

《Python Web 开发基础》教学大纲

目录

- 1. 大纲说明
- 2. 教学设计
- 3. 教学内容细化

1 大纲说明

《Python Web 开发基础》课程是河北师范大学软件学院软件工程专业机器学习方向的专业基础课，本教学大纲适用于相应专业方向的本科生教学。

学习本课程之前，学生应具备一定的数据结构、算法、数据库原理和计算机网络基础知识，在相应专业课程中成绩优良。

通过本课程的学习，要求学生达到下列基本目标：

1. 掌握简单的Python数据类型，以及基本的数学和文本操作
2. 掌握用Python内置的数据结构来处理数据
3. 掌握Python的代码结构和函数的用法
4. 可以使用模块和包编写大规模Python程序
5. 深入理解对象、类和其他面向对象特性
6. 学会使用普通文件、关系数据库和NoSQL数据库来存储数据
7. 管理系统任务，比如程序、进程和线程
8. 理解并发和网络编程的基础知识

2 教学设计

本课程在教学上采用理论与实践相结合的授课方式。理论部分通过多媒体课件来讲解基本的概念、思想、方法和原理，实践部分以课上和课下作业任务方式来使学生巩固理论知识并提高其实际编程能力。

1. 教学内容及学时分配

序号	主要内容	学时分配	作业量
1	Python语言初识	1	0
2	变量和数据类型	3	1
3	Python基本数据结构	4	2
4	代码结构	4	1

序号	主要内容	学时分配	作业量
5	模块、包和程序	3	1
6	对象和类	6	2
7	数据编码与处理	4	1
8	文件IO与数据库	6	1
9	系统管理	4	1
10	Web开发基础	4	1
11	并发和网络	8	2
12	Python开发最佳实践	3	0
合计		50	13

2. 课程考核

考核采用期末闭卷笔试的方式。本课程成绩由两部分构成：

- 1. 平时成绩：40分（作业、考勤、综合表现等）
- 2. 期末考试成绩：60分

3 教学内容细化

1. Python语言初识

1. 主要内容

- 真实世界中的Python
- Python与其他编程语言
- 为什么选择Python
- Python不擅长的
- Python 2与Python 3
- 安装、运行Python
- Python之禅

2. 教学目标

通过一些真实的Python程序来了解这门语言的概貌、能力以及在真实世界中的用途。

3. 重点难点

- Python开发环境配置
- 使用交互式解释器

2. 变量和数据类型

1. 主要内容

- 变量、名字和对象
- 数字

- 字符串和文本

2. 教学目标

了解掌握Python中最简单的数据类型：布尔值、整数、浮点数和文本字符串，学习基础的数学和文本操作。

3. Python基本数据结构

1. 主要内容

- 列表
 - 创建列表
 - 将其他数据类型转换成列表
 - 获取元素
 - 包含列表的列表
 - 修改元素
 - 指定范围并使用切片提取元素
 - 添加元素至尾部
 - 合并列表
 - 在指定位置插入元素
 - 删除指定位置的元素
 - 删除具有指定值的元素
 - 获取并删除指定位置的元素
 - 查询具有特定值的元素位置
 - 判断值是否存在
 - 记录特定值出现的次数
 - 拼接为字符串
 - 重新排列元素
 - 获取长度
 - 复制列表
- 元组
 - 创建元组
 - 元组与列表
- 字典
 - 创建字典
 - 转换为字典
 - 添加或修改元素
 - 合并字典
 - 删除具有指定键的元素
 - 删除所有元素

- 判断是否存在
- 获取元素
- 获取所有键
- 获取所有值
- 获取所有键值对
- 复制字典
- 集合
 - 创建集合
 - 将其他类型转换为集合
 - 测试值是否存在
 - 合并及运算符
- 数据结构比较
- 大型数据结构

4. 代码结构

1. 主要内容

- 控制流 if 、 elif 和 else
- 循环
 - while 循环
 - 使用 break 跳出循环
 - 使用 continue 跳到循环开始
 - 循环外使用 else
 - for 循环
 - 使用 break 跳出循环
 - 使用 continue 跳到循环开始
 - 循环外使用 else
 - 使用 zip() 并行迭代
 - 使用 range() 生成自然数序列
 - 其他迭代方式
- 推导式
 - 列表推导
 - 字典推导
 - 集合推导
 - 生成器推导
- 函数
 - 位置参数
 - 关键字参数

- 指定默认参数值
- 使用 * 收集位置参数
- 使用 ** 收集关键字参数
- 文档字符串
- 一等公民
- 内部函数
- 闭包
- 匿名函数: lambda() 函数
- 迭代器
- 生成器
- 装饰器
- 命名空间和作用域
- 异常处理
 - 使用 try 和 except 处理异常
 - 编写自己的异常

5. 模块、包和程序

1. 主要内容

- 独立程序
- 命令行参数
- 模块和 import 语句
 - 导入模块
 - 使用别名导入模块
 - 导入模块的一部分
 - 模块搜索路径
- 包
- Python标准库
 - 处理缺失键
 - 计数
 - 有序字典
 - 双端队列: 栈、队列
 - 使用 itertools 迭代代码结构
 - 使用 pprint() 友好输出

6. 对象和类

1. 主要内容

- 什么是对象
- 使用 class 定义类

- 继承
- 覆盖方法
- 添加新方法
- 使用 `super` 从父类得到帮助
- `self` 参数
- 使用属性对特性进行访问和设置
- 使用名称重整保护私有特性
- 方法的类型
- 鸭子类型
- 特殊方法
- 组合
- 何时使用类和对象而不是模块

7. 数据编码与处理

1. 主要内容

- 文本字符串
 - Unicode
 - 格式化
 - 正则表达式
- 二进制数据
 - 字节和字节数组
 - 使用 `struct` 转换二进制数据
 - 其他二进制数据工具
 - 使用 `binascii()` 转换字节 / 字符串
 - 位运算符

8. 文件IO与数据库

1. 主要内容

- 文件输入 / 输出
 - 写入文本文件
 - 读取文本文件
 - 写入二进制文件
 - 读取二进制文件
 - 自动关闭文件
- 结构化的文本文件
- 结构化二进制文件
- 关系型数据库
 - SQL

- DB-API
- SQLite
- NoSQL数据存储

9. 系统管理

1. 主要内容

- 文件管理
 - 创建文件
 - 检查文件是否存在
 - 检查是否为文件
 - 复制文件
 - 重命名文件
 - 创建链接
 - 修改权限
 - 修改所有者
 - 获取路径名
 - 获取符号的路径名
 - 删除文件
- 目录管理
 - 创建目录
 - 删除目录
 - 列出目录内容
 - 修改当前目录
 - 列出匹配文件
- 日期和时间
 - **datetime**模块
 - **time**模块
 - 读写日期和时间
 - 其他模块
- 程序和进程
 - 使用**subprocess**创建进程
 - 使用**multiprocessing**创建进程
 - 使用 `terminate()` 终止进程

10. Web开发基础

1. 主要内容

- Web客户端
 - 标准Web库

- **requests**

- Web服务端
 - 最简单的Python Web服务器
 - Web服务器网关接口
 - 框架
- Web服务和自动化

11. 并发和网络

1. 主要内容

- 并发
 - 队列
 - 进程
 - 线程
 - 生成器和协程
 - asyncio
- 网络
 - Web服务和API

12. Python开发最佳实践

1. 主要内容

- 寻找Python代码
- 安装包
 - 使用pip
 - 使用包管理工具
 - 从源代码安装
- 集成开发环境
 - PyCharm
- 命名和文档
- 虚拟环境
- 源码版本控制
 - Git