

# Module : JEE

## JSP

# Chapitre IV: JSP

## I- Introduction

## II-Les éléments de script JSP

- Balise de scripts JSP
- Balise d'expression
- Balise de déclaration

## III- Les directives JSP

## IV-Les objets implicites

## V- Les balises d'action JSP

# Chapitre IV: JSP

## I. Introduction

Java Server Pages (JSP) est une technologie de programmation côté serveur qui permet de créer des méthodes dynamiques et indépendantes de la plate-forme pour construire des applications Web. Les JSP ont accès à toute la famille des API Java, y compris l'API JDBC pour accéder aux bases de données d'entreprise.

# Chapitre IV: JSP

## I. Introduction

- JSP est basée sur la technologie des Servlets
- Le cycle de vie d'une JSP est le même que celui d'une Servlet
- Les fichiers JSP sont compilés sous forme de Servlets au premier appel
- Recompilés à chaque modification du code source
- Le code Java est intégré au code HTML avec des balises d'échappement `<%` et `%>`
- Les pages JSP sont plus faciles à maintenir que les Servlet car nous pouvons séparer la conception et le développement.
- JSP offre quelques fonctionnalités supplémentaires telles que le langage d'expression, les balises personnalisées, etc.

# Chapitre IV: JSP

## II. Les éléments de script JSP

Les éléments de script permettent d'insérer du code java à l'intérieur du jsp. Il existe trois types d'éléments de Scripting :

1. balise scriptlet
2. balise expression
3. balise de déclaration

# Chapitre IV: JSP

## II. Les éléments de script JSP

### **balise scriptlet**

Une balise scriptlet est utilisée pour exécuter un code source java dans un JSP.

Sa syntaxe est la suivante :

**<% écrire ici le code java %>**

# Chapitre IV: JSP

## II. Les éléments de script JSP

### balise scriptlet

```
2  <%@page contentType="text/html" pageEncoding="UTF-8"%>
3  <html>
4  <head>
5      <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
6      <title>JSP Page</title>
7  </head>
8  <body>
9      <h1>traitement des données reçu par JSP</h1>
10     <%
11         String name=request.getParameter("uname");
12         out.print("welcome "+name);
13     %>
14 </body>
15 </html>
```

# Chapitre IV: JSP

## II. Les éléments de script JSP

### **balise expression**

Le code placé dans la balise JSP expression est écrit dans le flux de sortie de la réponse. Il n'est donc pas nécessaire d'écrire `out.print()` pour écrire des données.

`<%= expression_ici %>`



# Chapitre IV: JSP

## II. Les éléments de script JSP

### balise expression

```
2  <%@page contentType="text/html" pageEncoding="UTF-8"%>
3  <html>
4  <head>
5      <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
6      <title>JSP Page</title>
7  </head>
8  <body>
9      <h1>traitement des données reçu par JSP</h1>
10     <% String nom=request.getParameter("nom");
11         String prenom=request.getParameter("prnom");
12         %> |
13
14     <%= "welcome "+nom %>
15 </body>
16 </html>
```

# Chapitre IV: JSP

## II. Les éléments de script JSP

### balise de déclaration

La balise de déclaration JSP est utilisée pour déclarer les champs et les méthodes.

`<%! Déclaration_de_champ_ou_methode %>`

# Chapitre IV: JSP

## II. Les éléments de script JSP

### balise de déclaration

```
8  <body>
9      <h1>traitement des données reçu par JSP</h1>
10
11      <%! String nom; %>
12
13      <%! int somme(int a, int b){
14
15          return a+b;
16      }
17      %>
18
19      <% nom=request.getParameter("nom");
20          String prenom=request.getParameter("prenom");
21      %>
22
23      <%= "welcome "+nom %> |
24
25      <%= "a+b="+somme(6,9) %>
26
27  </body>
```

## Chapitre IV: JSP

### III. Les directives JSP

Les directives JSP sont des messages qui indiquent au conteneur web comment traduire une page JSP à une servlet correspondante. La syntaxe utilisé est :

**<%@ directive attribute="value" %>**

Il en existe trois types:

- **page** directive
- **include** directive
- **taglib** directive

# Chapitre IV: JSP

## III. Les directives JSP

### page

La directive page définit les attributs qui s'appliquent à une page JSP entière.

`<%@ page attribute1="value1" attribute2="value2" ... %>`

Les 13 attributs de la directive page sont:

```
<%@ page language="java" import="java.util.Date" extends="monpackage.Etudiant" contentType="text/html"
    pageEncoding="UTF-8" info="description simple" buffer="15kb" isELIgnored="true"
    isThreadSafe="false" errorPage="pageException.jsp" isErrorPage="false"
%>

<html>
<body>
    Today is: <%= new java.util.Date() %>
</body>
</html>
```

# Chapitre IV: JSP

## III. Les directives JSP

### include

La directive **include** est utilisée pour inclure le contenu de toute ressource, qu'il s'agisse d'un fichier JSP, d'un fichier HTML ou d'un fichier texte.

`<%@ include file=" le fichier _a_ inclure " %>`

```
3
4 <html>
5   <body>
6     <%@ include file="menu.html" %>
7     <!--contenu de ma page ici-->
8   </body>
9 </html>
10
11
12
```

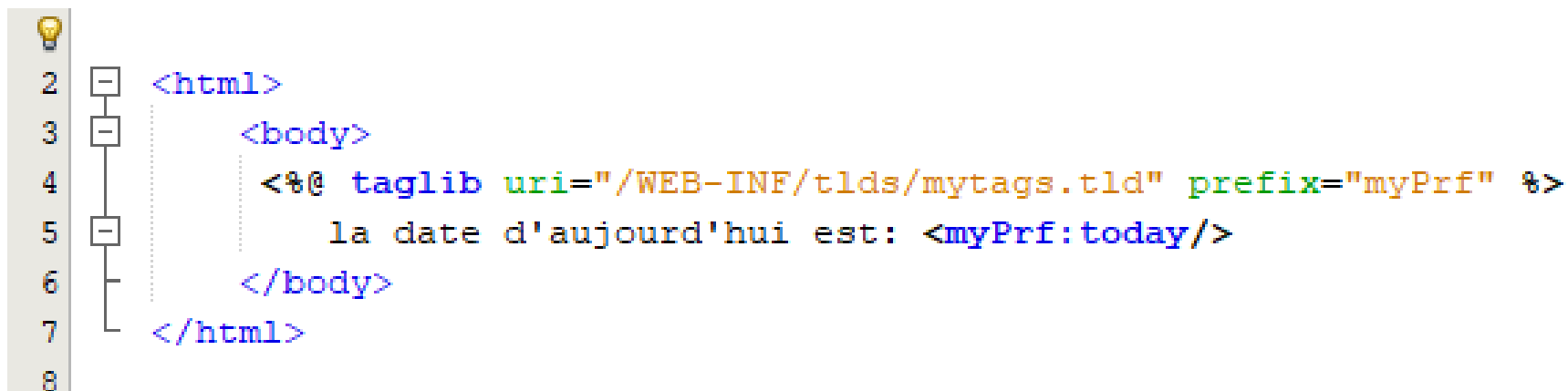
# Chapitre IV: JSP

## III. Les directives JSP

### taglib

La directive **taglib** est utilisée pour définir une bibliothèque qui définit de nombreuses balises. Nous utilisons le fichier TLD (Tag Library Descriptor) pour définir les balises.

`<%@ taglib uri="url_de_fichier_tld" prefix="mon_prefix" %>`



```
2 <html>
3   <body>
4     <%@ taglib uri="/WEB-INF/tlds/mytags.tld" prefix="myPrf" %>
5       la date d'aujourd'hui est: <myPrf:today/>
6     </body>
7   </html>
8
```



## Chapitre IV: JSP

### IV. Les objets implicites

Il existe 9 objets JSP implicites. Ces objets sont créés par le conteneur web et sont disponibles pour toutes les pages JSP.

Les objets implicites disponibles sont :



# Chapitre IV: JSP

## IV. Les objets implicites

Object	Sa classe
out	JspWriter
request	HttpServletRequest
response	HttpServletResponse
config	ServletConfig
application	ServletContext
session	HttpSession
pageContext	PageContext
page	Object
exception	Throwable

# Chapitre IV: JSP

## IV. Les objets implicites

### out

Pour écrire des données dans le tampon, JSP fournit un objet implicite nommé **out**. C'est l'objet de *JspWriter*. Dans le cas d'une servlet, vous devez écrire :

```
2 <%@page contentType="text/html" pageEncoding="UTF-8"%>
3 <html>
4   <body>
5     <% out.print("Today is:"+java.util.Calendar.getInstance().getTime()); %>
6   </body>
7 </html>
8
```

# Chapitre IV: JSP

## IV. Les objets implicites

### request

**request** est un objet implicite de la classe *HttpServletRequest* c'est-à-dire créé pour chaque requête JSP par le conteneur web. Il peut être utilisé pour obtenir des informations sur la requête telles que les paramètres, les informations d'en-tête, l'adresse distante, le nom du serveur...

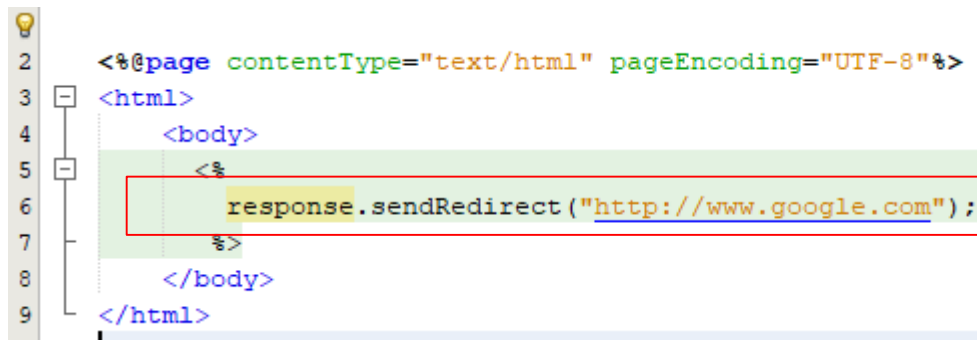
```
2  <%@page contentType="text/html" pageEncoding="UTF-8"%>
3  <html>
4      <body>
5          <%
6              String name=request.getParameter("nom");
7              out.print("ton nom est : "+name);
8          %>
9      </body>
10 </html>
```

# Chapitre IV: JSP

## IV. Les objets implicites

### response

**response** est un objet implicite de la classe *HttpServletResponse*. L'instance de HttpServletResponse est créée par le conteneur web pour chaque requête JSP.



```
2  <%@page contentType="text/html" pageEncoding="UTF-8"%>
3  <html>
4      <body>
5          <%
6              response.sendRedirect("http://www.google.com");
7          %>
8      </body>
9  </html>
```

The screenshot shows a code editor with a line number margin on the left (2-9). The code is a JSP page that sets the content type to text/html and page encoding to UTF-8. It contains an HTML structure with a body. Inside the body, there is a scriptlet that calls `response.sendRedirect("http://www.google.com");`. The `response` object is highlighted in yellow, and the entire scriptlet is enclosed in a red rectangular box.

# Chapitre IV: JSP

## IV. Les objets implicites

### session

**session** est un objet implicite de la classe *HttpSession*. Le développeur Java peut utiliser cet objet pour définir, obtenir ou supprimer un attribut ou pour obtenir des informations sur la session.

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<html>
  <body>
    <%
      String name=request.getParameter("uname");
      out.print("Welcome "+name);

      session.setAttribute("user",name);

    %>
    <a href="monprofil.jsp">passer à la page suivante</a>
  </body>
</html>
```

# Chapitre IV: JSP

## IV. Les objets implicites

### session

**session** est un objet implicite de la classe *HttpSession*. Le développeur Java peut utiliser cet objet pour définir, obtenir ou supprimer un attribut ou pour obtenir des informations sur la session.

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<html>
  <body>
    <%
      String name=request.getParameter("uname");
      out.print("Welcome "+name);

      session.setAttribute("user",name);

    %>
    <a href="monprofil.jsp">passer à la page suivante</a>
  </body>
</html>
```

## Chapitre IV: JSP

Tp\_4\_