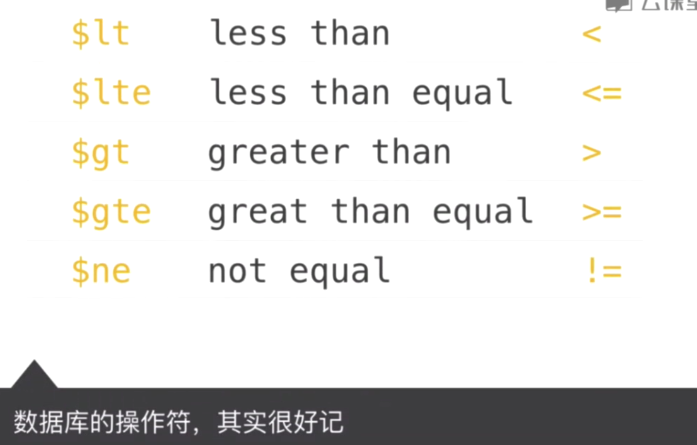
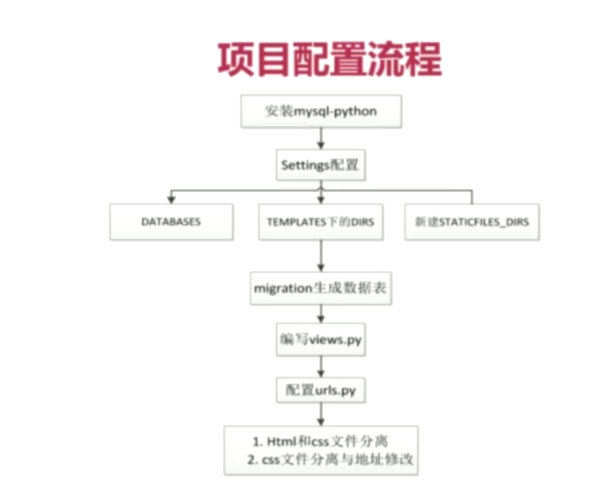


MongoDB数据库：





连接MySQL数据库



Django 知识点：

**static**

在settings添加

STATICFILES\_DIRS = (os.path.join(BASE\_DIR, 'static'),)

在templates 的html中，首行添加

{% load static%}

其余所有的css或者js样式，都要改为 “{% static 原文 %}”

**context:** 上下文，

非常注重 键，对于值没有那么重要

在基本环境

可以使用上下文。

view.py修改

def index(request):

context = {

'title':'Just a title',

'des':'Just a dec',

'score':'1.0'

}

return render(request,'index.html',context)

说明‘index.html’里面可以引用context里面的参数

templates里index.html修改的地方, 此处随便找个地方，如下。**{{ xxx }}**

'''

<h3><a href="#">{{ title }}</a></h3>

<p class="description">{{ des }}</p>

<span class="rate-score">{{ score }}</span>

'''

**连接MongoDB数据库：**

轮到model了

先在settings.py

增加

'''

#连接MongoDB数据库

from mongoengine import connect

connect('wbsite', host='127.0.0.1', port=27017)

'''

到model.py

#偷懒，全部导入

from mongoengine import\*

新建一个类

class ArtiInfo(Document):

des = StringField()

title = StringField()

scores = StringField()

tags = ListField(StringField())

meta = {

‘collection’: ‘arti\_info3’

#如果新建一个数据库，就不用写这句话，这个是已经存在的数据库

}

Django的模板语言

**分页：**

**from django.core.paginator import Paginator**

**iter="abcdefghijklmn"**

**paginator = Paginartor(iter,4) #将iter限制在每页为4条**

**paginator.page(1).object\_list**

**>>abcd**

**paginator.page(3).object\_list**

**>>ijkl**

**paginator.page(3).has\_next()**

**>>True**

**paginator.page(3).number #当前页数**

**>>3**

**paginator.num\_pages #总的页数**

**>>4**

**{% for i in a %} #循环可以嵌套，但每次都要结束endif**

**{%endfor%}**

**{%if a %}**

**{% endif %}**

**主模板与子模板：**

**字模板相等于继承主模板的的固定的部分，其余需要改变的地方，替换就好了**

**主模板**

**main.html**

**在需要替换的代码位置写上**

**{% block name1 %}**

**{% end block%}**

**{% block name2 %}**

**{% end block%}**

**子模板**

**small.html**

**{% extends 'main.html' %} #首行**

**{% block name1 %}**

**中间填写要替换的代码**

**{% end block %}**

**{% block name2 %}**

**中间填写要替换的代码**

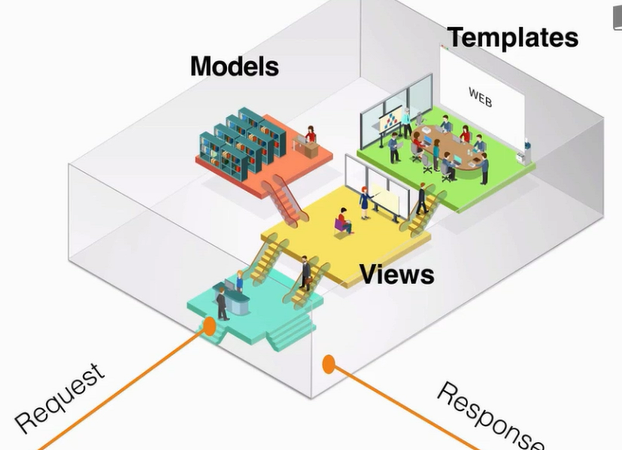
**{% end block %}**

**取出文件:为字典结构**

**xx.\_get\_collection().aggregate(xx)**

**semantic-UI 快速构建网页模板（进入官网查看套用就可以）**





views

templates

models

URL