Chapitre 3

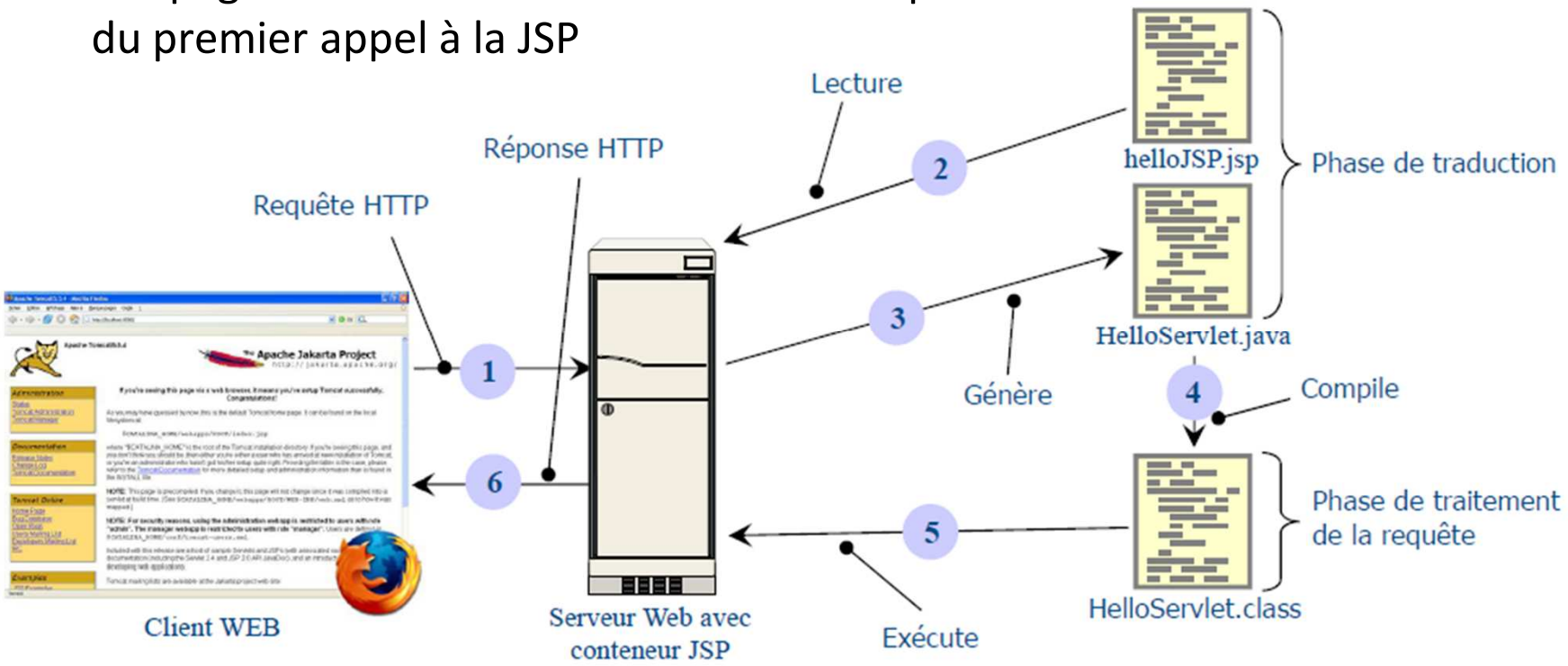
Java Server Pages (JSP)

Plan

1. **C'est quoi JSP ?**
2. **Les Tags: Eléments syntaxiques d'une JSP**
   - **Directives**
   - **Commentaires**
   - **Déclarations**
   - **Scriptlets**
   - **Expressions**
   - **Actions**
3. **Bilan : Génération de Servlet**
4. **Servlets et JSP : architecture MVC**

C'est quoi JSP ?

- JSP = Java Server Pages
- Une JSP est un fichier contenant du code HTML et des fragments de code Java exécutés sur le moteur de Servlets,
- Comparable aux langages côtés serveur de type PHP, ASP, …
- Les pages JSP sont converties en Servlet par le moteur de Servlets lors du premier appel à la JSP



HelloWorld avec une Servlet

- Besoin de modifier le fichier web.xml

```
public class HelloWorldServlet extends HttpServlet {

protected void doGet(HttpServletRequest req, HttpServletResponse res)
        throws ServletException, IOException {
    res.setContentType("text/html");
     PrintWriter out = res.getWriter();
     out.println("<html>");
     out.println("  <head>");
     out.println("     <title>Bonjour tout le monde</title>");
     out.println("  </head>");
     out.println("  <body>");
     out.println("     <h1>Bonjour tout le monde</h1>");
     out.println("     Nous sommes le " + (new java.util.Date().toString()) +
                 " et tout va bien.");
     out.println("  </body>");
     out.println("</html>");
     }
}
```

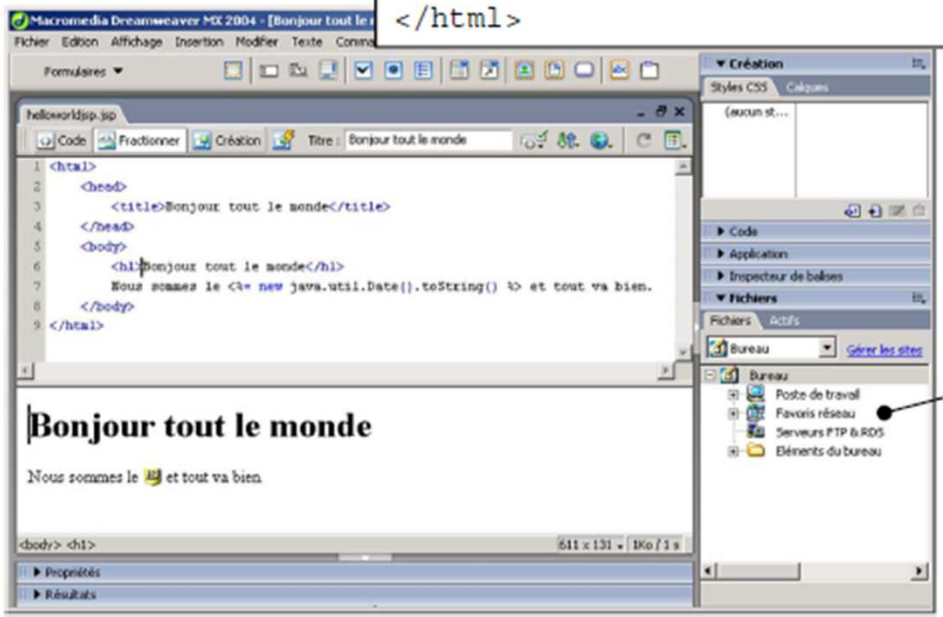
La partie structure du document HTML doit être précisée à l'aide de l'affichage de sortie : **devient vite contraignant**

HelloWorld avec une JSP

- helloworldjsp.jsp doit être placé à la racine de l'app, WEB
- Pas besoin de modifier le fichier web.xml

Ajout de fragment de code Java

```
<html>
<head>
<title>Bonjour tout le monde</title>
</head>
<body>
<h1>Bonjour tout le monde</h1>
Nous sommes le <%= new java.util.Date().toString() %> et tout va bien.
</body>
</html>
```



Utilisation d'un outil d'aide à la conception de page WEB avec prise en charge de code JSP

HelloWorld avec une JSP après la génération

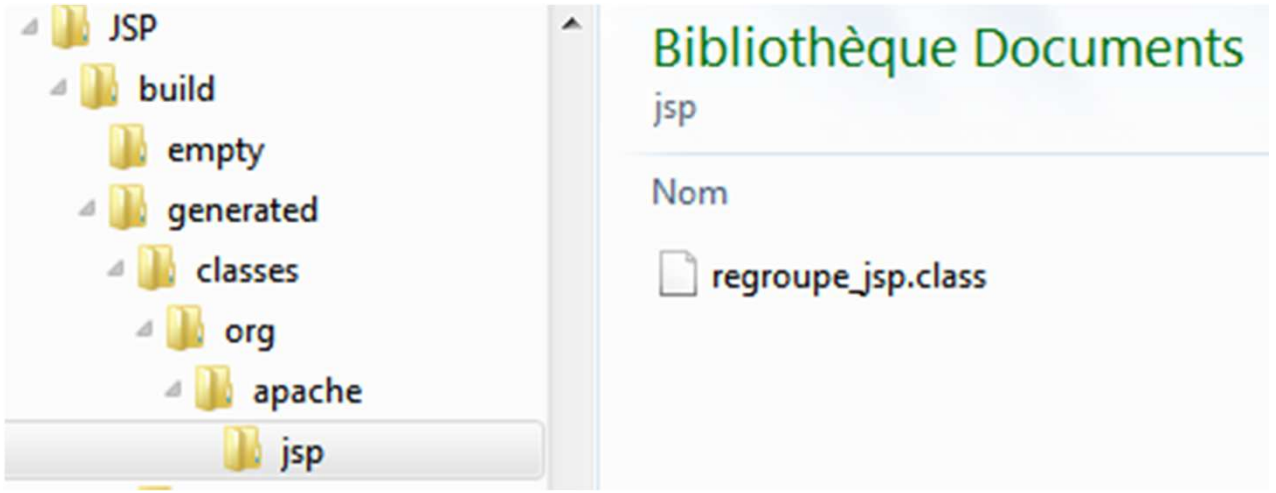
- Exemple : HelloWorld version Servlet

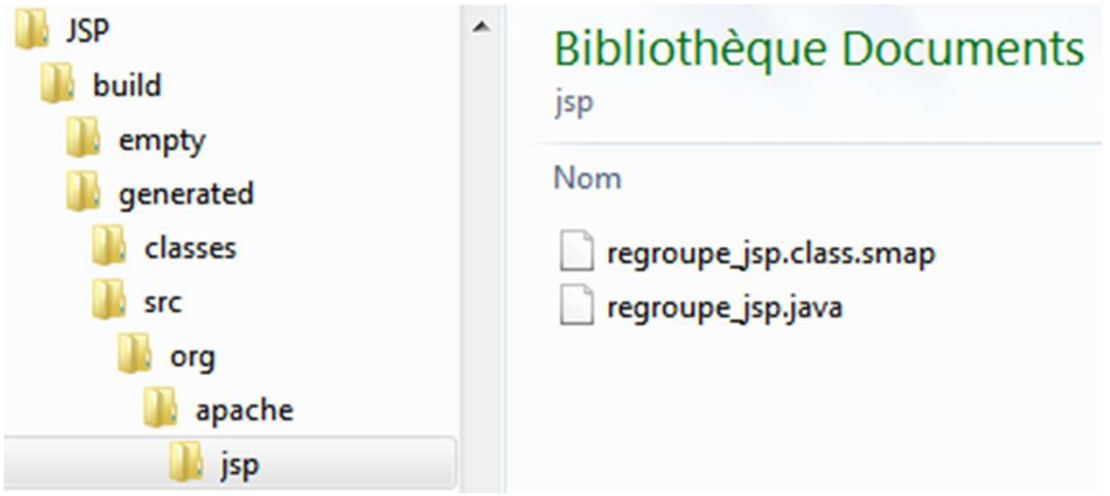
```
public final class helloworldjsp_jsp extends org.apache.jasper.runtime.HttpJspBase
    implements org.apache.jasper.runtime.JspSourceDependent {

  public void _jspService(HttpServletRequest request, HttpServletResponse response)
        throws java.io.IOException, ServletException {
    HttpSession session = null;
    ...
    try {
      ...
      _jspx_out = out;
      out.write("<html>\r\n");out.write("\t<head>\r\n");
      out.write("\t\t<title>Bonjour tout le monde</title>\r\n");
      out.write("\t</head>\r\n");out.write("\t<body>\r\n");
      out.write("\t\t<h1>Bonjour tout le monde</h1>\r\n");
      out.write("\t\tNous sommes le ");out.print( new java.util.Date().toString() );
      out.write(" et tout va bien.\r\n");out.write("\t</body>\r\n");out.write("</html>");
    } catch (Throwable t) {
      if (!(t instanceof SkipPageException)){
        out = _jspx_out;
        ...
        if (_jspx_page_context != null) _jspx_page_context.handlePageException(t);
      }
    } finally {
      if (_jspxFactory != null) _jspxFactory.releasePageContext(_jspx_page_context);
    }
  }
}
```

Hérite de *javax.servlet.jsp.HttpJspPage*
implémente la méthode *_jspService(...)*
équivalente à *service(...)*

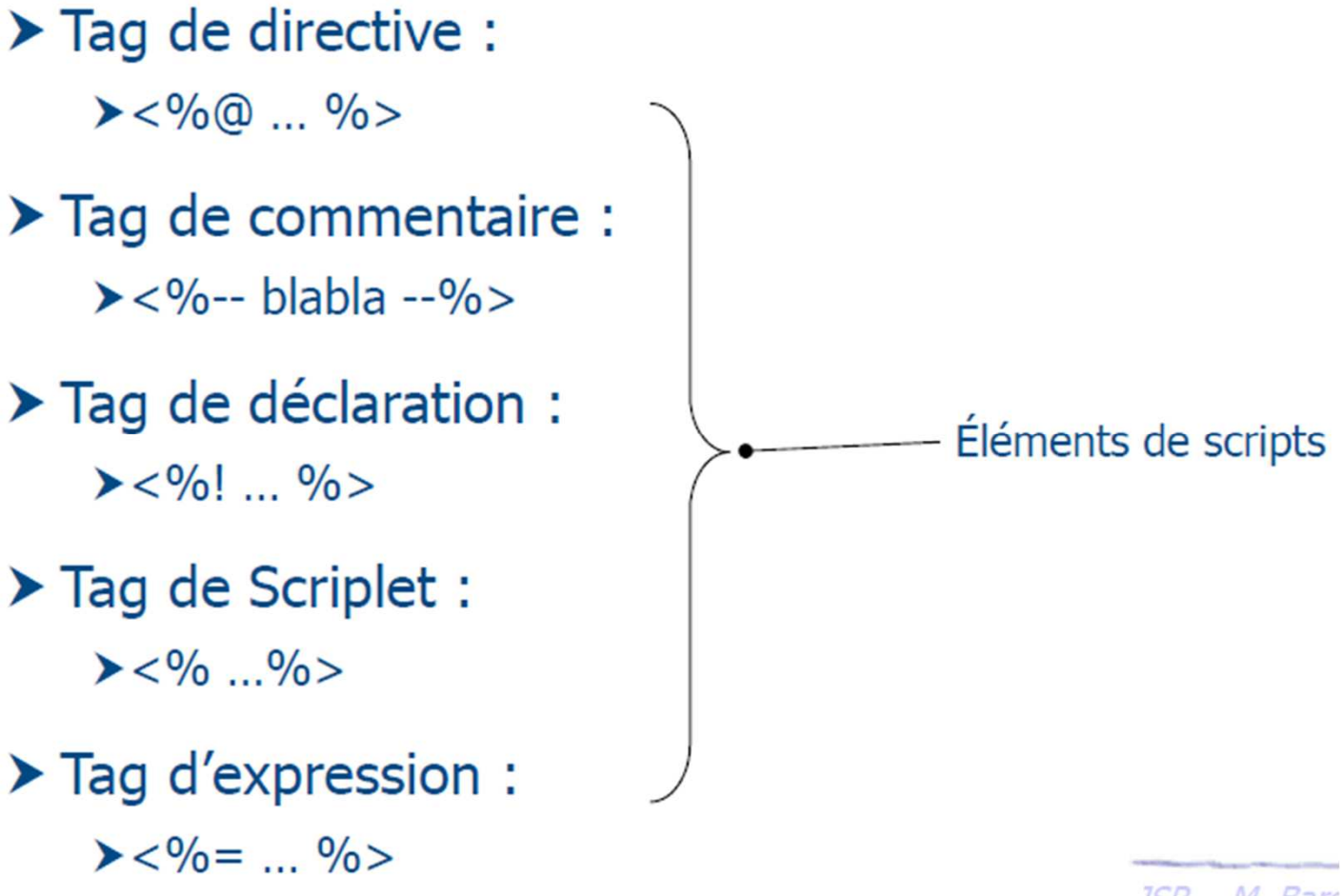
HelloWorld : JSP et Servlet générée





Les Tags: Eléments syntaxiques d'une JSP

- Les Tags permettent de différencier le code HTML au code Java



Directives dans JSP

- Les directives contrôlent comment le serveur WEB doit générer la Servlet
- Elles sont placées entre les symboles <%@ et %>
- Les directives suivantes sont disponibles
  - **include** : indique au compilateur d'inclure un autre fichier

```
<%@ include file="unAutreFichier" %>
```

  - **taglib** : indique une bibliothèque de balises a utiliser

```
<%@ taglib prefix="myprefix" uri="taglib/mytag.tld" %>
```

  - **page** : définit les attributs spécifiques à une page (voir après)

Directives dans JSP : include

- Cette inclusion se fait au moment de la conversion

```
<%@ include file="unAutreFichier" %>
```

- Tout le contenu du fichier externe est inclus comme s'il était saisi directement dans la page JSP
- Ne concerne que les ressources contenues dans le contexte
- La racine du chemin du fichier à inclure est la racine du contexte
- Pas de séparation de la portée des variables
- C'est souvent le cas avec les entêtes et les pieds de pages. Dans ce cas, codez ces parties dans des fichiers séparés et injectez les, via cette directive, dans tous les autre fichiers qui en ont besoin.

Directives dans JSP : include

- Exemple : inclusions par la directive JSP

Le fichier *entete.html*

```
<HTML>
<HEAD>
<TITLE>Page de démonstration</TITLE>
</HEAD>
<BODY>
```

```
<%@ include file = "/entete.html" %>
<%@ include file = "/corps.jsp" %>

Bonjour <%= mon_nom %>

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

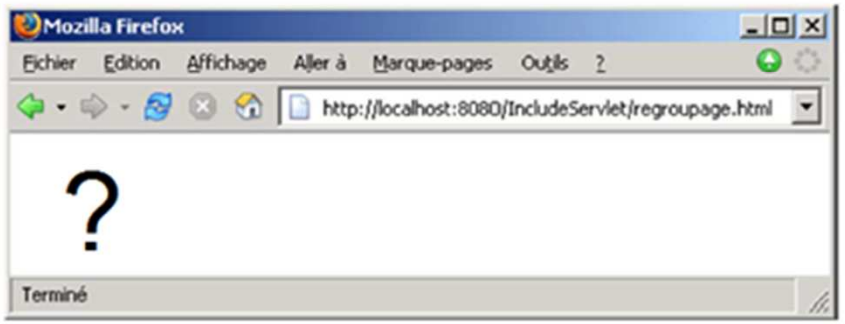
La variable mon_nom est définie dans *corps.jsp*

```
<%! String mon_nom; %>
<% mon_nom = " Etudiant MMI "; %>
```

Le fichier *corps.jsp*

```
Je suis dans le pied de page.
</BODY>
</HTML>
```

Le fichier *piedpage.html*



Directives dans JSP : page

- La directive page définit les attributs spécifiques à une page
  - **import** : importe un paquetage Java. Cette directive résulte en une instruction import dans la Servlet

```
<%@ page import="java.util.*, java.text.*" %>
```

  - **langage** : définit le langage de script utilisé dans la page
  - contentType : définit le type de contenu de la page générée

```
<%@ page contentType="text/plain" %>
```

  - **errorPage** : indique la page à afficher si une exception se produit pendant le traitement de la requête HTTP

```
<%@ page errorPage="toto.jsp" %>
```

  - **isErrorPage** : vaut true si la page est une erreur et false pour une page normale

```
<%@ page isErrorPage=false %>
```

Commentaires dans JSP

- Cet élément de script est utilisé pour faire un commentaire dans le code JSP
- Le texte dans un commentaire JSP ne sera pas envoyé au client ni compilé dans la Servlet
- Les commentaires sont placés entre les symboles <%-- et --%>

```
<html>
      <head>
            <title>Bonjour tout le monde</title>
      </head>
      <body>
            <!-- affichage d'un message classique -->
            <h1>Bonjour tout le monde</h1>
            Nous sommes le <%= new java.util.Date().toString() %> et tout va bien.
            <%-- Utilisation de la classe Date --%>
      </body>
</html>
```

Source de : http://localhost:8080/HelloWorldJSPServl

Fichier Edition Affichage

```
<html>
      <head>
            <title>Bonjour tout le monde</title>
      </head>
      <body>
            <!-- affichage d'un message classique -->
            <h1>Bonjour tout le monde</h1>
            Nous sommes le Wed Jan 26 18:54:15 CET 2005 et tout va bien.

      </body>
</html>
```

Les commentaires JSP n'apparaissent pas dans le code HTML du client

Déclarations dans JSP

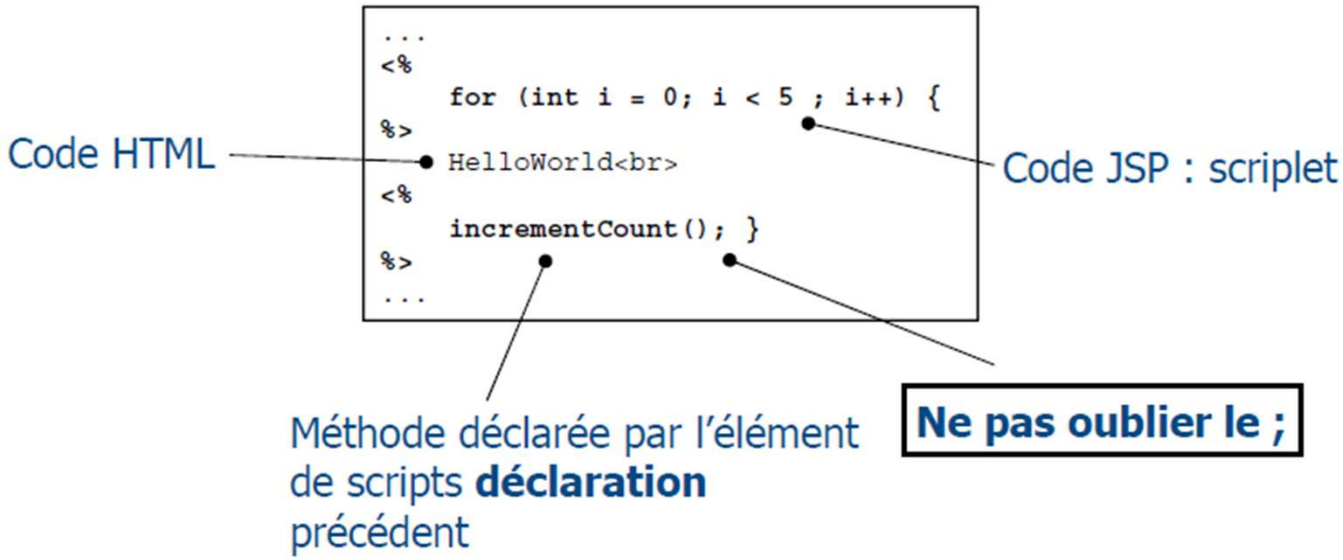
- Une déclaration permet d'insérer du code dans la classe de la Servlet
- Les déclarations sont placées entre <%! et %>
- Elle peut être utilisée pour
  - Déclarer un attribut de classe
  - Spécifier et implémenter des méthodes

```
<%!
    private int count = 0;
    private int incrementCount() { return count++; }
%>
```

- Les attributs et les méthodes déclarées dans la page JSP sont utilisables **dans toute la page JSP**
- Possibilité de redéfinir des méthodes *jspInit()* et *jspDestroy()*

Scriptlets dans JSP

- C'est un bloc de code Java qui est placé dans _jspService(…) de la Servlet générée (équivalent à service(...))
- Les scriplets sont placées entre les symboles <% et %>
- Tout code java a accès :
  - aux attributs et méthodes définis par le tag déclaration <%! … %>
  - aux objets implicites



Expressions dans JSP

- Sert à évaluer une expression et à renvoyer sa valeur
- Les expressions sont placées entre les symboles <%= %>
- Retourne une valeur String de l'expression
- Correspond à une scriplet comme <% out.println(…); %>
- Se transforme en out.println("…"); dans la méthode _jspService(…) après génération
- Comme l'expression est placée dans un appel de méthode, il est interdit de terminer l'expression via un point-virgule.
- Exemple : <%=new Date()%>
- Equivalent à: out.println(new Date()) ;

Actions dans JSP

- Les actions constituent une autre façon de générer du code Java à partir d'une page JSP.
- Les actions se reconnaissent facilement, syntaxiquement parlant : il s'agit de tag XML ressemblant à <jsp:tagName ... />.
- Cela permet d'indiquer que le tag fait partie du namespace (espace de noms) jsp. Le nom du tag est préétabli.
- Enfin, le tag peut, bien entendu comporter plusieurs attributs.

Actions dans JSP

- Il existe plusieurs actions différentes. Les principales sont les suivantes

<jsp:include> : Inclusion coté serveur
- Exemple : <jsp:include page= "entete.jsp" />

<jsp:forward> : Redirection vers une page
- Exemple : <jsp:forward page="affiche.jsp" />

<jsp:useBean> : Instanciation d'un objet java (java bean)
- Exemple :
  - <jsp:useBean id="jbName" class="TheClass" scope="session" />

<jsp:setProperty> : Cette action, permet de modifier une propriété sur un objet créé via l'action *<jsp:useBean ...>*
- Exemple :
  - <jsp:setProperty name="jbName" property="XXX" value="<%= javaExpression %>" />

<jsp:getProperty> : cette action est l'inverse de la précédente : elle permet de retourner dans le flux HTML, la valeur de la propriété considérée.
- Exemple :
  - <jsp:getProperty name="jbName" property="XXX" />

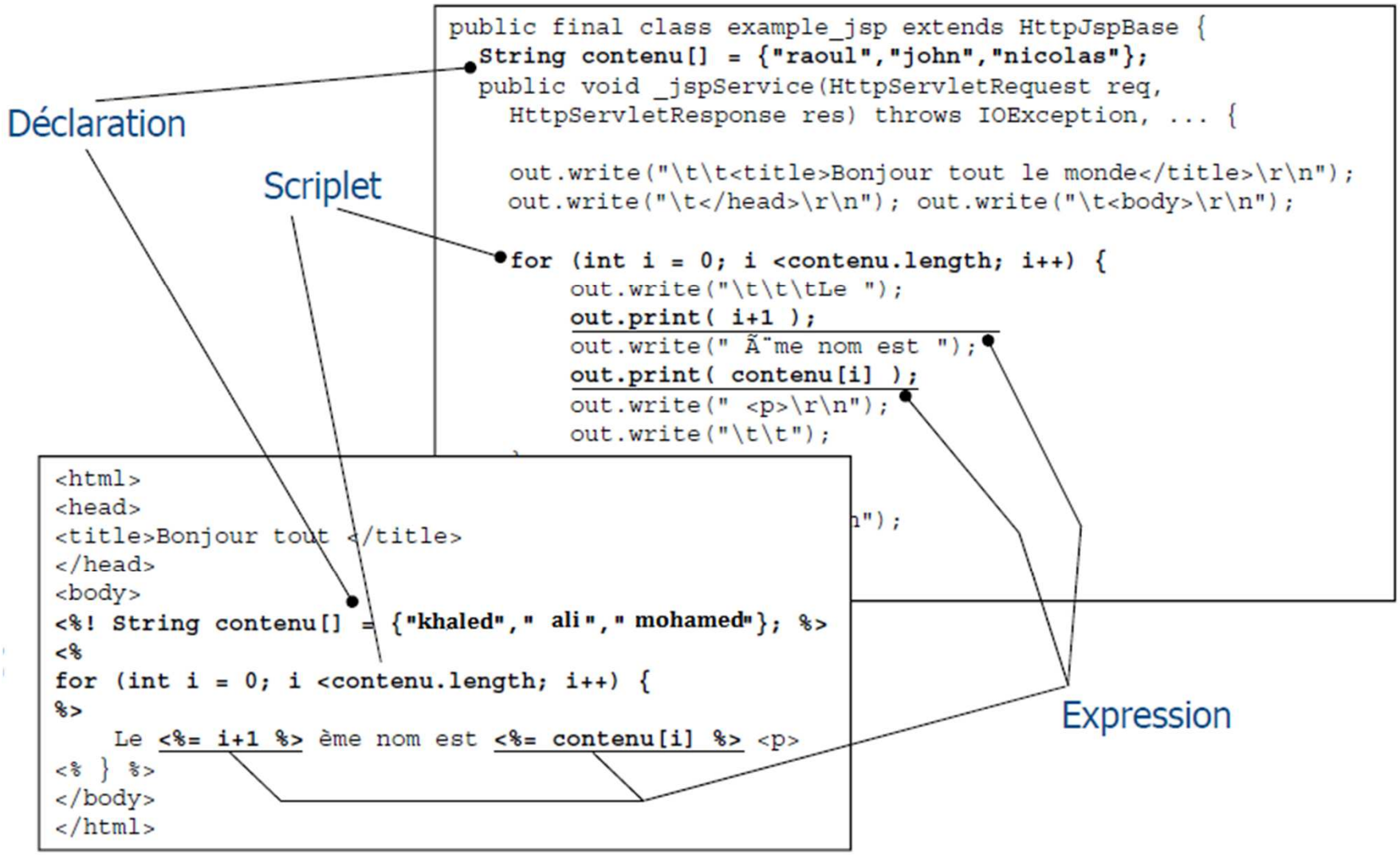
JSP include directive vs include action tag

La sortie obtenue des deux est la même, mais il existe peu de différences remarquables entre elles.

- La directive *'include'* inclut le fichier au moment de la traduction (la phase du cycle de vie d'un JSP où le JSP est converti en servlet équivalent) tandis que l'action *'include'* inclut le fichier au moment de l'exécution.
- Avec la directive *'include',* un changement dans le fichier inclus ne provoque pas une regénération et une compilation de la servlet correspondant à la JSP. Pour insérer un fichier dynamiquement à l'exécution de la servlet il faut utiliser le tag <jsp:include>.
- Différence de syntaxe: Include directive: <%@ include file="file_name" %> par contre, include action est comme suit: <jsp:include page="file_name" />
- Lors de l'utilisation de include action, nous pouvons également passer les paramètres à la page incluse à l'aide de l'action 'param', mais dans le cas d'une directive d'inclusion, ce n'est pas possible.

```
<jsp:include page="file_name" />
<jsp:param name="parameter_name" value="parameter_value" />
</jsp:include>
```

Bilan : Génération de Servlet



Servlets et JSP : architecture MVC

- Un constat : les pages JSP doivent contenir le moins de Java
- Besoin de structurer le code plus finement selon les compétences de chaque technologies (JSP, Servlet, …)
- Utilisation du modèle d'architecture : MVC (Modèle Vue Contrôleur)