



Python基础入门升级版

从零开始学 Python

课程介绍

(一) Python入门

七月在线 **David**
2017年11月25日



本节课程目标

- ❑ 对Python这门语言有初步的认识
- ❑ 搭建好Python交互式开发环境
- ❑ 了解Python运行的不同方式
- ❑ 了解基本语法，数值及字符串数值类型，会使用判断及循环，并有能力完成练习题



本节目录

- ☐ Python简介
- ☐ Python环境搭建，开发工具的使用
- ☐ 基础语法，标准操作符
- ☐ 循环与判断
- ☐ 标准数据类型：数值与字符串



Python简介

□ Python语言的历史

- 作者：**Guido van Rossum**（龟叔）
- 名字来源：Monty Python's Flying Circus
- ABC语言：Guido参与设计的是服务非专业程序员的语言。

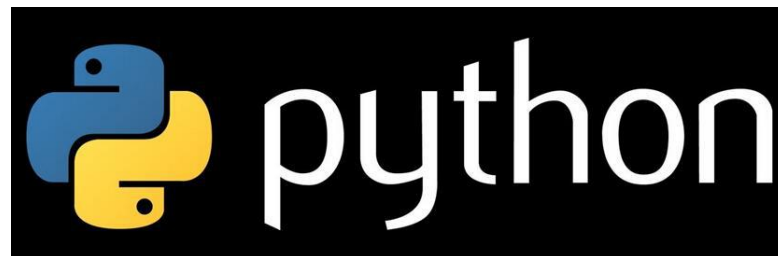


□ Python的特色

□ Python的应用

▪ 版本选择

- 优先至少选择3.5及以上版本
- 2和3的主要区别



Python开发环境搭建

- ☐ 直接安装Python官方环境[不推荐]



- ☐ 科学计算环境Anaconda简介与一键安装
- ☐ Anaconda下不同Python环境的安装
- ☐ 三方库的安装与卸载
- ☐ Jupyter Notebook启动及使用说明



Python常用库介绍

- Numpy简介

- NumPy是Python语言的一个扩充程序库。支持高级大量的维度数组与矩阵运算，此外也针对数组运算提供大量的数学函数库。
- 核心数据结构：ndarray与ufunc

- Pandas简介

- Pandas 纳入了大量库和一些标准的数据模型，提供了高效地操作大型数据集所需的工具。pandas提供了大量能使我们快速便捷地处理数据的函数和方法。
- 核心数据结构：Series, DataFrame和index对象

- Matplotlib简介

- 一个高级的绘图第三方库，可以支持种类繁多的绘图，自定义功能强大
- 基本图形例子：散点图、柱状图、3D图...



Python开发工具及运行方式

- 开发工具
 - cmd命令行
 - 自带及第三方IDE工具。如idle, pycharm
 - 推荐: Jupyter notebook

- 两种方式:
 - 脚本式: 使用场景: 应用使用
 - 交互式: 使用场景: 开发调试



Python基本语法规则

- ☐ 注释#,
- ☐ 代码分隔。继续上一行\
- ☐ 文本行分割。
- ☐ 多行文本赋值'''
- ☐ ; 多行语句写在一行
- ☐ : 分隔代码块 (头&体)
- ☐ 缩进 (缩进深度)
- ☐ 空行(用于分割, 函数与类, 阅读体验好, 空行也是程序的一部分)
- ☐ 模块



Python标识符命名规则

- 标识符:
 - 计算机语言中，允许作为名字的有效字符串集合
- 特殊标识符(关键字, BIFs, 专用下划线标识符)
- 创建合法的标识符规则
 - 第一个字符以字母或下划线开始
 - 其它部分可以是数字，字母或下划线
 - 大小写敏感
- 变量命名时建议采用驼峰表示法



Python中的关键字

- 常量
 - True , False , None
- 函数，类，对象和模块
 - def , class , import , from , del , global , return , lambda
- 判断与循环
 - If , elif , else , is , in , assert , and , or , not
- 循环
 - for , while , continue , break , pass
- 异常
 - with , raise , try , except , finally , as



BIFs: Python的自带电池

- ❑ 查看: `dir(__builtins__)`
- ❑ IO: `print()`, `open()`, `input()`
- ❑ 列表与帮助: `dir()`与`help()`
- ❑ 类型与身份: `type()`, `id()`, `memoryview()`, `isinstance()`, `issubclass()`,
- ❑ 数学运算类: `sum`, `pow`, `round`, `min`, `max`, `hash`等
- ❑ 类型转换: `int`, `bin`, `hex`, `oct`, `str`, `float`, `list`, `bytes`等
- ❑ 序列: `len()`, `range()`, `zip()`, `map()`, `reduce()`, `filter()`, `reversed()`, `sorted()`, `enumerate()`



标准操作符

- 算术运算符

- `+`, `-`, `*`, `/`, `//`取整, `%`取模(余数), `**`求幂

- 比较运算符

- `!=`, `==`, `>`, `>=`, `<`, `<=`

- 赋值运算符

- `=`, (增量赋值: `+=`, `-=`, `*=`, `/=`, `**=`, `//=`)

- 逻辑运算符

- `and`/`or`/`not`

- 成员运算符

- `in`/`not in`身份运算符

- 身份运算符

- `is`/`is not`身份运算符



变量赋值

- ❑ 赋值符 =
- ❑ 多重赋值, =
- ❑ 多元赋值
- ❑ 变量交换赋值



循环

- ☐ for迭代循环
- ☐ while条件循环
 - ☐ 死循环
 - ☐ 条件循环
- ☐ 循环控制
 - ☐ break
 - ☐ continue
 - ☐ pass



条件判断

- ☐ if
- ☐ if else
- ☐ if elif

- ☐ Python 三元表达式



Python标准数据类型

- ❑ 不可变类型:
 - ❑ 数值 **Number**
 - ❑ 字符串 **String**

- ❑ 容器类型 (可变类型) :
 - ❑ 列表 **List**
 - ❑ 元组 **Tuple**
 - ❑ 集合 **Set**
 - ❑ 字典 **Dictionary**



Python标准数据类型-数值型

- 数值类型:
 - int整数类型
 - float浮点
 - complex复数
- 操作
 - 数值类型间转换
 - 绝对值, 四舍五入
 - 其它操作



Python标准数据类型- 字符串型

- ☐ 字符串的初始化, 及转义
- ☐ 字符串长度
- ☐ 访问
 - ☐ 循环, 索引, 切片 (左闭右开)
- ☐ 格式化输出
- ☐ 拼接
- ☐ 查找与替换
- ☐ 大小写
- ☐ 编码与解码
- ☐ 其它操作dir()



练习1

- 求从1到100的数字中所有能既能被3整除，又能被5整合的数字有哪些。



练习2

- 输入一个字符串返回满足以下条件的字符串
 - 如果字符串长度大等于3，添加 'ing' 到字符串的末尾
 - 如果字符串是以 'ing' 结尾的，就在末尾添加 'ly'
 - 如果字符串长度小于3，返回原字符串



练习3

- 判断回文
 - 输入一个数字，如果是回文数字，返回True，否则 返回False
 - 提示：回文：62426是回文数字



练习4

- 输入一个字符串，返回满足以下条件字符串
 - 找到字符串中的子串 'not' 和 'bad'
 - 如果 'bad' 出现在 'not' 后面，就把 'not' ... 'bad' 之间包含的所有字符串替换成 'good'



练习5

- 输入一个字符串，把字符串拆分成两个等分
 - 如果字符串长度是偶数，前一半和后一半的长度是相同的
 - 如果字符串长度是奇数，则多出的一个字符加到前一半，如：'abcde'，前一半是'abc'，后一半是'de'



练习6

- 输入一个字符串返回满足以下条件的字符串
 - 找出与字符串的第一个字母相同的字母，把它们替换成 '*', 除了第一个字母本身以外
 - 例如: 输入 'babble', 返回 'ba**le'



练习7

- 输入一个字符串,返回满足以下条件的字符串
 - 由字符串的最前面两个字母和最后两个字母组成的字符串。
 - 例如: 'spring' 返回 'spng', 'is' 返回 'is'
 - 当输入的字符串长度小于2时, 返回空字符串



练习8

- 落球计算
- 一球从100米高度自由落下，假设每次落地后反跳回原高度的一半；再落下，再弹起。请问第6次落地后会弹起多少米？
- 请分别使用for循环与while循环。并使用break与continue流程控制关键字



作业

- ❑ 创建一个python文件，接收两个参数（均为数字）并找出两个数字间所有的素数（即只能被1和自己整除的数）
- ❑ 运行这个文件，并检查结果

