头到索引3）
print(numbers[7:])     # 输出：[7, 8, 9]（从索引7到结尾）
print(numbers[:])      # 输出：[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]（完整列表）
## 带步长的切片
print(numbers[::2])    # 输出：[0, 2, 4, 6, 8]（每隔一个元素）
print(numbers[::-1])   # 输出：[9, 8, 7, 6, 5, 4, 3, 2, 1, 0]（反向）
```

3.1.4 列表的基本操作

获取列表长度

使用len()函数可以获取列表中元素的数量：

```
colors = ["红", "绿", "蓝", "黄"]
print(len(colors))  # 输出：4

## 空列表的长度
empty_list = []
print(len(empty_list))  # 输出：0
```

3.1.4 列表的基本操作

检查元素是否存在

使用in和not in关键词可以检查元素是否在列表中：

```
fruits = ["苹果", "香蕉", "橙子"]


print("苹果" in fruits)    # 输出：True
print("西瓜" in fruits)    # 输出：False
print("香蕉" not in fruits)  # 输出：False
```

3.1.4 列表的基本操作

列表连接和重复

```
## 列表连接
list1 = [1, 2, 3]
list2 = [4, 5, 6]
combined = list1 + list2
print(combined)  # 输出：[1, 2, 3, 4, 5, 6]

## 列表重复
repeated = [0] * 5
print(repeated)  # 输出：[0, 0, 0, 0, 0]
```

实践练习

练习 3.1.1：创建个人信息列表

创建一个包含姓名、年龄、城市的列表，并访问和打印每个元素。

练习 3.1.2：列表切片练习

给定列表 [10, 20, 30, 40, 50, 60, 70, 80, 90]，使用切片获取：

1. 前三个元素
2. 后三个元素
3. 中间的元素（索引2到6）
4. 每隔一个元素

练习 3.1.3：购物清单

创建一个购物清单列表，检查特定商品是否在清单中，并计算清单中商品的总数量。