



基于Python的计算机教学研究

汇报人：

2024-01-26



目录

-
- Python语言基础与特点
 - Python在计算机教学中的应用
 - 基于Python的实验教学设计
 - Python编程技巧与规范
 - 基于Python的计算机考试系统设计与实现
 - 总结与展望



01

Python语言基础与特点

Chapter





Python语言概述

Python是一种解释型、面向对象、动态数据类型的高级程序设计语言。

02

Python由Guido van Rossum于1989年底发明，1991年第一次公开发布。

01



03

Python具有丰富和强大的库，常被称为“胶水语言”，能够把用其他语言制作的各种模块（尤其是C/C++）很轻松地联结在一起。

04

Python的设计哲学是“优雅”、“明确”、“简单”，让代码具备易读性和简洁性，尤其是使用空格缩进划分代码块，而非使用大括号或者关键词。



Python语法基础

01

Python采用缩进来表示代码块，不需要使用大括号 {}。



02

Python使用 # 来表示注释，注释不会被执行。



03

Python中的变量不需要声明，可以直接赋值，且变量名区分大小写。



04

Python中的数据类型包括数字、字符串、列表、元组、字典等。





Python程序控制结构

```
if __name__ == '__main__':  
    # 设置初始值  
    qMax = 100  
    pMax = 100  
    # 设置初始值  
    q = 0  
    p = 0  
    # 设置初始值  
    n = 1  
    # 开始循环  
    while n <= 100:  
        # 计算当前值  
        q = 2 * q + p  
        p = q - p  
        # 输出结果  
        print(q, p)  
        # 更新计数器  
        n = n + 1
```

01

Python中的条件语句使用 if 关键字，支持 elif 和 else 子句。

```
def setConfig(qMax, float pMax):  
    # 设置初始值  
    q = 0  
    p = 0  
    # 设置初始值  
    n = 1  
    # 开始循环  
    while n <= 100:  
        # 计算当前值  
        q = 2 * q + p  
        p = q - p  
        # 输出结果  
        print(q, p)  
        # 更新计数器  
        n = n + 1
```

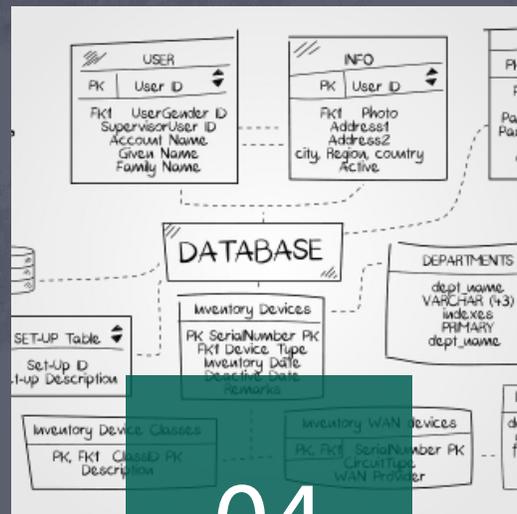
02

Python中的循环语句包括 for 循环和 while 循环，支持 break 和 continue 关键字来控制循环流程。

```
def main():  
    # 设置初始值  
    qMax = 100  
    pMax = 100  
    # 设置初始值  
    q = 0  
    p = 0  
    # 设置初始值  
    n = 1  
    # 开始循环  
    while n <= 100:  
        # 计算当前值  
        q = 2 * q + p  
        p = q - p  
        # 输出结果  
        print(q, p)  
        # 更新计数器  
        n = n + 1
```

03

Python中的函数使用 def 关键字定义，支持参数传递和返回值等操作。



04

Python中的异常处理使用 try/except 块来捕获和处理异常。



02

Python在计算机教学中的应用

Chapter





编程基础教学

通过Python学习变量、数据类型等基本概念，培养编程思维。

语法基础

Python语言简洁易懂的语法结构，适合作为编程初学者的入门语言。

变量与数据类型

控制流语句

利用Python的控制流语句（如if、for、while等）教授程序逻辑。

学习如何定义函数、调用函数，以及模块的导入与使用。

函数与模块



数据结构与算法教学

基本数据结构

通过Python实现列表、元组、字典等基本数据结构，理解数据组织方式。

高级数据结构

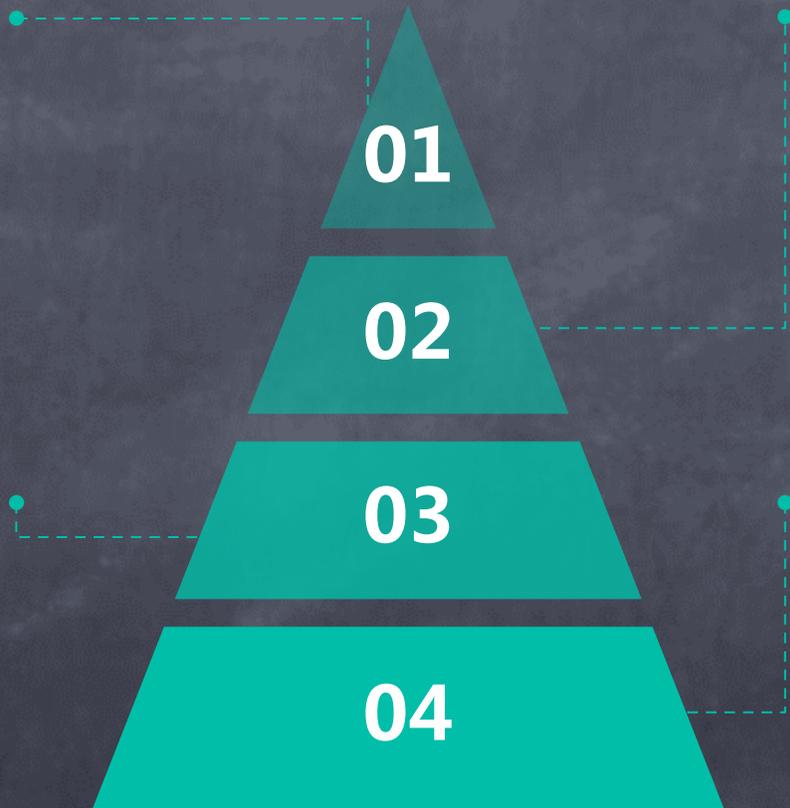
学习树、图等高级数据结构及其相关算法。

算法设计与分析

利用Python教授算法设计思想，如排序、查找等常用算法。

算法性能评估

通过Python进行算法性能分析，掌握时间复杂度和空间复杂度的概念。





软件开发实践教学

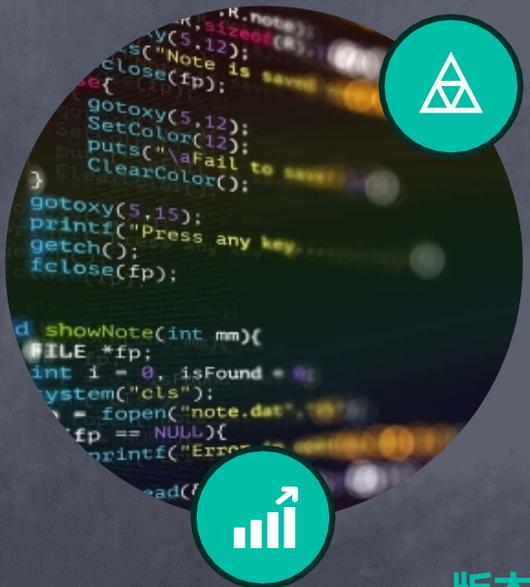
面向对象编程

通过Python学习类与对象的概念，掌握面向对象编程思想。



软件设计原则

教授软件设计的基本原则，如单一职责原则、开闭原则等。



版本控制与团队协作

利用Git等版本控制工具管理代码，实现团队协作开发。

软件开发流程

学习软件开发的基本流程，包括需求分析、设计、编码、测试等。



人工智能与机器学习教学

常用机器学习算法

教授分类、回归、聚类等常用机器学习算法的Python实现。

机器学习基础

通过Python学习机器学习的基本概念和原理。

A

B

C

D

深度学习入门

学习神经网络的原理，利用TensorFlow等框架实现简单的深度学习模型。

数据处理与可视化

利用Python进行数据清洗、特征工程以及数据可视化等操作，辅助机器学习和深度学习模型的训练与优化。



03

基于Python的实验教学设计

Chapter





实验教学内容与目标

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <title>My perfect website</title>
5 <meta charset="utf-8" />
6
7 <link rel="preconnect" href="https://s3.amazonaws.com/" />
8 <link rel="preconnect" href="https://www.mytag.com/" />
9
10 <meta name="viewport" content="width=device-width, initial-scale=1">
11
12
13 <script>
14 var mytag = mytag || {};
15 mytag.cmd = mytag.cmd || [];
16 {function() {
17   var gads = document.createElement('script');
18   gads.async = true;
19   gads.type = 'text/javascript';
20   var useSSL = 'https:' == document.location.protocol;
21   gads.src = (useSSL ? 'https' : 'http') + '://www.mytag.com/tag/js/gat.js';
22   var node = document.getElementsByTagName('script')[0];
23   node.parentNode.insertBefore(gads, node);
24 }
25 };
26 mytag.cmd.push(function() {
27   var homepageSquareSizeMapping = mytag.sizeMapping();
28   addSize([944, 280], [204, 280]);
29   addSize([0, 0], [360, 280]);
30   build();
31   mytag.defineSlot('/1023782/homepageDynamicSquare', [[360, 280], [280, 280]], 'reserved-div-1');
```

内容

涵盖Python基础语法、面向对象编程、常用库和框架等核心内容。

目标

培养学生掌握Python编程技能，具备独立解决问题的能力。



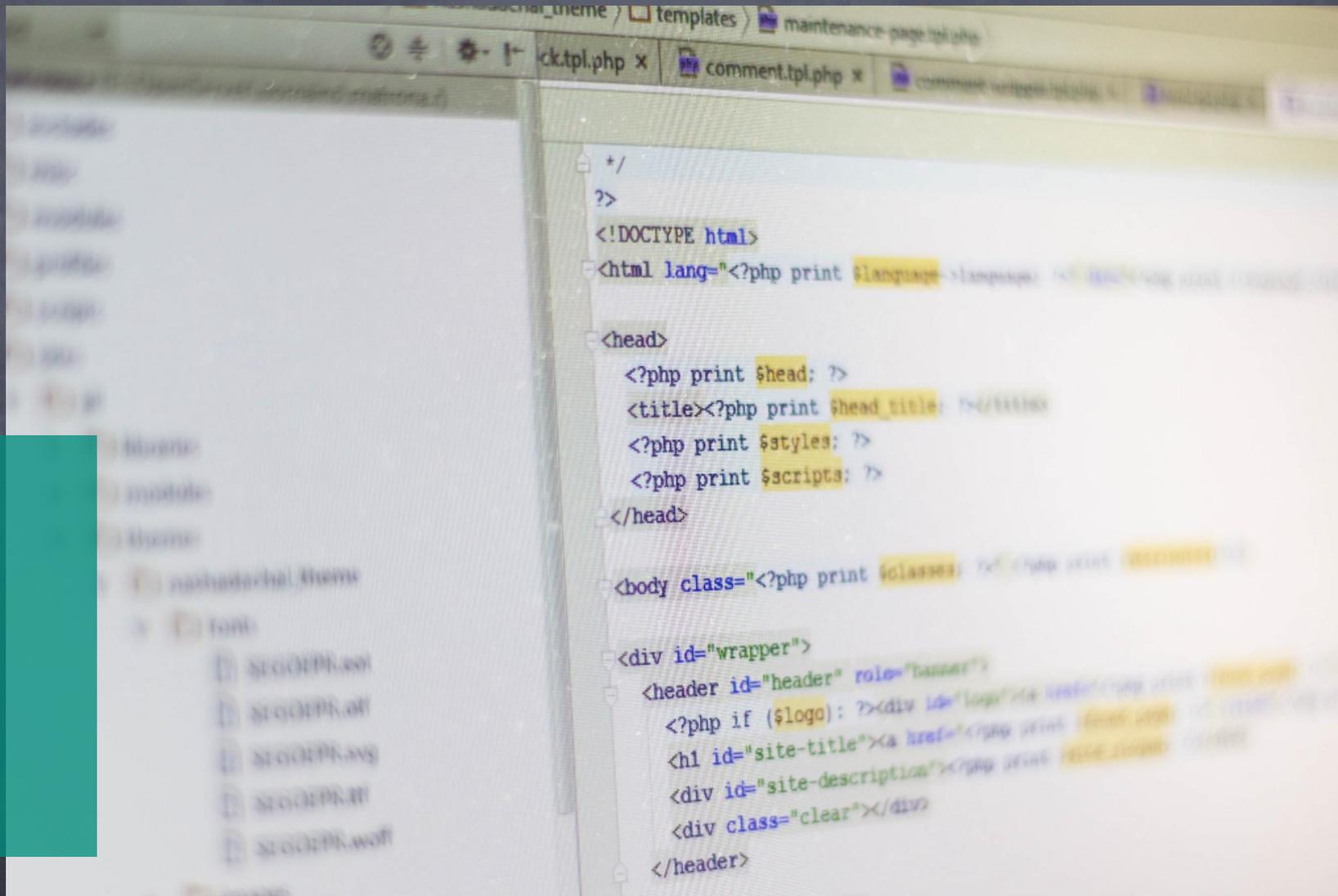
实验环境与工具配置

环境

推荐使用Python 3.x版本，配置好相应的开发环境。

工具

安装常用的Python IDE，如PyCharm、Visual Studio Code等。



以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：
<https://d.book118.com/788076077052006101>