

# JAVA/JEE

Les JSP + EL + JSTL

# JSP

---

- **Java Server Page**
- Page HTML dans lesquels du code JAVA peut être exécuté

- L'objet request est accessible

- Scriptlet: code JAVA exécuté dans la JSP

Syntaxe: `<% code JAVA %>`

- Expression: une seule instruction dont le résultat est converti en string

Syntaxe: `<%= instruction %>`

- Directive: instructions qui s'applique à toute la JSP (ex. import)

Syntaxe: `<%@ instruction(s) %>`

- Commentaire: commentaires non renvoyés côté client

Syntaxe: `<%-- commentaire --%>`

# Comment fonctionnent les JSP

---

- Les JSP sont compilés par Tomcat en Servlet
- Un cache des JSP compilés est conservé afin de servir rapidement les clients
- On peut lire le code transformé des JSP dans le répertoire work de Catalina  
`\work\Catalina\localhost\epsi-tp1\org\apache\jsp` de Tomcat

# Examples

```
<%@ page import="business.*, data.*, java.util.Date" %>
<%
    String firstName    = request.getParameter("firstName");
    String lastName     = request.getParameter("lastName");
    String emailAddress = request.getParameter("emailAddress");
%>

<p>This email address was added to our list on
    <%= new Date() %></p>
```

```
<%
    int numOfTimes = 1;
    while (numOfTimes <= 5){
%>
    <h1>This line is shown <%= numOfTimes %>
        of 5 times in a JSP.</h1>
<%
    numOfTimes++; }
%>
```

# Inclusion de JSP

```
<%@ include file="/includes/header.html" %>

<h1>Thanks for joining our email list</h1>

<p>Here is the information that you entered:</p>

<%@ page import="business.User" %>
<% User user = (User) request.getAttribute("user"); %>
<table cellspacing="5" cellpadding="5" border="1">
  <tr><td align="right">First name:</td>
    <td><%= user.getFirstName() %></td>
  </tr>
  <tr><td align="right">Last name:</td>
    <td><%= user.getLastName() %></td>
  </tr>
  <tr><td align="right">Email address:</td>
    <td><%= user.getEmailAddress() %></td>
  </tr>
</table>
<%@ include file="/includes/footer.jsp" %>
```

# Variables d'instance

---

- Il est possible de déclarer des variable d'instance avec la syntaxe suivante

Syntaxe: `<%! int maVaribale = 0; %>`

- A utiliser avec parcimonie car les JSP ne sont pas thread safe

# Expression language (EL)

- EL est une syntaxe particulière qui permet d'accéder plus facilement aux données depuis une JSP

```
<%@ page import="java.io.*,java.util.*" %>
<% String title = "Accessing Request Param"; %>
<html>
  <head>
    <title><% out.print(title); %></title>
  </head>
  <body>
    <center> <h1><% out.print(title); %></h1> </center>
    <div align="center"> <p>${param["username"]}</p> </div>
  </body>
</html>
```

# Expression language (EL)

- Plusieurs objets sont accessibles directement via EL:

Implicit object	Description
pageScope	Scoped variables from page scope
requestScope	Scoped variables from request scope
sessionScope	Scoped variables from session scope
applicationScope	Scoped variables from application scope
param	Request parameters as strings
paramValues	Request parameters as collections of strings
header	HTTP request headers as strings
headerValues	HTTP request headers as collections of strings
initParam	Context-initialization parameters
cookie	Cookie values
pageContext	The JSP PageContext object for the current page



# JSTL

---

- JSTL: **J**ava server page **S**tandard **T**ag **L**ibrary

Tags additionnels pour simplifier l'écriture d'une JSP:

- Tag de structure (itération, conditionnement ...)
- Internationalisation (i18n)
- Exécution de requêtes SQL
- Utilisation de documents XML

- S'appuie sur EL

# Exemples

```
<c:forEach begin="1" end="12" var="i" step="3" varStatus="vs">
  index = <c:out value="\${vs.index}"/> :
  count = <c:out value="\${vs.count}"/> :
  value = <c:out value="\${i}"/>
  <c:if test="\${vs.first}">
    : Premier element
  </c:if>
  <c:if test="\${vs.last}">
    : Dernier element
  </c:if>
  <br>
</c:forEach>
```

Nous sommes le

```
<fmt:formatDate value="\${now}" type="date" dateStyle="full"/>
```