



Les JSTL(Tag Lib)

Réalisé par: Meryem Sabiq / Imane Chawki

Définition de JSTL

La JSTL (JavaServer Pages Standard Tag Library) est une bibliothèque de tags (ou balises) prédéfinis qui permet de simplifier l'écriture de code dans les pages JSP (JavaServer Pages). Elle offre une collection de tags pour effectuer des opérations courantes telles que les boucles, les tests conditionnels, la gestion des chaînes de caractères, la manipulation des dates et des heures, la gestion des tableaux, et bien d'autres encore.



Avantages de la JSTL

La JSTL présente plusieurs avantages pour le développement d'applications web:

01

Réduction de la quantité de code HTML et Java dans les pages JSP

02

Séparation de la logique de présentation et de la logique métier

03

Réutilisation de code

Le langage EL (Expression Language).

JSTL propose un langage particulier constitué d'expressions qui permettent d'utiliser et de faire référence à des objets Java accessibles dans les différents contextes de la page JSP. Le but est de fournir un moyen simple d'accéder aux données nécessaires à une JSP.

Voici quelques exemples d'expressions JSTL :

`${variable}` : récupère la valeur de la variable.

`${objet.propriete}` : récupère la valeur de la propriété d'un objet.

`${variable1 + variable2}` : effectue une opération arithmétique sur les variables.

`${condition ? valeur_si_vrai : valeur_si_faux}` : teste une condition et retourne une valeur en fonction du résultat.

```
<p>Bonjour ${nom} !</p>
```

```
<p>${age >= 18 ? 'Vous êtes majeur' : 'Vous êtes mineur'}</p>
```

Architecture des JSTL

La JSTL possède quatre bibliothèques de tags :

Rôle	TLD	Uri
Fonctions de base	c.tld	http://java.sun.com/jstl/core
Traitements XML	x.tld	http://java.sun.com/jstl/xml
Internationalisation	fmt.tld	http://java.sun.com/jstl/fmt
Traitements SQL	sql.tld	http://java.sun.com/jstl/sql

les balises JSTL peuvent être utilisées dans la page JSP en utilisant une syntaxe spéciale qui inclut le préfixe de la bibliothèque de balises et le nom de la balise. Par exemple, la balise "if" de la bibliothèque de balises "core" est utilisée avec la syntaxe suivante :

php



Copy code

```
<c:if test="${condition}">
    <!-- Actions à effectuer si la condition est vraie -->
</c:if>
```

Comment intégrer les JSTL dans un projet

Pour intégrer la JSTL dans un projet, il est nécessaire de suivre les étapes suivantes :

1. Télécharger la bibliothèque JSTL.
2. Ajouter la bibliothèque JSTL au projet.
3. Inclure la directive "taglib".
4. Utiliser les balises JSTL.

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


Présentation des différents tags

Core

La bibliothèque de tags Core de la JSTL (JavaServer Pages Standard Tag Library) fournit des balises pour la gestion des flux de contrôle, la définition de variables, la manipulation de chaînes de caractères, la gestion des URL et des paramètres de requête, etc.

1. `c:if` : Permet d'exécuter une instruction conditionnelle.

```
<c:if test="{age} > 18">  
  <p>Vous êtes majeur.</p>  
</c:if>
```

2.` c:forEach` : Permet de parcourir une collection (tableau, liste, map...) et d'appliquer un traitement à chaque élément.

```
<c:forEach items="${listeObjets}" var="objet">
  <p>${objet.nom} - ${objet.prix} €</p>
</c:forEach>
```

3.` c:choose` : Permet d'effectuer des tests conditionnels multiples.

```
<c:choose>
  <c:when test="${age > 18}">
    <p>Vous êtes majeur.</p>
  </c:when>
  <c:when test="${age == 18}">
    <p>Vous venez d'avoir 18 ans.</p>
  </c:when>
  <c:otherwise>
    <p>Vous êtes mineur.</p>
  </c:otherwise>
</c:choose>
```

4. `c:set` : Permet de définir une variable ou de modifier la valeur d'une variable existante.

```
<c:set var="prix" value="15.99" />
<c:set var="quantite" value="{quantite + 1}" />

<p>Le prix unitaire est de ${prix} €.</p>
<p>Vous avez commandé ${quantite} produits.</p>
```

5 `c:url` : Permet de générer une URL avec des paramètres de requête.

```
<c:url var="urlProduit" value="/produit">
  <c:param name="id" value="123" />
</c:url>

<p>Le lien vers le produit est : <a href="{urlProduit}">Produit 123</a></p>
```

Formatting

La bibliothèque de balises JSTL formatting fournit des balises pour formater des nombres, des dates et des heures. Voici quelques exemples de balises de la bibliothèque

1. `<fmt:formatDate>` : permet de formater une date en utilisant un format spécifié

```
<p>La date du jour est : <fmt:formatDate value="${now}" pattern="EEEE dd MMMM yyyy" /></p>
```

2. `<fmt:parseNumber>` : permet de convertir une chaîne de caractères en un nombre en utilisant un format spécifié

```
<fmt:parseNumber var="quantite" value="${param.quantite}" />
```

xml

La bibliothèque de balises JSTL xml fournit des balises pour parcourir et traiter des documents XML.

1.x:parse : permet de parser un document XML et de stocker le résultat dans une variable.

```
<x:parse var="document" xml="${xmlDocument}" />
```

2.x:out : permet d'afficher une valeur d'un élément XML

```
<x:out select="$document//nom" />
```

sql

La bibliothèque de balises JSTL sql fournit des balises pour exécuter des requêtes SQL et traiter les résultats dans une page JSP

1. `sql:setDataSource` : permet de définir les paramètres de connexion à la base de données.

```
<sql:setDataSource var="dataSource" driver="com.mysql.jdbc.Driver"  
url="jdbc:mysql://localhost:3306/test" user="root" password="" />
```

2. `sql:query` : permet d'exécuter une requête SQL et de stocker les résultats dans une variable.

```
<sql:query var="resultat" dataSource="${dataSource}">  
    SELECT * FROM produits WHERE categorie = 'informatique'  
</sql:query>
```

Exemple:

```
public class JSTL extends HttpServlet {
    private static final long serialVersionUID = 1L;

    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

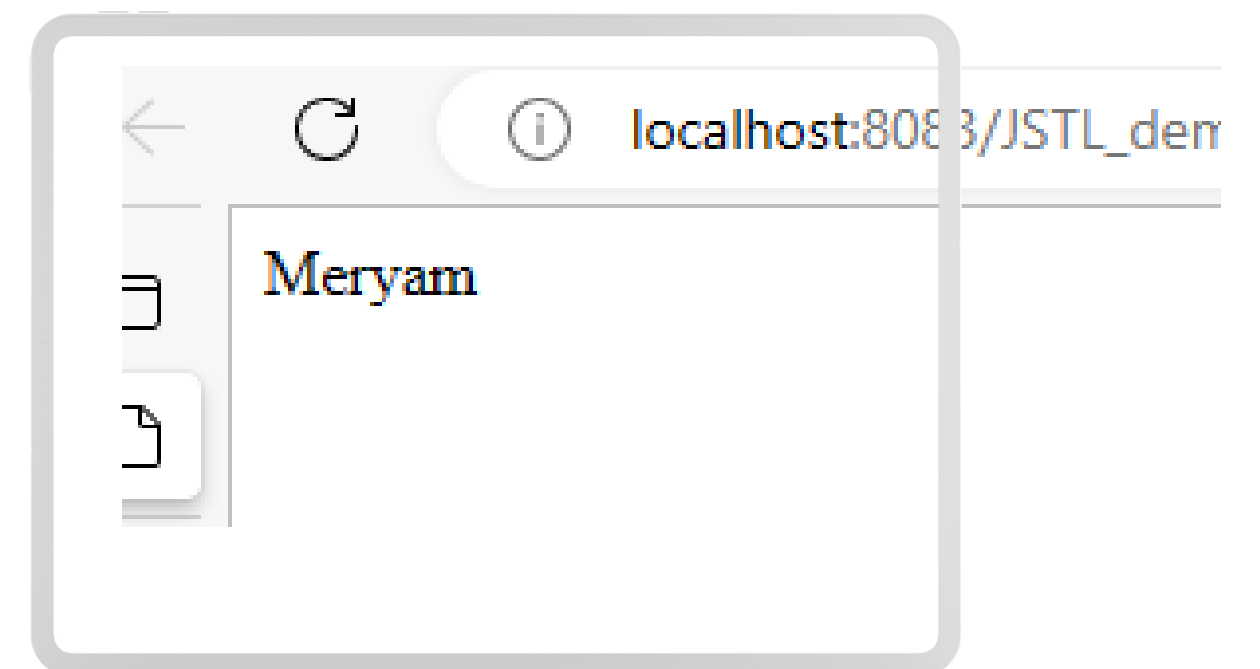
        String name = "Meryam";
        request.setAttribute("Nom", name);
        RequestDispatcher rd= request.getRequestDispatcher("resultat.jsp");
        rd.forward(request, response);

    }
}
```

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>

<%
    String nom =request.getAttribute("Nom").toString();
    out.print(nom);
%>

</body>
</html>
```



```

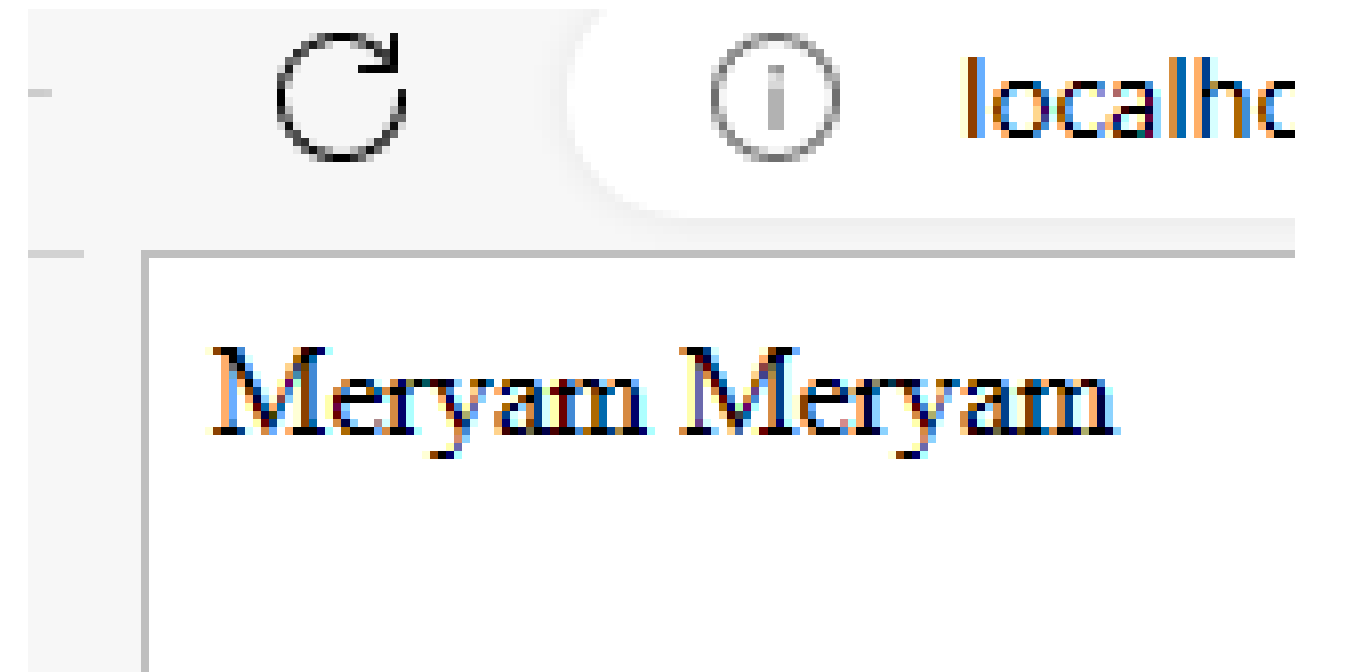
<body>

<%
    String nom =request.getAttribute("Nom").toString();
    out.print(nom);
%>

${Nom}

</body>

```

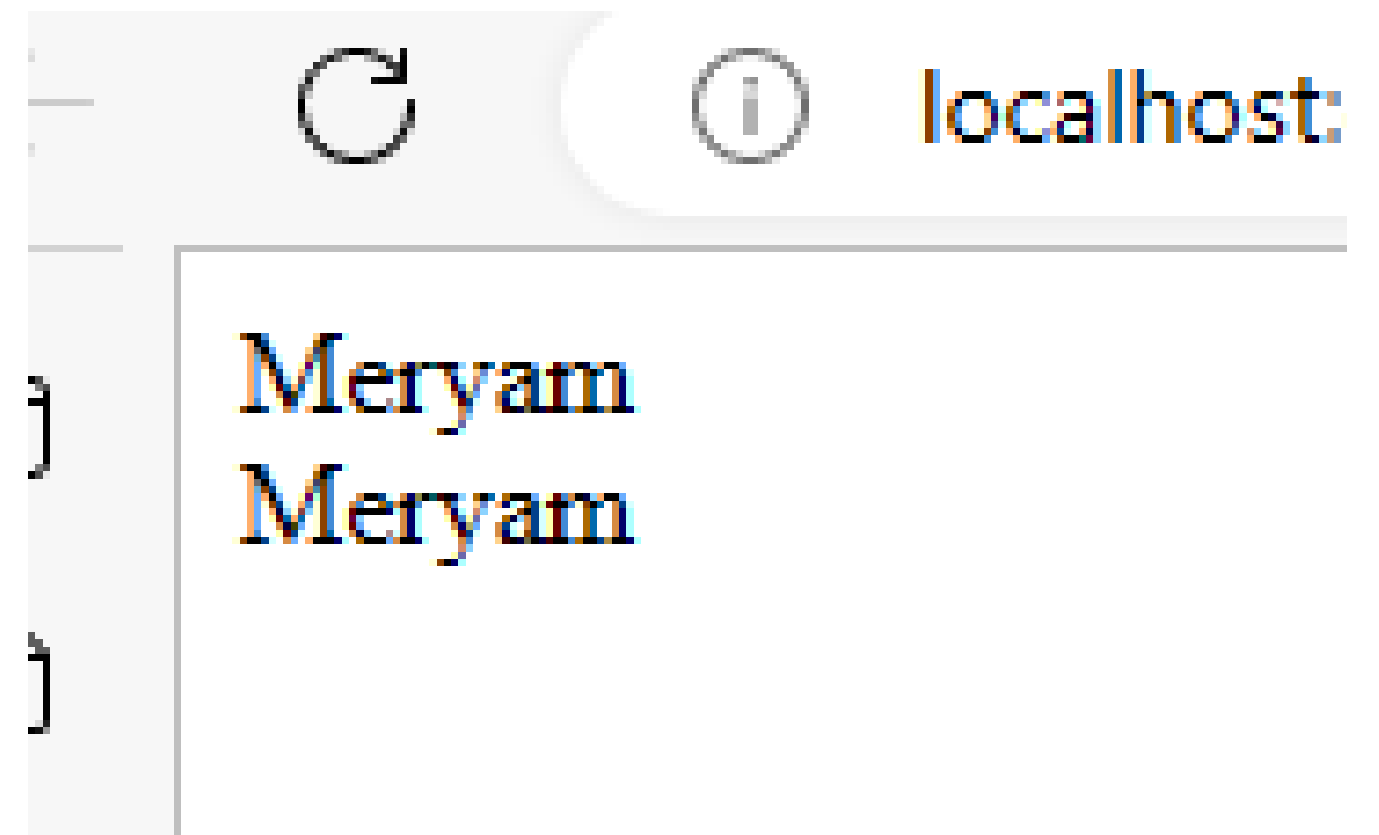


```

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
    ${Nom}<br>
    <c:out value="${Nom}" />

</body>
</html>

```





Merci