



Objetos Implícitos.



¿Qué son?

Los objetos implícitos son una serie de objetos asociados a las APIs de los servlets y las páginas JSP suministradas por el contenedor jsp, ya que estos pueden ser ocupados en cualquier scriptlet o expresión y aumentar su funcionalidad, con esto se consigue que no tengan que ser declarados de manera implícita.

Entre estos objetos se encuentran request, response, sesión, out, application, config, pageContext y page.

Estos permiten la el almacenamiento y recuperación de atributos, y se entiende por atributo a cualquier tipo de objeto de Java.



¿Qué son?

Su capacidad de almacenar y recuperar tiene gran importancia al tiempo de realizar interacción entre páginas JSP o Servlets.

Cada uno de estos objetos tiene métodos comunes para realizar acciones para el almacenamiento y recuperación de atributos, los cuales son: `setAttribute()`, `getAttribute()`, `setAttributeNames()` y `removeAttribute()`.



Métodos más Usuales

	Acción
<code>void setAttribute()</code>	Almacena un atributo de un objeto implícito.
<code>Object getAttribute()</code>	Devuelve dentro de un atributo almacenado en objeto implícito.
<code>Enumeración getAttributeNanem()</code>	Devuelve dentro de un objeto <code>java.util.Enumeration</code> los nombres de todos los atributos disponibles.
<code>void removeAttribute(String parametro)</code>	Elimina de la petición el atributo indicado por el parámetro.

The image features a large white circle centered on a black background. To the left of the white circle, there is a series of overlapping gray circles of varying shades, with a large white number '1' in the center of the darkest circle. To the right of the white circle, there is a series of concentric white circles of varying sizes.

1

REQUEST



REQUEST

Es una petición que realiza la pagina JSP, su clase o interfaz es: `javax.servlet.http.HttpServletResponse` y tiene la categoría de entrada y salida.

Este objeto contiene la información de la petición que un usuario ha realizado a la página JSP. El objeto request, contiene la información referida a los campos enviados de un formulario, cabeceras de petición HTTP, cookies, etc.



Asimismo, el objeto **request** permite el almacenamiento de información y recuperación de atributos, un atributo puede ser cualquier tipo de objeto Java. En consecuencia, al objeto **request** se puede añadir otros objetos en calidad de atributos y enviarlos a otras páginas JSP o a otros recursos del servidor.

Otro grupo de métodos del objeto **request** son aquellos que se utilizan para acceder a los parámetros de la petición, estos parámetros pueden ser los campos de un formulario o una cadena de consulta. Estos métodos son:

Enumeration getParameterNames(): devuelve en un objeto Enumeration los nombres de los distintos parámetros presentes en la petición.

String getParameter(String nombre): devuelve el valor del parámetro indicado, si no existiera devuelve el valor null.

String[] getParameterValues(String nombre): devuelve un array de objetos String los distintos valores que tiene un parámetro determinado, devuelve null en caso de que no exista el parámetro.

The image features a large white circle centered on a black background. To the left of the white circle, there is a series of overlapping circles in shades of gray, with the number '2' in white. To the right of the white circle, there is a series of concentric white circles.

2

RESPONSE



RESPONSE

Tiene un ámbito relativo a la pagina, y es una instancia de `javax.servlet.ServletResponse`. Encapsula la respuesta generada por la página JSP o servlet que será enviada al cliente en respuesta a su solicitud.



Los métodos más utilizados son:

setContentType(): se utiliza para establecer el tipo MIME y la codificación de caracteres de la página.

addCookie (cookie Cookie): se utiliza para agregar una nueva cookie a la respuesta.

setHeader (String nombre, String value)

sendRedirect (String): Se utiliza para reenviar la petición a otra url o servlet.

sendError (status_code int): Se utiliza para enviar como respuesta un error usando un estado específico.



3

OUT



OUT

Instancia de `javax.servlet.jsp.JspWriter`, este objeto permite escribir la respuesta o parte de ella, en HTML o cualquier otro contenido, como resultado en la petición cliente y enviarla al browser.



Los métodos más frecuentes son:

print: Escribe directamente en el archivo de salida HTML que se devolverá al cliente.

println: Escribe directamente en el archivo de salida HTML que se devolverá al cliente añadiendo una nueva línea.

newline(): Este método se utiliza para escribir una nueva línea. Su sintaxis es `out.newLine()`

Ejemplos.

Ejemplo sencillo con print o println:

```
<% out.println(«Hola mundo»); %>
```

También se puede agregar una marca HTML:

```
<% out.println(«<br><H2>Hola mundo</H2>»); %>
```

Al igual que se puede incorporar cualquier variable previamente declarada:

```
<% out.println(«<font color='red'>El valor del resultado es:>+resultado+«</font>»); %>
```



The image features a large white circle centered on a black background. To the left of the white circle, there is a series of overlapping circles in shades of gray, with the number '4' in white at the center. To the right of the white circle, there is a series of concentric white circles. The overall design is minimalist and geometric.

4

APPLICATION



APPLICATION

Representa a la propia aplicación web a la que pertenece la página JSP actual y su clase/interfaz es `javax.servlet.http.HttpSession` y esta en la categoría de contexto.

Una aplicación está definida en una carpeta dentro del contenedor web, por lo que una página JSP pertenecerá a una aplicación u otro dependiendo del URL.



El objeto `application` al ser una instancia de `ServletContext` ofrece todos los métodos de esta interfaz, además de los métodos de almacenar y recuperar atributos (`setAttribute` y `getAttribute`) en el ámbito de aplicación.

Ejemplos.

- **int getMajorVersion()**: Devuelve la versión superior del Servlet implementada por el contenedor web.
- **int getMinorVersion()**: Devuelve la versión inferior del Servlet implementada por el contenedor web.
- **String getServerInfo()**: Devuelve el nombre y la versión del contenedor web de la página que se está ejecutando.



A large white circle is centered on a black background. To its left, there is a series of overlapping gray circles of varying opacities, with the number '5' in white at the center. To its right, there are several thin, concentric white circles.

5

CONFIG



CONFIG

Este objeto es utilizado para enviar la información de configuración de la pagina JSP.

Config es una instancia a la interfaz de ServletConfig de servlet.



La información se envía a la página JSP a través del archivo web.xml. El objeto de configuración se utiliza para obtener información, su uso mas amplio es para la inicialización de parámetros.

El alcance de este objeto es solo para una sola página JSP



6

PAGE



PAGE

Es una instancia de la pagina JSP actual, su clase/interfaz es la `javax.lang.Object` y se relaciona con los servlets.

Es un obeejto JSP que contiene dos tipos de texto: datos estáticos que se pueden basar en cualquier formato de texto, y elementos JSP que construyen un contenido dinámico



La página puede estar compuesta por un archivo principal que incluye otros archivos que contienen una página JSP completa o un fragmento de una página JSP. La extensión recomendada para el archivo fuente de un fragmento de una página JSP es .jspx.

Los elementos JSP en una página JSP se pueden expresar en dos sintaxis, estándar y XML, una página JSP en sintaxis XML es un documento XML y puede manipularse mediante herramientas y API.



GRACIAS!

Kharites!