**Html**

## 一、Html介绍

1 超文本标记语言（英语：HyperText Markup Language，简称：HTML）**是一种用于创建网页的标准标记语言。可以使用 HTML 来建立自己的 WEB 站点，HTML 运行在浏览器上，由浏览器来解析。**

## 二、优点

1、静态HTML页面直接访问速度快;

2、生成的静态页面从理论上讲是没有攻击漏洞的，相对安全;

3、生成的静态页面访问时完全脱离了数据库访问的压力，不存在数据库查询操作;

4、目前生成静态网页是一种趋势;

5、静态网页的内容非常稳定，容易被搜索引擎检索，有利于SEO搜索引擎优化;

6、生成的静态页面可以降低服务器承受的能力，无需解析即可返回客户端，保证服务器正常运行。

## 三、缺点

1、信息量大时生成过程慢长，对服务器压力较大;

2、模板的制作和修改不直观，比传统的动态或静态页面难度大，不易于新手操作;

3、数据过多时，生成的HTML页面无形中也耗费大量的磁盘空间以存放这些静态文件;

4、只要是进行信息操作(添加/修改/删除)，就

## 四、Classify Compare

**1 HTML 和 XHTML**

HTML是一种基本的WEB网页设计语言，XHTML是一个基于XML的置标语言，看起来与HTML有些相象，只有一些小的但重要的区别，XHTML就是一个扮演着类似HTML的角色的XML，所以，本质上说，XHTML是一个过渡技术，结合了XML(有几分)的强大功能及HTML(大多数)的简单特性。HTML 和 XHTML 的区别简单来说，XHTML 可以认为是 XML 版本的 HTML，为符合XML 要求，XHTML 语法上要求更严谨些。以下是 XHTML 相对 HTML 的几大区别：

XHTML 要求正确嵌套

XHTML 所有元素必须关闭

XHTML 区分大小写

XHTML 属性值要用双引号

XHTML 用 id 属性代替 name 属性

XHTML 特殊字符的处理

**2 HTML 和 HTML5**

**1）. 简化的语法**

HTML5简化了很多细微的语法，例如doctype的声明，你只需要写<!doctype html>就行了。HTML 5 指定 UTF-8 编码的方式如下<meta charset="UTF-8">

**2）. <canvas>标签替代Flash**

Flash给很多Web开发者带来了麻烦，要在网页上播放Flash需要一堆代码和插件。<canvas>标签使得开发者只要使用一个标签就能和用户产生UI交互。

**3） 新增许多标签**

HTML5设计的一个原则是更好的体现网站的语义性，所以增加了<header>和<footer>这样的标签，用来明确表示网页的结构；新增 <section> 和 <article> 标签，<section>和<article>也有利于清晰化网页的结构，更有利于SEO；新增 <menu> 和 <figure> 标签，<menu>可以被用于创建传统的菜单，也可以用于工具栏和上下文菜单。<figure>标签使得网页文字和图片的排版更专业； 新增 <audio> 和 <video> 标签，这两个标签可能是HTML5里面最有用的两个标签了，使用起来时也异乎方便，如在HTML5页面中嵌入视频只需一小段

<video width="450" height="340" controls>

<source src="jamshed.mp4" type="video/mp4">

Your browser does'nt support video embedding feature.

</video>

**4）全新的表单**

HTML5对 <form> 和 <forminput> 标签进行了大量修改，添加了很多新的属性，也修改了很多属性，详细http://www.jb51.net/w3school/html5/html\_5\_form\_attributes.htm

**5）删除 <b> 和 <font> 标签，<frame>, <center>, <big> 标签**

**6）HTML5 支持了不同类型的存储类**型

HTML5 支持本地存储，在之前版本中是通过 Cookie 实现的。HTML5 本地存储速度快而且安全。并且HTML5有两种不同的对象可用来存储数据，HTML5通过JS来存储和访问数据localStorage 适用于长期存储数据，浏览器关闭后数据不丢失sessionStorage 只是针对一个session的数据存储，存储的数据在浏览器关闭后自动删除。总的来说，HTML5已经超越了标记语言的范畴，更富语义的标签将使得HTML5更有用处。Canvas+WEBGL等技术，实现无插件的动画以及图像、图形处理能力；本地存储，可实现offline应用；websocket，一改http的纯pull模型，实现数据推送的梦想；MathML，SVG等，支持更加丰富的render等等等等，现在对于对HTML5的了解还只停留在表面阶段，以后会多阅读一部分HTML5的书籍增加一下见识。

五、flash 代码的引用

**1 在网页中嵌入一段swf视频**，加载网页时不播放，点击下面的播放按钮实现播放

首先在网页中嵌入Object代码：

<object classid="clsid:d27cdb6e-ae6d-11cf-96b8-444553540000" width="400" height="300"

codebase="http://fpdownload.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=6,0,0,0" id="swf">

<param name="movie" value="images/zf03320.swf">

<param name="quality" value="high">

<param name="bgcolor" value="#F0F0F0">

<param name="menu" value="false">

<param name="wmode" value="opaque">

<param name="FlashVars" value="">

<param name="allowScriptAccess" value="sameDomain">

<param name="loop" value="false">

<param name="play" value=""false"">

<embed name="swf" src="images/zf03320.swf"

width="400" height="300"

align="middle"

quality="high"

menu="false"

play="false"

loop="false"

FlashVars=""

allowScriptAccess="sameDomain"

type="application/x-shockwave-flash"

pluginspage="http://www.adobe.com/go/getflashplayer">

</embed>

</object>

object标签是用于windows IE3.0及以后浏览器或者其它支持Activex控件的浏览器。“classid”和“codebase”属性必须要精确地按上例所示的写法写，它们告诉浏览器自动下载flash player的地址。如果你没有安装过flash player 那么IE3.0以后的浏览器会跳出一个提示框访问是否要自动安装flash player。当然，如果你不想让那些没有安装flash player的用户自动下载播放器，或许你可以省略掉这些代码。

embed标签是用于Netscape Navigator2.0及以后的浏览器或其它支持Netscape插件的浏览器。“pluginspage”属性告诉浏览器下载flash player的地址，如果还没有安装flash player的话，用户安装完后需要重启浏览器才能正常使用。

为了确保大多数浏览器能正常显示flash，你需要把EMBED标签嵌套放在OBJECT标签内，就如上面代码例子一样。支持Activex控件的浏览器将会忽略object标签内的embed标签。Netscape和使用插件的IE浏览器将只读取EMBED标签而不会识别OBJECT标签。也就是说，如果你省略了embed标签，那firefox就不能识别你的flash了（不过纳闷的是，省略了object只写embed，IE也能正常显示flash，呵呵，具体的再仔细看看了）。

下面列出了用于发布影片的object和embed标签的必要以及可选属性。

下面附带Object中的各个参数的意义表

**2 必需属性：**

calssid-设置浏览器的Activex控件，仅用于OBJECT标签。

·codebase-设置flash Activex控件的位置，因而如果浏览器如果没有安装的话，可以自动下载安装。仅用于OBJECT标签。

width-以百分比或象素指定flash影片的宽度。

height-以百分比或象素指定flash影片的高度。

src-指定影片的下载地址。仅用于EMBED标签。

pluginspage-设置flash 插件的位置，因而如果浏览器如果没有安装的话，可以自动下载安装。仅用于EMBED标签。

movie-指定影片的下载地址。仅用于OBJECT标签。

**3 可选属性及可用的值：**

id-设定变量名，用于脚本代码的引用。仅用于OBJECT。

name -设定变量名，用于脚本代码（如JavaScript）的引用。仅用于EMBED。

## 六、js控制flash

**1 代码**

<input type="button" value="播放SWF" id="playSWF">

<script type="text/javascript">

var btn=document.getElementById("playSWF");

btn.onclick=function()

{

document.swf.Play();

}

</script>

**2 一些函数**

Play() ---------------------------------------- 播放动画

StopPlay()------------------------------------停止动画

IsPlaying()----------------------------------- 动画是否正在播放

GotoFrame(frame\_number)---------------- 跳转到某帧

TotalFrames()------------------------------- 获取动画总帧数

CurrentFrame()------------------------------回传当前动画所在帧数-1

Rewind()-------------------------------------使动画返回第一帧

SetZoomRect(left,top,right,buttom)-------放大指定区域

Zoom(percent)------------------------------改变动画大小

Pan(x\_position,y\_position,unit)------------使动画在x,y方向上平移

PercentLoaded()----------------------------返回动画被载入的百分比

LoadMovie(level\_number,path)----------- 加载动画

TGotoFrame(movie\_clip,frame\_number)- movie\_clip跳转到指定帧数

TGotoLabel(movie\_clip,label\_name)------ movie\_clip跳转到指定标签

TCurrentFrame(movie\_clip)--------------- 回传movie\_clip当前帧-1

TCurrentLabel(movie\_clip)-----------------回传movie\_clip当前标签

TPlay(movie\_clip)---------------------------播放movie\_clip

TStopPlay(movie\_clip)----------------------停止movie\_clip的播放

GetVariable(variable\_name)-----------------获取变量

SetVariable(variable\_name,value)-----------变量赋值

TCallFrame(movie\_clip,frame\_number)---call指定帧上的action

TCallLabel(movie\_clip,label)----------------call指定标签上的action

TGetProperty(movie\_clip,property)--------获取movie\_clip的指定属性

TSetProperty(movie\_clip,property,number)-设置movie\_clip的指定属性

## 七、 实例分析

1 test.html

<script type="text/javascript" src="advertiser-flash.js"></script>

<div style="width:40%" id="content\_flash">

<script type="text/javascript" language="javascript">DefaultPlay("flash/1.swf");</script>

</div>

<div style="width:60%">

<ul class="left\_nav1" style="margin-top:10px;">

<li class="left\_nav1\_li left\_nav1\_li\_default " onclick="KeyPlay('flash/1.swf');">

</li>

<li class="left\_nav1\_li" onclick="KeyPlay('flash/2.swf');">

</li>

<li class="left\_nav1\_li" onclick="KeyPlay('flash/3.swf');">

</li>

<li class="left\_nav1\_li" onclick="KeyPlay('flash/4.swf');">

</li>

<li class="left\_nav1\_li" onclick="KeyPlay('flash/5.swf');">

</li>

</ul>

</div>

2 advertiser-Flash.js：

var w = 350;

var h = 400;

function swf(file)

{

document.write('<object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000" codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=11,0,1,152" width="' + w + '" height="' + h + '"> ');

document.write('<param name="movie" value="' + file + '">');

document.write('<param name="quality" value="high"> ');

document.write('<param name="wmode" value="transparent"> ');

document.write('<param name="menu" value="false"> ');

document.write('<embed src="' + file + '" quality="high" pluginspage="http://www.macromedia.com/go/getflashplayer" menu="false" wmode="transparent" type="application/x-shockwave-flash" width="' + w + '" height="' + h + '"></embed> ');

document.write('</object> ');

}

function NewComing(file)

{

var htmlStr = null;

htmlStr = '<object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000" codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=11,0,1,152"';

htmlStr += ' width="' + w + '" height="' + h + '"> ';

htmlStr += '<param name="movie" value="' + file + '">';

htmlStr += '<param name="quality" value="high">';

htmlStr += '<param name="wmode" value="transparent">';

htmlStr += '<param name="menu" value="false"> ';

htmlStr += '<embed src="' + file + '" quality="high" pluginspage="http://www.macromedia.com/go/getflashplayer" menu="false" wmode="transparent" ';

htmlStr += 'type="application/x-shockwave-flash" width="' + w + '" height="' + h + '"></embed>';

htmlStr += '</object>';

return htmlStr;

}

function DefaultPlay(file)

{

swf(file);

}

function KeyPlay(file)

{

document.getElementById("content\_flash").innerHTML = NewComing(file);

}

## 八、iframe基础概述

frame是一个非常有用的标签，先不说早期用它来模型Ajax效果，现在富本文编辑器它也绝对是主角。但是它又是一个特别的元素，最早出现IE4.0中，后纷纷被其他游览器吸纳，由于IE不开源，iframe在各游览器中都有很大差异。本文针对IE9、FF、Chrome三大浏览器进行分析。

**1.获得iframe对象**

<IFRAME frameBorder=0 frameSpacing=0 height=25 marginHeight=0 marginWidth=0 scrolling=no name=main src="bgm/bgm.html" width=300></IFRAME>

我们常用的获得iframe的方式一般有两种

var obj = document.getElementById(“iframe”);//获取对象

var dom = window.frames[“iframe”];//获取DO

如果只想改变iframe的 src 或者 border ，scrolling 等attributes，就需要用到第一种方法。但是如果想取得iframe所包含的页面(不是iframe本身)。就需要使用第二种方法，因为它取得的是一个完整的DOM模型，比如想得到iframe的document.body的内容操作iframe中的DOM元素，需要注意以下两点：必须先获取指定iframe的document对于1，必须在页面load完以后才能获取；chrome浏览器只能使用在HTTP协议中，在本地js文件不支持Iframe包含页面的访问。

**2.改变iframe本身的属性**

document.getElementById(“iframe”).src="a.html";

document.getElementById(“iframe”).frameborder="200";

**3 在父窗体中访问子窗体对象**

var value=window.frames[“frm”].document.getElementById("txt").value;

window.frames[“frm”]是获得子窗口的window对象

**4 在子窗口访问父窗口对象**

var value=window.parent.document.getElementById("txt").value;

其中：window.parent是获得父窗口的window对象。

**5 .iframe高度自适应**

scrollHeight返回的是一个数字，没有带单位在IE 下如果没有DTD声明，则按loose.dtd 解析，在设置对像的宽高时，会自动的加上单位"px"。将iframe的高度设为document.body的高度，也并不一定能成功,因为document.body的高度可能没有整个档的高度大，如使用了层。正确写法如下.在子页面中进行设置frae的高度

var frm=window.parent.document.getElementById("frm");//获得父窗口

frm.style.height=document.body.scrollHeight + "px";//设置父窗口的高度

**6 .contentWindow和contentDocument**

我们开始使用window.frames["name"]获得frame中的页面window对象使用document.getElementById(id)只能获得frame对象但是通过frame对象的contentWindow属性或contentDocument也可以获得在这里由于frame存在跨域访问限制，所以用法不一。

var dom=document.getElementById("frm");

var value;

if(dom.conentWindow)

{

value=dom.conentWindow.document.getElementById("txt").value;//IE

}else

{

value=dom.conentDocument.getElementById("txt").value;//FF,chrome

}

**7 优点**

1）.iframe能够原封不动的把嵌入的网页展现出来。

2）.程序调入静态页面比较方便;

3）.页面和程序分离;

**8 缺点**

1）.会产生很多页面，不容易管理。

2）.iframe框架结构有时会让人感到迷惑，如果框架个数多的话，可能会出现上下、左右滚动条，会分散访问者的注意力，用户体验度差。

3）.代码复杂，无法被一些搜索引擎索引到，这一点很关键，现在的搜索引擎爬虫还不能很好的处理iframe中的内容，所以使用iframe会不利于搜索引擎优化。

4）.很多的移动设备（PDA 手机）无法完全显示框架，设备兼容性差。

5）.影响主页面的渲染，阻塞页面加载，性能差。

**9 include和iframe**

（1）include 动态页面 asp aspx；iframe 动态 静态均可使用

（2）include是代码级组合,iframe是视图级组合.

（3）include 作为引用页的一部分，在本页执行，iframe 独立引用页，单独执行

（4）调用include是服务器将两个JSP页合并成一个页面解释后发送到客户端,对于浏览器来说他见到的就是一个页面而已,自然样式通用于整个页面.但是iFrame

实际上就是在页面上开了一个区域来显示另外一个页面,实际上还是两个单独的页面,样式不会互相影响

（5）iframe好在能够把原先的网页全部原封不动显示下来,但是如果用在首页,是搜索引擎最套讨厌的.那么你的网站即使做的在好,也排不到好的名次!

include呢就可以iframe会让前进后退失效，而且搜索引擎爬的不方便

（6）iframe标记未来会被淘汰，能不用就不用如果只跳转本页的某一个区域还是用iframe好。如果是整页的话还是用include file

## 九、iframe 和 frame

**1、frame不能脱离frameSet单独使用，iframe可以**

**2、frame不能放在body中,否则不能正常显示frame不能和body同时使用**

<frameset rows="50%,\*">

<frame name="frame1" src="test1.htm"/>

<frame name="frame2" src="test2.htm"/>

</frameset>

iframe可以和body同时使用

<body>

<frameset>

<iframe height="30%"name="frame1" src="test1.htm"/>

<iframe height="100"name="frame2" src="test2.htm"/>

</frameset>

</body>

**3、嵌套在frameSet中的iframe必需放在body中**,不嵌套在frameSet中的iframe可以随意使用

**4、frame的高度只能通过frameSet控制**；iframe可以自己控制，不能通过frameSet控制

区别：frame是整个页面的框架，iframe是内嵌的网页元素，也可以说是内嵌的框架，可以用它将一个HTML文档嵌入在另一个HTML中显示。

最大区别是网页中嵌入<Iframe>包含的内容与整个页面是一个整体，而<Frame>包含的内容是独立的个体，是可以独立显示的。

另外，Iframe还可以在页面中多次显示同一内容，而不必重复这段内容的代码。

**5、 iframe 可以放到表格里面。frame 则不行。**

<table>

<tr>

<td><iframe id="" src=""></iframe></td><td></td>

</tr>

</table>

**6、iframe是活动帧， 而frame是非活动帧**

iframe用起来更灵活，不需要frame那么多讲究，而且放的位置也可以自己设。

iframe是内嵌的，比较灵活，不过也有不好的地方，就是位置在不同的浏览器和分辨率下有可能不同，有时会把本来好好的页面搞得变形。

iframe是一个网页中的子框架,两网页间是父子关系，

iframe是一个浮动的框架,就是在你的页面里再加上一个页面。

frame用来把页面横着或竖着切开，<iframe>用来在页面中插入一个矩形的小窗口。

frame是把网页分成多个页面的页面。它要有一个框架集页面frameset。

**7 使用场景**

1） 嵌入网页

2） 跨域远程访问

## 九、FreeMarker简介

FreeMarker是一个模板引擎，一个基于模板生成文本输出的通用工具，使用纯Java编写。FreeMarker被设计用来生成HTMLWeb页面，特别是基于MVC模式的应用程序。所谓模板，就是一份已经写好了基本内容，有着固定格式的文档，其中空出或者用占位符标识的内容，由使用者来填充，不同的使用者给出的数据是不同的。在模板中的占位符，在模板运行时，由模板引擎来解析模板，并采用动态数据替换占位符部分的内容。FreeMarker不是一个Web应用框架，而适合作为Web应用框架一个组件，FreeMarker与Web容器无关，即在Web运行时，它并不知道Servlet或HTTP。它不仅可以用作表现层的实现技术，而且还可以用于生成XML，JSP或Java文件等。

**1、java操作**

（1）通过String来创建模版对象，并执行插值处理

public static void main(String[] args) throws Exception

{

//创建一个模版对象

Template t = new Template(null, new StringReader("用户名：${user};URL： ${url};姓名： 　${name}"), null);

//创建插值的Map

Map map = new HashMap();

map.put("user", "lavasoft");

map.put("url", "http://www.baidu.com/");

map.put("name", "百度");

//执行插值，并输出到指定的输出流中

t.process(map, new OutputStreamWriter(System.out));

}

（2）通过文件来创建模版对象，并执行插值操作

public void init() throws Exception

{

//初始化FreeMarker配置

//创建一个Configuration实例

cfg = new Configuration();

//设置FreeMarker的模版文件夹位置

cfg.setDirectoryForTemplateLoading(new File("G:\\testprojects\\freemarkertest\\src"));

}

public void process() throws Exception

{

//构造填充数据的Map

Map map = new HashMap();

map.put("user", "lavasoft");

map.put("url", "http://www.baidu.com/");

map.put("name", "百度");

//创建模版对象

Template t = cfg.getTemplate("test.ftl");

//在模版上执行插值操作，并输出到制定的输出流中

t.process(map, new OutputStreamWriter(System.out));

}

public static void main(String[] args) throws Exception

{

Test hf = new Test();

hf.init();

hf.process();

}

（3）创建模版文件test.ftl

<html>

<head>

<title>Welcome!</title>

</head>

<body>

<h1>Welcome ${user}!</h1>

<p>Our latest product:

<a href="${url}">${name}</a>!

</body>

</html>

**2、和spring集成**

（1）添加freemarker的jar，还需要额外添加spring-content-support的jar包，不然会报错

（2）bean.xml配置

<!-- 配置freeMarker的模板路径 -->

<bean class="org.springframework.web.servlet.view.freemarker.FreeMarkerConfigurer">

<property name="templateLoaderPath" value="WEB-INF/ftl/" />

<property name="defaultEncoding" value="UTF-8" />

</bean>

<!-- freemarker视图解析器 -->

<bean class="org.springframework.web.servlet.view.freemarker.FreeMarkerViewResolver">

<property name="suffix" value=".ftl" />

<property name="contentType" value="text/html;charset=UTF-8" />

<!-- 此变量值为pageContext.request, 页面使用方法：rc.contextPath -->

<property name="requestContextAttribute" value="rc" />

</bean>

（3）pojo类

package com.my.springmvc.bean;

public class User

{

private String username;

private String password;

public String getUsername()

{

return username;

}

public void setUsername(String username)

{

this.username = username;

}

public String getPassword()

{

return password;

}

public void setPassword(String password)

{

this.password = password;

}

}

（4）Controller类

@Controller

@RequestMapping("/home")

public class FreeMarkerController

{

@RequestMapping("/index")

public ModelAndView Add(HttpServletRequest request,HttpServletResponse response)

{

User user = new User();

user.setUsername("sg");

user.setPassword("1234");

List<User> users = new ArrayList<User>();

users.add(user);

ModelAndView mv = new ModelAndView();

mv.setViewName("index");

mv.addObject("users",users);

return mv;

}

}

（5）index.ftl

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>Insert title here</title>

</head>

<body>

<#list users as user>

username : ${user.username}<br/>

password : ${user.password}

</#list>

</body>

</html>

（6）效果

username : zhangsan

password : 1234

**3、模板语言**

（1）4个部分组成

@文本：直接输出的部分

@注释：<#– … –> 格式部分，不会输出

@插值：即 ${…} 或 #{…} 格式的部分，将使用数据模型中的部分替代输出

@指令：FreeMarker 指定，和 HTML 标记类似，名字前加 # 予以区分，不会输出

<html>

<head>

<title>Welcome!</title>

</head>

<body>

<#-- 注释部分 -->

<#-- 下面使用插值 -->

<h1>Welcome ${username} !</h1>

<p>We have these animals:</p>

<u1>

<!-- 使用FTL指令 -->

<#list animals as animal>

<li>${animal.name} for ${animal.price} Euros</li>

</#list>

</u1>

</body>

</html

（2）控制语句

<#if condition>

...

<#elseif condition2>

...

<#elseif condition3>

...

<#else>

<#switch value>

<#case refValue1>

...

<#break>

<#case refValue2>

...

<#break>

<#case refValueN>

...

<#break>

<#default>

...

</#switch>

（3）判断变量是否存在 : <#if readonly??></#if>

（4）防止空指针报错: 变量名后用 ! 加默认值：${foo!”Default”}，如果 foo 为 null 则输出 Default

（5）遍历List集合

<#list ["克里斯埃文斯", "斯嘉丽约翰逊", "小罗伯特唐尼"] as x>

${x}

</#list>

此外,迭代集合对象时,还包含两个特殊的循环变量:

item\_index:当前变量的索引值

item\_has\_next:是否存在下一个对象

也可以使用<#break>指令跳出迭代

（6）运算符

@算术运算符: +, - , \* , / , %

@比较运算符: =或者==,!=,>或者gt,<或者lt,>=或者gte,<=或者lte

@关系运算符：&&，|| ，!三种

（7）综合实例

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>Insert title here</title>

</head>

<body>

描述：${description}

<br/>

集合大小:${nameList?size}

<br/>

迭代list集合：

<br/>

<#list nameList as names>

这是第${names\_index+1}个人，叫做：<label style="color:red">${names}</label>

if判断：

<br/>

<#if (names=="陈靖仇")>

他的武器是: 十五~~

<#elseif (names=="宇文拓")> <#--注意这里没有返回而是在最后面-->

他的武器是: 轩辕剑~·

<#else>

她的绝招是：蛊毒~~

</#if>

<br/>

</#list>

迭代map集合：

<br/>

<#list weaponMap?keys as key>

key--->${key}<br/>

value----->${weaponMap[key]!("null")}

<#--

fremarker 不支持null, 可以用！ 来代替为空的值。

其实也可以给一个默认值

value-----${weaponMap[key]?default("null")}

还可以 在输出前判断是否为null

<#if weaponMap[key]??></#if>都可以

-->

<br/>

</#list>

include导入文件：

<br/>

<#include "include.html"/>

</body>

</html>

**4 使用场景**

1） 电商商品展示页面

2） 电子po

3） 静态化页面

**5 和其他对比。在java领域，表现层技术主要有三种：jsp、freemarker、velocity。**

（一） jsp优点

1）功能强大，可以写java代码

2）支持jsp标签（jsp tag）

3）支持表达式语言（el）

4）官方标准，用户群广，丰富的第三方jsp标签库

5）性能良好。jsp编译成class文件执行，有很好的性能表现

（二） jsp缺点

1）如使用不当容易破坏mvc结构。

2）第一次执行的时候需要转换成Servlet类，开发阶段进行功能调适时，需要频繁的修改JSP，每次修改都要编译和转换

3）没有分离表现层和业务逻辑，表现层和逻辑层混在一起。

（三）velocity优点

1）不能编写java代码，可以实现严格的mvc分离

2）性能良好，比jsp性能还要好些

3）使用表达式语言，据说jsp的表达式语言就是学velocity的

（四）velocity缺点

1）不是官方标准

2）用户群体和第三方标签库没有jsp多。

3）对jsp标签支持不够好

（五）FreeMarker优点

1）分离表现层和业务逻辑

使用JSP开发过程中在页面中大量的存在业务逻辑的代码，使得页面内容凌乱，在后期大量的修改维护过程中就变得非常困难。FreeMarker根本不支持Java脚本代码，

而是使用el表达式来输出展示数据。FreeMarker的设计初衷就是：模板+数据模型=输出，模板只负责数据在页面中的表现，不涉及任何的逻辑代码，而所有的逻辑都

是由数据模型来处理的。用户最终看到的输出是模板和数据模型合并后创建的。

2）提高开发效率

在我们以往的开发中，使用的都是JSP页面来展示数据的，即所谓的表现层。我们都知道，JSP在第一次执行的时候需要转换成Servlet类，开发阶段进行功能调适时，

需要频繁的修改JSP，每次修改都要编译和转换，那么试想一天中我们浪费在程序编译的时间有多少。相对于JSP来说，FreeMarker模板技术不存在编译和转换的问题

，所以就不会存在上述问题。而且开发过程中，我们在不必在等待界面设计开发人员完成页面原形后，我们再来开发程序。

3）分工明确

以往用JSP展现数据时，程序员并不熟悉界面设计技术，反之界面开发人员，也并不熟悉程序语言。协调工作很困难，使用FreeMarker后，作为界面开发人员，只专心

创建HTML文件、图像以及Web页面的其他可视化方面，不用理会数据；而程序开发人员则专注于系统实现，负责为页面准备要显示的数据。

（六）FreeMarker缺点

1）应用FreeMarker模板技术，在修改模板后，可能会看到已经过期的数据。如：生成静态的HTML页面后，如果一旦模板改变，而没有及时更新模板生成的HTML页面的话，

用户看到的就是过期的数据。

2）FreeMarker模板技术在应用过程中，FreeMarker中的变量必须要赋值，如果不赋值，那么就会抛出异常。想避免错误就要应用if/elseif/else 指令进行判段，如果对

每一个变量都判断的话，那么则反而增加了编程的麻烦。

3）FreeMarker的map限定key必须是string，其他数据类型无法操作

4）FreeMarker不支持集群应用。为了编成的方便性，把序列化的东西都放到了Session中，如Session，request等，在开发的过程中确实方便，但如果将应用放到集群中

，就会出现错误。

## 十、photoShop

Adobe Photoshop，简称“PS”，是由Adobe Systems开发和发行的图像处理软件。Photoshop主要处理以像素所构成的数字图像。使用其众多的编修与绘图工具，可以有效地进行图片编辑工作。ps有很多功能，在图像、图形、文字、视频、出版等各方面都有涉及。2003年，Adobe Photoshop 8被更名为Adobe Photoshop CS。2013年7月，Adobe公司推出了新版本的Photoshop CC，自此，Photoshop CS6作为Adobe CS系列的最后一个版本被新的CC系列取代。截止2016年12月Adobe PhotoshopCC2017为市场最新版本。Adobe支持Windows操作系统 、安卓系统与Mac OS， 但Linux操作系统用户可以通过使用Wine来运行Photoshop。