



Cours 02:

Java Front Office

Hayri ACAR
hayri.acar@reseau-cd.net

■ JSTL

- JSTL : bibliothèque, éviter l'utilisation de code Java dans les pages JSP.
- Lisibilité, maintenabilité, réduction de la taille de code.

```
<!-- boucle en Java -->
<%@ page import="java.util.List, java.util.ArrayList" %>
<%
    List<Integer> list = (ArrayList<Integer>)request.getAttribute("tirage");
    for(int i = 0; i < list.size();i++){
        out.println(list.get(i));
    }
%>
<!-- boucle en JSTL -->
<c:forEach var="item" items="${tirage}" >
    <c:out value="${item}" />
</c:forEach>
```

- Différentes versions : JSTL 1.0, JSTL 1.1 (intègre EL), JSTL 1.2

Configuration JSTL

- Inclure la bibliothèque core dans votre fichier jsp, directive taglib :
- `<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>`

uri : lien vers la définition de la bibliothèque.

prefix : alias pour faire appel aux balises de la bibliothèque. tagif s'écrit : `<c:if>`.

- Ou dans web.xml :

```
<jsp-config>
```

```
    <jsp-property-group>
```

```
        <url-pattern>*.jsp</url-pattern>
```

```
        <include-prelude>/WEB-INF/taglibs.jsp</include-prelude>
```

```
    </jsp-property-group>
```

```
</jsp-config>
```

/WEB-INF/ taglibs.jsp : `<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>`

- Ajouter dans le répertoire lib : <https://mvnrepository.com/artifact/javax.servlet/jstl/1.2>

La bibliothèque Core

- **Affichage d'une expression :**

```
<c:out value="test" /> <%-- Affiche test --%>
```

```
<c:out value="${ 'a' < 'b' }" /> <%-- Affiche true --%>
```

```
<%-- Cette balise affichera le mot 'test' si le bean n'existe pas : --%>
```

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

```
test
```

```
</c:out>
```

```
<%-- Elle peut également s'écrire sous cette forme : --%>
```

```
<c:out value="${bean}" default="test" />
```

Création ou modification d'une variable : <c:set>

<%-- Cette balise met l'expression "Bonjour" dans l'attribut "message" de la requête : --%>

```
<c:set var="message" value="« Bonjour" scope="request" />
```

<%-- Equivalent en scriptlet Java : --%>

```
<% request.setAttribute( "message", « Bonjour" ); %>
```

- var** : nom de la variable, **value** : valeur, **scope** : portée.

<%-- Affiche l'expression contenue dans la variable "message" de la requête --%>

```
<c:out value="${requestScope.message}" />
```

- requestScope** : permet de rechercher un objet dans la portée requête.
- Autre écriture pour initialiser une variable : `<c:set var="maVariable"> 12 </c:set>`
- Autre : `<c:set var="locale" scope="session"><c:out value="${param.lang}" default="FR"/></c:set>`

param : objet pour récupérer la valeur du paramètre de la requête nommé lang.

Modification des propriétés d'un objet

<%-- Crée un objet de type String --%>

<c:set scope="session" var="description" value= "Bonjour" />

<%-- Crée un objet du type du bean ici spécifié dans l'attribut 'value'--%>

<c:set scope="session" var= "bean" value="\${Bean}" />

target : contient le nom de l'objet dont la propriété sera modifiée ;

property : contient le nom de la propriété qui sera modifiée.

<!-- Définir ou modifier la propriété 'prenom' du bean 'auteur' -->

<c:set target="\${auteur}" property="prenom" value=« Patrice"/>

<!-- Définir ou modifier la propriété 'prenom' du bean 'auteur' via le corps de la balise -->

<c:set target="\${auteur}" property="prenom">

Wile E.

</c:set>

Supprimer une variable

`<%-- Supprime la variable "maVariable" de la session --%>`

`<c:remove var="maVariable" scope="session" />`

Les conditions

`<c:if test="{ 12 > 7 }" var="maVariable" scope="session">`

Ce test est vrai.

`</c:if>`

`<c:choose>`

`<c:when test="{expression}">Action ou texte.</c:when>`

...

`<c:otherwise>Autre action ou texte.</c:otherwise>`

`</c:choose>`

• Les boucles

<c:forEach> : parcourir une collection, <c:forTokens> : parcourir une chaîne de caractères.

En Java / en JSTL :

```
<%  
int[] cube= new int[8];  
/* Boucle calculant et affichant le cube des entiers de 0 à 7 */  
for(int i = 0 ; i < 8 ; i++)  
{  
    cube[i] = i * i * i;  
    out.println("<tr><td>" + i + "</td> <td>" + cube[i] + "</td></tr>");  
}  
%>
```

```
<c:forEach var="i" begin="0" end="7" step="1">  
    <tr>  
        <td><c:out value="${i}"/></td>  
        <td><c:out value="${i * i * i}"/></td>  
    </tr>  
</c:forEach>
```

begin : valeur de début du compteur

end : valeur de fin du compteur

step : pas d'incrémentation de la boucle

Itération

items : attribut qui indique la collection à parcourir, ici la liste maListe.

var : le nom de la variable qui sera liée à l'élément courant de la collection parcourue par la boucle, ici news.

Autre écriture :

`${news.titre}`

`${news.contenu}`

```
<%@ page import="java.util.*" %>
<%
    /* Création de la liste et des données */
    List<Map<String, String>> maListe = new ArrayList<Map<String, String>>();
    Map<String, String> news = new HashMap<String, String>();
    news.put("titre", "Titre de ma première news");
    news.put("contenu", "corps de ma première news");
    maListe.add(news);
    news = new HashMap<String, String>();
    news.put("titre", "Titre de ma seconde news");
    news.put("contenu", "corps de ma seconde news");
    maListe.add(news);
    pageContext.setAttribute("maListe", maListe);
%>

<c:forEach items="${maListe}" var="news">
<div class="news">
    <div class="titreNews">
        <c:out value="${news['titre']}" />
    </div>
    <div class="corpsNews">
        <c:out value="${news['contenu']}" />
    </div>
</div>
</c:forEach>
```

• Itération sur une chaîne de caractères

```
<%-- Affiche les différentes sous-chaînes séparées par une virgule ou un point-virgule --%>  
<c:forTokens var="sousChaine" items="salut; je m'appelle,Jean;Patrice+!" delims=";,+">  
  ${sousChaine}<br/>  
</c:forTokens>
```

• Les liens

```
<c:url>  
  
<%-- Génère une url simple, positionnée dans un lien HTML --%>  
<a href="<c:url value="test.jsp" />">lien</a>  
  
<%-- Génère une url et la stocke dans la variable lien --%>  
<c:url value="test.jsp" var="lien" />
```

• Imports

```
<%-- Importer une page distante dans une variable. --%>  
<c:import url="http://www.siteduzero.com/zozor/biographie.html" var="bio" scope="page"/>
```

Formulaires

```
<%@ page pageEncoding="UTF-8" %>
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Inscription</title>
    <link type="text/css" rel="stylesheet" href="form.css" />
  </head>
  <body>
    <form method="post" action="inscription">
      <p>Login</p>

      <label for="id">Nom : </label>
      <input type="text" id="nom" name="nom" value="" size="20" maxlength="60" />
      <br />
      <label for="mdp">Prenom : </label>
      <input type="text" id="mdp" name="mdp" value="" size="20" maxlength="60" />
      <br />
      <input type="submit" value="Inscription" class="sansLabel" />
      <br />
    </form>
  </body>
</html>
```

- La servlet :

```
public class Inscription extends HttpServlet {  
  
    public void doGet( HttpServletRequest request, HttpServletResponse response ) throws ServletException, IOException{  
  
        /* Affichage de la page d'inscription */  
  
        this.getServletContext().getRequestDispatcher("/WEB-INF/inscription.jsp" ).forward( request, response );  
  
    }  
}
```

- web.xml :

```
<servlet>  
  
    <servlet-name>Inscription</servlet-name>  
  
    <servlet-class>com.servlets.Inscription</servlet-class>  
  
</servlet>  
  
<servlet-mapping>  
  
    <servlet-name>Inscription</servlet-name>  
  
    <url-pattern>/inscription</url-pattern>  
  
</servlet-mapping>
```

■ Servlet.

```
public void doPost( HttpServletRequest request, HttpServletResponse response ) throws ServletException, IOException{  
    /* Récupération des champs du formulaire. */  
  
    String nom = request.getParameter("nom");  
  
    try {Utilisateur utilisateur = new Utilisateur();  
        utilisateur.validationNom( nom );  
    } catch (Exception e) { }  
  
    this.getServletContext().getRequestDispatcher("/WEB-INF/affichage.jsp").forward(request, response);  
}
```

■ Bean :

//Valide le nom d'utilisateur saisi.

```
private void validationNom( String nom ) throws Exception {  
    if ( nom != null && nom.trim().length() < 5 ) {  
        throw new Exception( "Le nom d'utilisateur doit contenir au moins 5 caractères." );  
    }  
}
```

Session

```
public void doPost( HttpServletRequest request, HttpServletResponse response ) throws ServletException, IOException {  
    String nom = request.getParameter("nom");  
  
    String prenom = request.getParameter("prenom");  
  
    HttpSession session = request.getSession();  
  
    session.setAttribute("nom", nom);  
    session.setAttribute("prenom", prenom);  
  
    this.getServletContext().getRequestDispatcher("/WEB-INF/test.jsp").forward(request, response);  
}
```

JSP :

```
<c:if test="${ !empty sessionScope.prenom && !empty sessionScope.nom }">  
    <p>Vous êtes ${ sessionScope.prenom } ${ sessionScope.nom } !</p>  
</c:if>
```

Cookie

```
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    Cookie[] cookies = request.getCookies();
    if (cookies != null) {
        for (Cookie cookie : cookies) {
            if (cookie.getName().equals("prenom")) {
                request.setAttribute("prenom", cookie.getValue());
            }
        }
    }
    this.getServletContext().getRequestDispatcher("/WEB-INF/test.jsp").forward(request, response);
}

public void doPost( HttpServletRequest request, HttpServletResponse response ) throws ServletException, IOException {
    String nom = request.getParameter("nom");
    String prenom = request.getParameter("prenom");

    Cookie cookie = new Cookie("prenom", prenom);
    cookie.setMaxAge(60 * 60 * 24 * 30);
    response.addCookie(cookie);

    this.getServletContext().getRequestDispatcher("/WEB-INF/test.jsp").forward(request, response);
}
```

Envoi de fichiers

JSP : <form method="post" action="bonjour" enctype="multipart/form-data">

<label for="fichier">Emplacement du fichier *</label>

<input type="file" id="fichier" name="fichier" />

web.xml : <servlet> <servlet-name>Test</servlet-name>

 <servlet-class>com.octest.servlets.Test</servlet-class>

 <multipart-config>

 <location>.....</location>

 <max-file-size>10485760</max-file-size> <!-- 10 Mo -->

 <max-request-size>52428800</max-request-size> <!-- 5 x 10 Mo -->

 <file-size-threshold>1048576</file-size-threshold> <!-- 1 Mo -->

 </multipart-config>

</servlet>

<servlet-mapping>

 <servlet-name>Test</servlet-name>

 <url-pattern>/bonjour</url-pattern>

</servlet-mapping>


```

public void doPost( HttpServletRequest request, HttpServletResponse response ) throws ServletException, IOException {
    // On récupère le champ du fichier
    Part part = request.getPart("fichier");
    // On vérifie qu'on a bien reçu un fichier
    String nomFichier = getNomFichier(part);
    // Si on a bien un fichier
    if (nomFichier != null && !nomFichier.isEmpty()) {
        String nomChamp = part.getName();
        // On écrit définitivement le fichier sur le disque
        ecrireFichier(part, nomFichier, CHEMIN_FICHIERS);
        request.setAttribute(nomChamp, nomFichier);
    }
    this.getServletContext().getRequestDispatcher("/WEB-INF/test.jsp").forward(request, response);
}

private void ecrireFichier( Part part, String nomFichier, String chemin ) throws IOException {
    BufferedInputStream entree = null;
    BufferedOutputStream sortie = null;
    try {
        entree = new BufferedInputStream(part.getInputStream(), 10240);
        sortie = new BufferedOutputStream(new FileOutputStream(new File(chemin + nomFichier)), 10240);

        byte[] tampon = new byte[10240];
        int longueur;
        while ((longueur = entree.read(tampon)) > 0) {
            sortie.write(tampon, 0, longueur);
        }
        sortie.close();
        entree.close();
    }

}

private static String getNomFichier( Part part ) {
    for ( String contentDisposition : part.getHeader( "content-disposition" ).split( ";" ) ) {
        if ( contentDisposition.trim().startsWith( "filename" ) ) {
            return contentDisposition.substring( contentDisposition.indexOf( '=' ) + 1 ).trim().replace( "\\\"", "" );
        }
    }
    return null;
}
}

```

Connexion BD

```
public class Connexion {  
    public void getConnection() {  
        Connection connexion = null;  
        Statement statement = null;  
        ResultSet resultat = null;  
        try {  
            // Chargement du driver  
            Class.forName("com.mysql.cj.jdbc.Driver");  
            connexion = DriverManager.getConnection("jdbc:mysql://localhost:3306/javaee", "root", "");  
            statement = connexion.createStatement();  
            // Exécution de la requête  
            resultat = statement.executeQuery("SELECT nom, prenom FROM noms;");  
            // Récupération des données  
            while (resultat.next()) {  
                String nom = resultat.getString("nom");  
                String prenom = resultat.getString("prenom");  
  
                Utilisateur utilisateur = new Utilisateur();  
                utilisateur.setNom(nom);  
                utilisateur.setPrenom(prenom);  
  
                utilisateurs.add(utilisateur);  
            }  
            resultat.close();  
            statement.close();  
            connexion.close();  
        }  
  
        public void ajouterUtilisateur(Utilisateur utilisateur) {  
            //faire la connexion  
  
            try {  
                PreparedStatement preparedStatement = connexion.prepareStatement("INSERT INTO noms(nom, prenom) VALUES(?, ?);");  
                preparedStatement.setString(1, utilisateur.getNom());  
                preparedStatement.setString(2, utilisateur.getPrenom());  
  
                preparedStatement.executeUpdate();  
            } catch (SQLException e) {
```

```

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    Noms tableNoms = new Noms();
    request.setAttribute("utilisateurs", tableNoms.recupererUtilisateurs());
    this.getRequestDispatcher("/WEB-INF/test.jsp").forward(request, response);
}

public void doPost( HttpServletRequest request, HttpServletResponse response ) throws ServletException, IOException {
    Utilisateur utilisateur = new Utilisateur();
    utilisateur.setNom(request.getParameter("nom"));
    utilisateur.setPrenom(request.getParameter("prenom"));

    Noms tableNoms = new Noms();
    tableNoms.ajouterUtilisateur(utilisateur);

    request.setAttribute("utilisateurs", tableNoms.recupererUtilisateurs());

    this.getRequestDispatcher("/WEB-INF/test.jsp").forward(request, response);
}

```

JSP :

```

<c:forEach var="utilisateur" items="${ utilisateurs }">
    <li><c:out value="${ utilisateur.prenom }" /> <c:out value="${ utilisateur.nom }" /></li>
</c:forEach>

```