

# **EL y JSTL**

## **- Tag Libraries-**

# ÍNDICE DE CONTENIDOS

- **Situación**
- **EL (Expression Language)**
- **Situación**
- **JSTL**
- **Referencia**
- **Ejemplo**

## Situación

“¿Qué pasa si los campos del Bean no son tipos “comunes” y tenemos que acceder a nuevos campos dentro de ellos.”

**NO NOS SIRVEN LOS BEANS!!!!!!**



## SINTAXIS GENERAL EL

 $\${firstPart.secondPart}$ 

Objetos implícitos

Atributos de  
page/request/session /  
application scopeAtributos de la  
primera parte*Ej:  $\${tren.maquinista.nombre}$*

## OPERADORES EL

### Arithmetic (5)

Addition:	+
Subtraction:	-
Multiplication:	*
Division:	/ and div
Remainder:	% and mod

### Logical (3)

AND:	&& and and
OR:	and or
NOT:	! and not

### Relational (6)

Equals:	== and eq
Not equals:	!= and ne
Less than:	< and lt
Greater than:	> and gt
Less than or equal to:	<= and le
Greater than or equal to:	>= and ge

## OBJETOS IMPLÍCITOS

- pageScope
- requestScope
- sessionScope
- applicationScope

- param
- paramValues

- header
- headerValues

- cookie
- initParam
- pageContext

MAPAS

## SINTAXIS GENERAL EL

**`${ firstPart[“secondPart”]}`**

***Ej: `${tren[“maquinista”][“nombre”]}`***



**Parece más trabajo pero, ¿qué pasa si accedo a Array, List, Map etc....**

## SINTAXIS GENERAL EL

```
String[] nombres = {"Pepe", "Manuel", "Lola"};  
request.setAttribute("nombres", nombres);
```

```
${nombres[0]}  
${nombres["0"]}
```

```
ArrayList<Vagon> vagones = new ArrayList<>();  
request.setAttribute("vagones", vagones);
```

```
${vagones[0].longitud}  
${vagones["0"]["longitud"]}
```

```
Map<String,String> mapa = new HashMap<String,String>;  
mapa.put("juan", "2222222X");  
request.setAttribute("personas", mapa);
```

```
${personas["juan"]}
```

## Situación

**“ Pese a EL seguimos teniendo código Java en nuestra vista (bucles, condicionales, recorrido de estructuras de datos...) ”**



“ Para minimizar el empleo de ese código Java en nuestra vista vamos a utilizar la **JSTL** (Java Server Pages **Standar** Tag Library). ”

## INSTALACIÓN DE JSTL

1. Instalar dependencias en el pom.xml (si proyectos Maven)
2. Añadir las librerías a los archivos jsp

## DEPENDENCIAS MAVEN

```
<dependency>  
  <groupId>org.glassfish.web</groupId>  
  <artifactId>jakarta.servlet.jsp.jstl</artifactId>  
  <version>3.0.1</version>  
</dependency>
```

```
<dependency>  
  <groupId>jakarta.servlet.jsp.jstl</groupId>  
  <artifactId>jakarta.servlet.jsp.jstl-api</artifactId>  
  <version>2.0.0</version>  
</dependency>
```

## DEPENDENCIAS MAVEN

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

**No vamos a utilizar todos**

## ETIQUETAS DEL CORE

### Propósito general

- `<c:out>`
- `<c:set>`

### Control de Flujo

- `<c:if>`
- `<c:choose>`
- `<c:foreach>`
- `<c:forTokens>`

### URLs

- `<c:import>`
- `<c:param>`

## ETIQUETAS SQL



¿Lo necesito?

- `<sql:setDataSource>`
- `<sql:query>`
- `<sql:update>` (insert/update/delete)
- `<sql:transaction>`
- `<sql:param>`

## ETIQUETAS `fmt`

- `<fmt:formatNumber>`
- `<fmt:formatDate>`
- `<fmt:setBundle>`
- `<fmt:message>`

# REFERENCIA

<https://docs.oracle.com/javaee/7/tutorial/jsf-el.htm>

<https://docs.oracle.com/javaee/5/jstl/1.1/docs/tlddocs/>





# EJEMPLOS



**END**



**jgardur081@g.educaand.es**