
Tedu.cn
达内教育
Python运维开发
NSD PYTHON
DAY01

内容

上午	09:00 ~ 09:30	python概述
	09:30 ~ 10:20	
	10:30 ~ 11:20	python起步
	11:30 ~ 12:20	
下午	14:00 ~ 14:50	数据类型
	15:00 ~ 15:50	
	16:00 ~ 16:50	判断语句
	17:00 ~ 17:30	总结和答疑

python概述

Tedu.cn 达内教育

python简介

Tedu.cn 达内教育

python起源

知识讲解

- 贵铎·范·罗萨姆（Guido van Rossum）于1989年底始创了python
- 1991年初，python发布了第一个公开发行版
- 为了更好的完成荷兰的CWI（国家数学和计算机科学研究院）的一个研究项目而创建

python的特点

知识讲解

- 高级：有高级的数据结构，缩短开发时间与代码量
- 面向对象：为数据和逻辑相分离的结构化和过程化编程添加了新的活力
- 可升级：提供了基本的开发模块，可以在它上面开发软件，实现代码的重用
- 可扩展：通过将其分离为多个文件或模块加以组织管理

python的特点（续1）

知识讲解

- 可移植性：python是用C写的，又由于C的可移植性，使得python可以运行在任何带有ANSI C编译器的平台上
- 易学：python关键字少、结构简单、语法清晰
- 易读：没有其他语言通常用来访问变量、定义代码块和进行模式匹配的命令式符号
- 内存管理器：内存管理是由python解释器负责的

Tedu.cn 达内教育

环境准备

Tedu.cn 达内教育

安装

- 下载：http://python.org
- Windows版本使用python.msi
- 类UNIX系统默认已经安装，或使用源码包

知识讲解

```
[root@py01 python]#  ./configure
[root@py01 python]#  make
[root@py01 python]#  make install
```

设置环境变量

Tedu.cn 达内教育

知识讲解

- 编写自动补全脚本

```
[root@py01 ~]#  vim  /usr/local/bin/tab.py
import readline
import rlcompleter

readline.parse_and_bind("tab: complete")
```

- 设置环境变量

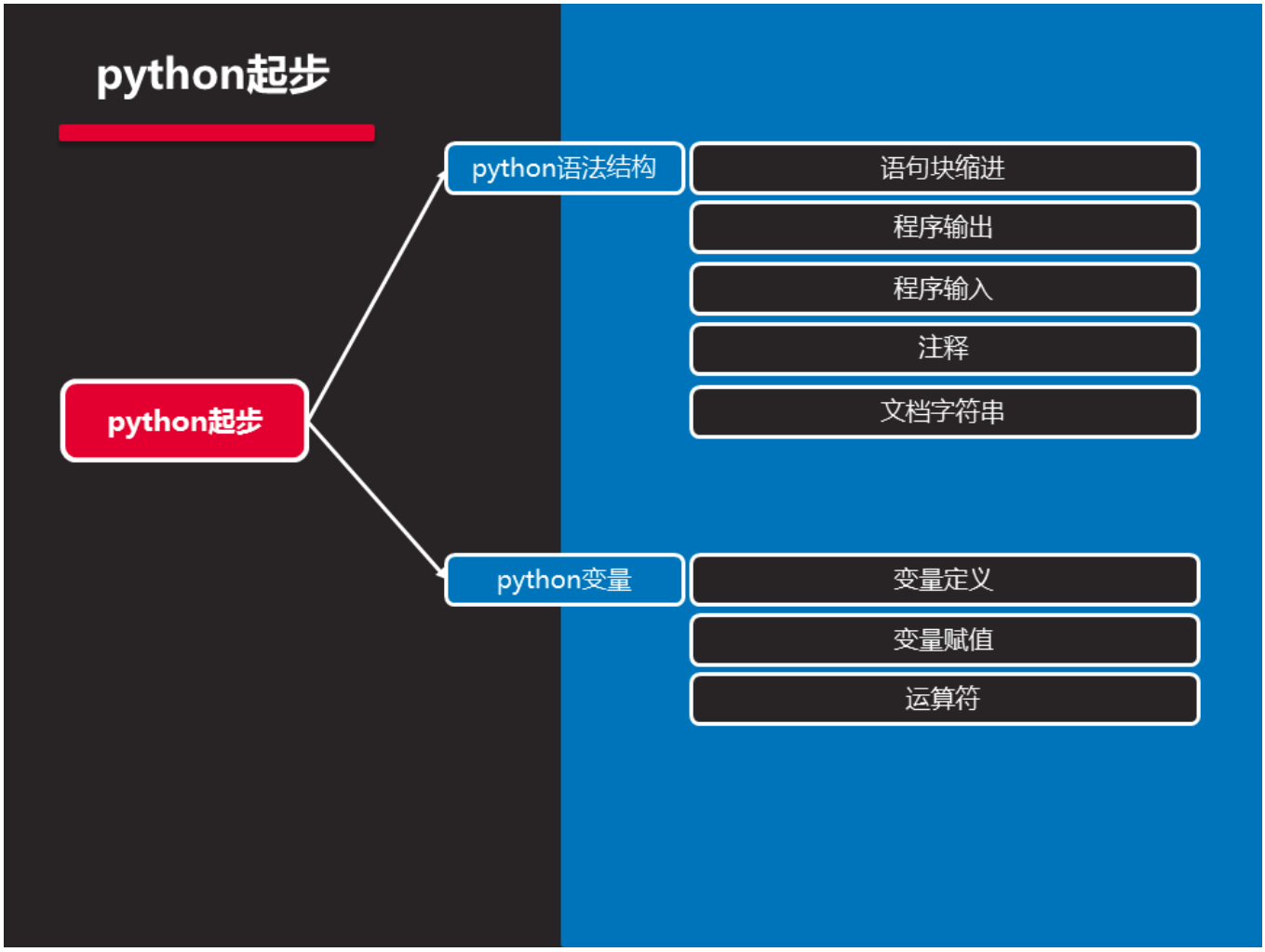
```
[root@py01 ~]#  vim  ~/.bash_profile
PYTHONSTARTUP=/usr/local/bin/tab.py
export PYTHONSTARTUP
[root@py01 ~]# source ~/.bash_profile
```

设置pycharm

Tedu.cn 达内教育

知识讲解

- Pycharm是由JetBrains打造的一款Python IDE
- 支持的功能有：
  - 调试、语法高亮
  - Project管理、代码跳转
  - 智能提示、自动完成
  - 单元测试、版本控制
- 下载地址：
  https://www.jetbrains.com/pycharm/download



Tedu.cn
达内教育

python语法结构

语句块缩进

Tedu.cn 达内教育

知识讲解

- python代码块通过缩进对齐表达代码逻辑而不是使用大括号
- 缩进表达一个语句属于哪个代码块
- 缩进风格
  - 1或2：可能不够，很难确定代码语句属于哪个块
  - 8至10：可能太多，如果代码内嵌的层次太多，就会使得代码很难阅读
  - 4个空格：非常流行， 范·罗萨姆支持的风格

程序输出

Tedu.cn 达内教育

知识讲解

- print将数据输出到屏幕

```
[root@py01 ~]#  python
>>> print 'Hello World!'
Hello World!
>>> print 'Hello', 'World'
Hello World
>>> print 'Hello' + 'World'
HelloWorld
```

程序输入

Tedu.cn 达内教育

知识讲解

- 使用raw_input()函数读取用户输入数据

```
[root@py01 ~]#  python
>>> user = raw_input('Enter your name: ')
Enter your name: bob
>>> print 'Your name is:', user
Your name is: bob
>>> i = raw_input('Input a number: ')
Input a number: 10
>>> i + 1
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: cannot concatenate 'str' and 'int' objects
```

案例1：程序输入输出

Tedu.cn 达内教育

课堂练习

1. 创建~/bin/login.py文件
2. 提示用户输入用户名
3. 获得到相关用户名后，将其保存在变量中
4. 在屏幕上输出Welcome 用户名

注释

Tedu.cn 达内教育

知识讲解

- 和大部分脚本及Unix-shell语言一样，python也使用#符号标示注释
- 从#开始，直到一行结束的内容都是注释
- 良好的注释习惯可以：
  - 方便其他人了解程序功能
  - 方便自己在日后读懂代码

文档字符串

Tedu.cn 达内教育

知识讲解

- 可以当作一种特殊的注释
- 在模块、类或者函数的起始添加一个字符串，起到在线文档的功能
- 简单的说明可以使用单引号或双引号
- 较长的文字说明可以使用三引号

文档字符串（续1）

Tedu.cn 达内教育

- 编写python程序模块

知识讲解

```
[root@localhost bin]# vim star.py
#!/usr/bin/env python
"""star module

just a sample module.
only include one function."""

def pstar():
    "do not accept args. Used to print 50 stars"
    print '*' * 50
```

文档字符串（续2）

Tedu.cn 达内教育

- 导入模块，查看在线文档

知识讲解

```
[root@localhost bin]# python
>>> import star
>>> help(star)
NAME
    star - star module
FILE
    /root/bin/star.py
DESCRIPTION
    just a sample module.
    only include one function.
FUNCTIONS
    pstar()
        do not accept args. Used to print 50 stars
```

Tedu.cn 达内教育

python变量

Tedu.cn 达内教育

变量定义

知识讲解

- 变量名称约定
  - 第一个字符只能是大小写字母或下划线
  - 后续字符只能是大小写字母或数字或下划线
  - 区分大小写
- python是动态类型语言，即不需要预先声明变量的类型

变量赋值

知识讲解

- 变量的类型和值在赋值那一刻被初始化
- 变量赋值通过等号来执行

```
>>> counter = 0
>>> name = 'bob'
```

- python也支持增量赋值

```
>>>  n += 1      #等价于n = n + 1
>>>  n *= 1      #等价于n = n * 1
>>> i++
  File "<stdin>", line 1
    i++
     ^
SyntaxError: invalid syntax
```

运算符

知识讲解

- 标准算术运算符

  + - * / // % **

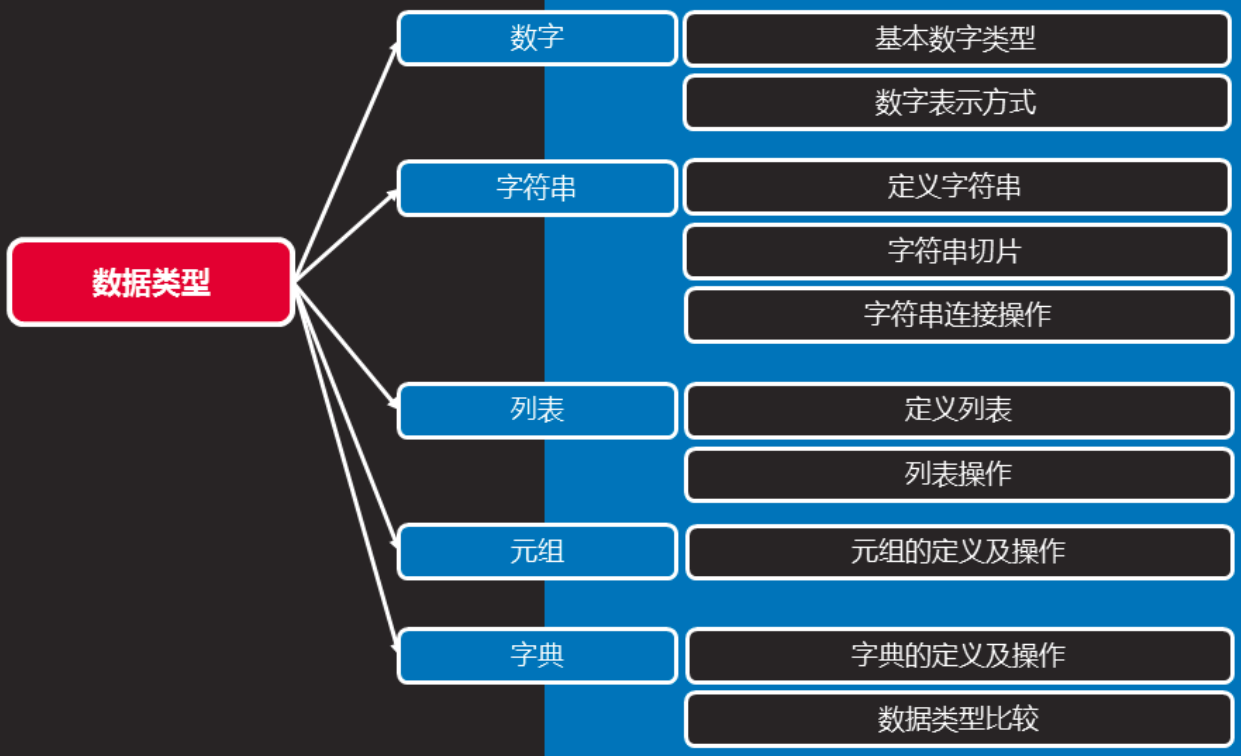
- 比较运算符

  < <= > >= == != <>

- 逻辑运算符

  and not or

数据类型



Tedu.cn
达内教育

数字

基本数字类型

Tedu.cn 达内教育

知识讲解

- int：有符号整数
- long：长整数
- bool：布尔值
  - True：1
  - False：0
- float：浮点数
- complex：复数

数字表示方式

Tedu.cn 达内教育

知识讲解

- python默认以十进制数显示
- 数字以0开头表示为8进制数
- 数字以0x或0X开头表示16进制数
- 数字以0b或0B开头表示2进制数

```
>>> 11
11
>>> 011
9
>>> 0x11
17
>>> 0b11
3
```

Tedu.cn 达内教育

字符串

Tedu.cn 达内教育

定义字符串

知识讲解

- python中字符串被定义为引号之间的字符集合
- python支持使用成对的单引号或双引号
- 无论单引号，还是双引号，表示的意义相同
- python还支持三引号（三个连续的单引号或者双引号），可以用来包含特殊字符
- python不区分字符和字符串

字符串切片

Tedu.cn 达内教育

知识讲解

- 使用索引运算符[ ]和切片运算符[:]可得到子字符串
- 第一个字符的索引是0，最后一个字符的索引是-1
- 子字符串包含切片中的起始下标，但不包含结束下标

```
>>> pyStr = 'python'
>>> pyStr[0]
'P'
>>> pyStr[-2]
'o'
>>> pyStr[2:4]
'th'
>>> pyStr[2:]
'thon'
>>> pyStr[:4]
'Pyth'
```

字符串连接操作

Tedu.cn 达内教育

知识讲解

- 使用+号可以将多个字符串拼接在一起
- 使用*号可以将一个字符串重复多次

```
>>> pyStr = 'python'
>>> isCool = 'is Cool‘
>>> print pyStr + ' ' + isCool
python is Cool
>>> pyStr * 2
'pythonpython'
```

Tedu.cn 达内教育

列表

Tedu.cn 达内教育

定义列表

知识讲解

- 可以将列表当成普通的“数组”，它能保存任意数量任意类型的python对象
- 像字符串一样，列表也支持下标和切片操作
- 列表中的项目可以改变

```
>>> aList = [1, "tom", 2, "alice"]
>>> aList[1] = 'bob‘
>>> aList[2:]
```

列表操作

- 使用in或not in判断成员关系
- 使用append方法向列表中追加元素

知识讲解

```
>>> aList = [1, "tom", 2, "alice"]
>>> 'tom' in aList
True
>>> 'alice' not in aList
False
>>> aList.append(3)
>>> aList[5] = 'bob'
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
IndexError: list assignment index out of range
```

元组

元组的定义及操作

- 可以认为元组是“静态”的列表
- 元组一旦定义，不能改变

知识讲解

```
>>> aTuple = (1, "tom", 2, "alice")
>>> 'tom' in aTuple
True
>>> aTuple[0] = 3
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: 'tuple' object does not support item assignment
```

字典

字典的定义及操作

Tedu.cn 达内教育

知识讲解

- 字典是由键-值(key-value)对构成的映射数据类型
- 通过键取值，不支持下标操作

```
>>> userDict = {'name':'bob', 'age':23}
>>> userDict['gender'] = 'male'
>>> 'bob' in userDict
False
>>> 'name' in userDict
True
>>> userDict[0]
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
KeyError: 0
```

数据类型比较

Tedu.cn 达内教育

知识讲解

- 按存储模型分类
  - 标量类型：数值、字符串
  - 容器类型：列表、元组、字典
- 按更新模型分类：
  - 可变类型：列表、字典
  - 不可变类型：数字、字符串、元组
- 按访问模型分类
  - 直接访问：数字
  - 顺序访问：字符串、列表、元组
  - 映射访问：字典

判断语句

Tedu.cn
达内教育

if语句

if语句语法结构

Tedu.cn 达内教育

知识讲解

- 标准if条件语句的语法

```
if expression:
    if_suite
else:
    else_suite
```

- 如果表达式的值非0或者为布尔值True, 则代码组if_suite被执行；否则就去执行else_suite
- 代码组是一个python术语，它由一条或多条语句组成，表示一个子代码块

if语句示例解析

Tedu.cn 达内教育

知识讲解

- 只要表达式数字为非零值即为True

```
>>> if 10:
...     print 'Yes'
Yes
```

- 空字符串、空列表、空元组，空字典的值均为False

```
>>> if   "":
...     print 'Yes'
... else:
...     print 'No'
No
```

案例2：判断合法用户

课堂练习

1. 创建~/bin/login2.py文件
2. 提示用户输入用户名和密码
3. 获得到相关信息后，将其保存在变量中
4. 如果用户输的用户名为bob，密码为123456，则输出Login successful，否则输出Login inorrect

扩展if语句

扩展if语句结构

Tedu.cn 达内教育

知识讲解

- 扩展if条件语句的语法

```
if expression1:
   if_suite
elif expression2:
   elif_suite
else:
   else_suite
```

- 只有满足相关条件，相应的子语句才会执行
- 没有switch/case这样的替代品

扩展if语句示例解析

Tedu.cn 达内教育

知识讲解

- 对于多个分支，只有一个满足条件的分支被执行

```
>>> if x > 0:
...     print 'Positive'
... elif x < 0:
...     print 'Negative'
... else:
...     print 'Zero'
```

案例3：编写判断成绩的程序

课堂练习

- 创建~/bin/grade.py脚本，根据用户输入的成绩分档，要求如下：
  1. 如果成绩大于60分，输出“及格”
  2. 如果成绩大于70分，输出“良”
  3. 如果成绩大于80分，输出“好”
  4. 如果成绩大于90分，输出“优秀”
  5. 否则输出“你要努力了”

总结和答疑

Tedu.cn 达内教育

判断失败

Tedu.cn 达内教育

问题现象

- 执行判断时出现以下错误

知识讲解

```
>>> a = 10
>>> if a = 10:
  File "<stdin>", line 1
    if a = 10:
         ^
SyntaxError: invalid syntax
>>>
```

故障分析及排除

Tedu.cn 达内教育

知识讲解

- 原因分析
  - 在python中，比较是否相等应该用双等号
  - 一个等号是赋值操作
- 解决办法
  - 将a=10改为a==10