

# Programmazione per il Web

## Note per la lezione del 30/03/2016

Igor Melatti

### Esempi di Pagine JSP

- Slide 218 – 286
- Slide 251 – 271 e 276 – 283: saltare, sono una copia
- `syntax.html`: i tag delle scriptlet vengono ignorati dall'HTML
- `syntax*.jsp`: vari aspetti della sintassi, soprattutto rispetto all'inaccurata slide 245
  - `syntax.0.jsp`: errore di *compilazione* (nessun problema per la traduzione). L'errore è abbastanza ovvio: `<br/>` è all'interno di una servlet, ma non è uno statement Java valido
  - `syntax.1.jsp`: errore di *traduzione* (quindi, alla compilazione non ci arriva proprio). L'errore è abbastanza ovvio: un commento `<%--` iniziato e mai chiuso. Fosse stato un commento HTML, almeno la risposta sarebbe stata inviata, ma con i commenti delle scriptlet gli errori sono fatali
  - `syntax.2.jsp`: corretto l'errore precedente, non ci sono più problemi. Da notare che l'espressione scriptlet `<%= "ci sono" %>` è effettivamente presente nella pagina HTML inviata in risposta (non è visualizzata dal browser, ma compare se si visualizza il sorgente della pagina); quindi, i commenti HTML non “annullano” la traduzione da JSP a servlet
  - `syntax.2.bis.jsp`: errore di traduzione, `\<%` non va bene
  - `syntax.2.ter.jsp`: a conferma che i commenti HTML non “annullano” la traduzione da JSP a servlet, ecco cosa succede se si introduce un errore all'interno di un commento HTML: errore di compilazione
  - in questi primi esempi, si è visto cosa si può mettere nel template HTML di una pagina JSP. In sostanza: rispetto a slide 245, l'unico problema è il primo “letterale”, `<%`, che va effettivamente sostituito con `<\%` se non si vuole incorrere in un errore di traduzione. Tutti gli altri letterali non danno problemi nel template text. Da qui

Table 1: Correzione di Slide 245. Nota: un caso a parte sono gli attributi di tag XML come nelle azioni standard e nei tag personalizzati: ci ritorneremo

Simboli	Permesso in template text	Permesso in costante stringa
<%	Solo se chiuso da un corrispondente %>; altrimenti, va corretto in <%	Sempre
%>	Sempre	Mai, va corretto in <%
'	Sempre	Sempre, ma può essere corretto in \'
"	Sempre	Mai, va corretto in \"
\	Sempre	Mai, va corretto in \\

in poi, si vede se danno problemi negli scripting elements JSP (per semplicità, in espressioni scriptlet). La situazione è l'opposta: adesso viene fuori un errore di *compilazione* per tutti i casi, tranne che per <% e '. Per gli ultimi 2 casi di slide 245 l'errore è ovviamente di Java, nel secondo (`syntax.4.jsp`) viene in realtà dalla traduzione da JSP, che confonde il primo %> che trova (anche se dentro una stringa costante) come una chiusura del <% (vedere la servlet generata). `syntax.8.jsp`: correggere <% (che non è sbagliato) in <% (come sarebbe necessario fare per il template text) dà errore di compilazione, perché in effetti Java non sa interpretare \%... vedere Tabella 1

- `direttive.jsp`: le altre direttive utili trattate finora; invoca `error.jsp`
- `error.jsp`: cattura l'eccezione nella pagina; notare che traduzione viene definito l'oggetto `exception`, altrimenti assente nelle altre traduzioni: è solo per quando c'è `<%@ page isErrorPage="true"%>`. Notare anche che `exception.printStackTrace` non scrive niente, perché di default va su standard error. Dato che non è possibile passargli direttamente out (che è di tipo `JspWriter`), occorre trasformarlo prima in stringa e poi scriverlo su out.
- `application.jsp`: esempio che usa l'oggetto implicito `application`
- `config.jsp`: esempio che usa l'oggetto implicito `config`; raramente importante
- `page.jsp`: in realtà (vedere il libro a pagina 245) è inutile; l'esempio di slide 272 è sbagliato, dato che `page` è di tipo `Object`; occorre fare quindi il cast; tanto vale usare `this`!