

Python程序设计及网络爬虫

课程复习





全课程总结

课程内容设计

□ 第一部分：Python快速入门

围绕2个具体实例，讲解Python基本语法元素，感性认识

□ 第二部分：Python基础语法

从5个方面讲解基础语法全体系，提供10个实例，理性学习

□ 第三部分：Python网络爬虫

从方法学角度开阔认识，提升整体编程能力，展望未来

课程内容设计

面向过程编程的"Python基础语法"全体系

- Python基础语法

- Python实例解析



- Python计算生态

课程内容设计

好的开始是成功的一半



Python基础语法 (全体系)

① 基本数据类型

- 整数、浮点数、复数
- 字符串

② 程序的控制结构

- 分支结构与异常处理
- 遍历循环、无限循环

③ 函数和代码复用

- 函数定义和使用
- 函数递归

④ 组合数据类型

- 集合类型
- 序列类型：元组和列表
- 字典类型

⑤ 文件和数据格式化

- 文件的使用
- 一二维数据的表示存储和处理

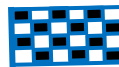
6 网络爬虫和自动化

- 集合类型
- 序列类型



程序设计基本方法

- 计算机的功能性和可编程性
- 编译和解释、静态语言和脚本语言
- IPO、理解问题的计算部分
- 掌握计算机编程的价值



Python基本语法元素

- 缩进、注释、命名、变量、保留字
- 数据类型、字符串、整数、浮点数、列表
- 赋值语句、分支语句、函数
- input()、print()、eval()、 print()格式化



and	elif	import	raise	global
as	else	in	return	nonlocal
assert	except	is	try	True
break	finally	lambda	while	False
class	for	not	with	None
continue	from	or	yield	async
def	if	pass	del	await

保留字



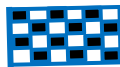
Python基本图形绘制

- 从计算机技术演进角度看待Python语言
- 海龟绘图体系及import保留字用法
- penup()、pendown()、pensize()、pencolor()
- fd()、circle()、seth()
- 循环语句：for和in、range()函数



基本数据类型

- 数据类型：整数、浮点数、复数及
- 数据类型运算操作符、运算函数
- 字符串类型：表示、索引、切片
- 字符串操作符、处理函数、处理方法、.format()格式化
- time库：time()、strftime()、strptime()、sleep()等



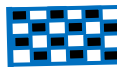
数字类型及操作

- 整数类型的无限范围及4种进制表示
- 浮点数类型的近似无限范围、小尾数及科学计数法
- +、-、*、/、//、%、**、二元增强赋值操作符
- abs()、divmod()、pow()、round()、max()、min()
- int()、float()、complex()



字符串类型及操作

- 正向递增序号、反向递减序号、<字符串>[M:N:K]
- +、*、len()、str()、hex()、oct()、ord()、chr()
- .lower()、.upper()、.split()、.count()、.replace()
- .center()、.strip()、.join() 、.format()格式化



程序的分支结构

- 单分支 *if* 二分支 *if-else* 及紧凑形式
- 多分支 *if-elif-else* 及条件之间关系
- *not and or > >= == <= < !=*
- 异常处理 *try-except-else-finally*



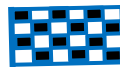
程序的循环结构

- *for...in* 遍历循环: 计数、字符串、列表、文件...
- *while* 无限循环
- *continue* 和 *break* 保留字: 退出当前循环层次
- 循环 *else* 的高级用法: 与 *break* 有关



函数的定义与使用

- 使用保留字`def`定义函数，`lambda`定义匿名函数
- 可选参数(赋初值)、可变参数(*b)、名称传递
- 保留字`return`可以返回任意多个结果
- 保留字`global`声明使用全局变量，一些隐式规则



代码复用与函数递归

- 模块化设计：松耦合、紧耦合
- 函数递归的2个特征：基例和链条
- 函数递归的实现：函数 + 分支结构



集合类型及操作

- 集合使用{}和set()函数创建
- 集合间操作：交(&)、并(|)、差(-)、补(^)、比较(>=<)
- 集合类型方法：.add()、.discard()、.pop()等
- 集合类型主要应用于：包含关系比较、数据去重



序列类型及操作

- 序列是基类类型，扩展类型包括：字符串、元组和列表
- 元组用`()`和`tuple()`创建，列表用`[]`和`set()`创建
- 元组操作与序列操作基本相同
- 列表操作在序列操作基础上，增加了更多的灵活性



python

语言程序设计

字典类型及操作

- 映射关系采用键值对表达
- 字典类型使用{}和dict()创建，键值对之间用:分隔
- d[key] 方式既可以索引，也可以赋值
- 字典类型有一批操作方法和函数，最重要的是.get()



文件的使用

- 文件的使用方式：打开-操作-关闭
- 文本文件&二进制文件，`open(,)`和`.close()`
- 文件内容的读取：`.read()` `.readline()` `.readlines()`
- 数据的文件写入：`.write()` `.writelines()` `.seek()`



一维数据的格式化和处理

- 数据的维度：一维、二维、多维、高维
- 一维数据的表示：列表类型(有序)和集合类型(无序)
- 一维数据的存储：空格分隔、逗号分隔、特殊符号分隔
- 一维数据的处理：字符串方法 `.split()` 和 `.join()`

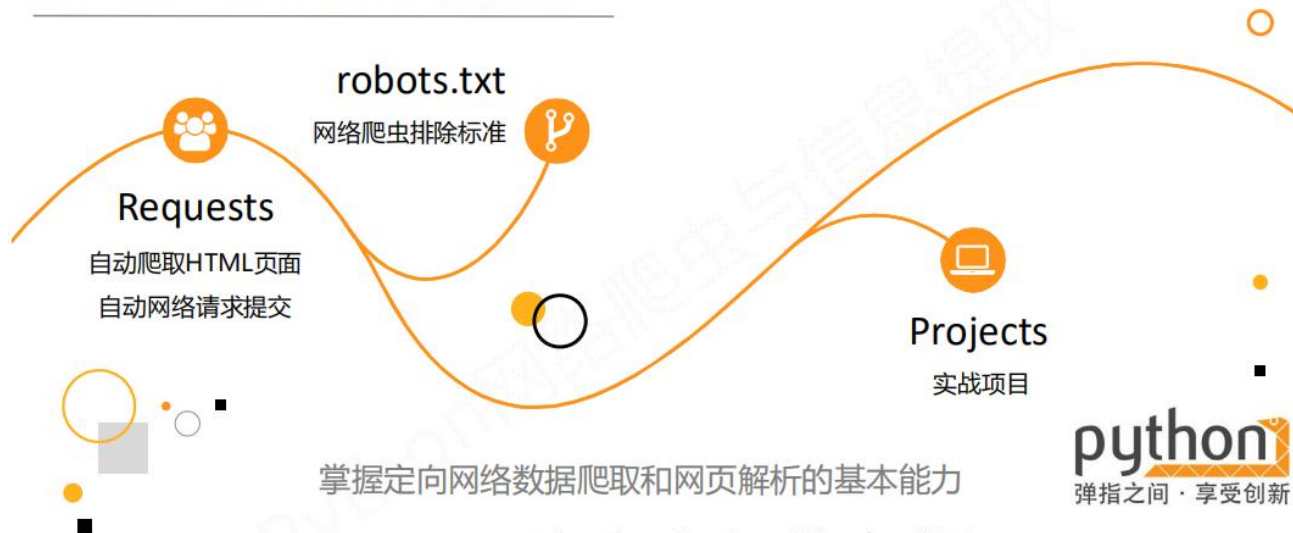


二维数据的格式化和处理

- 二维数据的表示：列表类型，其中每个元素也是一个列表
- CSV格式：逗号分隔表示一维，按行分隔表示二维
- 二维数据的处理：for循环+.split()和.join()



The Website is the API ...



Python网络爬虫与信息提取

python
弹指之间·享受创新

04X -Tian

爬取网页的通用代码框架

```
import requests
```

```
def getHTMLText(url):
```

```
    try:
```

```
        r = requests.get(url, timeout=30)
```

```
        r.raise_for_status() #如果状态不是200, 引发HTTPError异常
```

```
        r.encoding = r.apparent_encoding
```

```
        return r.text
```

```
    except:
```

```
        return "产生异常"
```

```
if __name__ == "__main__":
```

```
    url = "http://www.baidu.com"
```

```
    print(getHTMLText(url))
```

Python计算生态 (详解7个)

① turtle库

- 基本图形绘制

③ random库

- 随机数产生及应用

⑤ jieba库

- 简洁的中文分词

⑦ Requests库

- 网络爬虫



② time库

- 时间的基本处理

④ PyInstaller库

- 源代码打包为可执行文件

⑥ wordcloud库

- 中英文词云生成



Python计算生态 (概览一批)

① 从数据处理到人工智能

- 数据分析
- 数据可视化
- 文本处理
- 机器学习

② 从Web解析到网络空间

- 网络爬虫
- Web信息提取
- Web网站开发
- 网络应用开发

③ 从人机交互到艺术设计

- 图形用户界面
- 游戏开发
- 虚拟现实
- 图形艺术



Python实例解析 (15个)

- 实例1: 温度转换
- 实例2: Python蟒蛇绘制
- 实例3: 天天向上的力量
- 实例4: 文本进度条
- 实例5: 身体质量指数BMI
- 实例6: 圆周率的计算
- 实例7: 七段数码管绘制
- 实例8: 科赫雪花小包裹



Python实例解析 (15个)

- 实例9: 基本统计值计算
- 实例10: 文本词频统计
- 实例11: 自动轨迹绘制
- 实例12: 政府工作报告词云
- 实例13: 第三方库安装脚本
- 实例14: 电商商品页面爬取
- 实例15: 搜索关键词提交





课程考核

考核组成

平时 + (作业 + 大作业) + 期末考试

- **期末考试：考核Python基本语法，闭卷有监考**

判断题10分（10题）、简答题50分（10题）、编程题40分（4-5题）

- **大作业验收**



学习展望

应用深度

计算生态

语法深度

Python基础语法

Python进阶语法

Python高级语法

Python网络爬虫与信息提取

Python数据分析与展示

Python机器学习应用

Python科学计算三维可视化

...

Python+大数据+人工智能

Python+嵌入式+可编程硬件

学习展望

Python未来之路在哪里？

- Python Everywhere, Python无处不在
- Python Only Not Enough, 只有Python可以但不足够
- Python EcoSystem, Python计算生态将成为编程主流

"人生苦短，我学Python"

人生苦短
我学Python

