

Declaraciones, Expresiones y Scriptlet

Operaciones básicas que podemos realizar en una página JSP.

Comentarios

Son comentarios sobre el código, sirven para ayudar al programador. El compilador los ignora.

```
<!-- Esto es un comentario --%>

<% // comentario de una linea
/*
Comentario de
varias lineas
*/
%>
```

Declaraciones

Se utilizan para definir variables y métodos que se utilizarán en scriptlets y expresiones posteriores.

Su sintaxis es:

```
<%!
Declaraciones
%>
```

Ejemplo:

```
<%!
int contador = 5;
int sumar(int a, int b){
    return a+b;
}
%>
```

Expresiones

Son una evaluación de una expresión Java.

Su sintaxis es:

```
<%= contador %>
```

CONTADOR es una variable con un valor que se mostrará donde se haya colocado la expresión.

Scriptles

Pueden incluir cualquier tipo de código Java dentro.

Su sintaxis es:

```
<%  
  // aquí pondremos el código Java que necesitamos  
  out.println("Hola desde Java");  
%>
```

Directivas

Son etiquetas a partir de las cuales se genera información que puede ser utilizada por el motor JSP. No producen una salida visible al usuario sino que configuran cómo se ejecutará la página JSP.

Su sintaxis es:

```
<%@directiva atributo="valor" %>
```

Las directivas disponibles son:

- **include**

Incluye el contenido de un fichero en la página mediante el atributo file.

```
<%@include file="cabecera.html" %>
```

- **taglib**

Importa bibliotecas de etiquetas (Tag Libraries)

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

- **page**

Especifica atributos relacionados con la página a procesar. Los atributos son:

Atributo	Sintaxis	Utilización
import	<%@page import="java.util.Map"%>	Importa clases y paquetes Java para ser utilizadas dentro del fichero JSP.
session	<%@page session="true"%>	Especifica si utiliza los datos contenidos en sesión; por defecto "true".
contentType	<%@page contentType="text/html"%>	Especifica el tipo MIME del objeto "response"; por defecto "text/html; charset=ISO-8859-1".

buffer	<%@page buffer="12kb"%>	Buffer utilizado por el objeto writer "out"; puede tomar el valor de "none"; por defecto "8kb".
errorPage	<%@page errorPage="error.jsp"%>	Especifica la ruta de la página de error que será invocada en caso de producirse una excepción durante la ejecución de este fichero JSP.
isErrorPage	<%@page isErrorPage="true"%>	Determina si este fichero JSP es una página que maneja excepciones. Únicamente a este tipo de páginas pueden acceder a la variable implícita "exception", que contiene la excepción que provocó la llamada a la página de error.

EJERCICIOS BÁSICOS JSP

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@page import="java.text.*, java.util.*"%>
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>JSP Page</title>
  </head>
  <body>
    <h1>Fecha</h1>
    <%
      Date d = new Date();
      String hoy = DateFormat.getDateInstance().format(d);
      %>
    Hoy es <%=hoy%>
  </body>
</html>
```

```
<%
  String nombre = request.getParameter("NOMBRE");
  if(nombre!=null && nombre.length()>0){
    out.println("Hola " + nombre);
  }else{
    out.println("No tienes acceso. ");
  }
%>
```

Ejemplo 1: Hola Mundo.

Página HTML

```
<html>

<head>

<title>Primer Servlet</title>

</head>

<body>

<center><h2>Ejemplo de Ejecución de un Servlet</h2></center><br><br>

<form action=/PrimerServlet/PrimerServlet method=GET>

<Center>

<BR><input type=submit value="Ejecutar el Servlet"></form>

</center>

</body>

</html>
```

Código del Servlet

```
//Primer Servlet. Muy sencillo.

import java.io.*;

import javax.servlet.*;

import javax.servlet.http.*;

public class PrimerServlet extends HttpServlet

{

    public void doGet (HttpServletRequest req, HttpServletResponse res)

throws ServletException, IOException {

    res.setContentType("text/html");

    PrintWriter out = new PrintWriter(res.getOutputStream());

    out.println("<html>");
```

```
out.println("<head><title>HolaMundoServlet</title></head>");
out.println("<body>");
out.println("<h1><center>Hola Mundo desde el servidor
WEB</center></h1>");
out.println("</body></html>");
out.close();
}

public String getServletInfo() {
    return "Crea una página HTML que dice HolaMundo";
}
```

Ejemplo 2: Acceso básico a formularios.

Página HTML

```
<html>
<head>
<title>Segundo Servlet</title>
</head>
<body>
<h2><center>Segundo Servlet</center></h2>
<form action=/SegundoServlet/SegundoServlet method=POST>
<BR>
<BR>
<center>
Introduzca su nombre y pulse el botón de enviar<BR>
<BR><input type=text name=NOMBRE>
<BR>
<BR><input type=submit value="Enviar Nombre">
<input type=reset value="Borrar">
</center>
```

</form>

</body>

</html>

Código del Servlet

```
// SegundoServlet.java
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;

public class SegundoServlet extends HttpServlet {
    String nombre;

    public void service( HttpServletRequest petición, HttpServletResponse
    respuesta )
        throws ServletException, IOException {
        nombre = petición.getParameter("NOMBRE");
        ServletOutputStream out = respuesta.getOutputStream();
        out.println("<html>");
        out.println("<head><title>HolaTalServlet</title></head>");
        out.println("<body>");
        out.println("<p><h1><center>Su nombre es:
        <B>"+nombre+"</B></center></h1></p>");
        out.println("</body></html>");
        out.close();
    }
}
```


[illegible]

Código del Servlet

```
// CalculadoraServlet

import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class CalculadoraServlet extends HttpServlet {

    public void service (HttpServletRequest req, HttpServletResponse
    res) throws ServletException, IOException {

        double op1, op2, result;

        int operacion;

        String simb_op[] = {"+", "-", "*", "/"};

        ServletOutputStream out = res.getOutputStream();

        op1 = Double.parseDouble(req.getParameter("operando1"));
        op2 = Double.parseDouble(req.getParameter("operando2"));
        operacion = Integer.parseInt(req.getParameter("operacion"));
    }
}
```

```
result = calcula(op1, op2, operacion);
```

```
out.println("<html>");
```

```
out.println("<head><title>Resultado de calcular con  
Servlet</title></head>");
```

```
out.println("<body BGCOLOR = \"#E0E0FF\"          TEXT= \"blue\">");
```

```
out.println("<h1><center>La operacion efectuada  
es:</center></h1>");
```

```
out.println("<h2> <b><center>"+ op1+"          "+ simb_op[operacion-  
1] +
```

```
"  "+ op2 +
```

```
"  =  "+ result +
```

```
"</center></b></h2>");
```

```
out.println("</body>");
```

```
out.println("</html>");
```

```
out.close();
```

```
}
```

```
public double calcula(double op1, double op2, int operacion){
```

```
double result = 0;
```

```
switch (operacion){
```

```
case 1:
```

```
return op1 + op2;
```

```
case 2:
```

```
return op1 - op2;
```

```
case 3:
```

```
return op1 * op2;
```

```
case 4:
```

```
return op1 / op2;}
```

```
return result;
    }
}
```

Ejemplo 4: Primitiva.

Página HTML

```
<html>
<head>
<title>Primitiva Servlet</title>
</head>
<body>
<h2><center>Primitiva Servlet</center></h2>
<form action=/PrimitivaServlet/PrimitivaServlet method=POST>
<BR>
<BR>
<center>
Introduce tu combinación y pulsa el botón de enviar<BR>
<BR>NUM1:<input type=text name=NUM1>
<BR>NUM2:<input type=text name=NUM2>
<BR>NUM3:<input type=text name=NUM3>
<BR>NUM4:<input type=text name=NUM4>
<BR>NUM5:<input type=text name=NUM5>
<BR>NUM6:<input type=text name=NUM6> <BR>
<BR><input type=submit value=Enviar Combinación>
<input type=reset value=Borrar>
</center>
</form>
</body>
</html>
```

Código del Servlet

```
// PrimitivaServlet.java

import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
import java.util.*;

public class PrimitivaServlet extends HttpServlet{

    int primi[] = new int[6], combiUsuario[] = new int[6];
    int i, contador=0, aux, aciertos=0;
    Random rand = new Random();

    public void init(ServletConfig config) throws ServletException {
        super.init(config);
        //generamos los números
        while (contador<6){
            aux = rand.nextInt(48) + 1;
            if (!comprueba(primi,aux)){
                primi[contador] = aux;
                contador++; }
        }
        //ordenamos el array
        Arrays.sort(primi);
    }

    private boolean comprueba(int array[], int num){
        for (int i=0; i<=5; i++){
            if (primi[i]==num) return true;
        }
        return false;
    }
}
```

```
public void service( HttpServletRequest petición, HttpServletResponse
respuesta ) throws ServletException, IOException {
    aciertos=0;

    ServletOutputStream out = respuesta.getOutputStream();
    combiUsuario[0] = Integer.parseInt(petición.getParameter("NUM1"));
    combiUsuario[1] = Integer.parseInt(petición.getParameter("NUM2"));
    combiUsuario[2] = Integer.parseInt(petición.getParameter("NUM3"));
    combiUsuario[3] = Integer.parseInt(petición.getParameter("NUM4"));
    combiUsuario[4] = Integer.parseInt(petición.getParameter("NUM5"));
    combiUsuario[5] = Integer.parseInt(petición.getParameter("NUM6"));

    out.println("<h2><center>Primitiva Servlet</center></h2>");
    //imprimimos todos los números de la combinación del usuario
    out.print("<p>Tu combinación es:</p><B>");
    for (i=0; i<6; i++) {
        out.print(" "+combiUsuario[i]);
    }
    out.print("</B>");
    //comprobamos la combinación
    for (i=0; i<=5; i++) {
        if (Arrays.binarySearch(primi,combiUsuario[i])>=0) {
            out.println("<p>Número acertado:
            <B>"+combiUsuario[i]+"</B></p>");
            aciertos++;
        }
    }
    out.println("<p>Números acertados: <B>"+aciertos+"</B></p>");

    //imprimimos todos los números de la combinación ganadora
```

```
out.print("<p>La combinación ganadora es:</p><B>");
for (i=0; i<6; i++) {
    out.print(" "+primi[i]);
} out.print("</B>");
out.close();
}
}
```

Ejemplo 5: Acceso a Ficheros de Texto Plano.

Página HTML

```
<html>
<head>
<title>Segundo Servlet</title>
</head>
<body>
<h2><center>Segundo Servlet</center></h2>
<form action=/FicheroServlet/FicheroServlet method=POST>
<BR>
<BR>
<center>
Introduzca su nombre y pulse el botón de enviar<BR>
<BR><input type=text name=TEXTO>
<BR>
<BR><input type=submit value=Enviar Nombre>
<input type=reset value=Borrar>
</center>
</form>
</body>
</html>
```

Código del Servlet

```
//    FicheroServlet.java

import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;

public class FicheroServlet extends HttpServlet {

    StringBuffer mensaje = null;
    FileOutputStream fos = null;
    String[] strTEXTO;

    public void init(ServletConfig config) throws ServletException {
        super.init(config);
        FileInputStream fis = null;
        try{
            fis = new FileInputStream("c:\\tmp\\texto.txt");
        } catch(java.io.FileNotFoundException e) {
            e.printStackTrace();
        }

        mensaje = new StringBuffer();
        try{
            int caracter;
            while ( (caracter = fis.read()) != -1 ){
                mensaje.append((char)caracter);
            } fis.close();
        }
        catch(java.io.IOException e) {
            e.printStackTrace();
        }
    }
}
```

```
Try {
fos = new FileOutputStream("c:\\tmp\\log.txt");
}
catch(java.io.IOException e) {
    e.printStackTrace(); }
}

public void service( HttpServletRequest petición,
HttpServletResponse respuesta )
throws ServletException, IOException {

strTEXTO = petición.getParameterValues("TEXTO");
ServletOutputStream out = respuesta.getOutputStream();

out.println("<p>" + mensaje + "</p>");
out.println("<p>Su nombre es: " + strTEXTO[0] + "</p>");
out.close();

registrar();
}

public void destroy() {
try {
fos.close(); }
catch(java.io.IOException e) {
    e.printStackTrace(); }
}

public synchronized void registrar() {
try {
fos.write(strTEXTO[0].getBytes());
```



```
}  
  
catch(java.io.IOException e) {  
    e.printStackTrace(); }  
  
}  
  
}
```

Ejemplo 6: Ejemplo JSP de uso de la directiva INCLUDE

```
<HTML>  
  
<head>  
  
<title> Página de prueba de directivas de compilación </title>  
  
</head>  
  
<body>  
  
<h1> Página de prueba de directivas de compilación </h1>  
  
<%@ include file="/fich1.html" %>  
  
<%@ include file="/fich2.jsp" %>  
  
</body>  
  
</HTML>
```

· fich1.html

```
<HTML>  
  
<head> <title> Hola, Mundo </title> </head>  
  
<body> <h1> ¡Hola, Mundo! </h1>  
  
</body>  
  
</HTML>
```

· fich2.jsp

```
<%@ page info="Un ejemplo Hola Mundo" import="java.util.Date" %>  
  
La fecha de hoy es: <%= new Date().toString() %>
```

Ejemplo 7: Ejemplo JSP de uso de formularios

Página comprobar.html.

```
<HTML>
<head> <title> Ejemplo de uso del forward </title>
</head>

<body>

<h1> Ejemplo de uso del forward </h1>
<form method="post" action="comprobar.jsp">

<input type="text" name="Nombre">
<br> y clave:
<input type="password" name="password"> </p>
<p><input type="submit" name="login">
</form>

</body>
</HTML>
```

Página comprobar.jsp.

```
<% if
((request.getParameter("Nombre").equals("Roberto")) &&
(request.getParameter("password").equals("xyzyz"))) {
%>

<jsp:forward page="saludoforward.jsp"/>
```

```
<% } else { %>
```

```
<%@ include file="comprobar.html"%>
```

```
<% } %>
```

Página saludoforward.jsp.

```
<HTML>
```

```
<head>
```

```
<title> Saludo al cliente </title>
```

```
</head>
```

```
<body>
```

```
<h1> Saludo al cliente</h1>
```

```
<%
```

```
out.println("Bienvenido a la nueva aplicación");
```

```
%>
```

```
</body>
```

```
</HTML>
```

Ejemplo 8: Ejemplo JSP de uso de objetos implícitos

Página HTML

```
<HTML>
```

```
<head>
```

```
<title> Formulario de petición de nombre </title>
```

```
</head>
```

```
<body>

<h1> Formulario de petición de nombre </h1>

<!-- Se envía el formulario al JSP "saludo.jsp" -->

<form method="post" action="saludo.jsp">
<p> Por favor, introduce tu nombre:
<input type="text" name="nombre">
</p>
<p> <input type="submit" value="enviar información">
</form>

</body>

</HTML>
```

Página JSP –saludo.jsp

```
<HTML>

<head> <title> Saludo al cliente </title>

</head>

<body>

<h1> Saludo al cliente</h1>

<%-- Los parámetros que le pasa el cliente en la petición se obtienen del
objeto implícito request --%>

<%
String nombre = request.getParameter("nombre");
out.println("Es un placer conocerte, " + nombre); %>

<%-- Al evaluarse el código hay que escribir explícitamente en la salida
(objeto implícito out) --%>

</body>

</HTML>
```