

Les JSP : Introduction

Les JSP ont été créés afin de concurrencer directement les langages ASP et PHP. En effet, même si la construction de servlet reste simple, l'ergonomie de l'ASP et du PHP, directement intégrés dans une page HTML a longtemps retardé l'utilisation des servlets. Le but des JSP est d'offrir exactement la même facilité d'écriture.

Avantage de JSP contre ASP : Premièrement c'est écrit en Java et non dans un langage du type VBScript ou d'autres. Java étant un langage de développement très performant, les applications en JSP sont plus puissantes et mieux adaptées aux applications complexes et réutilisables. Ensuite, le JSP est bien plus portable que l'ASP.

Avantage de JSP contre PHP : Les deux se valent mais l'API java reste plus étendue que les fonctionnalités offertes par PHP. De plus, Java est naturellement objet alors que PHP peine à se mettre à "niveau". De fait, JSP sera recommandé pour les grandes applications.

Avantage de JSP contre les Servlets pures : Le JSP permet de séparer le contenu statique d'une page HTML, du contenu dynamique de celui-ci. Et il est plus simple d'écrire du HTML à l'aide de JSP que d'une servlet pure.

De plus les pages JSP n'ont pas besoin d'être compilées ni ne nécessitent de changement complexe comme la redéfinition du CLASSPATH. On doit simplement les mettre où l'on souhaite dans notre site, et elles marcheront si le serveur supporte les JSP ...

Les pages JSP sont en fait des servlets !! Elles sont traduites en servlet qui sont compilées la première fois qu'un utilisateur la regarde.

JSP utilise un premier moyen d'insertion de code : les scripting elements. A l'instar des balises PHP (type `<?php ?>`), ils sont constituées de balises semblables à du code HTML.

Ainsi JSP propose une balise pour entrer directement du code : `<% %>`. On peut également utiliser la balise suivante pour afficher directement une valeur : `<%= expr %>`. L'expression `expr` est alors évaluée et directement envoyée dans le flux HTML. Si l'on souhaite effectuer des déclarations alors il faut utiliser la balise `<%! %>`.

Enfin pour générer un commentaire qui ne soit pas transmis à la page HTML, on peut utiliser la balise `<%-- --%>`

Exercice 1

Soit le programme Java suivant qui calcule et imprime la conversion de degrés Fahrenheit en degrés Celsius.

```
import java.text.NumberFormat;

import java.text.DecimalFormat;

public class F2C {

    public static double f2c(int f) {

        return ((f - 32) * 5) / 9.0;

    }

    public static void main(String[] args) {

        System.out.println("Fahrenheit Celsius");

        NumberFormat fmt = new DecimalFormat("##0.000");

        for (int f = 32; f <= 212; f += 20) {

            String cs = fmt.format(f2c(f));

            System.out.println(f + " " + cs);

        } } }
```

Ecrivez le code JSP qui fait le même calcul en mettant le résultat dans une table où les données sont centrées dans les cellules.

Exercice 2 - JSP Sélection

Téléchargez le fichier [selection.jsp](#). Vous devez compléter ce fichier de manière à afficher la couleur sélectionnée. Pour cela, il vous faut introduire un scriptlet qui doit tester si la requête est bien du type POST. Si c'est le cas, il faut afficher le texte "Vous avez choisi la couleur <couleur>" où <couleur> sera la valeur du paramètre "couleur" de la requête.

Rappels :

- Dans le scriptlet, vous pouvez accéder à l'objet représentant la requête par le mot-clé request
- Pour savoir quelles méthodes sont invocables sur cet objet, consultez l'API de sa classe (javax.servlet.http.HttpServletRequest) comme d'habitude !

Exercice 2

Soit un formulaire avec les champs suivants :

-Nom

- Langages de programmations connus : Java, C, C++, VisualBasic, Lisp, Prolog, Autres (à cocher).

-Soumission

Ecrivez le code de la page JSP qui affiche les valeurs reçues par le formulaire.

Exercice 3 - Redirection des exceptions

Copiez le code selection1.jsp dans un fichier selection2.jsp. Modifiez le code pour ajouter le choix "Orange". Modifiez le scriptlet de manière à lever une exception avec pour texte descriptif "Cette couleur n'est pas belle" si le choix "Orange" est validé.

Faites-en sorte que cette exception soit envoyée à une page erreur.jsp. Ecrivez la page erreur.jsp qui se contentera d'afficher le texte de l'exception.

Exercice 4 - Portée des variables

Créez un Java Bean "Compteur" avec une unique propriété "nombre" qui est un nombre entier initialisé à 0. Compteur doit disposer d'une méthode qui permet d'incrémenter sa propriété nombre.

Ecrivez ensuite une page compteur.jsp. Dans cette page, créez 3 objets Counter avec les portées respectives "application", "session" et "request". Dans le code de la page, invoquez la méthode d'incrémentation puis affichez la valeur de la propriété nombre de chaque compteur.

Pour bien comprendre les différences de portée, constatez l'évolution des compteurs en rechargeant la page de votre navigateur plusieurs fois, puis en quittant ce navigateur et en le redémarrant.

Exercice 5

a) Définir un bean qui calcule le poids idéal d'une personne à partir de son sexe et de sa taille.

Voici la formule pour calculer le poids idéal en fonction du sexe et de la taille :

Poids idéal femme = $(72.7 * \text{taille}) - 58$

Poids idéal homme = $(62.1 * \text{taille}) - 44.7$

Voici les propriétés du bean qui calcule le poids idéal :

Nom de la propriété	Mode	Type
sexe	lecture/écriture	char
taille	lecture/écriture	double
poids idéal	lecture seulement	double

b) Dans quelle répertoire de notre environnement de développement local faut-il mettre le code du bean ?

c) Ecrivez le code de la page JSP poidsIdeal.jsp pour :

-créer et initialiser le bean correspondant aux paramètres de la requête de la manière la plus concise possible

- calculer et imprimer le poids idéal

Exercice 6

Soit la fonction Math.random() qui affiche un nombre aléatoire entre 0 et 1.

Soit le code la page exercice4a.jsp :

```
<%!
private static double nombreAleatoireClasse = Math.random();
%>
<html>
<head>
<title> Exercice 4a </title>
</head>
<body>
Voici le nombre aléatoire: <%= nombreAleatoireClasse %>
</body>
</html>
```

Soit le code la page exercice4b.jsp :

```
<%
double nombreAleatoireClasse = Math.random();
%>
<html>
<head>
```

```
<title> Exercice 4-b </title>
```

```
</head>
```

```
<body>
```

Voici le nombre aléatoire: <%= nombreAleatoireClasse %>

```
</body>
```

```
</html>
```

a) Que pouvons-nous dire par rapport à l'affichage de la valeur aléatoire dans toutes les requêtes à la page exercice4a.jsp jusqu'à ce que le conteneur soit arrêté ?

b) Que pouvons-nous dire par rapport à l'affichage de la valeur aléatoire dans toutes les requêtes à la page exercice4b.jsp jusqu'à ce que le conteneur soit arrêté ?

c) Expliquez de manière concise la différence entre les pages exercice4a.jsp et exercice4b.jsp par rapport à l'affichage de la valeur aléatoire

Exercice 7 (Base de Donnée)

Une clinique médicale regroupe plusieurs médecins. Sa base de données gère les informations relatives à tous les médecins de la clinique et tous les patients, en indiquant les rendez-vous pris avec les médecins. Soit les données suivantes :

- pour les médecins de la clinique : un numéro d'identification, le nom et la spécialité (gynécologie, urologie, etc) ;
- pour les patients : un numéro d'identification, le nom, le numéro de téléphone ;
- pour un rendez-vous pris par un patient avec un médecin : la date et l'heure.

Ecrire toutes les commandes de création de tables pour représenter les données de cette clinique médicale.

Exercice 8

Soit le fichier bd.inc.jsp :

```
<%!
```

```
public static void setDriver(String driver) throws ClassNotFoundException {
```

```
Class.forName(driver);
```

```
}
```

```
public static Connection getConnection(String url, String username, String password)
```

```
throws SQLException {
```

```
String user = "?user=" + username;
```

```
String pword = "&password=" + password;
```

```
return DriverManager.getConnection(url + user + pword);
```

```
} %>
```

En utilisant les fonctions définies dans le fichier bd.inc.jsp et la base de données définie dans l'exercice précédent, écrire le code d'une page JSP qui affiche le résultat d'une requête SQL avec les noms des médecins, les noms des clients et les dates de leur rendez-vous ordonnées par le nom du médecin.

Exercice 9

Soit le code du TimeBean qui contient l'heure et les minutes au moment de sa création :

```
package examens;

import java.util.*;

public class TimeBean {

    private int hours;

    private int minutes;

    public TimeBean() {

        Calendar now = Calendar.getInstance();

        this.hours = now.get(Calendar.HOUR_OF_DAY);

        this.minutes = now.get(Calendar.MINUTE);

    }

    public int getHours() {

        return hours;

    }

    public int getMinutes() {

        return minutes;

    }

}
```

Soit le code de la page JSP suivante où quatre beans de la classe TimeBean sont créés avec différentes valeurs de l'attribut scope :

```
<jsp:useBean id="time1" class="examens.TimeBean" scope="page" />

<jsp:useBean id="time2" class="examens.TimeBean" scope="request" />

<jsp:useBean id="time3" class="examens.TimeBean" scope="session" />

<jsp:useBean id="time4" class="examens.TimeBean" scope="application" />

<html>
```

```
<head> <title> Exercice sur portee 2 </title> </head>

<body>

<p> Time1

<jsp:getProperty name="time1" property="hours" />:

<jsp:getProperty name="time1" property="minutes" /> </p>

<p> Time2

<jsp:getProperty name="time2" property="hours" />:

<jsp:getProperty name="time2" property="minutes" /> </p>

<p> Time3

<jsp:getProperty name="time3" property="hours" />:

<jsp:getProperty name="time3" property="minutes" /> </p>

<p> Time4

<jsp:getProperty name="time4" property="hours" />:

<jsp:getProperty name="time4" property="minutes" /> </p>

</body>

</html>
```

- a) Quelles sont les valeurs affichées quand nous appelons la page tout de suite après le démarrage du conteneur JSP à 10h00 ?
- b) Quelles sont les valeurs affichées quand nous rafraichissons la page 5 minutes après le démarrage du conteneur JSP dans le même navigateur ?
- c) Quelles sont les valeurs affichées quand nous appelons la page 15 minutes après le démarrage du conteneur JSP dans un autre navigateur ?