
JSP

Rappels

- Application web
- Serveurs Web, moteur de servlets
- Fonctionnement

Exercice 1 : De HTML au JSP

Soit la page HTML suivante intitulé *bienvenue.html* :

```
<HTML>
      <BODY> Bienvenue au cours Applications Web </BODY>
</HTML>
```

Q1.1. Placer le fichier html résultant sur le serveur et accéder à la page html ainsi créée via un navigateur.

Q1.2. Reprendre l'exercice 1 en remplaçant l'extension .html par .jsp Effectuer les mêmes opérations en considérant le .jsp au lieu du .html. Que constater vous ?

Les JSP ne se réduisent pas à remplacer l'extension .html en .jsp. Nous allons voir par la suite comment utiliter des JSP pour la création de pages web dynamiques.

Exercice 2 : Contenu dynamique & Expressions

Q2.1. Manipulation des dates

Afficher la date de chargement de la page. Pour cela on crée date.jsp suivant :

```
<HTML>
      <BODY> Aujourd'hui est le <%= new java.util.Date() %> </BODY>
</HTML>
```

Q2.2. Accéder à cette page jsp via un navigateur. Que se passe t-il ?

Exercice 3 : Scriptlets

La programmation n'est pas réduite à de simple ajout d'expressions dans des pages html. En effet, Il est possible d'écrire des bloques de codes Java dans une JSP. Un bloque Java (scriptlet) est exécuté à chaque fois que la JSP est invoquée.

On reprend la jsp de la question 2.1. et on souhaite maintenant de manipuler la date à l'aide de scriptlets.

Q3.1. Modifier la jsp de la question 2.1 pour pouvoir afficher la date y ajoutant du code java

- Créer une variable '*date'* de type *java.util.Date*
- Instancier la variable '*date'* à l'aide de *java.util.Date();*
- Ecrire la date (valeur de la variable '*date'*)

Q3.2. Générer une table en html. Par exemple, la table contiendra les nombres de 1 à n (ici = = 5). Pour cela, vous devez écrire un scriptlet qui initialise la variable n. Puis afficher tous les entiers de 1 à n sous forme du tableau ci-contre.

Nombre	1
Nombre	2
Nombre	3
Nombre	4
Nombre	5

Q3.3. Générer une page selon la valeur d'une variable booléenne. Pour cela, il faut d'abord écrire un scriptlet qui initialise une variable booléenne 'statuser', Puis on génère une page HTML qui affiche 'bienvenue' si 'statuser' est vraie et 'au revoir' dans le cas contraire.

Exercice 4 : Directives

Une directive JSP a la syntaxe suivante : <%@ chaine_de_caractères

Nous allons reprendre l'exemple concernant la date de la question Q3.1. et lui ajouter à la première ligne la directive '@ page' :

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

Q.4.1. Modifier la JSP en conséquence (tenir compte de l'import).

Nous souhaitons maintenons importer la classe 'hello' du package 'bonjour' défini par le programme 'hello.java' suivant :

```
package bonjour;

public class hello{
  public String affiche(){
    return "Bonjour";
  }
}
```

Q.4.2. Ecrire la page JSP d'importer la classe 'hello' et d'exécuter la classe 'affiche'.

Nous allons maintenant considérer l'attribut contenType de la directive page qui spécifie le type MIME et l'encodage de caractère utilisé par la réponse JSP. Le type Mime par défaut et 'text/html' et l'encodage de caractère est 'ISO-8859-1'.

Nous considérons ici deux JSP qui affichent différents résultats après affectation de la valeur de l'attribut contentType de la directive page. Dans le premier cas, la page script-page-contenttype-html.jsp affiche une table sous format html (contentType a la valeur 'text/html') alors que le second JSP, script-page-contenttype-xml.jsp, affiche un le résultat sous format xml (contentType a la valeur 'text/xml').

Q4.3. Ecrire script-page-contenttype-html.jsp et script-page-contenttype-xml.jsp

Q4.4. Ecrire une JSP et y inclure le fichier de la JSP 'bienvenue.jsp' (voir question Q1.2.) à l'aide de la directive '@ include' :

```
<%@ include file="bienvenue.jsp" %>
```

Exercice 5 : Déclarations

Les JSP sont compilées en classes et tous les scriptlets de la JSP sont placés dans une seule méthode de la classe. On peut alors déclarer des variables et des méthodes et les utiliser dans vos scriptlets et expressions.

Q5.1. Reprendre la JSP de l'exercice 4 dédiée à la manipulation de la date. Nous allons la modifier avec des déclarations pour effectuer la même tâche (afficher la date) :

- Déclarer une variable theDate et l'initialiser avec la date courante
- Ecrire la fonction getDate() qui retourne la date (valeur de la variable theDate)
- Afficher la date en utilisant une expression qui permet d'afficher la date en appelant la fonction getDate()

Accéder à la JSP ainsi modifiée avec un navigateur. Que constatez –vous ?

Q5.2. Modifier la JSP précédente pour remédier à ses limites.

Exercice 6 : Tags

Q6.1. Reprendre les exercices concernant les directive include et forward et réécrire les pages JSP en utilisant cette fois les balises jsp:include et jsp:forward.

Exercice 7 : Sessions & Cookies

Q7.1. Ecrire une page HTML représentant un formulaire qui permet de saisir le nom d'une personne.

Q7.2. Ecrire une JSP qui de sauver le nom saisi précédemment dans une session en utilisant l'attribut 'theName'. Cette JSP affiche aussi un lien avec la page suivante intitulée savename.jsp.

Q7.3. Ecrire la page nextpage.jsp qui affiche le nom saisi au début.

Q7.4. Reprendre cet exercice en considérant cette fois en plus du nom de la personne, son prénom, son adresse, son âge et sa situation actuelle.

Les cookies sont des données envoyées par le serveur web au navigateur client. Les cookies sont stockés dans le disque dur du client sous forme de petits fichiers textes. Ils aident le serveur web à identifier les internautes. Il devient alors de tracer les internautes.

Nous allons manipuler les cookies dans des pages JSP. Pour cela, nous allons d'abord ajouter un cookie par une page JSP, puis afficher les valeurs du même cookie par une autre page JSP.

Q7.5. Commencer par écrire une page (cookie-form.html) qui demande à l'utilisateur de saisir son nom.

Q7.6. Une fois le nom de l'utilisateur saisi, la page setcookie.jsp effectue ensuite l'ajout du cookie. Il faudra utiliser

- La fonction getParameter de l'objet 'request' pour récupérer le nom saisi par l'utilisateur et le stocker dans la variable 'username'
- Initialiser la variable timestamp avec la date courante
- Créer ensuite la cookie : Cookie cookie = new Cookie ("username",username);
- Définir l'âge maximum du cookie (délais avant son expiration) en utilisant la fonction setMaxAge de l'objet cookie.
- Ajouter le cookie à l'aide de la fonction addCookie de l'objet response.

Cette page JSP affiche aussi un lien vers la JSP qui a pour rôle d'afficher les valeurs du cookie.

Q7.7. Ecrire enfin la JSP qui affiche les valeurs du cookie.