

Python 程序设计教案

学院(部): 信息工程学院

课程名称	Python 程序设计基础	课程代码	z01100443	学 时 分 配	总学时：56
		学分	3.5		讲 授：24
课程类别	必修课（ 是 ）		选修课（ ）		实 验：32
授课专业	计算机应用技术	授课班级	2018 计算机 应用技术 AB		课堂讨论：
任课教师	徐博龙	职称	高工		习题课：
所选教材	《Python 程序设计基础与应用》，董付国，机械工业出版社				机 动：
本课程教学 目的与要求	通过本课程的学习，使得学生能够理解 Python 的编程模式，熟练运用 Python 列表、元组、字典、集合等基本数据类型以及相关列表推导式、切片等特性来解决实际问题，熟练掌握 Python 分支结构、循环结构、函数设计以及类的设计与使用，熟练使用正则表达式处理字符串，同时使得学生掌握数据采集、数据分析、数据可视化等领域中的实际问题，同时还应培养学生的代码优化与安全编程意识。				
本课程教学 重点与难点	Python 编程模式中非常重要的一条是代码简单化、问题简单化，应保证代码具有较强的可读性。在教学过程中，一定不要在 Python 程序中带有其他编程语言的痕迹，要尽量从最简单的角度去思考和解决问题、实现自己的想法和思路，保证代码的优雅、简洁，让代码更加 Pythonic。 使用 Python 编程解决问题应充分借鉴和使用成熟的标准库和扩展库，尽量避免自主编写完整的业务逻辑，部分具体操作完全可以使用扩展库来实现，大幅度提高开发效率。例如，如果需要进行排序则直接使用内置函数或方法进行排序，切不可再使用选择法排序、冒泡法排序或堆排序等排序算法自己编写代码实现。类似地，要习惯于使用 python-docx 处理 Word 文 档 、 openpyxl 处 理 Excel 文 档 ， 使用 urllib、requests、scrapy 编写网络爬虫程序，使用 pandas 处理和分析数，使用 matplotlib 对数据进行可视化，等等。教学过程中要多演示，让学生多练习，力求让大部分学生掌握书中的案例，少部分同学能够进行发挥和实现特定的任务。				

参考书目	1. 董付国.《Python 可以这样学》，清华大学出版社，2017 2. 董付国,《Pyth 程序设计开发宝典》，清华大学出版社，2017
------	---

Python 程序设计教案

本次授课内容	1.1 Python 语言简介 1.2 Python 版本简介 1.3 Python 开发环境安装与配置	
本次课的教学目的	了解Python 语言版本 熟悉Python 开发环境	
本次课教学重点与难点	如何选择合适的Python 版本 Windows 平台的Python 安装 不同开发环境的配置和简单使用	
教学方法 教学手段	PPT、边讲边练	
课堂教学 时间分配	教学内容	时间分配（分）
课堂教学设计	首先介绍Python 版本历史演变 然后介绍Python 的安装和使用	
实 验	安装不同版本的Python	
思考题及作业题	1. 下载安装Python。 3. 下载并安装Anaconda3。	
备 注		

教学后记		
	第 一 节 课	

- Python 是一门跨平台、开源、免费的解释型高级动态编程语言。
- Python 支持命令式编程（How to do）、函数式编程（What to do），完全支持面向对象程序设计，拥有大量扩展库。

▪ 问题解决：把列表中的所有数字都加5，得到新列表。（命令式编程）

```
>>> x = list(range(10))
>>> x
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
>>> y = []
>>> for num in x:
    y.append(num+5)
>>> y
[5, 6, 7, 8, 9, 10, 11, 12, 13, 14]
>>> [num+5 for num in x]
[5, 6, 7, 8, 9, 10, 11, 12, 13, 14]
```

创建列表

空列表

循环，遍历x中的每个元素

列表方法，在尾部追加元素

列表推导式

▪ 问题解决：把列表中的所有数字都加5，得到新列表。（函数式编程）

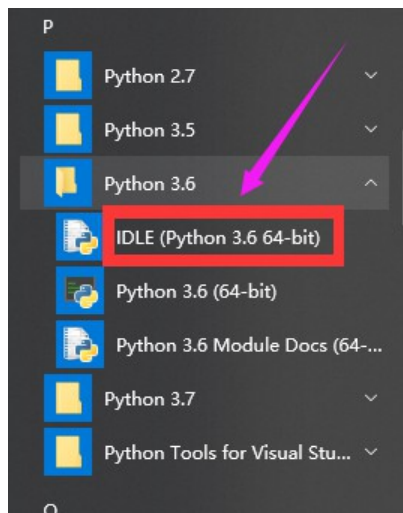
```
>>> x = list(range(10))
>>> x
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
>>> def add5(num):
    return num+5
>>> list(map(add5, x))
[5, 6, 7, 8, 9, 10, 11, 12, 13, 14]
>>> list(map(lambda num: num+5, x))
[5, 6, 7, 8, 9, 10, 11, 12, 13, 14]
```

定义函数，接收一个数字，加5后返回

把函数add5映射到x中的每个元素

lambda表达式，等价于函数add5

- 默认编程环境: **IDLE**
- 其他常用开发环境:
 - **Eclipse+PyDev**
 - **pyCharm**
 - wingIDE
 - **Eric**
 - PythonWin
 - **Anaconda3** (内含Jupyter和Spyder)
 - zwPython



```
Python 3.6.0 Shell
File Edit Shell Debug Options Window Help
Python 3.6.0 (v3.6.0:41df79263a11, Dec 23 2016, 08:06:12) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more
information.
>>> print('Hello World!')
Hello World!
>>> import math
>>> print(math.sqrt(9))
3.0
>>>
```

Python 程序设计教案

本次授课内容	1.4 Python 编程规范 1.5 扩展库安装方法 1.6 标准库与扩展库对象的导入与使用 1.7 Python 程序的__name__属性	
本次课的教学目的	了解 Python 编码规范 掌握扩展库安装方式 掌握标准库对象与扩展库对象的导入和使用	
本次课教学重点与难点	缩进对 Python 代码的影响 扩展库安装方式 标准库对象与扩展库对象的三种导入方法之间的区别	
教学方法 教学手段	PPT、边讲边练	
课堂教学 时间分配	教学内容	时间分配（分）
课堂教学设计	首先介绍 Python 编程规范，让学生重点理解代码缩进的作用，体会续行符的作用。 然后重点讲解扩展库的安装方法，让学生实验在线安装、离线安装，并注意扩展库的安装位置。 接下来讲解 import ...、from ... import ... 和 from ... import *这几种用法的区别。 最后简单演示 Python 程序__name__属性的功能。	
实 验	使用 pip 安装 Python 扩展库。	
思考题及作业题	2. 使用 pip 工具安装扩展库 pandas、openpyxl 和 pillow。 4. 解释导入标准库与扩展库对象的几种方法之间的区别。 5. 解释一下 Python 程序中__name__属性的作用	
备 注		
教学后记		

第一节 课

一、Python 编码规范

- (1) 类定义、函数定义、选择结构、循环结构、with 块，行尾的冒号表示缩进的开始。
- (2) 每个 import 语句只导入一个模块，最好按标准库、扩展库、自定义库的顺序依次导入。
- (3) 最好在每个类、函数定义和一段完整的功能代码之后增加一个空行，在运算符两侧各增加一个空格，逗号后面增加一个空格。
- (4) 尽量不要写过长的语句。如果语句过长，可以考虑拆分成多个短一些的语句，以保证代码具有较好的可读性。如果语句确实太长而超过屏幕宽度，最好使用续行符（line continuation character）“\”，或者使用圆括号将多行代码括起来表示是一条语句。
- (5) 虽然 Python 运算符有明确的优先级，但对于复杂的表达式建议在适当的位置使用括号使得各种运算的隶属关系和顺序更加明确、清晰。
- (6) 以符号 # 开始，表示本行 # 之后的内容为注释。包含在一对三引号'''...'''或"""..."""之间且不属于任何语句的内容将被解释器认为是注释。

二、导入和使用标准库对象的方法

```
>>> import math                #导入标准库 math
>>> math.sin(0.5)              #求 0.5（单位是弧度）的正弦
0.479425538604203
>>> import random             #导入标准库 random
>>> n = random.random()        #获得[0, 1) 内的随机小数
>>> n = random.randint(1,100)  #获得[1, 100]区间上的随机整数
>>> n = random.randrange(1, 100) #返回[1, 100)区间中的随机整数
>>> import os.path as path     #导入标准库 os.path，并设置别名为 path
>>> path.isfile(r'C:\windows\notepad.exe')
True
>>> import numpy as np         #导入扩展库 numpy，并设置别名为 np
>>> a = np.array((1, 2, 3, 4))  #通过模块的别名来访问其中的对象
>>> a
array([1, 2, 3, 4])
>>> print(a)
[1 2 3 4]
>>> from math import sin       #只导入模块中的指定对象，访问速度略快
>>> sin(3)
0.1411200080598672
>>> from math import sin as f  #给导入的对象起个别名
>>> f(3)
0.1411200080598672
>>> from os.path import isfile
>>> isfile(r'C:\windows\notepad.exe')
True
>>> from math import *         #导入标准库 math 中所有对象
>>> sin(3)                    #求正弦值
0.1411200080598672
>>> gcd(36, 18)               #最大公约数
18
>>> pi                        #常数  $\pi$ 
3.141592653589793
>>> e                        #常数 e
2.718281828459045
```

>>> log2(8)	#计算以 2 为底的对数值
3.0	
>>> log10(100)	#计算以 10 为底的对数值
2.0	
>>> radians(180)	#把角度转换为弧度
3.141592653589793	

第 二 节 课

一、安装 Python 扩展库

pip 在线安装（命令提示符环境，建议切换至 Python 安装目录中的 scripts 文件夹执行）

pip 离线安装：<https://www.lfd.uci.edu/~gohlke/pythonlibs/>

exe 安装，不是每个扩展库都支持

conda 在线安装

如果机器上安装了多个 Python 开发环境，那么在一个环境下安装的扩展库无法在另一个环境下使用，需要分别安装。

pip 命令示例	说明
pip download SomePackage[==version]	下载扩展库的指定版本，不安装
pip freeze [> requirements.txt]	以 requirements 的格式列出已安装模块
pip list	列出当前已安装的所有模块
pip install SomePackage[==version]	在线安装 SomePackage 模块的指定版本
pip install SomePackage.whl	通过 whl 文件离线安装扩展库
pip install package1 package2 ...	依次（在线）安装 package1、package2 等扩展模块
pip install -r requirements.txt	安装 requirements.txt 文件中指定的扩展库
pip install --upgrade SomePackage	升级 SomePackage 模块
pip uninstall SomePackage[==version]	卸载 SomePackage 模块的指定版本