



EBook Gratis

# APRENDIZAJE

## jsp

Free unaffiliated eBook created from  
**Stack Overflow contributors.**

#jsp

# Tabla de contenido

<b>Acerca de</b>	<b>1</b>
<b>Capítulo 1: Empezando con jsp</b>	<b>2</b>
Observaciones	2
Examples	2
Hola Mundo	2
JavaScript	2
Instalando JSP	2
<b>Capítulo 2: Comentarios en JSP</b>	<b>3</b>
Introducción	3
Observaciones	3
Examples	3
Ejemplo de comentarios JSP	3
Comentario HTML en un JSP	4
<b>Capítulo 3: Creación de bibliotecas de etiquetas personalizadas utilizando JSP 2.0 (sin cl</b>	<b>5</b>
Introducción	5
Observaciones	5
Examples	5
Un ejemplo básico de usar una etiqueta personalizada sin ningún código Java	5
<b>Capítulo 4: Directivas</b>	<b>7</b>
Sintaxis	7
Examples	7
Ejemplo simple	7
<b>Capítulo 5: Objetos implícitos</b>	<b>8</b>
Introducción	8
Examples	8
JSP sale del objeto implícito, simplemente muestra la fecha y la hora	8
solicitar objeto en JSP	8
<b>Creditos</b>	<b>9</b>

---

## Acerca de

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [jsp](#)

It is an unofficial and free jsp ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official jsp.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to [info@zzzprojects.com](mailto:info@zzzprojects.com)

---

# Capítulo 1: Empezando con jsp

## Observaciones

JSP es una **tecnología de vista Java que se** ejecuta en un servidor que le permite escribir texto de plantilla en idiomas del lado del cliente como [HTML](#) , [CSS](#) , [JavaScript](#) , etc. JSP es compatible con las denominadas *listas de etiquetas* que están respaldadas por piezas de código Java con las que puede controlar el flujo de páginas y / o la salida de forma dinámica (mediante programación). Un taglib bien conocido es [JSTL](#) . JSP también es compatible con [Expression Language](#) ( [EL](#) ), con una sintaxis como `${}` que se puede usar para acceder a datos de back-end (en realidad, los atributos que están disponibles en la página, solicitud, sesión y ámbitos de aplicación), principalmente en combinación con taglibs.

## Examples

### Hola Mundo

El contenido se ha movido de nuevo a la buena [página de wiki de JSP](#)

### JavaScript

El contenido se ha movido de nuevo a la buena [página de wiki de JSP](#)

### Instalando JSP

El contenido se ha movido de nuevo a la buena [página de wiki de JSP](#)

Lea Empezando con jsp en línea: <https://riptutorial.com/es/jsp/topic/1017/empezando-con-jsp>

---

# Capítulo 2: Comentarios en JSP

## Introducción

Hay varias formas de agregar comentarios en la página JSP, ya que es la tecnología de vista java que se enfoca principalmente en la parte de vista de la aplicación web que incluye html principalmente y, al ser la tecnología java, también puede contener código java. Entonces, uno necesita entender cómo usar comentarios apropiados en jsp.

Hay 3 tipos de comentarios que podemos usar en la página JSP:

Comentario JSP

Comentario HTML

Comentario de Java (que puede usarse dentro de scriptlets)

## Observaciones

Para ver el código fuente de la clase java traducido por el contenedor de la página JSP:

Eclipse:

WORKSPACE\.metadata\plugins\org.eclipse.wst.server.core\tmp0\work\Catalina\localhost\PROJECT\_NAME\org\

Netbeans: Haga clic derecho en la página y seleccione ver servlet

## Examples

### Ejemplo de comentarios JSP

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html">
<title>JSP Comments</title>
</head>
<body>
    <%-- JSP comments --%> -- Ignored by container, you can't see this comment in source code
    of JSP_java class translated by container, doesn't include in response

    <!-- HTML comments --> -- Ignored by browser, You can see this comment in source code of
    html page but browser ignores it to print on page, included in response

    <% // java comments %> -- Ignored by JRE, This comment can be seen in JSP_java class
    translated by container, doesn't include in response

    <!-- Today is <%= new java.util.Date() %> -->

</body>
</html>
```

Cuando ejecute el código anterior, el resultado será una página en blanco. Para comprender la diferencia entre estos comentarios, debe ver la fuente de la página web y el código fuente de la clase JSP\_Java traducidos por contenedor.

## Comentario HTML en un JSP

También puede incluir comentarios HTML en JSP. Usas la sintaxis de comentario html básica:

```
<!--Comment goes here-->
```

La diferencia entre el uso de comentarios de estilo JSP y comentarios de estilo HTML es que los JSP no se incluirán cuando se genere el HTML y los comentarios de estilo HTML. Por lo tanto, depende de si desea que se vean si alguien ve el código fuente.

Lea Comentarios en JSP en línea: <https://riptutorial.com/es/jsp/topic/9040/comentarios-en-jsp>

---

# Capítulo 3: Creación de bibliotecas de etiquetas personalizadas utilizando JSP 2.0 (sin clases java)

## Introducción

Para usar etiquetas personalizadas en JSP, solíamos tener archivos TLD, junto con una clase de Java para definir las etiquetas personalizadas, pero publicar las especificaciones de JSP 2.0, se puede lograr lo mismo, sin la necesidad de ningún archivo de código fuente, lo que facilita la tarea. para mantener y también puede ser creado por alguien con conocimientos básicos de etiquetas JSP sin necesidad de código java.

## Observaciones

Por lo general, para agregar contenido estático a una página HTML, usamos

link rel = etiqueta "import" (en HTML5) o jQuery .load ("")

para incluir cualquier contenido externo. En el caso de cualquier contenido dinámico, utilizando JSP, usamos

jsp: incluir

para incluir dicho contenido. Pero para los casos, cuando queremos agregar contenido personalizado, que pueda interactuar con el contenido existente, usamos etiquetas JSP personalizadas. Nos permiten generar contenido personalizado, basado en los parámetros de la persona que llama y también devolvemos valores procesado en la etiqueta personalizada de nuevo a la persona que llama.

Referencias:

1. [Etiquetas personalizadas en páginas JSP.](#)
2. [Diferencia entre las etiquetas JSP e incluye](#)
3. [Variable variable de la etiqueta](#)

## Examples

### Un ejemplo básico de usar una etiqueta personalizada sin ningún código Java

Para crear una etiqueta de uso, debemos seguir un par de pasos:

1. Cree un archivo de etiquetas, definiendo los atributos que utiliza y las variables que utilizará la etiqueta.

a. Los atributos deben tener un nombre, su tipo y un campo obligatorio con un valor booleano

segundo. Las variables se definirán con un nombre junto con un cierto alcance-

NESTED (Disponible en la etiqueta del cuerpo),

AT\_BEGIN (dentro de la etiqueta hasta el final del alcance) y

AT\_END (Desde el final de la etiqueta hasta el final del alcance)

```
<%@ attribute name="name" required="true" type="java.lang.String" description="Name
of User" %>
<%@ attribute name="role" required="true" type="java.lang.String" description="Role
of User" %>
<%@ variable name-given="passBack" scope="AT_BEGIN"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>

<h1>${name}</h1>
<h2>${role}</h2>

<%String backToCaller="Back to caller";%>
<c:set var="passBack" value="Pass back successful"/>
```

2. Defina la entrada para importar etiqueta, donde tagdir será el directorio que contiene el archivo de etiqueta con extensión .tag

```
<%@ taglib prefix="tags" tagdir="/WEB-INF/tags" %>
```

3. Finalmente use la etiqueta, pasando los atributos requeridos, según la definición de la etiqueta. Aquí `customTag` es el nombre del archivo de etiqueta, junto con el prefijo como se define

```
<tags:customTag name="Hello Tag!!" role="I am the boss here"/>
```

4. El conjunto de variables de la etiqueta se puede recuperar en el jsp de la persona que llama utilizando el lenguaje de expresión

Hola \$ {passBack}

Lea Creación de bibliotecas de etiquetas personalizadas utilizando JSP 2.0 (sin clases java) en línea: <https://riptutorial.com/es/jsp/topic/8764/creacion-de-bibliotecas-de-etiquetas-personalizadas-utilizando-jsp-2-0--sin-clases-java->



---

# Capítulo 4: Directivas

## Sintaxis

- `<% @ policyName attributeName = "valor"%>`

## Examples

### Ejemplo simple

Las directivas, como su nombre indica, son instrucciones o instrucciones que debe seguir el contenedor al traducir una JSP a un servlet. Hay 3 directivas a saber, `page`, `include` y `taglib` que puede usar en su JSP.

A continuación se muestra un ejemplo simple de uso de la directiva de `page` :

```
<%@ page isErrorPage="true" %>
```

Esto indicaría al contenedor que el archivo JSP que contiene esta línea es una página de error.

Las directivas se pueden usar en cualquier lugar del archivo JSP, pero la convención es colocar al principio / principio del archivo JSP

Lea Directivas en línea: <https://riptutorial.com/es/jsp/topic/2756/directivas>

# Capítulo 5: Objetos implícitos

## Introducción

Objetos Java creados por el contenedor web. Hay 9 objetos implícitos | objeto | Tipo | | ----- | --  
----- | | fuera | JspWriter | | solicitud | HttpServletRequest | | respuesta |  
HttpServletResponse | | config | ServletConfig | | aplicacion | ServletContext | | sesión |  
HttpSession | | pageContext | PageContext | | pagina | Objeto | | excepción | Throwable |

## Examples

JSP sale del objeto implícito, simplemente muestra la fecha y la hora.

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

### solicitar objeto en JSP

se utiliza para obtener información de solicitud, como parámetro, información de encabezado, dirección remota, nombre del servidor, puerto del servidor, tipo de contenido, codificación de caracteres.

#### ***Index.html***

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

#### ***req.jsp***

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

Lea Objetos implícitos en línea: <https://riptutorial.com/es/jsp/topic/8205/objetos-implicitos>

# Creditos

S. No	Capítulos	Contributors
1	Empezando con jsp	<a href="#">BalusC</a> , <a href="#">CodeWarrior</a> , <a href="#">Community</a> , <a href="#">Jozef Chocholacek</a> , <a href="#">Menuka Ishan</a> , <a href="#">shanmuga</a> , <a href="#">Shashanth</a> , <a href="#">Squidward</a> , <a href="#">user7491506</a> , <a href="#">zack6849</a>
2	Comentarios en JSP	<a href="#">Naman</a> , <a href="#">user7491506</a>
3	Creación de bibliotecas de etiquetas personalizadas utilizando JSP 2.0 (sin clases java)	<a href="#">Shivam Aggarwal</a>
4	Directivas	<a href="#">Abubakkar</a>
5	Objetos implícitos	<a href="#">KIRAN KUMAR MATAM</a>