**Web.xml pour config :**

<jsp-config>

<jsp-property-group>

<url-pattern>\*.jsp</url-pattern>

<el-ignored>true</el-ignored>

</jsp-property-group>

</jsp-config>

**Objet Implicites de la JSPype** de l'objet

Description

| **Identifiant** | **Type de l'objet** | **Description** |
| --- | --- | --- |
| **pageContext** | PageContext | Il fournit des informations utiles relatives au contexte d'exécution. Entre autres, il permet d'accéder aux attributs présents dans les différentes portées de l'application. Il contient également une référence vers tous les objets implicites suivants. |
| **application** | ServletContext | Il permet depuis une page JSP d'obtenir ou de modifier des informations relatives à l'application dans laquelle elle est exécutée. |
| **session** | HttpSession | Il représente une session associée à un client. Il est utilisé pour lire ou placer des objets dans la session de l'utilisateur courant. |
| **request** | HttpServletRequest | Il représente la requête faite par le client. Il est généralement utilisé pour accéder aux paramètres et aux attributs de la requête, ainsi qu'à ses en-têtes. |
| **response** | HttpServletResponse | Il représente la réponse qui va être envoyée au client. Il est généralement utilisé pour définir le Content-Type de la réponse, lui ajouter des en-têtes ou encore pour rediriger le client. |
| **exception** | Throwable | Il est uniquement disponible dans les pages d'erreur JSP. Il représente l'exception qui a conduit à la page d'erreur en question. |
| **out** | JspWriter | Il représente le contenu de la réponse qui va être envoyée au client. Il est utilisé pour écrire dans le corps de la réponse. |
| **config** | ServletConfig | Il permet depuis une page JSP d'obtenir les éventuels paramètres d'initialisation disponibles. |
| **page** | objet this | Il est l'équivalent de la référence this et représente la page JSP courante. Il est déconseillé de l'utiliser, pour des raisons de dégradation des performances notamment. |

**-------------------------------------PAREIL MAIS POUR EXPRESSION LANGUAGE-------------------------------------**

| **Catégorie** | **Identifiant** | **Description** |
| --- | --- | --- |
| JSP | **pageContext** | Objet contenant des informations sur  l'environnement du serveur. |
| Portées | **pageScope** | Une Map qui associe les noms et valeurs des attributs  ayant pour portée la page. |
|  | **requestScope** | Une Map  qui associe les noms et valeurs des attributs ayant pour portée la requête. |
|  | **sessionScope** | Une Map  qui associe les noms et valeurs des attributs ayant pour portée la session. |
|  | **applicationScope** | Une Map  qui associe les noms et valeurs des attributs ayant pour portée l'application. |
| Paramètres de requête | **param** | Une Map qui associe les noms et valeurs des paramètres  de la requête. |
|  | **paramValues** | Une Map  qui associe les noms et multiples valeurs \*\* des paramètres de la requête sous forme de tableaux de String. |
| En-têtes de requête | **header** | Une Map qui associe les noms et valeurs des paramètres  des en-têtes HTTP. |
|  | **headerValues** | Une Map  qui associe les noms et multiples valeurs \*\* des paramètres des en-têtes HTTP sous forme de tableaux de String. |
| Cookies | **cookie** | Une Map qui associe les noms et instances des cookies. |
| Paramètres d’initialisation | **initParam** | Une Map qui associe les données contenues dans les  champs <param-name> et <param-value> de  la section <init-param> du fichier web.xml. |

**JSTL CORE**

**Liste**

<c:forEach var="item" items="${tirage}" **begin**="0" **end**="9" > (Limiter éléments)

<c:out value="${item}" />

</c:forEach>

--------------------------------------------------------------

<table>

<tr>

<th>Valeur</th>

<th>Cube</th>

</tr>

<c:forEach var="i" begin="0" end="7" step="1">

<tr>

<td><c:out value="${i}"/></td>

<td><c:out value="${i \* i \* i}"/></td>

</tr>

</c:forEach>

</table>

-----------------------------------------------------------------

Pour une MAP :

<c:forEach items="${maListe}" var="news" varStatus="status">

<div class="news">

<div class="titreNews">

<c:out value="${news['titre']}" />

</div>

<div class="corpsNews">

<c:out value="${news['contenu']}" />

</div>

</div>

</c:forEach>

----------------------------------------------------------------------------

Chaine de caractères

<c:forTokens var="sousChaine" items="salut; je suis un,gros;zéro+!" delims=";,+">

${sousChaine}<br/>

</c:forTokens>

**Affectation Variable**

<c:set var="message" value="Salut les zéros !" scope="request" />

<c:out value="${requestScope.client.nom}" default="Aucune Valeur n'a été rentrée"/>

**Affectation en variable Session**

<c:set var="locale" scope="session">

<c:out value="${param.lang}" default="FR"/>

</c:set>

**MODIFICATION**

<c:set target="${coyote}" property="prenom" value="Wile E."/>

**SUPPRESSION Variable**

<c:remove var="maVariable" scope="session" />

**Test IF**

<c:if test="${ 12 > 7 }" var="maVariable" scope="session">

Ce test est vrai.

</c:if>

**IF / ELSE IF / ELSE**

<c:choose>

<c:when test="${expression}">Action ou texte.</c:when>

...

<c:otherwise>Autre action ou texte.</c:otherwise>

</c:choose>

**LIEN**

<c:url value="/monSiteWeb/countZeros.jsp">

<c:param name="nbZeros" value="${countZerosBean.nbZeros}"/>

<c:param name="date" value="22/06/2010"/>

</c:url>

**REDIRECTION**

<c:redirect url="http://www.siteduzero.com">

<c:param name="mascotte" value="zozor"/>

<c:param name="langue" value="fr"/>

</c:redirect>

**INCLUSION Page**

<%-- Importer une page distante dans une variable

Le scope par défaut est ici page si non précisé. --%>

<c:import url="http://www.siteduzero.com/zozor/biographie.html" var="bio" scope="page"/>

<%-- Les attributs valables pour <c:url/> le sont aussi pour la redirection.

Ici par exemple, l'utilisation de paramètres --%>

<c:import url="footer.jsp">

<c:param name="design" value="bleu"/>

</c:import>

**JSTL XML**

<news>

<article id=*"1"*>

<auteur>Pierre</auteur>

<titre>Foo...</titre>

<contenu>...bar !</contenu>

</article>

<article id=*"27"*>

<auteur>Paul</auteur>

<titre>Bientôt un LdZ J2EE !</titre>

<contenu>Woot ?</contenu>

</article>

<article id=*"102"*>

<auteur>Jacques</auteur>

<titre>Coyote court toujours</titre>

<contenu>Bip bip !</contenu>

</article>

</news>

**Parse le doc XML**

<!-- Contenu Brut -->

<c:import url=*"monDocument.xml"* varReader=*"monReader"*>

<!-- Contenu parsé XML : on stocke la structure du XML dans le var-->

<x:parse var=*"doc"* doc=*"*${monReader}*"* scope=*"request"*/>

<x:out select=*"count($doc/news/article/auteur)"* />

</c:import>

l'expression XPath *($doc/news/article/auteur)* renvoie bel et bien un ensemble de nœuds, en l'occurrence les nœuds "auteur" ; cet ensemble de nœuds est stocké dans une structure de type NodeSet, un type propre à XPath qui implémente le type Java standard NodeList.

Dans une expression XPath, pour faire référence à une variable nommée nomVar on n'utilise pas${nomVar} comme c'est le cas dans une EL, mais $nomVar

<%-- Sélectionne le nœud 'article' dont l'attribut 'id' a pour valeur le contenu de la variable

nommée 'idArticle' qui a été passée en paramètre de la requête, via l'objet implicite param --%>

<x:out select="$doc/news/article[@id=$param:idArticle]"/>

**IF**

<%-- Afficher le titre de la news postée par 'Paul' --%>

<x:if select="$doc/news/article[auteur='Paul']">

Paul a déjà posté une news dont voici le titre :

<x:out select="$doc/news/article[auteur='Paul']/titre" />

</x:if>

**IF / ELSE IF**

<%-- Affiche le titre de la news postée par 'Nicolas' si elle existe, et un simple message sinon --%>

<x:choose>

<x:when select="$doc/news/article[auteur='Nicolas']">

Nicolas a déjà posté une news dont voici le titre :

<x:out select="$doc/news/article[auteur='Nicolas']/titre" />

</x:when>

<x:otherwise>

Nicolas n'a pas posté de news.

</x:otherwise>

</x:choose>

**BOUCLE**

<!-- Affiche les auteurs et titres de tous les articles -->

<p>

<x:forEach var="element" select="$doc/news/article">

<strong><x:out select="$element/auteur" /></strong> :

<x:out select="$element/titre" />.<br/>

</x:forEach>

</p>

**FORMAT XSL**

<?xml version="1.0" encoding="utf-8"?>

<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

<xsl:template match="/">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>Mise en forme avec XSLT</title>

</head>

<body>

<table width="1000" border="1" cellspacing="0" cellpadding="0">

<tr>

<th scope="col">Id</th>

<th scope="col">Auteur</th>

<th scope="col">Titre</th>

<th scope="col">Contenu</th>

</tr>

<xsl:for-each select="/news/article">

<tr>

<td>

<xsl:value-of select="@id" />

</td>

<td>

<xsl:value-of select="auteur" />

</td>

<td>

<xsl:value-of select="titre" />

</td>

<td>

<xsl:value-of select="contenu" />

</td>

</tr>

</xsl:for-each>

</table>

</body>

</html>

</xsl:template>

</xsl:stylesheet>

Application transformation :

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

<%@ taglib uri="http://java.sun.com/jsp/jstl/xml" prefix="x" %>

<c:import varReader="xslFile" url="test.xsl">

<c:import varReader="xmlFile" url="monDocument.xml">

<x:transform doc="${xmlFile}" xslt="${xslFile}"/>

</c:import>

</c:import>

----------------------------------------------------------------------------------------------------

<c:import var="xslFile" url="test.xsl"/>

<c:import var="xmlFile" url="monDocument.xml"/>

<x:transform doc="${xmlFile}" xslt="${xslFile}">

<x:param name="couleur" value="orange" />

</x:transform>

**WEB.XML**

<web-app>

<jsp-config>

<jsp-property-group>

<url-pattern>\*.jsp</url-pattern>

<include-prelude>/WEB-INF/taglibs.jsp</include-prelude>

</jsp-property-group>

</jsp-config>

**FORMULAIRES**

<p class="${empty erreurs ? 'succes' : 'erreur'}">${resultat}</p> // Test Unitaire sur class