

Java WEB

Java Enterprise Edition



Tutorial 02

JSTL em 21 mins

A. Como disponibilizar?

Importar para o projeto as bibliotecas `jstl.jar` e `standard.jar`.

B. Como interagir com o Banco de Dados?

Basicamente a página só precisará do acesso a Core e a SQL, colocado desta maneira

```
<%@ taglib uri='dbsql' prefix='sql' %>
<%@ taglib uri='core' prefix='c' %>
```

Para realizar a conexão:

```
<sql:setDataSource var="conn"
  driver="org.gjt.mm.mysql.Driver"
  url="jdbc:mysql://localhost:3306/dfjug?user=root&useUnicode=true"
/>
```

Para executar uma query:

```
<sql:query var="rSet" dataSource="${conn}">
  SELECT * FROM tabela
</sql:query>
```

Para obter os dados:

1º Modo: arma-se um comando for, do seguinte modo:

```
<c:forEach var='row' items='${rSet.rowsByIndex}'>
  <c:forEach var='rowData' items='${row}'>
    <c:out value='${rowData}'/>
  </c:forEach>
</c:forEach>
```

O primeiro for conta as linhas, enquanto que o segundo verifica as colunas.

2º Modo: O mesmo laço pode obter pelos nomes da coluna:

```
<c:forEach var='row' items='${rSet.rows}'>
  <c:out value='${row.NomeColuna}'/>
</c:forEach>
```

3º Modo: Também é possível pegar pelo número do campo:

```
<c:forEach var='row' items='${rSet.rowsByIndex}'>
  <c:out value='${row[numColuna]}'/>
</c:forEach>
```

4º Modo: Pegando apenas uma única linha:

```
<c:out value='${rSet.rowsByIndex[numLinha][numColuna]}'/>
```

C. Como são os comandos básicos?

C1. Criando uma variável

```
<c:set var="curso" value="JSP"/>
```

C2. Mostrando a variável na página

```
<c:out value="${curso}"/>
```

C3. Removendo variáveis

```
<c:remove var="curso"/>
```

C4. Trilhar erros de variáveis que não existem

```
<c:out var="{curso}" escapeXml="false">
  Curso não existe
</c:out>
```

C5. Executando um if além de guardar seu resultado em uma variável

```
<c:if test="{curso == 'JSP'}" var="teste">
  São iguais
</c:if>
Resultado do IF: <c:out value="{teste}"/>
```

C6. Usando um if mais complexo, é possível inclusive encadear

```
<c:choose>
  <c:when test="{curso == 'JSP'}">
    São iguais
  </c:when>
  <c:otherwise>
    São diferentes
  </c:otherwise>
</c:choose>
```

C7. Fazendo importações de JSPF com teste para verificar possíveis erros:

```
<c:catch var="erro">
  <c:import url="pagina.jspf"/>
</c:catch>
<c:if test="{not empty erro}">
  Não consegui encontrar a página especificada
</c:if>
```

C8. Fazendo redireções

```
<c:redirect url="pagina.jsp"/>
```

C9. Obtendo parametros

```
<c:out value='{param.nome}'/>
```

D. Usar Variáveis definidas no scriptlet?

D1. De Scriptlet para TagsStandards

```
<%
String s = "teste";
pageContext.setAttribute("nome", s);
%>
<c:out var="{nome}"/>
```

D2. De TagsStandards para Scriptlet

```
<c:set var="curso" value="JSP"/>
<%=pageContext.getAttribute("curso")%>
```

E. Usar variáveis do tipo Inteiro?

```
<% pageContext.setAttribute("idt", new Integer(2)); %>
<c:if test="{idt > 1}">
  É maior que 1
```

```
</c:if>
```

Pode-se usar os seguintes comparadores:

```
! not empty < lt > gt <= le >= ge == eq != ne
```

F. Usar variáveis do tipo Lógico?

```
<c:set var='liga' value='false'/>
<c:set var='liga' value='${!liga}'/>
<c:if test="${liga}">
  Tá ligada
</c:if>
```

G. Como usar a formatação?

G1. Usando o formato Brasileiro

```
<%@ taglib uri='format' prefix='fmt' %>
<fmt:setLocale value='pt_BR'/>
```

G2. De campos numéricos

```
<fmt:formatNumber value='121212.122' pattern='0.000' maxFractionDigits='3'/>
<fmt:formatNumber value='121212.122' pattern='0.00' maxFractionDigits='2'/>
```

G3. De campos data

```
<jsp:useBean id="now" class="java.util.Date"/>
<fmt:formatDate value='${now}'/>
<fmt:formatDate value='${now}' type='both' dateStyle='full' timeStyle='full'/>
```

Valores possíveis são: default, short, medium, long e full.

H. Como usar em Beans ?

H1. Construindo o Bean

Os beans deverão ter métodos padrões definidos com **set** (entrada) e **get** (saída) por exemplo:

```
package beans;
public class MeuBean {
  private String nome;
  public void setNome(String newValue) { nome = newValue; }
  public String getMostra() { return nome; }
}
```

H2. Acessando este bean pela aplicação

```
<jsp:useBean id='bnProjeto' class='beans.MeuBean'>
  <c:set target='${bnProjeto}' property='nome' value='Fernando'/>
</jsp:useBean>
<c:out value='${bnProjeto.mostra}'/>
```