

# JSP

## PARTE 2

Vista (JSTL – Standard Tag Library – Core Tags)

Con Apache Tomcat  
(Servidor Web y JSP)





# JSP

# JSTL

## JSTL

### Etiquetas más sencillas de integrar dentro de la vista

- Es una colección de etiquetas JSP que permite el manejo del Core JSP de una manera más sencilla e integrada
- Instalación :
  - Descargar la librería desde <https://tomcat.apache.org/taglibs/standard/> (JSLT 1.2)
  - Copiar las librerías en el proyecto donde las quaramos usar (**webapps\WEB-INF\lib**)
- Funcionalidades disponibles (Sólo vamos a ver las Core Tags y las Formatting Tags):
  - Core Tags
  - Formatting tags
  - SQL tags
  - JSLT Funcions
- Para poder usarlas solo es necesario declarar la parte de la funcionalidad que queramos usar , en la cabecera de la página JSP

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



## JSP

## JSTL

## JSTL

### Core Tags

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

- **<c:out>** : muestra el resultado de evaluar una expresión, igual que hace <%=....%>

The <c:out> tag has the following attributes -

Attribute	Description	Required	Default
Value	Information to output	Yes	None
default	Fallback information to output	No	body
escapeXml	True if the tag should escape special XML characters	No	true

```
<c:out value="Esto es una prueba"/>
```

```
<c:out value="${param.pa}"/>
```



## JSP

## JSTL

## JSTL

### Core Tags

- **<c:set>** : establece el resultado de la evaluación de una expresión en una variable

The **<c:set>** tag has the following attributes –

Attribute	Description	Required	Default
Value	Information to save	No	body
target	Name of the variable whose property should be modified	No	None
property	Property to modify	No	None
var	Name of the variable to store information	No	None
scope	Scope of variable to store information	No	Page

```
<c:set var="Income" scope="session" value="{4000*4}"/>
<c:out value="{Income}"/>
```



## JSP

## JSTL

## JSTL

### Core Tags

- **<c:remove>** : elimina una variable del scope indicado

The **<c:remove>** tag has the following attributes –

Attribute	Description	Required	Default
var	Name of the variable to remove	Yes	None
scope	Scope of the variable to remove	No	All scopes

```
<c:set var="income" scope="session" value="${4000*4}"/>
<p>Before Remove Value is: <c:out value="${income}"/></p>
<c:remove var="income"/>
```



JSP

JSTL

## JSTL

### Core Tags

- **<c:catch>** : captura una excepción dentro del código incluido dentro del elemento

The **<c:catch>** tag has the following attributes –

Attribute	Description	Required	Default
var	The name of the variable to hold the java.lang.Throwable if thrown by elements in the body.	No	None

```
<c:catch var="catchtheException">
  <% int x = 2/0;%>
</c:catch>

<c:if test = "${catchtheException != null}">
  <p>The type of exception is : ${catchtheException} <br />
  There is an exception: ${catchtheException.message}</p>
</c:if>
```



## JSP

## JSTL

## JSTL

### Core Tags

- **<c:if>** : sentencia condicional if, evalua lo incluido en el atributo **test**

The **<c:if>** tag has the following attributes –

Attribute	Description	Required	Default
test	Condition to evaluate	Yes	None
var	Name of the variable to store the condition's result	No	None
scope	Scope of the variable to store the condition's result	No	page

```
<c:set var="income" scope="session" value="${4000*4}"/>
<c:if test="${income > 8000}">
  <p>My income is: <c:out value="${income}"/><p>
</c:if>
```



JSP

JSTL

## JSTL

### Core Tags

- `<c:choose>`, `<c:when>`, `<c:otherwise>` : sentencia condicional entre varias opciones (switch)

```
<c:choose>
  <c:when test="${income <= 1000}">
    Income is not good.
  </c:when>
  <c:when test="${income > 10000}">
    Income is very good.
  </c:when>
  <c:otherwise>
    Income is undetermined...
  </c:otherwise>
</c:choose>
```





# JSP

# JSTL

## JSTL

### Core Tags

- **<c:import>** : inserta el contenido de una página en otra

The **<c:import>** tag has the following attributes –

Attribute	Description	Required	Default
url	URL to retrieve and import into the page	Yes	None
context	/ followed by the name of a local web application	No	Current application
charEncoding	Character set to use for imported data	No	ISO-8859-1
var	Name of the variable to store imported text	No	Print to page
scope	Scope of the variable used to store imported text	No	Page
varReader	Name of an alternate variable to expose java.io.Reader	No	None

```
<c:import var="data" url="http://www.javatpoint.com"/>  
<c:out value="${data}"/>
```



## JSP

## JSTL

## JSTL

### Core Tags

- **<c:forEach>** : iteración sobre colecciones y conjuntos

The **<c:forEach>** tag has the following attributes –

Attribute	Description	Required	Default
items	Information to loop over	No	None
begin	Element to start with (0 = first item, 1 = second item, ...)	No	0
end	Element to end with (0 = first item, 1 = second item, ...)	No	Last element
step	Process every step items	No	1
var	Name of the variable to expose the current item	No	None
varStatus	Name of the variable to expose the loop status	No	None

```
<c:forEach var="j" begin="1" end="3">  
  Item <c:out value="{j}" /><p>  
</c:forEach>
```

```
<c:forEach var="lang" items="{paramValues.langChoice}">  
  <font color="#00CC00"><c:out value="{lang}" />,</font>  
</c:forEach>
```



JSP

JSTL

## JSTL

### Core Tags

- **<c:forTokens>** :Iteración sobre tokens según el delimitador indicado

The **<c:forTokens>** tag has similar attributes as that of the **<c:forEach>** tag except one additional attribute **delims** which specifies characters to use as delimiters.

Attribute	Description	Required	Default
delims	Characters to use as delimiters	Yes	None

```
<c:forTokens items="Rahul-Nakul-Rajesh" delims="-" var="name">
  <c:out value="${name}"/><p>
</c:forTokens>
```



JSP

JSTL

## JSTL

### Core Tags

- **<c:url>** : crea un enlace a una URL con los parámetros indicados

The **<c:url>** tag has the following attributes –

Attribute	Description	Required	Default
Value	Base URL	Yes	None
context	/ followed by the name of a local web application	No	Current application
var	Name of the variable to expose the processed URL	No	Print to page
scope	Scope of the variable to expose the processed URL	No	Page

```
<a href = "<c:url value = "/jsp/index.htm"/>">TEST</a>
```



JSP

JSTL

## JSTL

### Core Tags

- **<c:param>** : añade un parámetro a un elemento **url**

The **<c:param>** tag has the following attributes –

Attribute	Description	Required	Default
name	Name of the request parameter to set in the URL	Yes	None
value	Value of the request parameter to set in the URL	No	Body

```
<c:url value="/index1.jsp" var="completeURL"/>  
<c:param name="trackingId" value="786"/>  
<c:param name="user" value="Nakul"/>  
</c:url>
```



## JSP

## JSTL

## JSTL

### Core Tags

- **<c:redirect>** : redirige a la URL pasada como parámetro

The **<c:redirect>** tag has the following attributes –

Attribute	Description	Required	Default
url	URL to redirect the user's browser to	Yes	None
context	/ followed by the name of a local web application	No	Current application

```
<c:set var="url" value="0" scope="request" />
<c:if test="${url<1}">
  <c:redirect url="http://javatpoint.com"/>
</c:if>
<c:if test="${url>1}">
  <c:redirect url="http://facebook.com"/>
</c:if>
```



## JSP

## JSTL

## JSTL

### Formatting Tags

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

- **<fmt:formatNumber>** : usada para formater números, porcentajes y valores monetarios.

The **<fmt:formatNumber>** tag has the following attributes –

Attribute	Description	Required	Default
Value	Numeric value to display	Yes	None
type	NUMBER, CURRENCY, or PERCENT	No	Number
pattern	Specify a custom formatting pattern for the output.	No	None
currencyCode	Currency code (for type = "currency")	No	From the default locale
currencySymbol	Currency symbol (for type = "currency")	No	From the default locale
groupingUsed	Whether to group numbers (TRUE or FALSE)	No	true
maxIntegerDigits	Maximum number of integer digits to print	No	None
minIntegerDigits	Minimum number of integer digits to print	No	None
maxFractionDigits	Maximum number of fractional digits to print	No	None
minFractionDigits	Minimum number of fractional digits to print	No	None
var	Name of the variable to store the formatted number	No	Print to page
scope	Scope of the variable to store the formatted number	No	page

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

<html>
<head>
<title>fmt:parseNumber tag</title>
</head>
<body>
<h3>The fmt:parseNumber tag Example is:</h3>

<c:set var="Amount" value="786.970" />

<fmt:parseNumber var="j" type="number" value="${Amount}" />
<p><i>Amount is:</i> <c:out value="${j}" /></p>

<fmt:parseNumber var="j" integerOnly="true" type="number" value="${Amount}" />
<p><i>Amount is:</i> <c:out value="${j}" /></p>
</body>
</html>
```



## JSP

## JSTL

## JSTL

### Formatting Tags

- **<fmt:parseNumber>** : usada para parsear números, porcentajes y valores monetarios.

The **<fmt:parseNumber>** tag has the following attributes –

Attribute	Description	Required	Default
Value	Numeric value to read (parse)	No	Body
type	NUMBER, CURRENCY, or PERCENT	No	number
parseLocale	Locale to use when parsing the number	No	Default locale
integerOnly	Whether to parse to an integer (true) or floating-point number (false)	No	false
pattern	Custom parsing pattern	No	None
timeZone	Time zone of the displayed date	No	Default time zone
var	Name of the variable to store the parsed number	No	Print to page
scope	Scope of the variable to store the formatted number	No	page

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

<html>
<head>
<title>fmt:parseNumber tag</title>
</head>
<body>
<h3>The fmt:parseNumber tag Example is:</h3>

<c:set var="Amount" value="786.970" />

<fmt:parseNumber var="j" type="number" value="${Amount}" />
<p><i>Amount is:</i> <c:out value="${j}" /></p>

<fmt:parseNumber var="j" integerOnly="true" type="number" value="${Amount}" />
<p><i>Amount is:</i> <c:out value="${j}" /></p>
</body>
</html>
```





JSP

JSTL

## JSTL

### Formatting Tags

- **<fmt:formatDate>** : usada para formatear fechas de varias formas

The **<fmt:formatDate>** tag has the following attributes –

Attribute	Description	Required	Default
Value	Date value to display	Yes	None
type	DATE, TIME, or BOTH	No	date
dateStyle	FULL, LONG, MEDIUM, SHORT, or DEFAULT	No	default
timeStyle	FULL, LONG, MEDIUM, SHORT, or DEFAULT	No	default
pattern	Custom formatting pattern	No	None
timeZone	Time zone of the displayed date	No	Default time zone
var	Name of the variable to store the formatted date	No	Print to page
scope	Scope of the variable to store the formatted date	No	page

The pattern attribute is used to specify even more precise handling of the date –

Code	Purpose	Sample
G	The era designator	AD
y	The year	2002
M	The month	April & 04
d	The day of the month	20
h	The hour(12-hour time)	12
H	The hour(24-hour time)	0
m	The minute	45
s	The second	52
S	The millisecond	970
E	The day of the week	Tuesday
D	The day of the year	180
F	The day of the week in the month	2 (2nd Wed in month)
w	The week in the year	27
W	The week in the month	2
a	The a.m./p.m. indicator	PM
k	The hour(12-hour time)	24
K	The hour(24-hour time)	0
z	The time zone	Central Standard Time
'		The escape for text
"		The single quote



# JSP

# JSTL

## JSTL

### Formatting Tags

- **<fmt:formatDate>** : usada para formatear fechas de varias formas

```
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt"%>
<html>
<head>
<title><fmt:formatDate</title>
</head>
<body>
<h2>Different Formats of the Date</h2>
<c:set var="Date" value="1" />
<p>
Formatted Time :
<fmt:formatDate type="time" value="${Date}" />
</p>
<p>
Formatted Date :
<fmt:formatDate type="date" value="${Date}" />
</p>
<p>
Formatted Date and Time :
<fmt:formatDate type="both" value="${Date}" />
</p>
<p>
Formatted Date and Time in short style :
<fmt:formatDate type="both" dateStyle="short" timeStyle="short"
value="${Date}" />
</p>
<p>
Formatted Date and Time in medium style :
<fmt:formatDate type="both" dateStyle="medium" timeStyle="medium"
value="${Date}" />
</p>
<p>
Formatted Date and Time in long style :
<fmt:formatDate type="both" dateStyle="long" timeStyle="long"
value="${Date}" />
</p>
</body>
</html>
```



GOBIERNO  
DE ESPAÑA

MINISTERIO  
DE INDUSTRIA, COMERCIO  
Y TURISMO

EQI

Escuela de  
organización  
industrial



Unión Europea  
Fondo Social Europeo  
Iniciativa de Empleo Juvenil  
El FSE invierte en tu futuro





## JSP

## JSTL

## JSTL

### Formatting Tags

- **<fmt:parseDate>** : usada para parsear fechas

The **<fmt:parseDate>** tag has the following attributes –

Attribute	Description	Required	Default
Value	Date value to read (parse)	No	Body
type	DATE, TIME, or BOTH	No	date
dateStyle	FULL, LONG, MEDIUM, SHORT, or DEFAULT	No	Default
timeStyle	FULL, LONG, MEDIUM, SHORT, or DEFAULT	No	Default
parseLocale	Locale to use when parsing the date	No	Default locale
pattern	Custom parsing pattern	No	None
timeZone	Time zone of the parsed date	No	Default time zone
var	Name of the variable to store the parsed date	No	Print to page
scope	Scope of the variable to store the formatted date	No	page

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

<html>
<head>
  <title>fmt:parseDate Tag</title>
</head>
<body>
  <h3>Parsed Date:</h3>
  <c:set var="date" value="12-08-2016" />

  <fmt:parseDate value="${date}" var="parsedDate" pattern="dd-MM-yyyy" />
  <p><c:out value="${parsedDate}" /></p>

</body>
</html>
```



JSP

JSTL

## JSTL

### Formatting Tags

- `<fmt:setBundle>`, `<fmt:message>` : usada para los idiomas (Internacionalización)
- Primeramente crearemos los siguientes ficheros de idioma en el paquete **com.beca.idiomas**

✓ com.beca.idiomas

- idiomas\_en.properties
- idiomas\_es.properties
- idiomas.properties

- Cada fichero tendrá una lista de **claves=valor** donde **clave** será la etiqueta que representa a una palabra y **valor** será el valor de esa palabra en el idioma correspondiente

idiomas.properties

```
1 hola=HOLAS
2 adios=ADIOS
3 hastaLuego=HASTA LUEGO
```

idiomas\_es.properties

```
1 hola=HOLA
2 adios=ADIOS
```

idiomas\_en.properties

```
1 hola=HELLO
2 adios=BYE
```



GOBIERNO  
DE ESPAÑA

MINISTERIO  
DE INDUSTRIA, COMERCIO  
Y TURISMO



Escuela de  
organización  
industrial



Unión Europea  
Fondo Social Europeo  
Iniciativa de Empleo Juvenil  
El FSE invierte en tu futuro





JSP

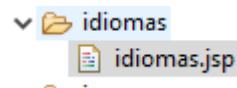
JSTL

## JSTL

### Formatting Tags

- `<fmt:setBundle>`, `<fmt:message>` : usada para los idiomas (Internacionalización)
- El fichero **idiomas.properties** contendrá los valores por defecto, es decir, si no se selecciona ningún idioma o no encuentra alguna etiqueta en algún idioma, mostrará lo que haya en este fichero

- Crearemos una página JSP nueva





## JSP

## JSTL

## JSTL

### Formatting Tags

- **<fmt:setBundle>**, **<fmt:message>** : usada para los idiomas (Internacionalización)

```
idiomas.jsp
1 <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
2   pageEncoding="ISO-8859-1"%>
3   <%@ taglib uri="http://java.sun.com/jsp/jstl/fmt" prefix="fmt" %>
4   <!DOCTYPE html>
5   <html>
6   <head>
7     <meta charset="ISO-8859-1">
8     <title>Insert title here</title>
9   </head>
10  <body>
11
12   <!-- Establecemos los nombres de los ficheros de idiomas -->
13   <fmt:setBundle basename="com.beca.idiomas.idiomas" />
14
15   <!-- Mostramos la clave hola -->
16   <p><fmt:message key="hola"/></p>
17
18   <!-- Seleccionamos el idioma en (idiomas_en.properties) -->
19   <fmt:setLocale value="en"/>
20
21
22   <!-- Mostramos la clave hola -->
23   <p><fmt:message key="adios"/></p>
24 </body>
25 </html>
```

## JSTL

### Formatting Tags

- `<fmt:timeZone>` : especifica la time zone que se va a usar

The `<fmt:timeZone>` tag has the following attributes –

Attribute	Description	Required	Default
Value	Time zone to apply to the body	Yes	None

```
<c:set var="str" value="%<new java.util.Date()%>" />
<table border="2" width="100%">
  <tr>
    <td width="100%" colspan="2" bgcolor="#FF7F50">
      <p align="center">
        <b>
          <font color="#000000" size="6">Formatting:
          <fmt:formatDate value="${str}" type="both"
            timeStyle="long" dateStyle="long" />
          </font>
        </b>
      </p>
    </td>
  </tr>
</table>
```

```
<c:forEach var="zone"
  items="%<java.util.TimeZone.getAvailableIDs()%>">
  <tr>
    <td width="50%" bgcolor="#C0C0C0">
      <c:out value="${zone}" />
    </td>
    <td width="50%" bgcolor="#FFEB3D">
      <fmt:timeZone value="${zone}">
        <fmt:formatDate value="${str}" timeZone="${zn}"
          type="both"/>
      </fmt:timeZone>
    </td>
  </tr>
</c:forEach>
</table>
```



JSP

JSTL

## JSTL

### Formatting Tags

- **<fmt:requestEncoding>** : especifica codificación usada en los formularios para realizar el envío de las peticiones

The **<fmt:requestEncoding>** tag has the following attributes –

Attribute	Description	Required	Default
key	Name of character encoding you want to apply when decoding request parameters.	Yes	None

```
<fmt:requestEncoding value = "UTF-8" />
```





## JSP

## Ejercicios

## JSTL

### Core Tags / Formatting Tags

- Crear un Bean llamado **Cliente** con los atributos (dni, nombre, apellidos, teléfono)
- Implementar un formulario HTML (**cliente.html**) que nos permita introducir los atributos del Bean **Cliente**
- Configurar la acción del formulario anterior para que lo envíe a la página JSP (**cliente.jsp**)
- Cuando la página JSP reciba los parámetros, que los vuelva a insertar en una nueva instancia de **Cliente** y los visualice
- Configurar **cliente.jsp** para que en función de un parámetro que le llegue llamado **locale**, establezca el idioma según el valor indicado en el parámetro recibido (locale=es, locale=en,...)
- Declarar en los ficheros de idiomas las etiquetas del formulario y probar que funciona correctamente



## JSP

### Ejercicios

## JSTL

### Core Tags / Formatting Tags

- Crear una página HTMML llamada **edad.html** que tenga un formulario para recoger un campo edad
- Configurar la acción del formulario a **edad.jsp**
- En **edad.jsp** recibir el parámetro edad y en función del valor recibido mostrar un mensaje por pantalla

edad < 18 : **MENOR DE EDAD**

18 <= edad <= 65 : **ADULTO EN EDAD DE TRABAJAR**

65 <= edad : **JUBILADO**

- Usar la internacionalización como en el caso anterior



## JSP

## Ejercicios

## JSTL

### Core Tags / Formatting Tags

- Crear un fichero HTML llamado **espia.html** con un formulario que permita introducir un mensaje cifrado
- Cifrar el mensaje, sustituyendo cada espacio por el carácter #
- Configurar el formulario para que envíe una petición a la página **espia.jsp**
- En la página **espia.jsp** descifrar el mensaje y mostrarlo (sin los #)



JSP

JSTL

## JSTL

### Formatting tags

- Consultar :

- <https://docs.oracle.com/javaee/5/tutorial/doc/bnakw.html#bnalc>
- [https://www.tutorialspoint.com/jsp/jsp\\_standard\\_tag\\_library.htm](https://www.tutorialspoint.com/jsp/jsp_standard_tag_library.htm)



JSP

JSTL

JSTL

SQL tags

- Consultar :

- <https://docs.oracle.com/javaee/5/tutorial/doc/bnald.html>
- [https://www.tutorialspoint.com/jsp/jsp\\_standard\\_tag\\_library.htm](https://www.tutorialspoint.com/jsp/jsp_standard_tag_library.htm)



JSP

JSTL

JSTL

XML tags

- Consultar :

- <https://docs.oracle.com/javaee/5/tutorial/doc/bnakq.html>
- [https://www.tutorialspoint.com/jsp/jsp\\_standard\\_tag\\_library.htm](https://www.tutorialspoint.com/jsp/jsp_standard_tag_library.htm)