

nsd1905_py02_day04

复杂的列表排序

- 列表的sort方法有一个名为key的参数
- key可以是一个函数，该函数处理列表中的每一项，返回值作为排序依据

```
>>> ips = {'172.40.58.150': 10, '172.40.58.124': 6, '172.40.58.101': 10, '127.0.0.1': 121,
'192.168.4.254': 103, '192.168.2.254': 110, '201.1.1.254': 173, '201.1.2.254': 119,
'172.40.0.54': 391, '172.40.50.116': 244}
>>> iplist = list(ips.items())
>>> iplist
[('172.40.58.150', 10), ('172.40.58.124', 6), ('172.40.58.101', 10), ('127.0.0.1', 121),
('192.168.4.254', 103), ('192.168.2.254', 110), ('201.1.1.254', 173), ('201.1.2.254', 119),
('172.40.0.54', 391), ('172.40.50.116', 244)]
>>> def get_data(seq):
...     return seq[-1]
>>> iplist.sort(key=get_data, reverse=True)
>>> iplist
[('172.40.0.54', 391), ('172.40.50.116', 244), ('201.1.1.254', 173), ('127.0.0.1', 121),
('201.1.2.254', 119), ('192.168.2.254', 110), ('192.168.4.254', 103), ('172.40.58.150',
10), ('172.40.58.101', 10), ('172.40.58.124', 6)]

>>> iplist = list(ips.items())
>>> iplist.sort(key=lambda seq: seq[-1])
>>> iplist
[('172.40.58.124', 6), ('172.40.58.150', 10), ('172.40.58.101', 10), ('192.168.4.254',
103), ('192.168.2.254', 110), ('201.1.2.254', 119), ('127.0.0.1', 121), ('201.1.1.254',
173), ('172.40.50.116', 244), ('172.40.0.54', 391)]
```

安装模块

- python有大量的模块，这些模块没有全部作为标准安装的一部分。使用时，需要单独安装
- 官方站点：<https://pypi.org/>
- 安装python包，用pip命令。方法与rpm包类似

```
# 在线安装软件包，默认联系的是国外官方站点，慢
# 可以使用国内镜像站点加速
[root@room8pc16 ~]# mkdir ~/.pip/
[root@room8pc16 ~]# vim ~/.pip/pip.conf
[global]
index-url = http://pypi.douban.com/simple/
[install]
trusted-host=pypi.douban.com

# 例如：在线安装ipy模块
(nsd1905) [root@room8pc16 day04]# pip install ipy
```

```
>>> import IPy

# 本地安装
# tar xzf /linux-soft/05/zzg_pypkgs.tar.gz -C ~
(nsd1905)[root@room8pc16 ~]# pip install zzg_pypkgs/pymysql_pkgs/*
>>> import pymysql
```

数据库

创建示例数据库

需要为一个小公司创建数据库，数据库中需要记录员工信息和发工资的情况。

需要的字段有：姓名、性别、联系方式、部门、出生日期、工资日、基本工资、奖金、实发工资

数据库范式：数据库范式有第一范式、第二范式、第三范式、巴斯-科德范式、第四范式、第五范式六种。如果到第五范式就是完美范式，但是很难实现。关系型数据库至少要满足第三范式。

所谓第一范式（1NF）是指在关系模型中，对域添加的一个规范要求，所有的域都应该是原子性的，即数据库表的每一列都是不可分割的原子数据项。联系方式由家庭住址、email、电话号码等组成，它不满足原子性，需要拆分。

第二范式（2NF）是在第一范式（1NF）的基础上建立起来的，即满足第二范式（2NF）必须先满足第一范式（1NF）。第二范式（2NF）要求数据库表中的每个实例或记录必须可以被唯一地区分。简单来说，就是表需要一个主键。

员工表：员工ID、姓名、出生日期、email、部门ID

部门表：部门ID、部门名称

工资表：id、工资日、员工ID、基本工资、奖金、实发工资

第三范式（3NF）是第二范式（2NF）的一个子集，即满足第三范式（3NF）必须满足第二范式（2NF）。简而言之，非主信息不能依赖于其非主信息。如，工资表中的实发工资，它依赖基本工资和奖金。因此，这个字段不应该出现在数据库中，应该是由程序计算。

准备mariadb数据库

```
[root@room8pc16 ~]# mysql -uroot -ptedu.cn
MariaDB [(none)]> CREATE DATABASE nsd1905 DEFAULT CHARSET utf8;
```

通过pymysql操作数据库

操作数据库的步骤：

- 创建到数据库服务器的连接
- 创建游标。游标可以理解为和文件对象一样。通过文件对象实现对文件的读写，通过游标可以实现对表的增删改查
- 能过游标执行SQL语句
- 关闭游标、关闭连接

SQLAlchemy

- 不需要书写SQL语句
- 不限于mysql数据库，还可以连接oracle / sqlite / sql server / postgresql / mysql
- 使用ORM

ORM

- Object：面向对象编程中的对象
- Relationship：关系，关系型数据库
- Mapper：映射
- 数据库中表和sqlalchemy中的类映射，一张表对应一个类
- 类中的类属性和表中的字段映射，字段名和类变量名一样
- 表的每个字段都对应Column的一个实例
- 数据库中每种数据类型也都映射为一个sqlalchemy的类
- 类的实例，映射为表中的一行记录

sqlalchemy的应用

安装

```
(nsd1905)[root@room8pc16 ~]# pip install zzg_pypkgs/sqlalchemy_pkgs/*
```

创建数据库

```
MariaDB [nsd1905]> CREATE DATABASE tedu1905 DEFAULT CHARSET utf8;
```