JSP Expression Languaje (EL)

- Expression Language (EL) simplifica el despliegue de los datos almacenados en los componentes JavaBeans en los JSPs.
- En una aplicación MVC son utilizados para desplegar los atributos compartidos por los Servlets en las páginas JSP sin importar el alcance de los atributos (request, session, application).
- Con EL no es posible modificar los atributos de los Javabeans (métodos set) únicamente se pueden leer sus atributos (métodos get).
- Con EL no es posible iterar objetos como Arrays, Lists, etc. Para esto, es necesario combinar EL con JSTL (en la lección de JSTL se verá un ejemplo).
- Para utilizarse no es necesario hacer ninguna declaración especial en los JSP. El único requisito es que sea un archivo JSP válido (ej. myJSP.jsp).
- Se pueden utilizar objetos implícitos (objetos que no necesitan declararse con New), operadores, y palabras reservadas en EL.
- EL fue agregada a partir de la tecnología JSP 2.0.

Sintáxis para EL en un JSP

${ expression }

${ javabean.property}

Ejemplo de EL

- Supongamos que tenemos una clase llamada Producto (Javabean) con sus métodos set y get para cada atributo privado.
- Después en un Servlet se crea un objeto (Javabean) de tipo Producto (aquí se esta generando en código duro, aunque bien podría ser creado con información de una bd). Posteriormente, el Servlet comparte dicho objeto en el alcance de request y hace una redirección a la vista (detalle.jsp).

Código del Servlet

```
Producto prod = new Producto();
prod.setDescripcion = "Televisión Sony";
prod.setPrecioVenta(100.5);
request.setAttribute("producto",prod);
RequestDispatcher rd = request.getRequestDispatcher("detalle.jsp");
rd.forward(request,response);
```

- Después de que el Servlet hace la redirección a la vista ("detalle.jsp"), en este JSP, estará disponible el JavaBean con el nombre de "producto" para poder ser mostrado con EL.

Código del archive detalle.jsp

```
<html>
<body>
  <h1>Detalle del producto</h1>
  <h2>Descripcion:  ${producto.descripcion}
  <h2>Precio:  ${producto.precioVenta}
  <h2>Precio:  ${producto.activo}
</body>
</html>
```

EL manda llamar los métodos de forma interna:
getDescripcion();
getPrecioVenta();
isActivo();

Nota: Podemos observar que para usar EL en un JSP no se necesita declarar nada en especial en el archivo JSP. Solo utilizar la sintaxis ${ javabean.attributo} o ${ expression }

Ejemplo – Sintáxis Expression Language

- El archivo el-ejemplo1-src.zip contiene un ejemplo completo de la diapositiva anterior.
- Para ver el ejemplo solo descomprime el archivo y haz clic en el botón “Open Project” en Netbeans y después en “Run”.

JSP Operadores en Expression Languaje.

Expression Language (EL) como su nombre lo índica sirve evaluar expresiones en los JSPs. Esas expresiones son evaluadas al momento de ser ejecutado el archivo JSP en el servidor, y posteriormente el resultado de dicha expresión, es reemplazado en el lugar donde fue colocada la expresión. Finalmente ese resultado es el enviado al cliente, junto con el HTML del JSP.

- Con EL no solo podemos desplegar los Javabeans. También podemos utilizar una serie de operadores, en combinación con los Javabeans para poder crear contenido más dinámico.
- La siguiente tabla muestra algunos de los operadores (aritméticos y lógicos) más utilizados en EL.

Operadores en Expression Language (EL)

Operator	Description
.	Access a bean property or Map entry
[]	Access an array or List element
( )	Group a sub-expression to change the evaluation order
+	Addition
-	Subtraction or negation of a value
*	Multiplication
/ or div	Division
%	Modulo (remainder)
== or eq	Test for equality
!= or ne	Test for inequality
< or lt	Test for less than
> or gt	Test for greater than
<= or le	Test for less than or equal
>= or gt	Test for greater than or equal

Ejemplo – Operadores en EL

- El archivo el-operadores-src.zip contiene un ejemplo completo de manejo de operadores.
- Para ver el ejemplo solo descomprime el archivo y haz clic en el botón “Open Project” en Netbeans y después en “Run”.

Objetos implícitos en los JSPs

- Hay una serie de objetos implícitos en los JSPs los cuales son creados por el contenedor web que están disponibles para todos los JSPs.
- Objetos implícitos significa que no es necesario crearlos con New para poder utilizarlos.
- Algunos de estos objetos se pueden manejar por medio de EL.
- La siguiente tabla muestra los objetos implícitos más utilizados en los JSPs.

Objeto	Tipo
out	JspWriter
request	HttpServletRequest
response	HttpServletReponse
application	ServletContext
session	HttpSession

Objeto implícito (out)

- **Sirve para escribir datos al buffer (en el caso de los JSPs, la salida es directamente al navegador). En los JSPs dicho objeto se llama out.**
- **Si se va usar en los Servlets, se necesita escribirlo de la siguiente forma:**

```
PrinterWriter out = response.getWriter();
```

Ejemplo del objeto implícito out

En este ejemplo solo se esta mostrando la fecha actual del Sistema.

out.jsp

```
<html>
<body>

<% out.print("Today is:"+java.util.Calendar.getInstance().getTime()); %>

</body>
</html>
```

Objeto implícito (request)

- **El objeto implícito request de tipo HttpServletRequest es creado en cada JSP por el contenedor web. Puede ser usado para obtener informacion del request, como por ejemplo parámetros, las cabeceras, dirección IP remota, nombre del servidor, puerto del servidor, el content type, etc.**
- **También puede ser usado para crear, o eliminar atributos de los JSP con alcance de request.**

Ejemplo del objeto implícito request

En este ejemplo se envía un formulario a un JSP, y se despliega el valor del campo del formulario.

request.html

```
<form action="welcome.jsp">
   <input type="text" name="uname">
   <input type="submit" value="go"><br/>
</form>
```

welcome.jsp

```
<%
  String name=request.getParameter("uname");
  out.print("welcome "+name);
%>
```

Objeto implícito (response)

En los JSPs hay un objeto implícito llamado response de tipo HttpServletResponse el cual es creado por el contenedor web. Puede ser usado para manipular la respuesta, por ejemplo con un redireccionamiento a otro recurso, o enviar un error.

Ejemplo del objeto implícito response

En este ejemplo se utiliza el objeto response para enviar una respuesta como un redireccionamiento a un sitio externo (amazon.com)

response.html

```
<form action="redirect.jsp">
   <input type="text" name="uname">
   <input type="submit" value="go"><br/>
</form>
```

redirect.jsp

```
<%
  response.sendRedirect("http://www.amazon.com");
%>
```

Objeto implícito (application)

En los JSPs application es un objeto implícito de tipo ServletContext. La instancia de este objeto es creada solo una vez por el contenedor web cuando la aplicación o proyecto es deployed en el servidor. Este objeto puede ser usado para obtener parámetros de inicialización para toda la aplicación que son declarados en el deployment descriptor (web.xml)

application.html

```
<form action="ruta.jsp">
 <input type="text" name="uname">
 <input type="submit" value="go">
</form>
```

ruta.jsp

```
<%
  out.print("Welcome "+request.getParameter("uname")+"<br>");
  String ruta = application.getInitParameter("rutaDescargas");
  out.print("La ruta de descargas es =" + ruta);
%>
```

web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app . . .>
    <context-param>
    <param-name>rutaDescargas</param-name>
    <param-value>public/downloads</param-value>
  </context-param>
</web-app>
```

Objeto implícito (session)

En los JSPs session es un objeto implícito de tipo HttpSession. Es usado para obtener, crear, o eliminar variables con información acerca del usuario actual. Este objeto lo utilizaremos para administrar la autenticación en las aplicación que desarrollaremos llamada "Publicación de vacantes"

session.html

```
<form action="setSessionVar.jsp">
  <input type="text" name="uname">
  <input type="submit" value="go">
</form>
```

setSessionVar.jsp

```
<%
  String name=request.getParameter("uname");
  out.print("Welcome "+name);
  session.setAttribute("user",name);
%>
<a href="second.jsp">Otra pagina JSP</a>
```

second.jsp

```
<h3>Segunda pagina</h3>
<%
String name=(String)session.getAttribute("user");  out.print("Hello "+name);
%>
```

Administrador de aplicaciones de tomcat para ver información de la sesiones.

Tomcat Web Application Manager

Message:	OK

Manager			
List Applications	HTML Manager Help	Manager Help	Server Status

Applications

Path	Version	Display Name	Running	Sessions	Commands
/	*None specified*	Welcome to Tomcat	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/docs	*None specified*	Tomcat Documentation	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/el	*None specified*		true	1	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes

No. de Sessiones activas para esta aplicación.

Sesiones activas para la aplicación

Sessions Administration for /el

Tips:

- Click on a column to sort.
- To view a session details and/or remove a session attributes, click on its id.

Active HttpSessions informations

Refresh Sessions list 1 active Sessions

Session Id	Type	Guessed Locale	Guessed User name	Creation Time	Last Accessed Time	Used Time	Inactive Time	TTL
937B8FFD566A3170487EBD70F77D7664	Primary		Ivan Eliseo Tinajero Diaz	2016-02-26 10:03:26	2016-02-26 10:23:14	00:19:47	00:00:07	00:29:52

Invalidate selected Sessions

Detalles de una sesión

Details for Session 937B8FFD566A3170487EBD70F77D7664

Session Id	937B8FFD566A3170487EBD70F77D7664
Guessed Locale	
Guessed User	Ivan Eliseo Tinajero Diaz
Creation Time	2016-02-26 10:03:26
Last Accessed Time	2016-02-26 10:23:14
Session Max Inactive Interval	00:30:00
Used Time	00:19:47
Inactive Time	00:00:47
TTL	00:29:12

ID de la Sesión

Refresh

1 ATTRIBUTES

Remove Attribute	Attribute name	Attribute value
Remove	user	Ivan Eliseo Tinajero Diaz

Nombre del atributo creado

session.setAttribute("user",name);

Valor del atributo

En este caso, esto viene del formulario

String name=request.getParameter("uname");

Return to session list

Ejemplo – Objetos implícitos en los JSP

- El archivo el-objetos-implicitos-src.zip contiene un ejemplo de los objetos implícitos en los JSP.
- Para ver el ejemplo solo descomprime el archivo y haz clic en el botón “Open Project” en Netbeans y después en “Run”.

Secuencia del alcance que Java usa para buscar el atributo

Cuando se usa la sintaxis ${attributo} EL busca el atributo iniciando con el alcance más pequeño (page scope) y después hasta llegar al mayor alcance (application scope).

Scope	Description
page	The bean is stored in the implicit PageContext object.
request	The bean is stored in the HttpServletRequest object.
session	The bean is stored in the HttpSession object.
application	The bean is stored in the ServletContext object.

Ejemplo: código Servlet

JavaBeans alcance request

```
Categoria cat = new Categoria("Electronica");
request.setAttribute("categoria", cat);
```

Alcance request: el atributo esta disponible solo para la petición actual.

JavaBeans alcance session

```
Usuario user = new Usuario("ivan","administrador");
HttpSession session = request.getSession();
session.setAttribute("usuario",user);
```

Alcance session: el atributo esta disponible durante toda la sesión. Puede ser accedido en todos los Servlets y JSPs.

Ejemplo – Alcance de los atributos.

- El archivo el-scope-src.zip contiene un ejemplo para utilizar diferentes alcances de atributos.
- Para ver el ejemplo solo descomprime el archivo y haz clic en el botón “Open Project” en Netbeans y después en “Run”.

Más sobre Expression Language…

http://www.tutorialspoint.com/jsp/jsp_expression_language.htm