```
Django模板语言,简写DTL
                                         动态部分,就是模板语言
                           模板包含两部分
                                         静态部分,包含html、css、js
                                  模版致力于表达外观,而不是程序逻辑
                                   模板的设计实现了业务逻辑view与显示内容template的解耦
                           概念
                                   一个视图可以使用任意一个模板,一个模板可以供多个视图使用
                                   作为Web框架,Django提供了模板,用于编写html代码
                                     然后在templates目录下创建应用同名目录
                 模板介绍
                                                                  用于区分哪个模板是属于哪个应用的
                                     然后在应用同名目录下创建模板文件, test.html
                           模板配置
                                     首先在项目的根目录下创建templates目录
                                     最后在settings.py文件中,配置模板加载路径即可
                                                                     os.path.join(BASE_DIR, 'templates')
                                                为了减少开发人员重复编写加载、渲染的代码,Django提供了简写函数render,用于加载、渲染模板
                           Django处理模板分为两个阶段
                                                1.加载:根据给定的路径找到模板文件,编译后放在内存中
                                                2.渲染: 使用上下文数据对模板传值并返回动态生成的网页
                           作用
                                  获取、处理传入到模板中的上下文数据,动态的产生页面内容
                                          语法
                                                  {% 代码段 %}
                                                               {% for item in 列表 %}
                                                               执行循环逻辑
                                                                {{ forloop.counter }}获取当前是第几次循环,从1开始
                                                    for循环标签
                                                               {% empty %}
                                                               列表为空或不存在时执行此逻辑
                                                               {% endfor %}
                                                            {% if ... %}
                                                            逻辑1
                                   标签
                                                            {% elif ... %}
                                          常用标签
                                                            逻辑2
                                                    if标签
                                                            {% else %}
                                                            逻辑3
                                                            {% endif %}
                                                                  运算符左右两侧必须有空格
                                                    比较运算符标签
                                                                                 > <= >=
                                                    布尔运算符标签
                                                                  and
                                                                                 not
                                                   模板语言中不允许带参数的函数,如果要对变量进行处理,就需要使用过滤器
                                           作用
                                                   使用管道符号 | 来应用过滤器,用于进行计算、转换操作,可以使用在变量、标签中
                                                   两个参数时: 变量|过滤器:参数
                                           语法
                                                   一个参数时: 变量|过滤器
                                                     第一个参数就是变量
                                   过滤器
                                                     如果过滤器需要追加参数,则使用冒号:传递参数
                                           语法说明
                                                     冒号后面最多可以传一个参数
                                                                                        变量|length
                                                      length,返回字符串、列表、元组、字典的元素个数
                                           常用过滤器
                                                                                  变量|default:默认值
                                                      default,如果变量不存在时则返回默认值
                                                      date,用于对日期类型的值进行字符串格式化
                                                 {{ 变量 }}
                                          语法
                                                     由字母、数字、下划线和点组成
                                          变量名规则
                                                      变量名不能以下划线开头
                 模板语言
                           分类
                                                       当模版引擎遇到点如book.name
                                                       2.先属性后方法,将book当作对象,查找属性name,如果没有再查找方法name()
                                          变量解析规则
                                   变量
                                                      1.字典book['name']
                                                      3.如果是格式为book.0则解析为列表book[0]
                                                  如果变量不存在则插入空字符串 ''
                                          提示
                                                  如果要使用带参数的方法,需要使用过滤器
                                                  在模板中调用方法时不能传递参数
                                               当内建过滤器无法满足需求时, 需要自定义过滤器
                                               过滤器就是python中的函数,注册后就可以在模板中当作过滤器使用
                                                          创建python包命名为templatetags
                                                          在新建的templatetags包中新建一个py文件
                                                                   from django.template import Library
                                                                  # 创建注册对象,命名为register
                                                                   register = Library()
                                               自定义过程
                                   自定义过滤器
                                                                   # 把以下函数注册成过滤器
                                                          编码
                                                                   @register.filter
                                                                   # 定义求奇偶数的函数
                                                                   def mod(num):
                                                                    # 如果为偶数返回NO。反之,返回YES
                                                                    return num%2
                                                       模板中需要使用load标签加载自定义过滤器
                                               使用
                                                       {#加载过滤器#}
                                                       {% load customfillter %}
                                          在模板中使用模板注释,注释掉某一段代码,那么这段代码不会被编译,不会输出到客户端
                                          html注释只能注释html内容,不能注释模板语言
                                          注释可以包含任何模版代码,有效的或者无效的都可以注释掉
                                   注释
                                                    {#...#}
                                          单行注释
模板(Template)
                                                    {% comment %}
                                          多行注释
                                                    {% endcomment %}
                           模板继承和类的继承含义是一样的,主要是为了提高代码重用,减轻开发人员的工作量
                           典型应用: 网站的头部、尾部信息
                                           如果发现一段代码在多个模板中出现,那就应该把这段代码内容定义到父模板中
                                    前提
                                           {% block 名称 %}
                                           预留区域,可以编写默认内容,也可以没有默认内容
                                    语法
                                           {% endblock 名称 %}
                                           父模板中也可以使用上下文中传递过来的数据
                           父模板
                                    特点
                                           父模板定义在templates文件目录下
                                              block标签:用于在父模板中预留区域,留给子模板填充差异性的内容
                                              可以给预留区域定义名字, 但多个预留区域名字不能相同
                                    block标签
                                              为了更好的可读性,建议给endblock标签写上名字,这个名字与对应的block标签名字相同
                                           子模板定义在templates/应用文件目录下
                 模板继承
                                           extends标签:继承,写在子模板文件的第一行
                                    特点
                                           子模版不用填充父模版中的所有预留区域,如果子模版没有填充,则使用父模版定义的默认值
                                                       {#继承父模板#}
                           子模板
                                           继承父模板
                                                       {% extends 'base.html' %}
                                                      {% block 名称 %}
                                    语法
                                                       实际填充内容
                                           子模板使用
                                                      {{ block.super }} 用于获取父模板中block的内容,也可以不获取
                                                      {% endblock 名称 %}
                                    父模板定义在templates文件目录下
                                    子模板定义在templates/应用文件目录下
                           注意点
                                    子模版继承父模板之后,本身原有的内容是不起作用的
                                    父模板不能把上下文传给子模板,只继承内容不继承上下文
                                      小于号 < 转换为 &lt;
                                      大于号 > 转换为 >
                            转义符号
                                      单引号 '转换为 '
                                      双引号"转换为 "
                                      与符号 & 转换为 & amp;
                            作用
                                   转义后标记代码不会被直接解释执行,而是被直接呈现,防止客户端通过嵌入js代码攻击网站。
                 HTML转义
                            概念
                                   在视图中,通过调用模板传递上下文,模板对上下文传递的字符串进行输出时,会对以下字符自动转义
                                      {{ 变量|escape }}
                                                    过滤器escape:可以实现对变量的HTML转义,默认模板就会转义,一般省略
                                      {{ 变量|safe }}
                                                   过滤器safe:禁用转义,告诉模板这个变量是安全的,可以解释执行
                            防止转义
                                                        标签autoescape:设置一段代码都禁用转义,接受on、off参数
                                      {% autoescape off %}
                                                        on: 启用转义
                                      {% endautoescape %}}
                                                        off: 禁用转义
                                    Django默认开启了CSRF安全验证功能
                                                               在中间件设置项中可以看到CSRF安全验证功能的存在
                                     当启用中间件并加入csrf_token标签后,服务器会向客户端浏览器中写入一条Cookie信息
                         防止被攻击
                                    这条Cookie信息的值与csrf_token标签生成的隐藏域的value是一致的
                                                                                     如果对比失败则返回403页面,而不会进行后续的处理
                                    csrf_token标签生成的隐藏域的value提交到服务器后会先由csrf中间件进行验证
                                                                                     如果对比成功就继续处理请求
                                 CSRF全拼为Cross Site Request Forgery, 译为跨站请求伪造
                 CSRF
                                 CSRF指攻击者盗用了你的身份,以你的名义发送恶意请求
                         概念
                                 包括:以你名义发送邮件,发消息,盗取你的账号,甚至于购买商品,虚拟货币转账......
                                 造成的问题:个人隐私泄露以及财产安全
                         容易被攻击
                                    客户端访问服务器时没有同服务器做安全验证
                         提示
                                 重要信息发送POST请求提交到服务器
                                 在用户注册、登录页面,为了防止暴力请求,可以加入验证码功能,如果验证码错误,则不需要继续处理,可以减轻业务服务器、数据库服务器的压力
                          提示
                                                  用于制作验证码
                                    安装Pillow3.4.1包
                                    定义生成验证码的视图
                 验证码
                          实现步骤
                                    准备输入验证码的网页
                                                                                   验证成功
                                                                            一致
                                    接收客户端发送的验证码
                                                      并与服务器生成的验证码比较
                                                                            不一致
                                                                                    验证失败
                                                      http://127.0.0.1:8000/fan1
                                     使用请求地址匹配正则
                                                      url(r'^fan1/$', fan1)
                                                   当正则变化,地址不变就会匹配失败
                                     可能出现的问题
                                                   正则变化 url(r'^fan123/$', fan1)
                           常规解析
                                                                              再匹配 http://127.0.0.1:8000/fan1 就出错
                                     设想
                                             是否有一种机制
                                                          当正则变化,地址根据正则的变化自动生成
                                             反向解析
                                                      根据正则动态生成请求地址
                 反向解析
                                     解决
                                             反向解析应用在模板中的超链接, 视图中的重定向
                                                                  # 常用解析
                                                                  url(r'^', include('Book.urls')),
                                             在项目中添加namespace属性
                                                                  # 反向解析
                                                                  url(r'^', include('Book.urls', namespace='Book'))
                                     步骤
                           反向解析
                                                               url(r'^fan123/$', views.fan1, name='fan1'),
                                             在应用中添加name属性
```

动态地址跳转到fan1

模板中动态生成超链接