# 模板T

## 模板语言

1. 模板变量：

数字、字母、下划线和点组成，不能下划线开头

使用模板变量：

{{ 模板变量名 }}

解析顺序：

如：{{ book.btitle }}

把book看成字典，btitle为键名

把book当成对象，btitle为属性

把book当成对象，btitle为方法

如：{{ book.0 }}

把book当成字典，0为键名

把book当成列表，0为下标

如果解析失败，返回空字符串填充模板变量

1. 模板标签：

{% 代码段 %}

for循环：

{% for x in 列表 %}....

{% empty %}

列表为空时执行

..{% endfor %}

{% forloop.counter %}可以得到for循环到了第几次

条件判断：

{% if 条件 %}

{% elif 条件 %}

{% else %}

{% endif %}

关系比较操作符：

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

**比较操作符两边必须有空格**

逻辑运算：

Not and or

1. 过滤器：用于对模板变量进行操作

data: 改变日期的显示格式

length: 求长度，针对字符串和列表

default: 设置模板变量的默认值，不设置的话空字符串填充

格式：模板变量|过滤器:参数

## 模板继承

编写base.html

预留块：

{% block 块名 %}

可写可不写内容

{% endblock 块名 %}

**路径是从templates下开始**

子模板文件:

{% extends ‘booktest/base.html’ %}

{% block 块名 %}

重写内容

{% endblock 块名 %}

## html转义

（使用模板时)在模板上下文中的html标签默认会被转义

小于 < 转换为 &lt；

大于 > 转换为 &gt；

单引号 ‘ 转换为 &#39；

双引号 “ 转换为 &quot；

与符号 & 转换为 &amp；

要关闭**模板上下文字符串**的转义：可以使用**{{ 模板变量|safe }}**

也可以：

{% autoescape off %}

模板语言代码

{% endautoescape off %}

模板硬编码中的字符串默认不会转义：{{ test|default:’<h1> ...</h1>’}}

手动转义：{{ test|default:’&lt;h1&gt; ...&lt;/h1&gt;’}}

## 四、csrf攻击

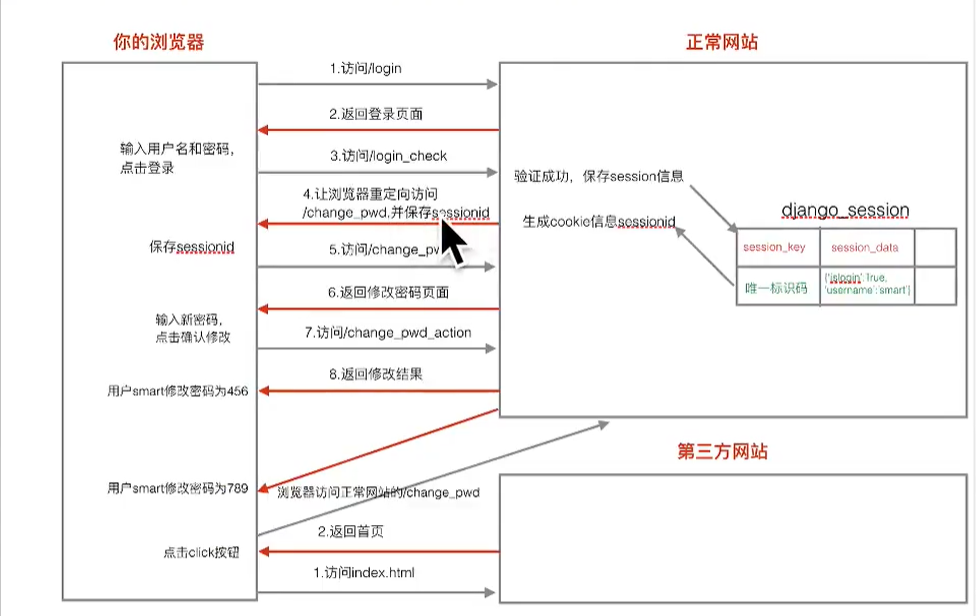
1、跨站请求伪造

1）当你登录某个正常网站后，你的浏览器保存了seeionid，且登录状态仍保持即没有退出

2）你又访问了某个网站，其中你进行某个操作会发起对你之前登录的网页中某些修改信息的请求

<input type=”hidden” name=”pwd” value=”789”>

3）此时，由于登录状态保持，url请求也是基于你的浏览器，那么就可以进入并进行信息的修改



1. Django本身默认启用了**csrf防护中间件**，只针对post提交，所以重要数据用post提交

自己的网站在所有表单post提交的时候加上一个模板标签：

<form method=”post” action=”/......”>

正式编写时用反向解析

{% csrf\_token %}

.......

防护原理：

1、渲染模板文件时{% csrf\_token %}被替换为一个csrfmiddlewaretoken的隐藏域

<input type=”hidden” name=”csrfmiddlewaretoken” value=”...”>

2、服务器会交给浏览器保存一个名为csrftoken的cookie信息，这条信息的值与隐藏域input元素的value属性是一致的

3、提交表单时，csrfmiddlewaretoken的隐藏域的value值和cookie值都会发给服务器，提交到服务器后会先由csrf中间件进行验证，如果对比失败，则返回403页面，而不会进行后续的处理。

一个网站是不好拿到**其他人**访问网站的隐藏域的

## 五、反向解析

具体可以看**1-项目开发流程.docx**