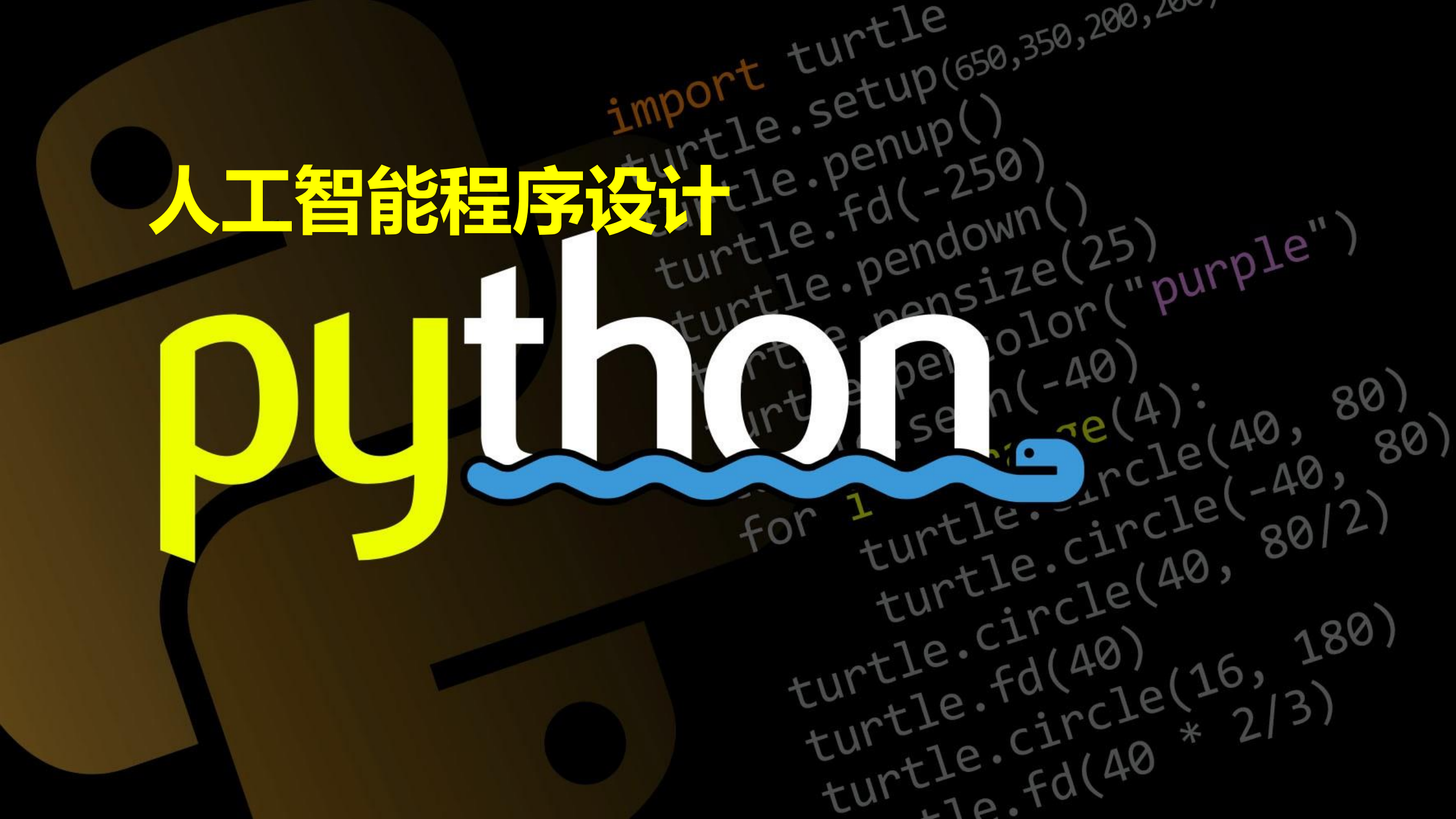
人工智能程序设计


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

3.3 FOR循环遍历列表

北京石油化工学院 人工智能研究院

刘 强

3.3 for循环遍历列表

Python的for循环是专门为遍历序列设计的循环结构

它能够自动从列表中取出每个元素并执行相应的操作。相比其他循环方式，for循环的语法更加简洁直观，特别适合处理列表、字符串、元组等序列类型的数据。

遍历（iteration）提供了一种自动化的方式来依次访问列表中的每个元素。

3.3.1 for循环

for循环的基本语法

for循环的基本语法结构为：

```
for 变量 in 序列:
    循环体
```

for循环会自动从序列中取出每个元素，赋值给循环变量，然后执行循环体中的代码。

3.3.1 for循环

简单遍历

最基本的列表遍历方式是直接遍历列表中的元素：

```
fruits = ["苹果", "香蕉", "橙子", "葡萄"]
for fruit in fruits:
    print(f"我喜欢吃{fruit}")
## 输出：
## 我喜欢吃苹果
## 我喜欢吃香蕉
## 我喜欢吃橙子
## 我喜欢吃葡萄
```

3.3.1 for循环

数值列表遍历

```
numbers = [1, 2, 3, 4, 5]
total = 0

for number in numbers:
    total += number
    print(f"当前数字：{number}，累计和：{total}")

print(f"最终总和：{total}")
```

3.3.2 遍历中的操作

条件判断

在遍历过程中可以结合条件语句：

```
scores = [85, 92, 78, 96, 88, 73, 90]
print("优秀学生（90分以上）：")
for score in scores:
    if score >= 90:
        print(f"分数：{score}")
print("\n需要补考的学生（75分以下）：")
for score in scores:
    if score < 75:
        print(f"分数：{score}")
```

3.3.2 遍历中的操作

遍历时修改列表

在for循环遍历列表的过程中，不要直接修改正在遍历的列表（如添加、删除元素），这会导致索引混乱和意外结果。正确的做法是创建新列表来存储处理后的数据：

```
original_prices = [100, 200, 150, 300, 250]
discounted_prices = []
## 计算打折后的价格
for price in original_prices:
    discounted_price = price * 0.8  # 8折
    discounted_prices.append(discounted_price)
print(f"原价格：{original_prices}")
print(f"折后价格：{discounted_prices}")
```

3.3.3 enumerate()函数

当需要在遍历时获取元素的索引时，使用enumerate()函数：

```
colors = ["红", "绿", "蓝", "黄"]
for index, color in enumerate(colors):
    print(f"索引{index}：{color}")
## 输出：
## 索引0：红
## 索引1：绿
## 索引2：蓝
## 索引3：黄
```

3.3.4 zip()函数与并行遍历

zip()函数用于将多个列表"配对"，实现并行遍历。它会将对应位置的元素组合在一起，形成元组。

```
employees = ["张华", "李梅", "王强", "赵敏"]
sales = [850, 920, 780, 960]  # 销售额（千元）
print("销售业绩报告：")
for salesperson, sale in zip(employees, sales):
    level = "优秀" if sale >= 900 else "良好" if sale >= 800 else "达标"
    print(f"{salesperson}：{sale}千元（{level}）")

## 输出：
## 销售业绩报告：
## 张华：850千元（良好）
## 李梅：920千元（优秀）
## 王强：780千元（达标）
## 赵敏：960千元（优秀）
```

实践练习

练习 3.3.1：分数转换

给定一个百分制分数列表，遍历列表并将每个分数转换为五分制分数。转换规则：90-100→5分，80-89→4分，70-79→3分，60-69→2分，0-59→1分

练习 3.3.2：销售数据统计

有两个列表分别存储销售员姓名和销售额，使用zip()函数遍历并统计：

1. 达标销售员数量（≥10000元）
2. 优秀销售员姓名（≥20000元）

练习 3.3.3：商品库存管理

给定商品名称、价格和库存数量三个列表，遍历并输出：

1. 库存不足的商品（数量<10）
2. 总库存价值