# JSTL et EL

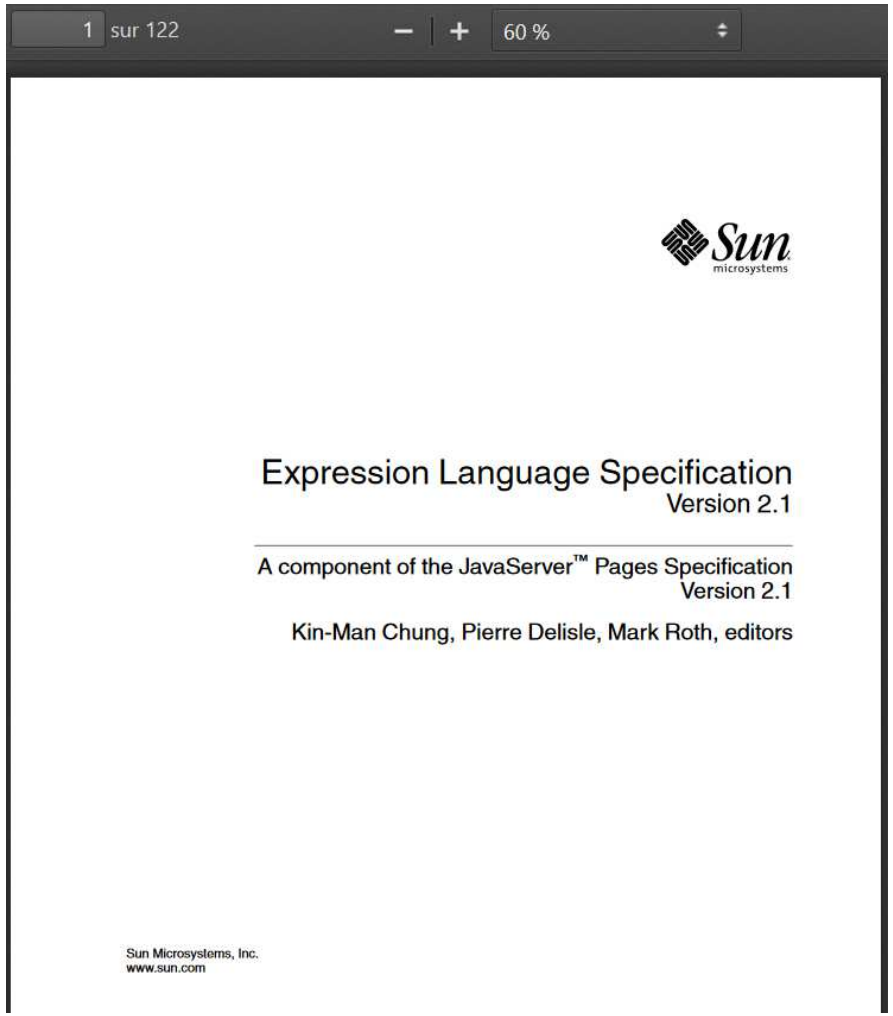
RAMZI FARHAT

- ❑ EL
- ❑ JSTL
- ❑ MVC
- ❑ Exercice

# Introduction EL

## EL (Expression Language)

- Langage permettant, au sein d'une page JSP, de :
  - *Évaluer des expressions*
  - *Manipuler des objets*
  - *Manipuler les attributs de requêtes*
- Syntaxe :
  - *Evaluation immédiate (compilation avec la page JSP) :*
    - ${expression}
  - *Evaluation différée (compilation au moment d'utilisation de l'expression :*
    - #{expression}

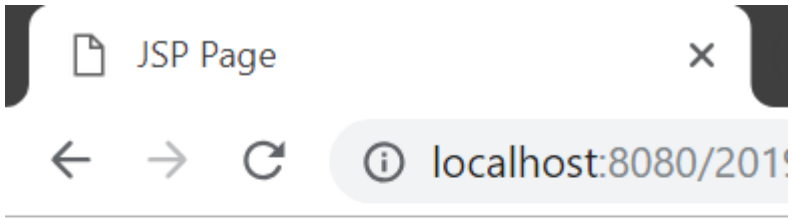



https://download.oracle.com/otn-pub/jcp/jsp-2.1-fr-spec-oth-JSpec/jsp-2_1-fr-spec-el.pdf

# Exemple

```jsp
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
    </head>
    <body>
        <ol>
            <li>${'Hello EL!'}</li>
            <li>${"Hello EL!"}</li>
        </ol>

    </body>
</html>
```



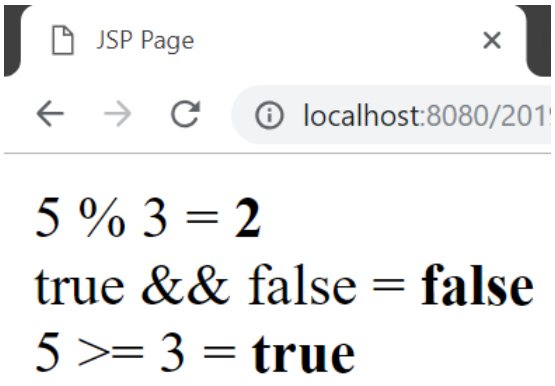

1. Hello EL!
2. Hello EL!

# Opérateurs

Les opérateurs suivants peuvent êtres utilisés pour définir une expression :

| | |
|---|---|
| Opérateurs arithmétiques | + - * / % Mod |
| Opérateurs logiques | && \|\| ! |
| Opérateurs relationnels | equals() compareTo() == eq ! ne < lt > gt <= le >= ge |

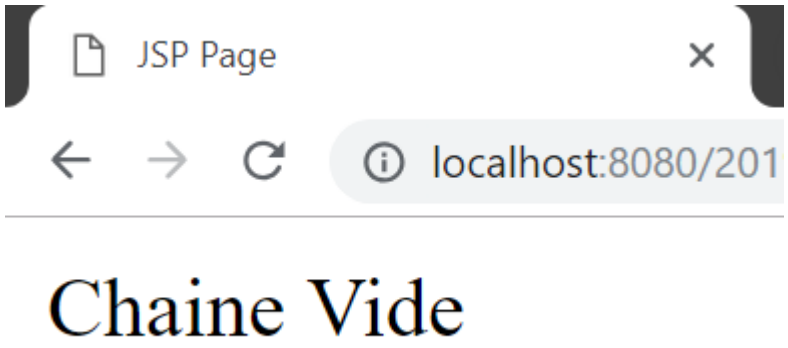
```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>JSP Page</title>
    </head>
    <body>
        <div>5 % 3 = <strong>${5 % 3}</strong></div>
        <div> true && false = <strong>${true && false}</strong></div>
        <div>5 >= 3 = <strong>${5 >= 3}</strong></div>
    </body>
</html>
```



# Tester des valeurs

| Test ternaire | Test ? valeurARetournerSiVrai: ValeurARetournerSiFaux |
|---|---|
| Vérifier qu'un objet est vide | empty |

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>JSP Page</title>
    </head>
    <body>
        <div>${empty ""?"Chaine Vide":"Chaine non vide"}</div>
    </body>
</html>
```

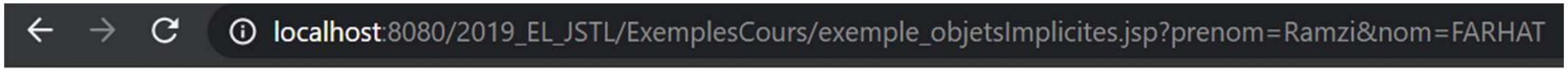


# Objets implicites

| | |
|---|---|
| pageContext | objet contenant des informations sur l'environnement du serveur. |
| pageScope | Map qui associe les noms et les valeurs des attributs de portée page |
| requestScope | Map qui associe les noms et les valeurs des attributs de portée requête |
| sessionScope | Map qui associe les noms et les valeurs des attributs de portée session |
| applicationScope | Map qui associe les noms et les valeurs des attributs de portée application |
| param | Map qui associe les noms et les valeurs des paramètres de la requête |
| paramValues | Map qui associe les noms et les multiples valeurs des paramètres de la requête sous forme de tableaux de String |
| header | Map qui associe les noms et les valeurs des paramètres des en-têtes HTTP |
| headerValues | Map qui associe les noms et les valeurs des paramètres des entêtes HTTP sous forme de tableaux de String |
| cookie | Map qui associe les noms et les instances des cookies |
| initParam | Map qui associe les données contenues dans les champs <param-name> et <param-value> de la section <init-param> du fichier web.xml |

# Exemple

```jsp
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
    </head>
    <body>
        <h1>Bienvenue ${param.prenom} ${param.nom}</h1>
    </body>
</html>
```

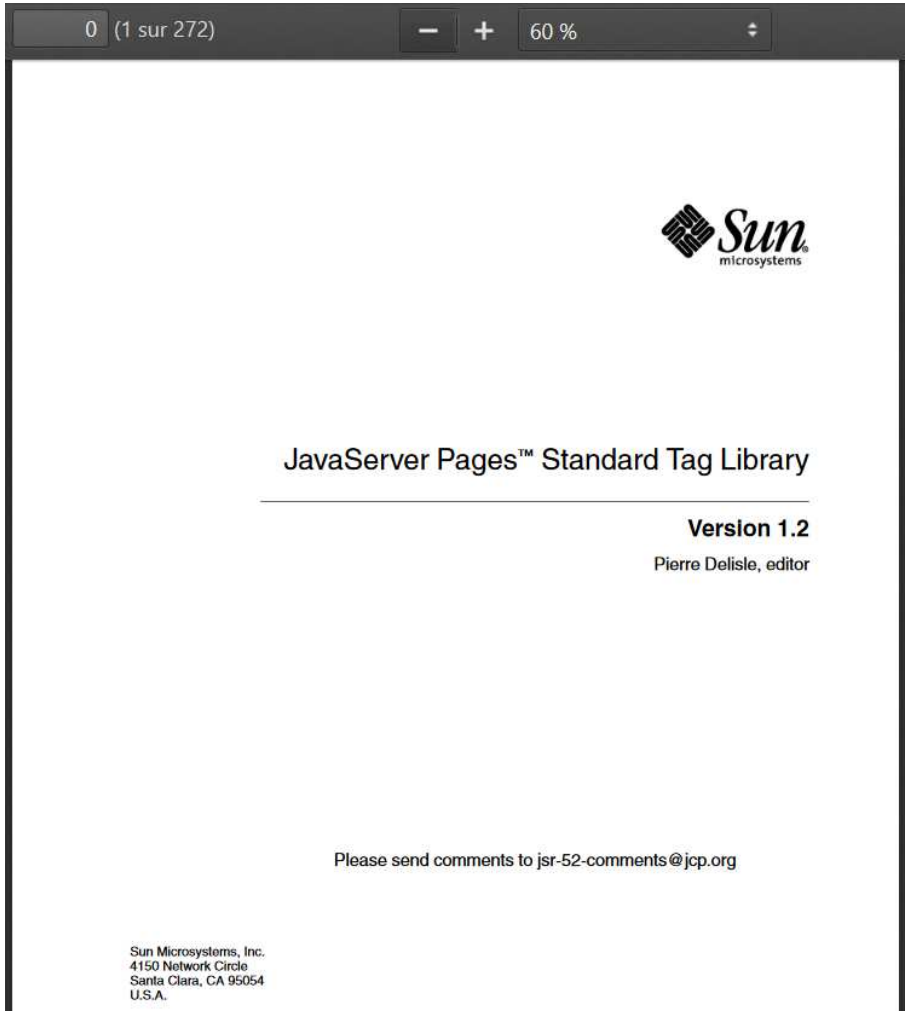



**Bienvenue Ramzi FARHAT**

# Introduction JSTL

## JSTL (JSP Standard Tag Library)

- Complément de l'EL permettant de développer des vues JSP sophistiqués.
- Bibliothèques :
  - *core : fonctionnalités de base*
  - *fmt : formatage et analyse des données*
  - *sql : interactions avec les BD*
  - *xml : traitement de données XML*




https://download.oracle.com/otn-pub/jcp/jstl-1.2-mrel2-oth-JSpec/jstl-1_2-mrel2-spec.pdf

# JSTL core

- Affichage
  - *Syntaxe*

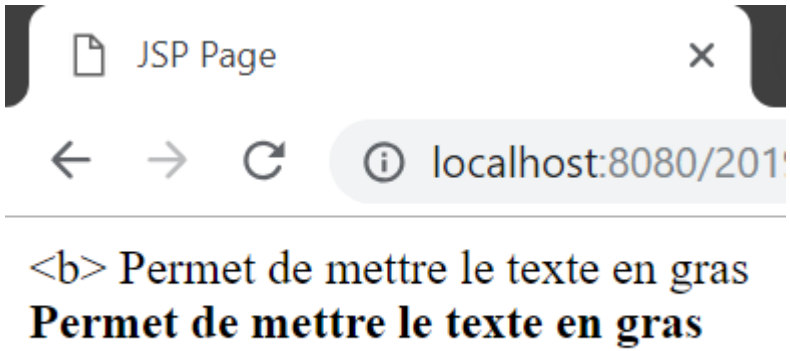
Chaîne de caractères ou EL

expression alternative à afficher si l'expression de value ne retourne rien

*<c:out value="<expression>" [default="<expression>"] [escapeXml="{false | true}]"/>*

"true" permet d'afficher des balises sans les interpréter par le navigateur (par défaut)

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
    </head>
    <body>
        <div>
        <c:out
            value="${beanInvalide}"
            default="<b> Permet de mettre le texte en gras"/>
        </div>
        <div>
        <c:out
            value="<b> Permet de mettre le texte en gras"
            escapeXml="false"/>
        </div>
    </body>
</html>
```



# JSTL core

- Gestion des variables
  - *Création d'une variable*

*page ou request ou session ou application*

*<c:set var="<nomVariable>" [value="<valeurVariable>"] [scope="<portéeVariable>"]/>*

  - *Suppression d'une variable*

*<c:remove var="<nomVariable>" [scope="<portéeVariable>"]/>*

```
<c:set var="msg" value="Mon message 1 (scope request)" scope="request"/>
<c:set var="msg" value="Mon message 2 (scope session)" scope="session"/>
<h3>msg :</h3> <c:out value="${msg}"/>
<h3>sessionScope.msg :</h3> <c:out value="${sessionScope.msg}"/>
<h3 style="color:red;">Suppression de la variable msg de la portée request</h3>
<c:remove var="msg" scope="request"/>
<h3>msg : </h3><c:out value="${msg}"/>
<h3>sessionScope.msg : </h3><c:out value="${sessionScope.msg}"/>
```

JSP Page

localhost:808

**msg :**

Mon message 1 (scope request)

**sessionScope.msg :**

Mon message 2 (scope session)

**Suppression de la variable msg de la portée request**

**msg :**

Mon message 2 (scope session)

**sessionScope.msg :**

Mon message 2 (scope session)

# JSTL core

- Gestion des JavaBeans
  - *Création d'une reference à un JavaBean*

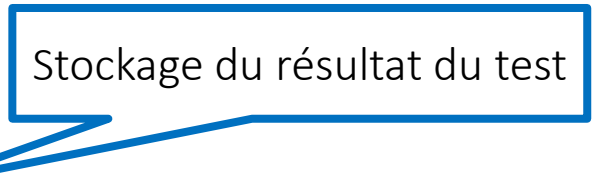
*<c:set var=”<nomReference>” value=”${<identifiantDuBean>}” [scope=”<portée>”]/>*

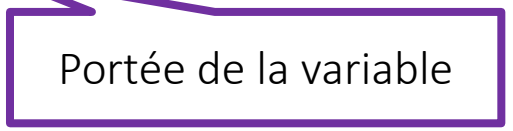
  - *Utilisation d'un JavaBean pour modifier la valeur d'une propriété*

*<c:set target="<nomReference>" property="<nomPropriété>" value="<valeur>"/>*

# JSTL core

- Conditionnel (1/2)
  - *Syntaxe*


```
<c:if test="<expression>" [var="<identifiant>" [scope="<portée>"]]>
   Contenu à afficher
</c:if>
```

  - *Exemple*

```
<c:if test=”${7>2}>
        Mon test fonctionne
</c:if>
```

# JSTL core

- Conditionnel (2/2)
  - *Syntaxe*

Si l'expression du test est évaluée à vrai alors exécuter et fin de ***choose***, sinon passer au suivant ***when***

```
<c:choose>
  <c:when test="<expression>">Action ou Affichage d'un texte</c:when>
 [ ...
  <c:otherwise>Autre action ou texte.</c:otherwise>]
</c:choose>
```

Exécution si aucune expression n'est évaluée à vrai

  - *Exemple*

```
<c:choose>
   <c:when test="${empty param['couleur']}"> couleur non spécifié</c:when>
   <c:when test="${empty param['prix']}"> prix non spécifié</c:when>
   <c:otherwise> "tous les paramètres sont initialisés</c:otherwise>
</c:choose>
```

# JSTL core

- Itératif (1/2)
  - *Syntaxe*

```
<c:forEach
    [var="<nomVariable>"]
    begin="<valeurDébut>"
    end="<valeurFin>"
    [step="<pasAvancement"]
    [varStatus="<nomVariableEtat>"]>
    ...
</c:forEach>
```

  - *Syntaxe pour parcourir une liste*

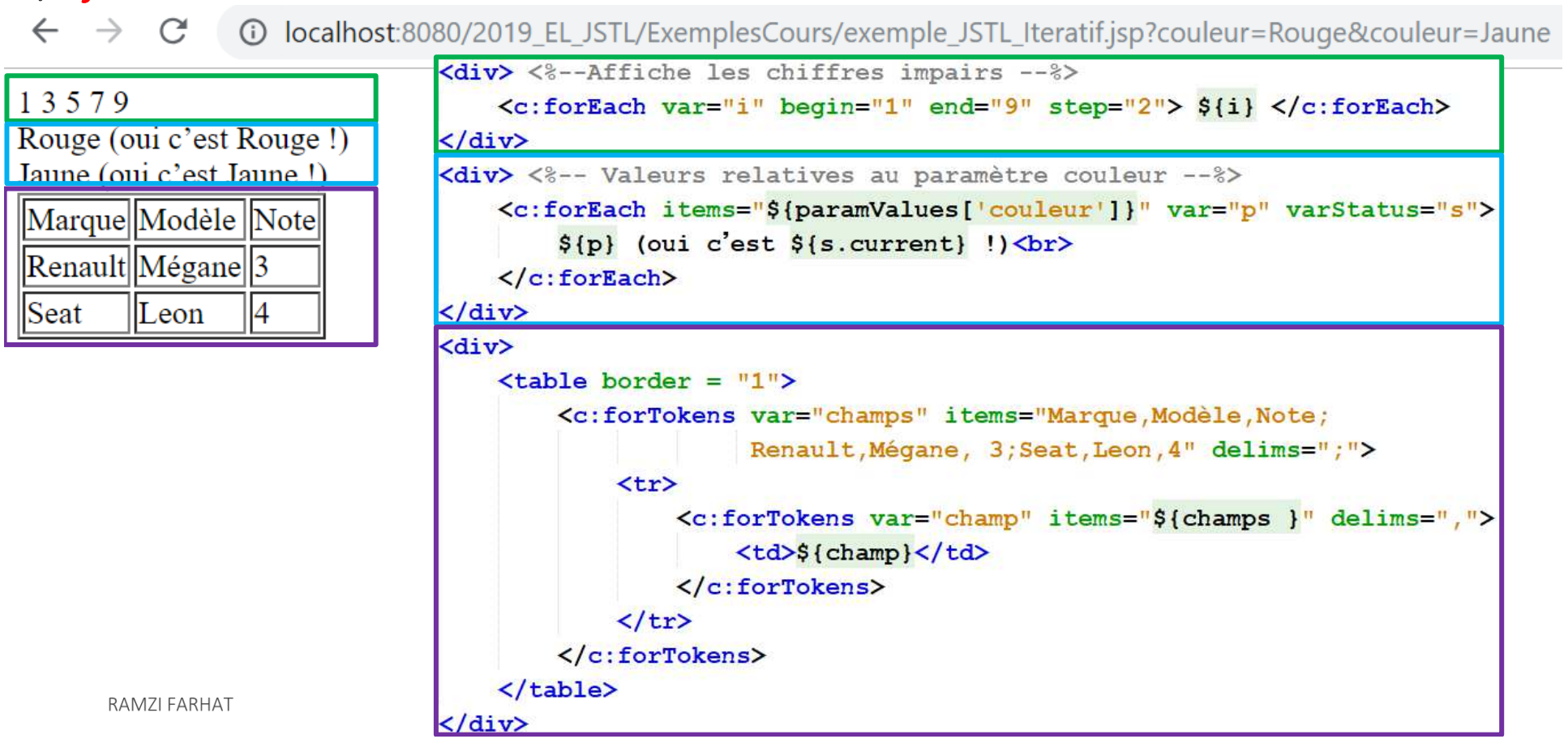
```
<c:forEach
    items="<une liste de valeurs>"
    [var="<nom de la variable>"]
    [begin="<valeur de début"]
    [end="valeur de fin"]
    [varStatus="<nom de la variable d'état>"]>
    ...
</c:forEach>
```

Propriétés de la variable d'état :
- **begin** : valeur de l'attribut begin
- **end** : valeur de l'attribut end
- **step** : valeur de l'attribut step
- **first** : retourne vrai si c'est la première itération
- **last** : retourne vrai si c'est la dernière itération
- **count** : retourne le numéro de l'itération en cours (démarre de 1)
- **index** : retourne l'index de l'itération (démarre de 0)
- **current** : retourne l'élément courant de la collection parcourue

# JSTL core

- Itératif (2/2)
  - *Syntaxe pour parcourir une chaîne de caractères*

*<c:forTokens var="<nomVariable>" items="<chaîneDeCaractères"*
*delims="<unOuPlusieursDélimiteurs"> ...*
*</c:forTokens>*



# JSTL core

- Gestion des URL (1/3)
  - *Ajouter un lien hypertexte*

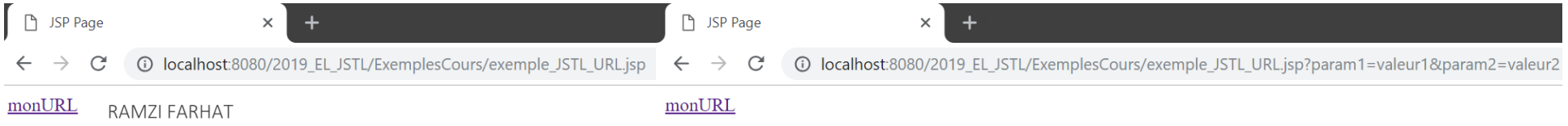
*<c:url value="<lien relatif ou absolue interne>" var="<variable contenant le lien>">*

*[<c:param name="<nom du paramètre>" value="<valeur du paramètre>"/>…]*

*</c:url>*

  - *Avantages*
    - *ajout automatique du contexte pour les URL absolues,*
    - *ajout automatique de l'identifiant de session*
    - *encodage des caractères spéciaux des paramètres*

```
<c:url value="exemple_JSTL_URL.jsp" var="monURL">
    <c:param name="param1" value="valeur1"/>
    <c:param name="param2" value="valeur2"/>
</c:url>
<a href="${monURL}">monURL</a>
```



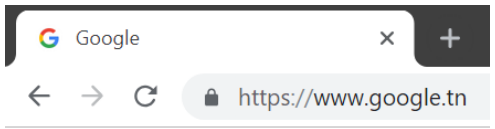
# JSTL core

- Gestion des URL (2/2)
  - *Redirection*

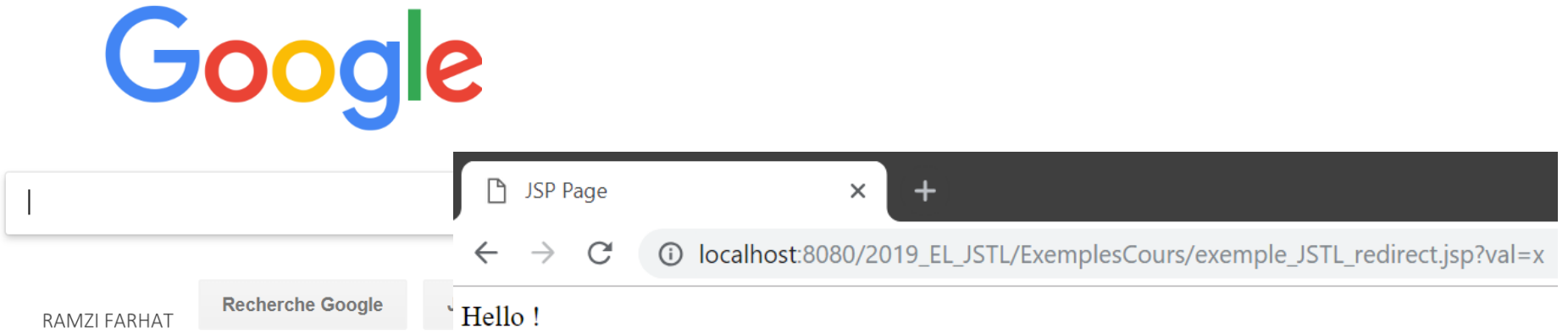
*<c:redirect url="<URL>">*

*[<c:param name="<nomParamètre>" value="<valeurParamètre>"/>...]*

*</c:redirect>*




```
<c:if test="${empty param.val}">
    <c:redirect url="http://www.google.tn"/>
</c:if>
Hello !
```



# JSTL core

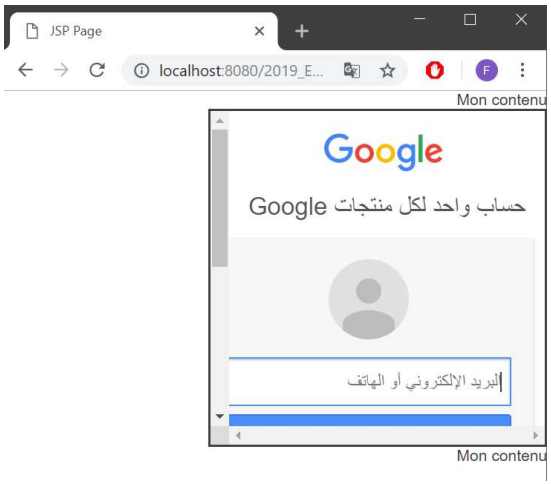
- Gestion des URL (3/3)
  - *Importation*

*<c:import url="<URL>" [var="<nomVariable>"] [scope="<portéVariable"] ...>*

*[<c:param .../>]**

*</c:import>*

```
<body>
    <c:import url="https://www.gmail.com" var="monURL"/>
    <div>Mon contenu</div>
    <div style=" overflow:scroll ;width:300px; height: 300px; border-style: solid;">
        ${monURL}
    </div>
    <div>Mon contenu</div>
</body>
```

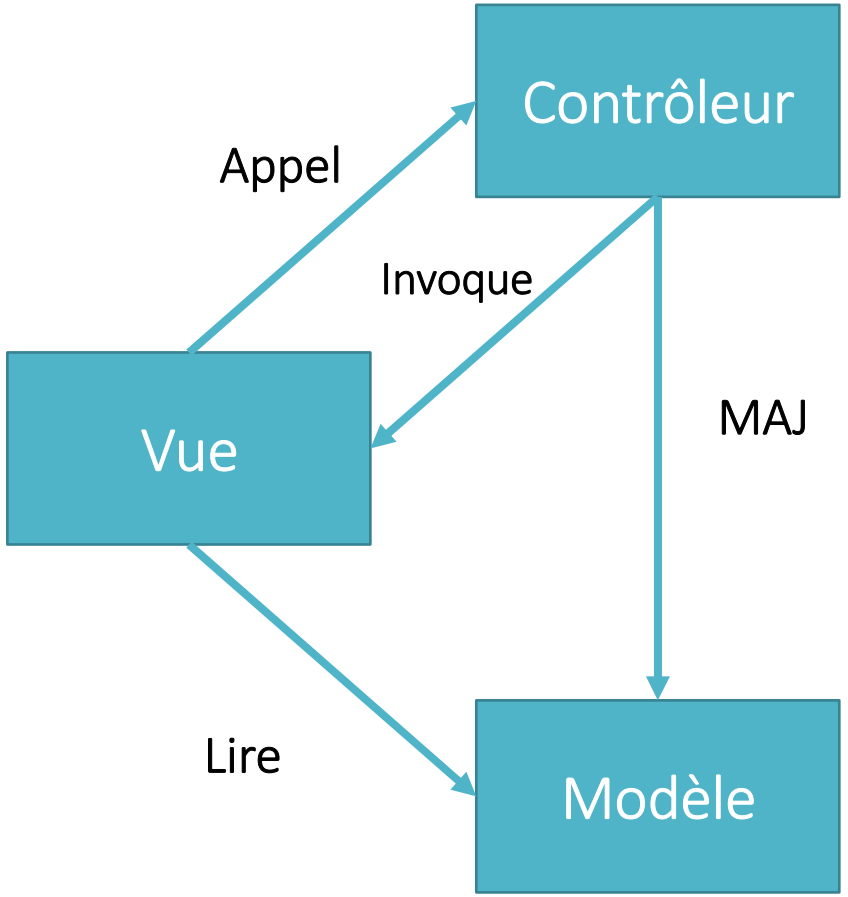


# Patron de conception

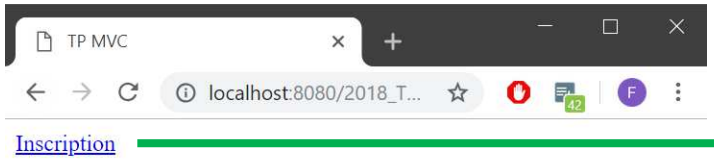
- Patron de conception
  - Design pattern en anglais
  - Bonnes pratiques pour un problème de conception donné
  - Types :
    - *Creational (Singleton, Factory, Constructor, etc.)*
      - Création d'objets
    - *Structural (Decorator, Façade, Flyweight, etc.)*
      - Relations entre objets
    - *Behavioral (Command, Mediator, Observer, etc.)*
      - Communication entre objets
    - *Architectural (MVC, MVVM, SOA, etc.)*
      - Structuration de l'application

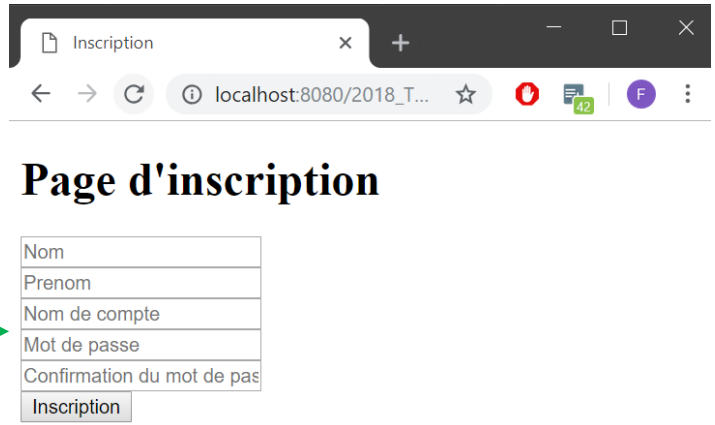
# Patron de conception MVC

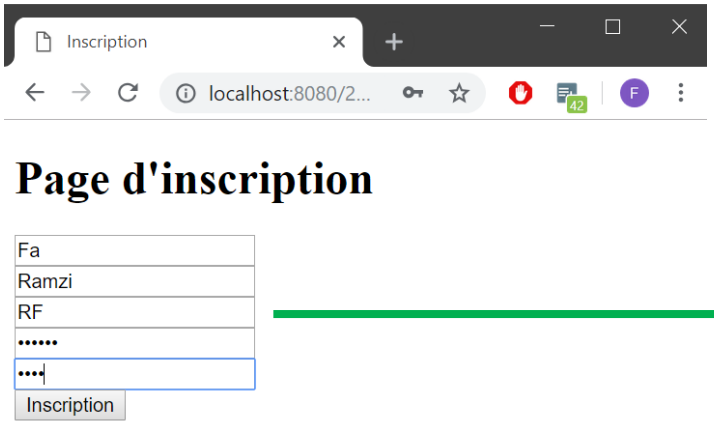
- Patron de conception MVC
  - M : Modèle
    - *Contient les données et les traitements*
    - *EJB, JavaBeans*
  - V : Vue
    - *Interface graphique*
    - *JSP, EL, JSTL*
  - C : Contrôleur
    - *Traite les actions de l'utilisateur pour mettre à jour le modèle et/ou de la vue*
    - *Servlet*
  - Types
    - *MVC : plusieurs contrôleurs*
    - *MVC 2 : un seul contrôleur*



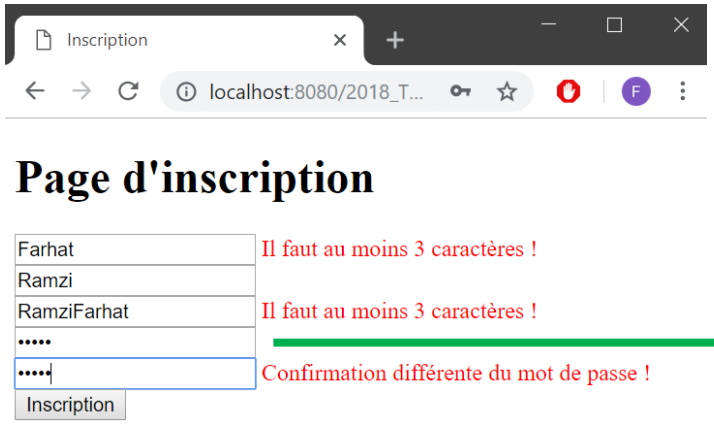
# Exercice


TP MVC
localhost:8080/2018_T...
Inscription


Inscription
localhost:8080/2018_T...
Page d'inscription
Nom
Prenom
Nom de compte
Mot de passe
Confirmation du mot de pas
Inscription


Inscription
localhost:8080/2...
Page d'inscription
Fa
Ramzi
RF
Inscription


Inscription
localhost:8080/2018_T...
Page d'inscription
Fa
Il faut au moins 3 caractères !
Ramzi
RF
Il faut au moins 3 caractères !
Mot de passe
Confirmation du mot de pas
Confirmation différente du mot de passe !
Inscription


Inscription
localhost:8080/2018_T...
Page d'inscription
Farhat
Il faut au moins 3 caractères !
Ramzi
RamziFarhat
Il faut au moins 3 caractères !
Confirmation différente du mot de passe !
Inscription


Inscription
localhost:8080/2018_T...
Page d'inscription
Vous êtes inscrit avec les informations suivantes :
Nom : Farhat
Prenom : Ramzi
Nom de compte : RamziFarhat

- EL
  - Introduction
  - Opérateurs
  - Tests
  - Objets implicites
- JSTL
  - Introduction
  - JSTL core : Affichage
  - JSTL core : Variables
  - JSTL core : JavaBeans
  - JSTL core : Conditionnel
  - JSTL core : Itératif
  - JSTL core : URL
- MVC
  - Patrons de conception
  - Patron de conception MVC
- Exercice