

Qué es un scriptlet, expresiones y expresiones

Section 6, Lecture 24

Un **scriptlet** en JSP es un fragmento de código Java escrito entre los símbolos `<%` y `%>`, es decir

```
<% código Java %>
```

en general, puede escribir cualquier código Java y esto, el hecho de disponer de todo el lenguaje Java detrás, es precisamente lo que hace que JSP sea el lenguaje de script de servidor mas potente. Más que ASP o PHP.

Ejemplo1:

Queremos producir un programa JSP que devuelva al cliente la fecha y hora del servidor, produciendo la salida,

Para hacer esto iniciamos una sesión de NetBeans y hacemos desde el menú principal "Archivo", "Proyecto Nuevo", "Java Web", "Web Application" y damos nombre al archivo jsp en la forma de scriptlet1, finalmente sustituimos el archivo index.jsp por el siguiente código,

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Scriptlet en páginas JSP</title>
  </head>
  <body bgcolor="ffffcc">
    <center>
      <%
        java.util.Date FechaHora = new java.util.Date();
        out.print("La fecha y Hora son: "+ FechaHora);
      %>
```

```
    </center>
  </body>
</html>
```

En JSP se utilizan las **expresiones** para insertar valores, obtenidos con Java, directamente a la salida que se envía al cliente o solicitante.

Tiene dos sintaxis:

```
<%= expresión Java %>
```

(Note el signo igual (=) que acompaña justo detrás del inicio de código JSP y asegúrese que dicho signo acompaña al de porcentaje (%)).

Existe otra sintaxis que es la siguiente

Cualquiera de las dos sintaxis se corresponde con una **expresión** en JSP. ¿Cómo funcionan las expresiones?. Pues muy sencillo, la expresión es evaluada en el Motor de JSP, se obtiene un valor de la expresión y éste se sustituye justo donde se utiliza.

Veamos un ejemplo sencillo, abrimos una nueva sesión de NetBeans 6.9, hacemos Nuevo Proyecto, Java Web, Web Application y le damos nombre expresiones1, cuando NetBeans 6.9 nos muestre la plantilla del archivo index.jsp lo cambiamos por este otro código.

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>JSP (Expresiones)</title>
  </head>
  <body>
    <h1>
      <center>
```

```
        Hola. La fecha y la hora son <%= new java.util.Date() %>
    </center>
</h1>
</body>
</html>
```

En el código anterior, tenemos una expresión. Esta expresión básicamente consiste en crear una nueva instancia de la clase Date, con lo cual la expresión new java.util.Date() producirá una llamada a la fecha del sistema por parte del Motor JSP desde el servidor, y esa fecha obtenida se sustituirá por todo lo que hay entre las etiquetas de comienzo y fin del lenguaje JSP.

La salida en el navegador será similar a la siguiente (con el correspondiente cambio de fecha y hora).

Las **declaraciones** de variables o métodos se hacen en JSP en la forma

```
<%! declaración de variable o método; %>
```

Note el signo (!) de admiración que acompaña justo detrás del signo porcentaje (%), allí donde comienza del código JSP.

Estas variables o métodos así definidos serán globales y por tanto accesibles desde cualquier lugar de la página JSP. Hay que tener en cuenta que el servidor a través del motor JSP transforma la página JSP en un servlet, y éste es usado por múltiples peticiones, lo que provoca que este tipo de variables conserven su valor entre sucesivas llamadas o ejecuciones. Las declaraciones se ejecutan una única vez, en la primera llamada al servlet equivalente.

En el caso de una declaración múltiple,

```
<%! variable1; [variable2;] ... %>
```

También puede inicializarse las variables en el momento de la declaración, como por ejemplo:

```
<%! int contador = 0; %>
```

Los ocho tipos de datos primitivos incluidos en el lenguaje de programación Java que pueden declararse para una variable son: **byte, short, int, long, float, double, boolean** y **char**.

También puede usarse esta otra sintaxis

Ejemplo1:

Queremos declarar y asignar valores a tres variables numéricas a través de declaraciones JSP para calcular finalmente el valor medio.

Iniciamos una nueva sesión con Netbeans IDE, abrimos un proyecto nuevo del tipo “Java Web” – “Web Application”, le damos por nombre **declaraciones1** y sustituimos el archivo index.jsp por el siguiente código:

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Ejemplo Declaraciones1</title>
  </head>
  <body>
    <%! double num1=2.2, num2=4.4, num3=6.0;%>
    <h2>
      <center>
        Los números a promediar son: <%=num1%>, <%=num2%> y
        <%=num3%><br><hr>
        <%! public double media(double n1,double n2,double n3){ return
(n1+n2+n3)/3; } %>
        Media = <%=media(num1,num2,num3)%>
      </center>
    </h2>
  </body>
</html>
```