

nsd1907_py02_day04

复杂列表的排序

- 列表的sort方法接受一个名为key的参数
- key可以是一个函数。该函数处理列表中的每一项，将处理结果(函数返回值)作为排序依据

```
>>> adict = {'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}
>>> alist = list(adict.items())
>>> alist
[('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 func1(seq):
...     return seq[-1]
...
>>> alist.sort(key=func1)
>>> alist
[('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)]
# 改为匿名函数，并且实现降序排列
>>> alist.sort(key=lambda seq: seq[-1], reverse=True)
>>> alist
[('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)]
```

模块安装

- python包官方站点：<https://pypi.org/>

在线安装软件包

```
# pip install xxxx
```

在线安装，默认使用的是国外网站，也可以设置，使用国内镜像站点：

```
[root@room8pc16 day02]# mkdir ~/.pip/
[root@room8pc16 day02]# vim ~/.pip/pip.conf
[global]
index-url = http://pypi.douban.com/simple/
[install]
trusted-host=pypi.douban.com
```

本地安装

注意：需要在虚拟环境中安装

```
# ls /linux-soft/05/
zzg_pypkgs.tar.gz
# tar xf zzg_pypkgs.tar.gz
# cd zzg_pypkgs
# pip install pymysql_pkgs/*
```

数据库

你正在为一个小型公司编写基础数据库，数据库需要记录员工、部门和发工资的信息。

收集字段：姓名、性别、联系方式、部门、出生日期、工资日、底薪、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。

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。如联系方式不能成为一个字段，需要将其拆分成家庭住址、电话号码、email等。

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。简单来说就是每个表需要有个主键

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

工资表：id、工资日、员工ID、底薪、奖金、实发工资

第三范式（3NF）要求一个关系中不包含已在其它关系已包含的非主关键字信息。简单来说，非主字段，不能依赖其他非主字段。实发工资是由底薪、奖金算出来的，所以它不应该出现在数据库表中。用的时候，应该是通过程序计算。

创建数据库：

```
[root@room8pc16 day02]# mysql -uroot -ptedu.cn
MariaDB [(none)]> CREATE DATABASE nsd1907 DEFAULT CHAR SET utf8;
```

pymysql操作数据库

- 连接

- 创建游标。游标就像是文件对象，通过文件对象可以对文件实施读写操作，通过游标就可以对数据库进行增删改查。
- 执行SQL语句
- 如果执行的语句是增删改，必须commit确认
- 关闭

sqlalchemy

- sqlalchemy采用ORM管理数据库
- O : Object对象
- R : Relationship关系
- M : Model模型
- sqlalchemy将一个Class与数据中的一张表相关联
- sqlalchemy将类变量与表的字段关联
- sqlalchemy将数据库的各种数据类型定义成了相关的Class
- sqlalchemy将Class的实例与表的记录关联

安装

```
(nsd1907) [root@room8pc16 day04]# pip install /var/ftp/pub/zzg_pypkgs/sqlalchemy_pkgs/*
```

创建新的数据库

```
MariaDB [nsd1907]> CREATE DATABASE tedu1907 DEFAULT CHAR SET utf8;
```