1. JSP 概述

与Servlet的关系

- JSP 是服务端脚本语言
- JSP 被翻译成 `Servlet` 后才编译运行的，能实现 `Servlet` 实现的所有功能
- JSP 源码是纯文本

JSP 页面的处理过程

- 用户第一次访问 JSP 页面时
    - 客户端 $\Longrightarrow$ JSP 文件 $\Longrightarrow$ 翻译JSP $\Longrightarrow$ `.java` 文件
    - $\Longrightarrow$ `.class` 文件 $\Longrightarrow$ HTML 响应 $\Longrightarrow$ 响应回客户端
- JSP 页面被再次请求时
    - 客户端请求 $\Longrightarrow$ JSP 文件 $\Longrightarrow$ 查找 `.class` 文件 $\Longrightarrow$ 执行 `HTML` 响应 $\Longrightarrow$ 响应客户端

JSP 基本结构

- 是带有 JSP 元素的常规 `Web` 页面，由模版文本和 `JSP` 元素组成
- 在一个 JSP 页面中，所有非 JSP 元素内容称为模版文本
- JSP 并不依赖 `HTML`，在处理一个 JSP 页面请求时，模版文本和 JSP 元素生成的内容会合并，合并后的结果作为响应内容发送给浏览器
- JSP 元素
    - 脚本元素
    - 指令元素
    - 动作元素

2. 脚本元素

- 允许用户将小段代码加入到 `JSP` 中，例如加入 `if` 根据具体情况产生不同 `HTML` 代码
- 在页面被请求时执行
- 脚本 `<% 脚本代码 %>`
    - HTML 中插入 java
    - 根据逻辑产生不同内容

```
<%
  if(date == x) {
%>
上午好!
<%
  } else {
%>
下午好!
<%
  }
%>
```

- 表达式：`<%= 变量 %>`
  - 一种简单的输出形式，一定要有一个可以输出的值
  - 直接输出给定的值或变量的值
- 声明：`<%! 变量 %>`
  - 类似类的属性，全局变量，对所有 JSP 请求共享
  - 不输出任何文本到输出流，在声明元素中声明的变量和方法将**在 JSP 页面初始化时进行初始化**
- 注释：`<%-- content --%>`

3. 指令元素

- 用来向 JSP 容器提供编译信息
- 指令不向客户端产生任何输出，所有的指令都只在当前页面有效

- `<%@page 属性列表 %>`
  - 描述和页面相关的信息，如：导入所需类包、指明输出内容类型、控制 Session 等。
  - 可以出现多次，但在每个 page 指令中，每一种属性只能出现一次，重复的会覆盖之前的
  - `import` 导入包和类
  - `contentType` 指定输出内容的 MIME 类型 和字符编码方式，用于将网站的表格等内容获取并下载成指定格式
  - 控制 `Session` ：指明是否使用 `Session` 对象
- `<%@include file = "文件" %>`
- 在页面翻译期间引入另一个页面，可以是 `JSP、HTML`
  - 规则是将两个页面变成一个再翻译
  - 两个页面**不能还有同名的变量等规则**
- `<%@taglib uri = "标签库URI" prefix = "标签前缀" %>`
  - 用于指定 JSP 页面使用的标签库，通过该指令可以在 JSP 页面中使用标签库中的标签

4. 动作元素

- `<jsp : include>`
  - 不会把动作指令 `include` 指定的文件与原 JSP 页面合并，而是告诉 Java 解释器，这个文件 JSP 运行时才处理
  - 包含的是普通文本，将其发送到客户端，由客户端负责显示
  - JSP 文件，JSP 容器就执行这个文件，然后将结果发送到客户端，客户端负责显示这些结果
- `<jsp : forward>`：把请求转发到另一个页面

```
<jsp:forward page = "target.jsp">
  <jsp:param name = "" value = ""/>
</jsp:forward>
```

  - 只输出转发的结果页面
  - 调用者和被调用者共享一个 `request` 对象

- `<jsp : useBean>`：用于查找或实例化一个 `JavaBean`
- `<jsp : setProperty>`：用于设置 `JavaBean` 的属性
- `<jsp : getProperty>`：输出 `JavaBean` 的属性