<%@ page language="java" contentType="text/html; charset=utf-8" pageEncoding="utf-8"%>

jsp 的 page 指令可以修改 jsp 页面中一些重要的属性，或者行为。

1. language属性 表示 jsp 翻译后是什么语言文件。暂时只支持 java。
2. contentType属性 表示 jsp 返回的数据类型是什么。也是源码中 response.setContentType()参数值
3. pageEncoding属性 表示当前 jsp 页面文件本身的字符集。 iv.
4. import属性 跟 java 源代码中一样。用于导包，导类。

==========两个属性是给 out 输出流使用=========

1. autoFlush属性 设置当 out 输出流缓冲区满了之后，是否自动刷新冲级区。默认值是 true。 vi.
2. buffer属性 设置 out 缓冲区的大小。默认是 8kb

==========两个属性是给 out 输出流使用=========

1. errorPage属性 设置当 jsp 页面运行时出错，自动跳转去的错误页面路径。
2. isErrorPage属性 设置当前 jsp 页面是否是错误信息页面。默认是 false。如果是 true 可以 获取异常信息。
3. session属性 设置访问当前 jsp 页面，是否会创建 HttpSession 对象。默认是 true。
4. extends属性 设置

contentType=“text/html”：网页类型html  
charset=”utf-8"网页编码类型  
language="java"网页编程语言  
<% @ page%>编程语法

第一阶段：**将jsp编译成Servlet（.java）文件。用到的指令是pageEncoding**，根据pageEncoding＝“XXX”的指示，找到编码的规则为“XXX”，服务器在将JSP文件编译成.java文件时会根据pageEncoding的设定读取jsp，结果是由指定的编码方案翻译成统一的UTF-8编码的JAVA源码（即.java）。

第二阶段：**从Servlet文件（.java）到Java字节码文件（.class），从UTF-8到UTF-8**。在这一阶段中，不论JSP编写时候用的是什么编码方案，经过这个阶段的结果全部是UTF-8的encoding的java源码。JAVAC用UTF-8的encoding读取java源码，编译成UTF-8编码的二进制码（即.class），这是JVM对常数字串在二进制码（java encoding）内表达的规范。这一过程是由JVM的内在规范决定的，不受外界控制。

第三阶段：**从服务器到浏览器，这在一过程中用到的指令是contentType**。服务器载入和执行由第二阶段生成出来JAVA二进制码，输出的结果，也就是在客户端可见到的结果，在这次输出过程中，由contentType属性中的charset来指定，将UTF8形式的二进制码以charset的编码形式来输出。如果没有人为设定，则默认的是ISO-8859-1的形式。

pageEncoding **：说明jsp内容的编码。**

contentType：设置JSP源文件和响应正文的字符集编码及MIME类型，  contentType的**charset是指服务器发送给客户端时的内容编码**  
contentType 内容类型  
text/html 说明是网页 其实还有text/css 那就是样式  
charset 编码方式

若未添加，网页可能乱码