



Java avancé

JSP avancées

Objectifs

- Découvrir des technologies complémentaires
- Savoir utiliser les principales balises JSTL



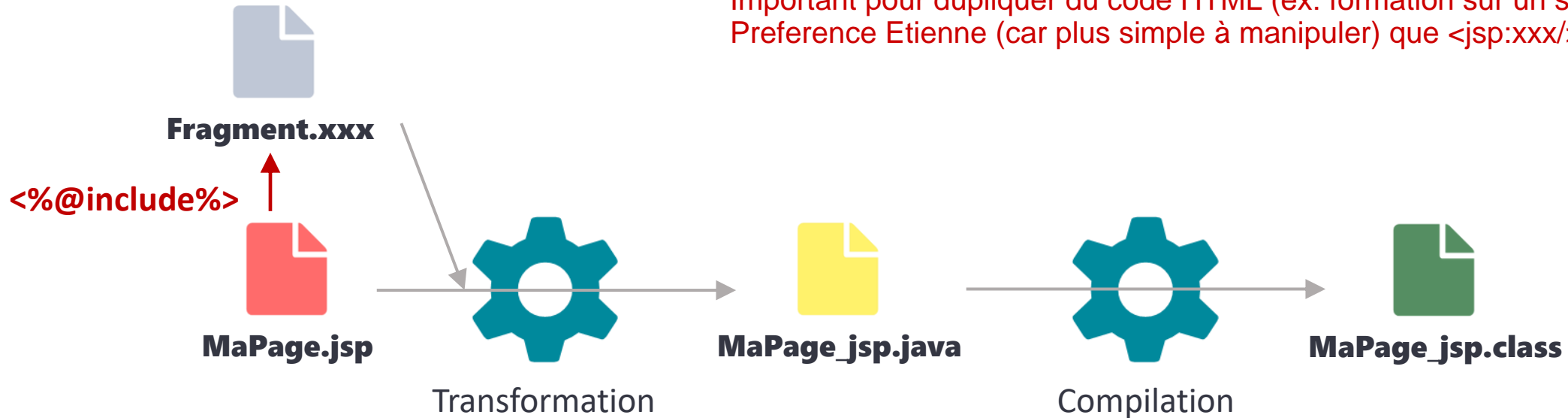
Les principales technologies disponibles

- La directive `<%@include%>`
- Les balises JSP
- Les librairies JSTL



La directive `<%@include%>`

- Les JSP avancées



`<%@include%>` :

Important pour dupliquer du code HTML (ex: formation sur un site internet)
Preference Etienne (car plus simple à manipuler) que `<jsp:xxx/>`

`<%@include%>` :
inclut une JSP dans une autre jsp

a la compilation JEE, un code avec html et java sera générée



Les balises JSP

- Elles cachent du code Java en le remplaçant

```
<jsp:xxx/>
```

- Inclusions

```
<jsp:include page="url de la page">
```

```
    <jsp:param value="valeur" name="parametre"/>
```

```
</jsp:include>
```

Adapté lorsque les éléments sont relativement fixes (ex: header, footer)

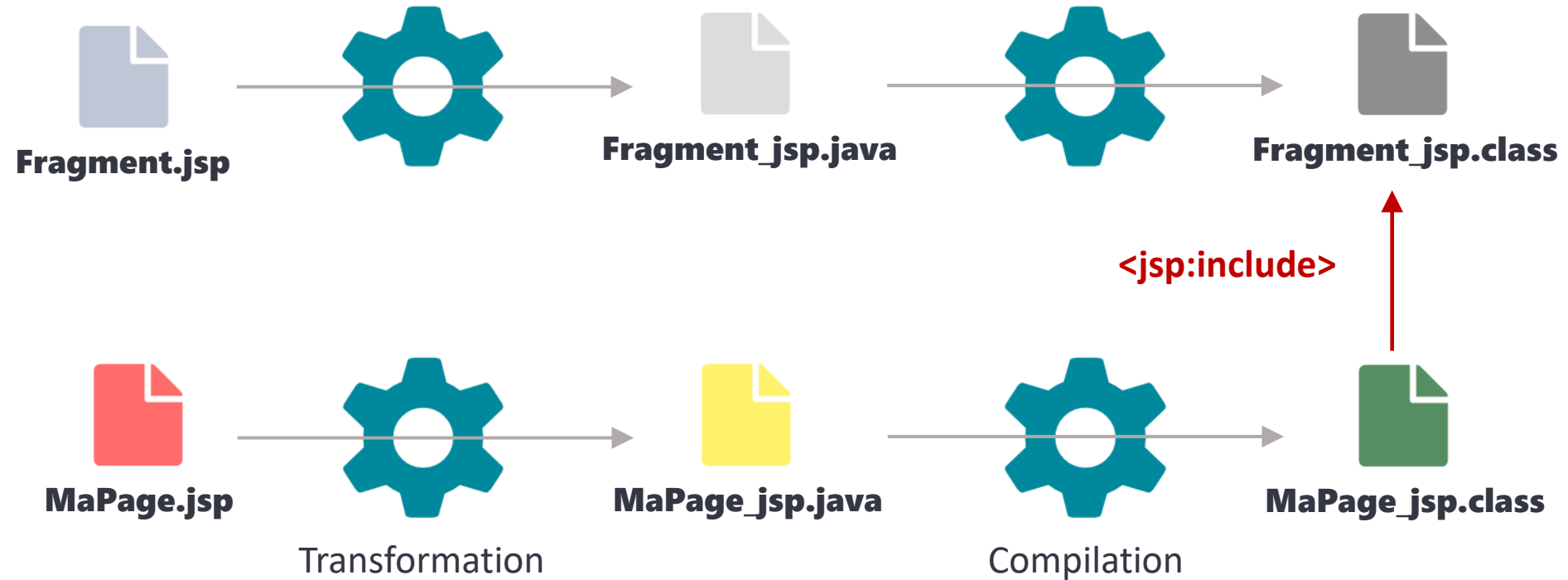
- Lecture de données

Abordée juste
après...



La balise <jsp:include>

- Les JSP avancées



Demo

Les inclusions dans les JSP



- Syntaxe simple

`${expression}`

- Nomenclature pointée

`${monObjet.maVariable}`

`${monObjet.monTableau[0]}`

`${monObjet.maMethode() }`

`toString()` par exemple

- Structure de code impossible

- Pas d'alternative
- Par de boucle

- Ne lève jamais d'exception



Lecture d'attributs avec l'EL

- Lecture ciblée dans un contexte

```
${pageScope['nomAttribut']} OU ${pageScope.nomAttribut}  
${requestScope['nomAttribut']} OU ${requestScope.nomAttribut}  
${sessionScope['nomAttribut']} OU ${sessionScope.nomAttribut}  
${applicationScope['nomAttribut']} OU ${applicationScope.nomAttribut}
```

- Lecture non ciblée

```
${nomAttribut}
```

- Recherche dans l'ordre suivant : page, request, session et application

Préciser le lieu de recherche (page, request,...) si on a plusieurs attributs du même nom



Les principaux autres objets disponibles

- Pour la lecture des paramètres de requête

```
${param.nomDuParametre}
```

- Pour la lecture des en-têtes de requête

```
${header.nomEntete}
```

- Pour la lecture des cookies

```
${cookie.nomCookie}
```

- Pour la lecture du contexte de page

```
${pageContext.request.contextPath}
```

permet de récupérer le nom de l'application (sert au niveau de toutes les URL)

-Les liens (par exemple chemin d'accès vont fonctionner même si on renomme un projet.war)

De plus si un client renomme l'application, les liens fonctionnent quand même



Demo

Retour sur l'EL



Java Server Pages Standard Tag Library 1.2



JSR 52



Les bibliothèques JSTL disponibles

!	Nom de la librairie	Rôle
	core	Librairie de base Outillage pour alternatives, répétitives...
	x	Librairie de manipulation XML
	fmt	Librairie d'internationalisation et de formatage
	sql	Librairie pour faire des traitements SQL
	functions	Librairie de fonctions utilitaires disponibles pour les instructions EL

on ne met pas de SQL dans le JSP



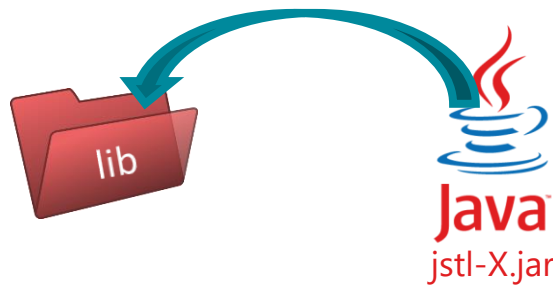
La directive `<%@ taglib %>`

- Permet l'import d'une librairie JSTL
- Syntaxe

```
<%@ taglib prefix="XXX" uri="namespace de la librairie" %>
```

- Utilisation de la librairie

```
<XXX:balise attribut="valeur"></XXX:balise>
```



La librairie core

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

- Les conditionnelles

```
<c:if test="${expression EL}">
```

...

```
</c:if>
```

l'uri est fixe:

<http://java.sun.com/jsp/jstl/core>

c: raccourci de core
f ou fun pour function

<c:else n'existe pas
<c:otherwise

```
<c:choose>
```

```
<c:when test="${expression EL}">
```

...

```
</c:when>
```

```
<c:otherwise>
```

...

```
</c:otherwise>
```

```
</c:choose>
```



La librairie core

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

- Les répétitives

```
<c:forEach var="element" items="${collection}">  
    ${element.propriete}  
</c:forEach>
```

permet d'afficher des listes (de contacts par exemple
comme dans TpAnnuaire)

<%@ taglib prefix="c"
utilisation de librairie core



Quelques opérateurs utiles de l'EL

- Les JSP avancées

Domaine	Opérateurs disponibles
Les opérateurs de comparaison	== (eq), !=(ne), <(lt), >(gt), <=(le), >=(ge), empty
Les opérateurs logiques	&&(and), (or), !(not)
L'opérateur conditionnel	? :
Les opérateurs arithmétiques	+, -, *, /(div), %(mod)
⚠ L'opérateur de concaténation	+=

On modifie la donnée avec l'op de concaténation, mais normalement on ne le fait pas dans une JSP

```
<c:if test="$ {maVariable==12}">
```

```
<c:if test="$ {!empty maVariable}">
```

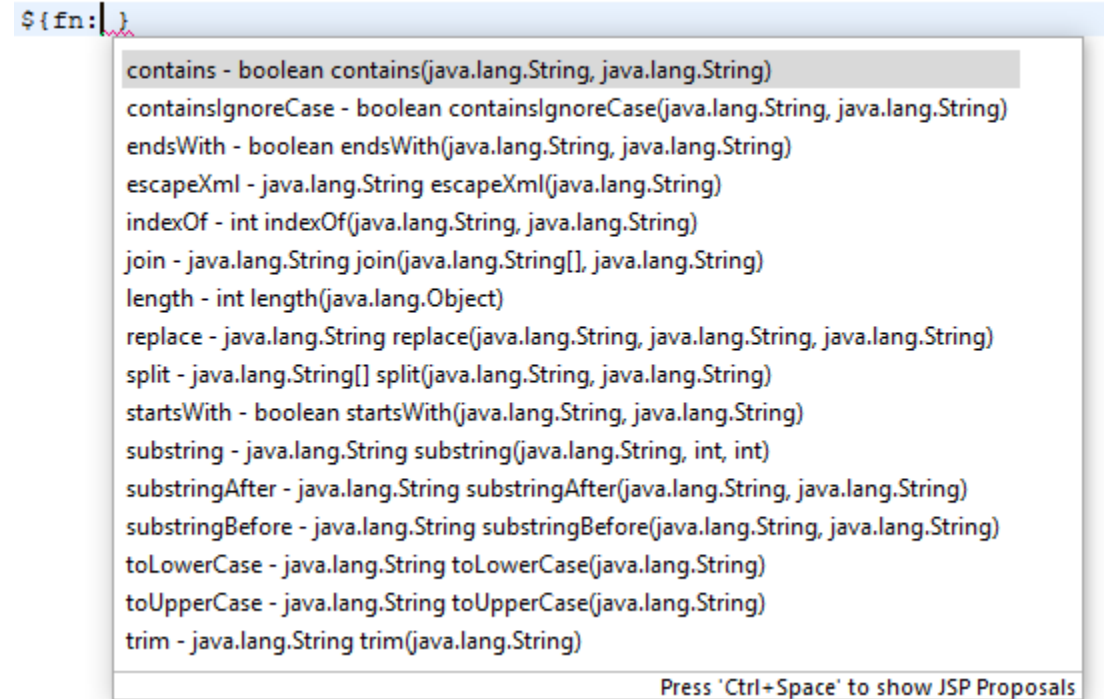


La librairie functions

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

- Pas de balise
- Utilisation dans les expressions EL

Met à disposition une série de fonctionnalités



Demo

Les balises JSTL



Atelier Annuaire pro





Java avancé

JSP avancé