

Diseño y Desarrollo de Sistemas de Información.

Sesión Práctica 5. Introducción a JSP (Java Server Pages)

Objetivo

El propósito principal de este tutorial es el de proporcionar a los nuevos usuarios las habilidades necesarias para construir páginas JSP sencillas. Este documento explica cómo utilizar el entorno visual de edición de JSP y HTML de JDeveloper 11g, y guía al usuario a través de un conjunto de pasos comunes para el desarrollo de aplicaciones web, tal como el formateo de HTML, el trabajo con imágenes, CSS, elementos de formulario HTML y cómo trabajar con etiquetas JSP personalizadas en el editor visual.

Índice

En este documento trataremos los siguientes puntos:

- Introducción
- Paso 1 – Formateando HTML
 - o Crear una nueva aplicación, proyecto y JSP
 - o Formateando HTML
 - o Formateando una lista
 - o Añadiendo un enlace
- Paso 2 – Usando imágenes.
- Paso 3 – Usando CSS
- Paso 4 – Tablas HTML
- Paso 5 - Formularios HTML
- Paso 6 – Etiquetas JSP
- Resumen

Introducción

Este tutorial presenta una introducción al entorno de edición visual de JDeveloper 11g para JSP y HTML. Guiará a los usuarios a través de los pasos básico para trabajar con los elementos habituales de JSP y HTML, lo que incluye el formateo de HTML, el trabajo con imágenes, CSS, tablas y formularios HTML. Adicionalmente se proporciona una introducción básica al trabajo con etiquetas y elementos JSP sin necesidad de tener conocimientos sobre Java.

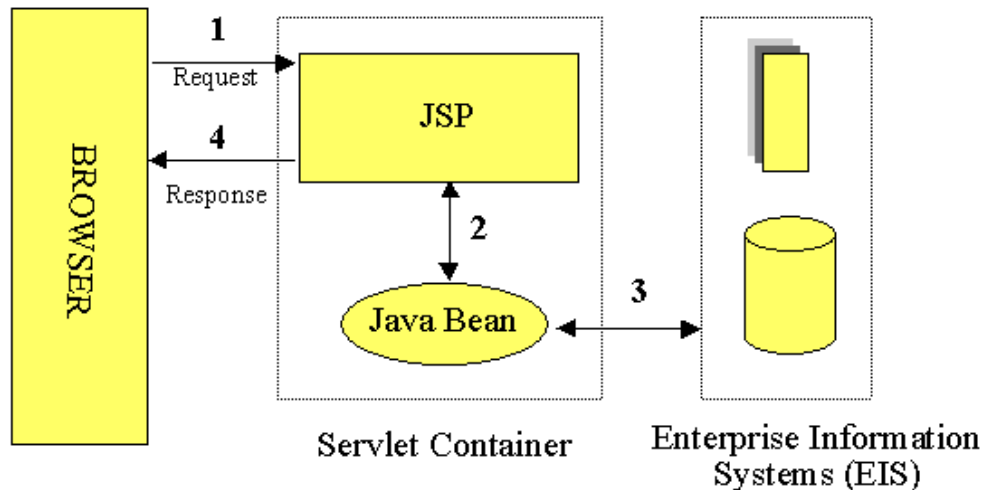
¿Qué son las JSP?

Las Java Server Pages (JSPs) son un método de creación de páginas web dinámicas en la parte del servidor usando el lenguaje Java. En ese sentido son similares a otros métodos o lenguajes tales como el PHP, los CGI (Common Gateway Interface) que son programas que generan páginas web en el servidor, o los ASP (Active Server Pages), un método específico de Microsoft. Sin embargo, se diferencian de ellos en otras cosas.

Para empezar, las páginas JSPs se ejecutan en una máquina virtual Java, lo cual permite que, en principio, se puedan usar en cualquier tipo de plataforma, siempre que exista una máquina virtual Java para ella. Cada JSP

se ejecuta en su propia hebra, es decir, en su propio contexto; pero no se comienza a ejecutar cada vez que recibe una petición, sino que persiste de una petición a la siguiente. Su persistencia le permite realizar ciertas tareas de forma más eficiente: conexión a bases de datos y manejo de sesiones, por ejemplo.

Los JSPs son en realidad servlets: un JSP se compila a un programa en Java la primera vez que se invoca, y del programa en Java se crea una clase que se empieza a ejecutar en el servidor como un servlet. La principal diferencia entre los servlets y los JSPs es el enfoque de la programación: un JSP es una página Web con etiquetas especiales y código Java incrustado, mientras que un servlet es un programa que recibe peticiones y genera a partir de ellas una página web



Ambos necesitan un programa que los contenga, y sea el que envíe de forma efectiva páginas web al servidor, y reciba las peticiones, las distribuya entre los servlets, y lleve a cabo todas las tareas de gestión propias de un servidor web.

Mientras que servidores como el Apache están especialmente pensados para páginas web estáticas CGI, y programas ejecutados por el servidor tales como PHP, hay otros servidores específicos para servlets y JSPs llamados contenedores de servlets (servlet containers) o servlet engines.

El contenedor de Servlets mas difundido actualmente es Jakarta Tomcat, también de la fundación Apache. Tomcat es Open Source, y está continuamente en desarrollo, existe una gran variedad de desarrollos adicionales a su alrededor como puede ser Cocoon que se emplea para servir páginas XML. Puede servir páginas sólo o bien como un añadido al servidor Apache. Además es relativamente rápido, y fácil de instalar.

Sintaxis de las páginas JSP

Las páginas JSP no son más que páginas HTML a las que se les ha incrustado código Java, que genera valores de forma dinámica. Por tanto podemos decir que JSP no es más que una mezcla de HTML y Java, aunque con algunas particularidades.

El código java debe ir separado del contenido HTML por medio de unos delimitadores especiales:

```
<% /*Código Java*/ %>
```

Además las páginas JSP incluyen otro tipo de sentencias como pueden ser:

- **Directivas**

Modifican el comportamiento, y por tanto el resultado final de la página. Se comienzan a ejecutar antes de comenzar el procesamiento de la página.

Las directivas son `include`, `page` y `taglib`, cada una de ellas contiene una serie de posibles atributos, aquí vemos un ejemplo de sintaxis para la directiva `page`.

```
<%@ page language='java'
    contentType='text/html'
    info='Comentario'
    errorPage='paginaError.jsp' %>
```

- **Declaraciones**

Declaración de variables globales persistentes de una ejecución a otra de la página

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

- **Expresiones JSP**

Las expresiones JSP, son expresiones que devuelven un resultado. Lo que se hace es evaluar la expresión y posteriormente se imprime en la página el resultado. No es necesario que estas expresiones terminen con ;

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

- **Scriptlets**

Fragmentos de código Java intercalados en el código HTML, que pueden contener cualquier tipo de instrucción Java, estructuras condicionales, bucles, etc...

```
<% if(contador>5){ %>
    <h2>Número mayor que 5</h2>
<% }
    else{ %>
        <h1>Número menor que 5</h1>
<% } %>
```

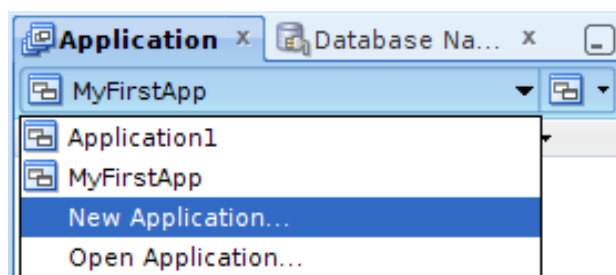
Paso 1 – Formateo de HTML

Este paso guiará al usuario primero por la creación de un Workspace, un proyecto y una página inicial JSP. Demostrará como realizar el formateo HTML tal como añadir negritas, cambiar el color de las fuentes, la alineación, etc...

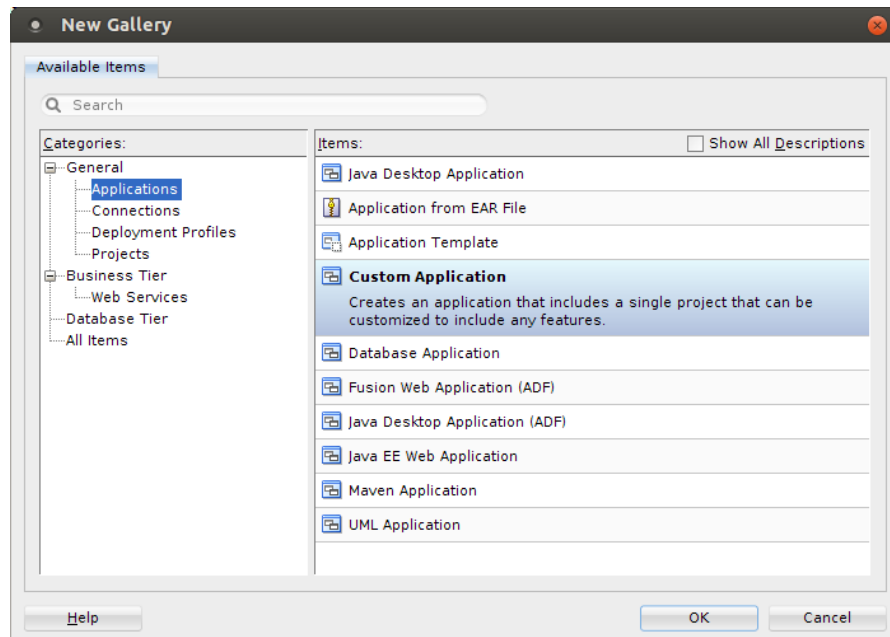
- Crear un nuevo Workspace, proyecto y JSP
- Formateo de HTML
- Formateando una Lista
- Añadiendo un enlace

Crear una nueva aplicación, proyecto y JSP

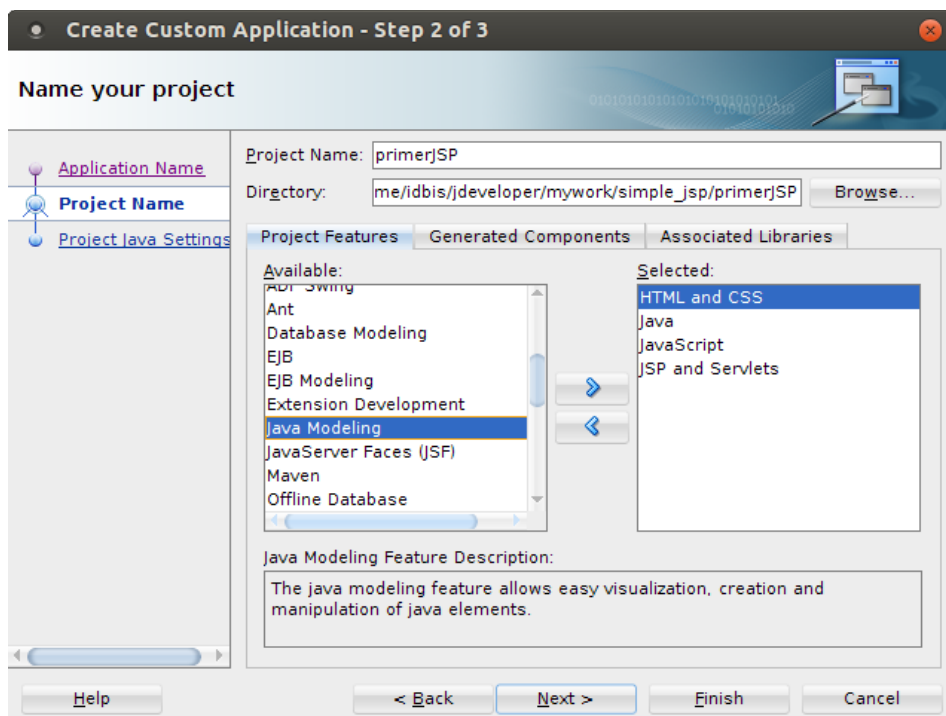
1. En el navegador de Aplicaciones, en el desplegable de aplicaciones seleccione **New Application...**



2. En la New Gallery seleccionamos la categoría Applications y el elemento Custom Application , pulsamos OK.



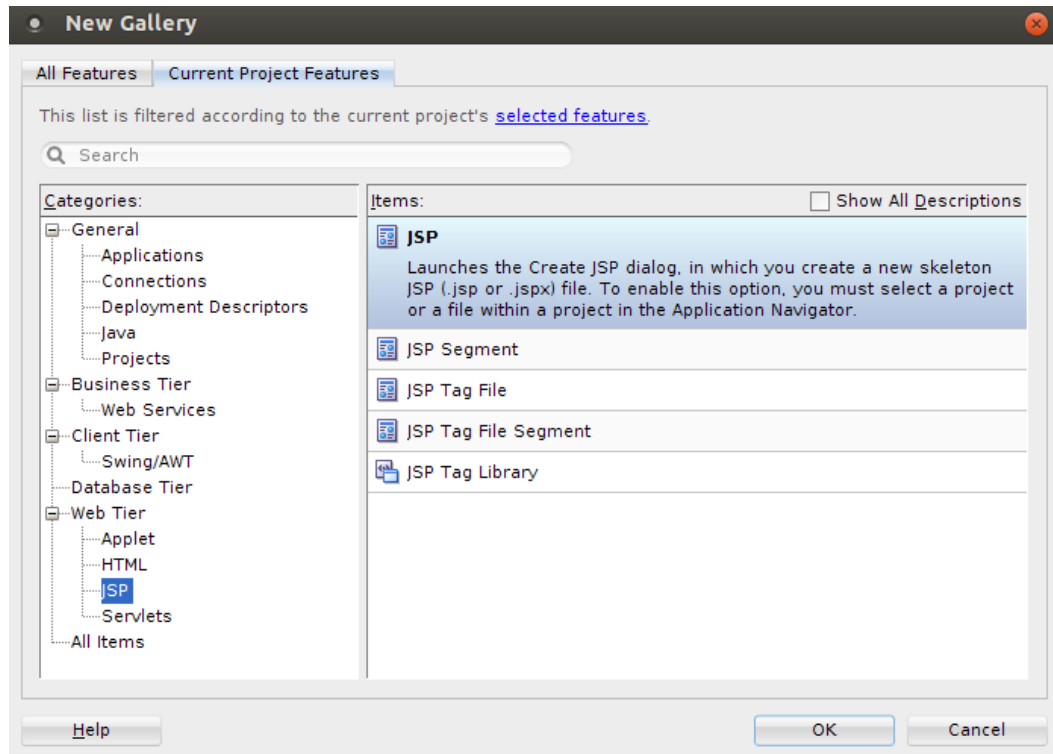
3. Le damos el nombre **simple_jsp** a la aplicación. Llamamos al proyecto **primerJSP**. Seleccionamos las siguientes tecnologías: HTML and CSS y JSP and Servlets. De forma automática se añadirá Java y Javascript. Pulse finalizar, se creará una nueva aplicación y un nuevo proyecto de tipo web.



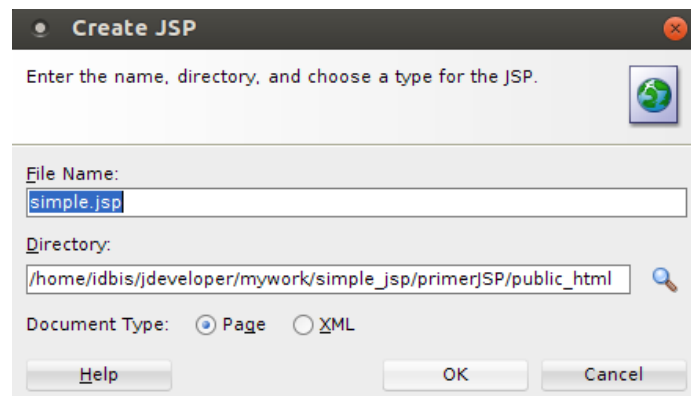
4. En el

Navegador haga pulse con el botón derecho sobre el nuevo proyecto vacío, y elija **New...** para que aparezca la New Gallery. En la New Gallery, abra el nodo **Web Tier**, seleccione la categoría **JavaServer Pages (JSP)** y haga doble clic sobre el elemento **JSP Page**.

5.

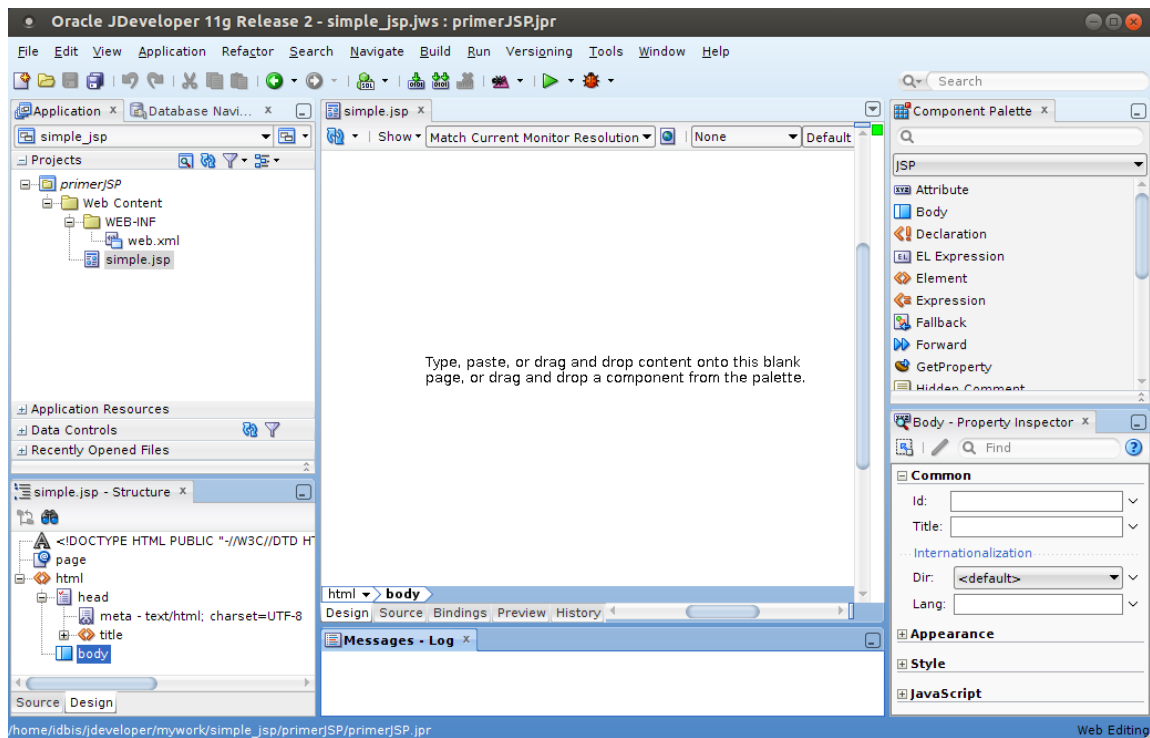


Nombre el fichero JSP como **simple.jsp**, y pulse **Ok** para generar una página JSP vacía.

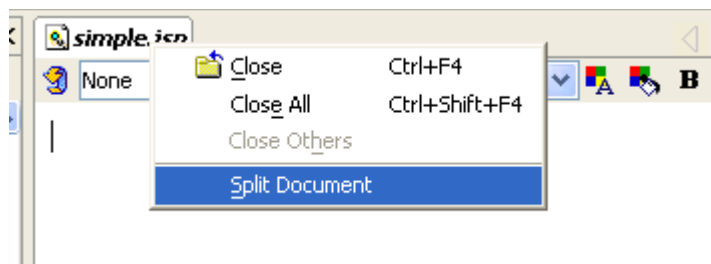


La nueva página JSP aparecerá en el editor Visual JSP.

6. Observe que el entorno de edición visual JSP/HTML es en realidad una combinación de los editores visual y de código, el Panel de Estructura en la parte baja a la izquierda, que permite el acceso a través de una jerarquía a todos los elementos de la página, el Inspector de Propiedades en la parte baja a la derecha, que permite cambiar el valor de los atributos y la Paleta de Componentes en la parte superior a la derecha, que permite insertar nuevas etiquetas/elementos en la página JSP/HTML.



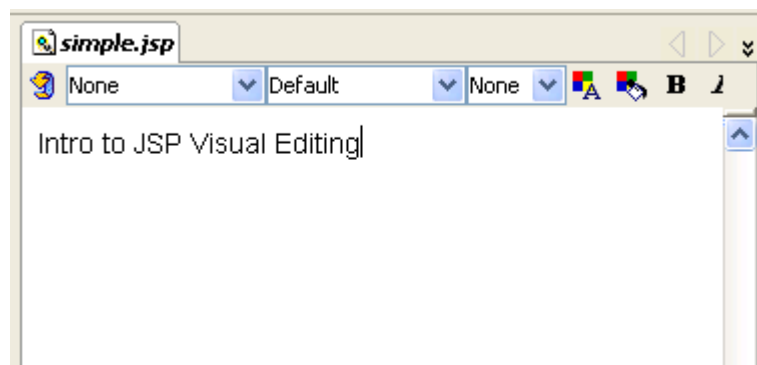
7. Como un paso opcional puede dividir la vista del editor para mostrar tanto el editor visual como el de código. En la pestaña simple.jsp en la parte superior del editor, pulse con el botón derecho y seleccione **Split Document**. JDeveloper mostrará ambas vistas de forma simultánea.



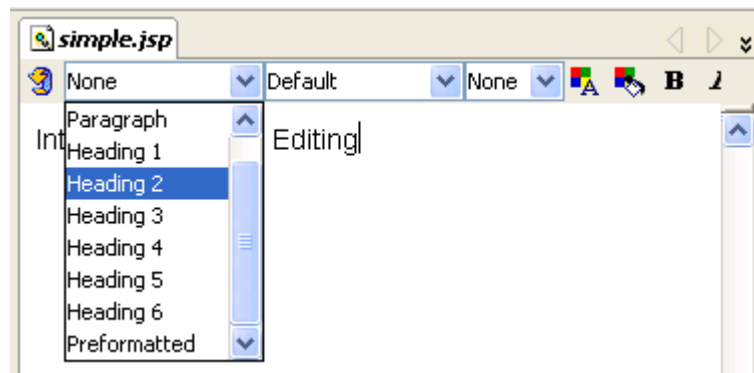
Puede seleccionar **Unsplit Document** para volver a una vista, o puede mantener la vista de código y editar JSP de forma simultánea.

Formateando HTML

1. Pulse dentro del editor visual JSP e introduzca algún texto.

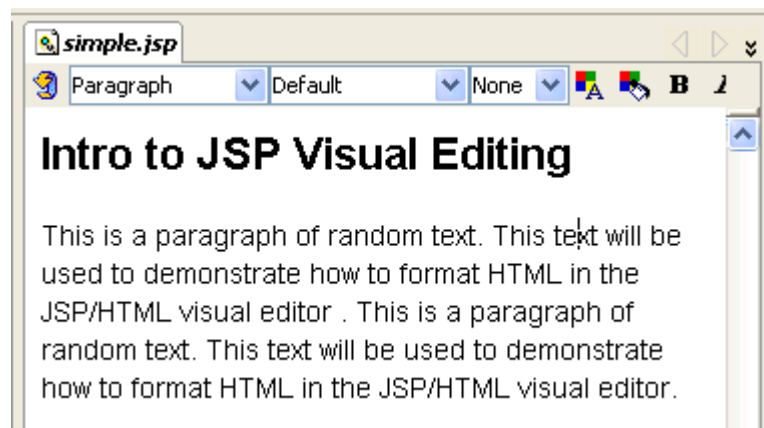


- Tras introducir el texto, pulse en el menú desplegable de la izquierda en el la barra de herramientas superior y seleccione **Heading 2** para formatear el texto a H2.

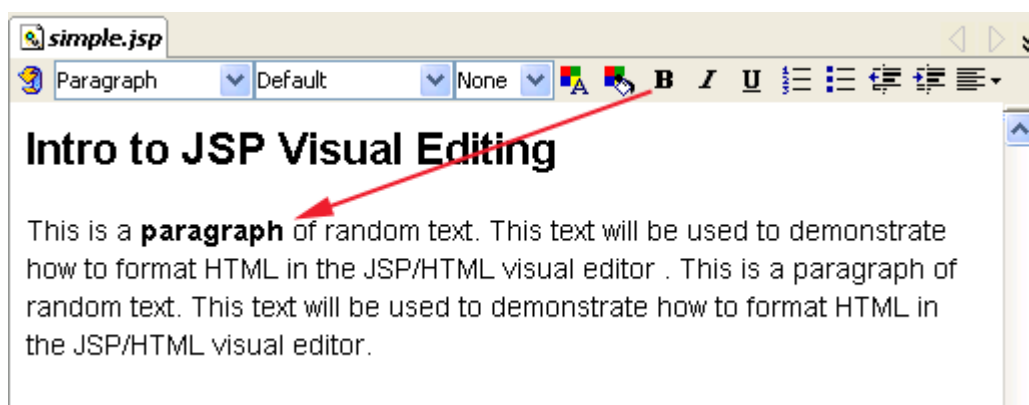


Tras la selección, el texto aparecerá con el nuevo formato H2.

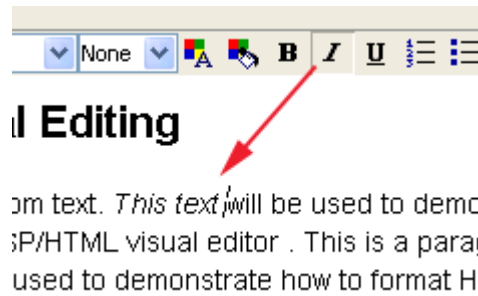
- Introduzca un retorno de carro y cambie el formato de nuevo a **Paragraph**. Introduzca un párrafo de texto aleatorio.



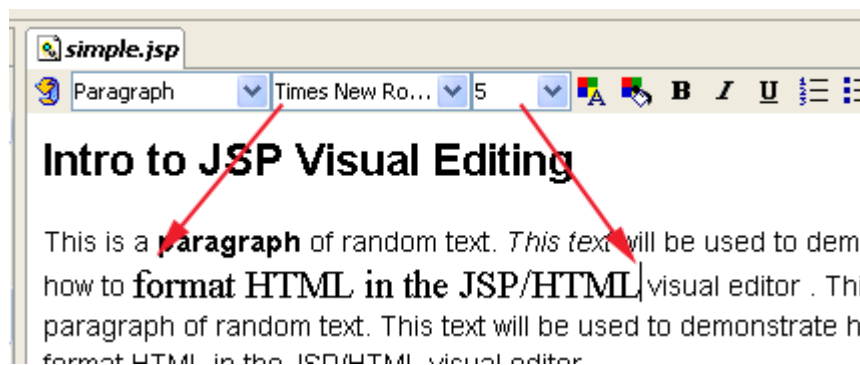
- Seleccione una porción de texto con el ratón y pulse el botón **B** de la barra superior para convertirlo a negrita.



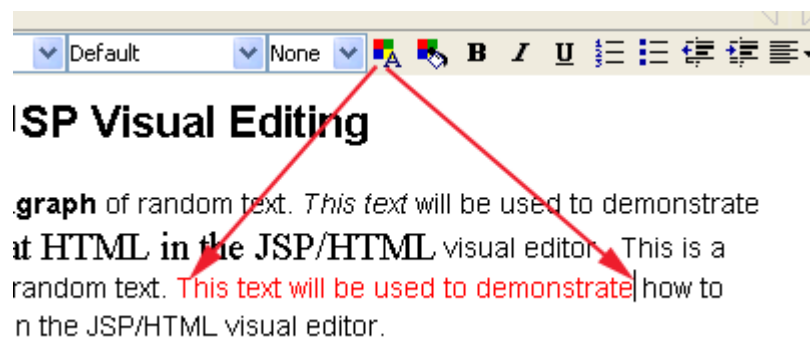
- Seleccione otra porción de texto y pulse sobre el botón **I** para ponerlo en cursiva.



6. Aplique formato a la fuente seleccionando texto y eligiendo una fuente **Times New Roman** y especificando un tamaño de **5**.

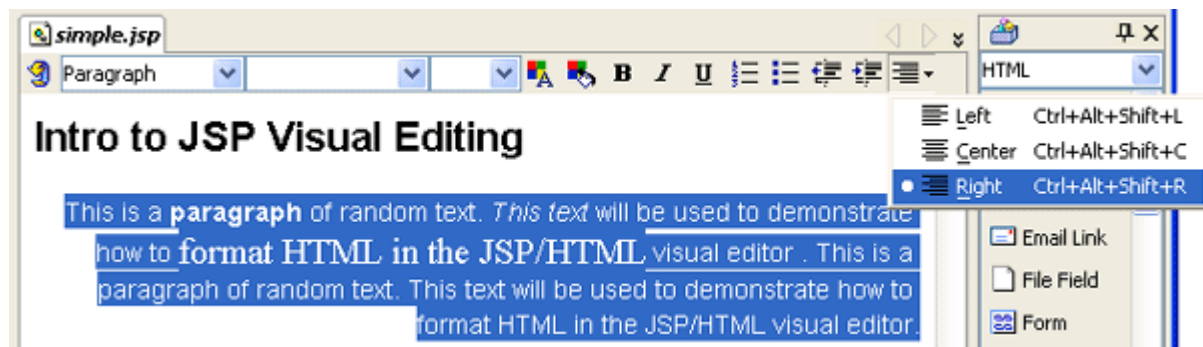


7. Ahora cambie el color del texto. Seleccione de nuevo una parte del texto y pulse sobre **Foreground Color** y seleccione el color a aplicar al texto.



8. Experimente con diferentes combinaciones de formato.

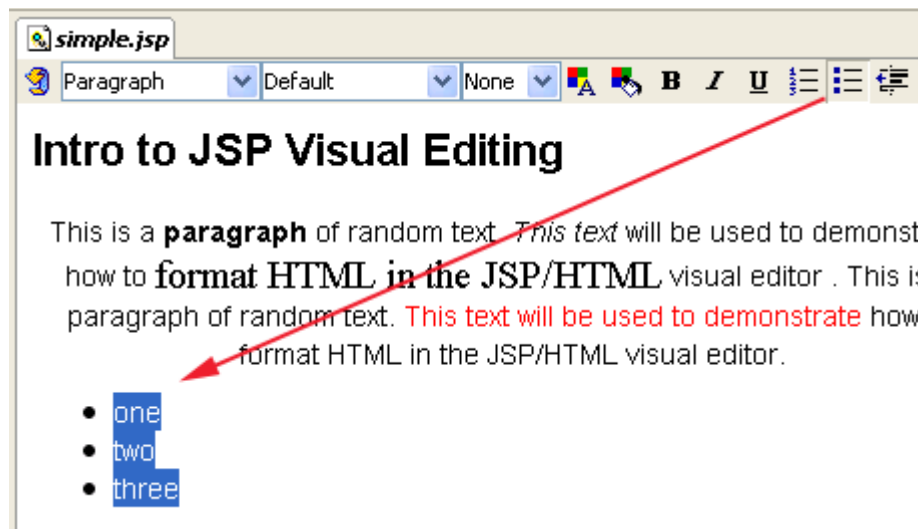
Ahora aplicaremos formato a la alineación de los párrafos. Seleccione el párrafo entero y pulse sobre el botón **Align Right** en la barra.



Seleccione otros alineamientos y observe los cambios que producen en el código fuente.

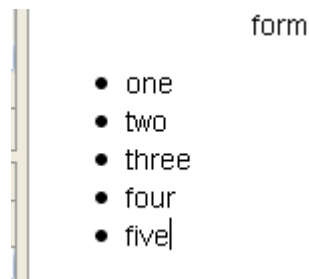
Formateando una lista

1. Bajo el párrafo que hemos editado anteriormente, introduzca texto en párrafos separados (Tras cada párrafo introduzca un retorno de carro). Seleccione los nuevos párrafos y pulse sobre el botón de lista no ordenada de la barra de herramientas.

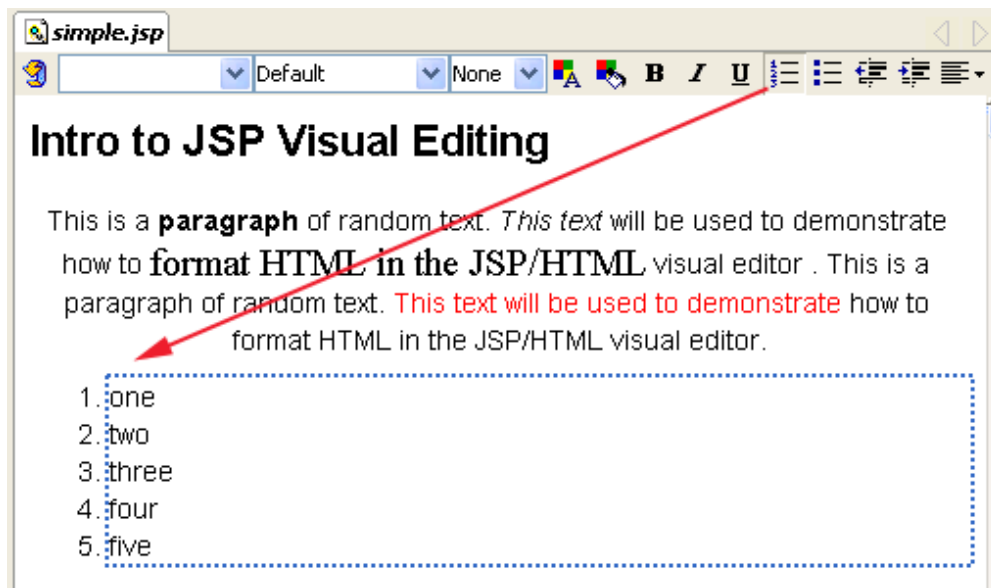


Observe los cambios del código fuente en la vista de código.

2. Introduzca nuevos elementos en la lista. Observe cómo el editor visual realiza el formateo de la lista.



3. Seleccione la lista completa en el editor visual, o en el Panel de Estructura. Pulse sobre el botón de lista ordenada de la barra de herramientas.



4. Seleccione un subconjunto de los elementos de la lista y aplique una sangría pulsando sobre el botón correspondiente de la barra de herramientas. Cambie la sublista a una lista no ordenada.



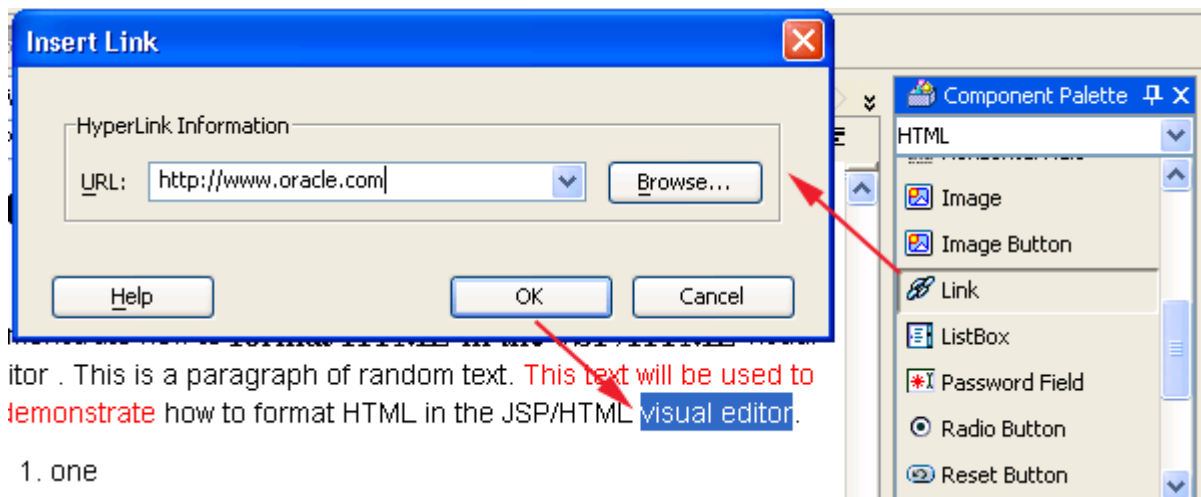
Experimente algo más aplicando y eliminando sangrías a los elementos de la lista.

Añadiendo un Enlace

1. La próxima tarea muestra como asociar un enlace (una URL de Internet) a un texto.

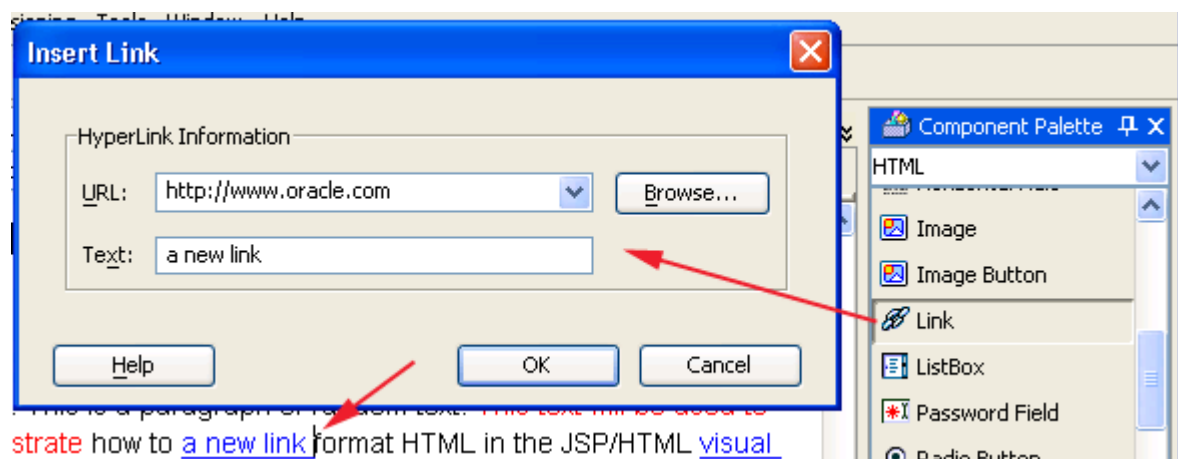
Seleccione una porción de texto o una palabra en el editor visual utilizando el ratón.

En la Paleta de Componentes (*Component Palette*), seleccione la paleta **HTML** de la lista desplegable de la parte superior. Pulse en el elemento **Link** de la paleta. Aparecerá un diálogo en el que podrá introducir la **URL**:. Introduzca una URL válida, como por ejemplo: <http://oracle.com>.




Tras pulsar **OK** para introducir la información del enlace, verá que el texto seleccionado se muestra como en enlace en el editor.

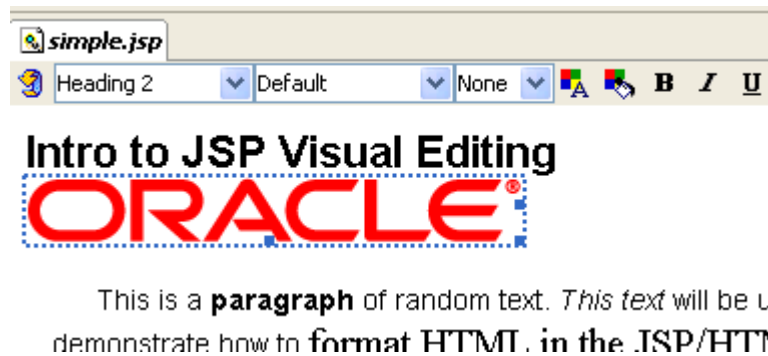
2. También es posible insertar tanto texto como un enlace asociado en su página sin necesidad de seleccionar texto previamente. La forma de hacerlo es arrastra el elemento **Link** desde la paleta de componentes a cualquier área de la página y luego introducir el **texto** y la **URL** para el enlace.



Paso 2 – Usando Imágenes

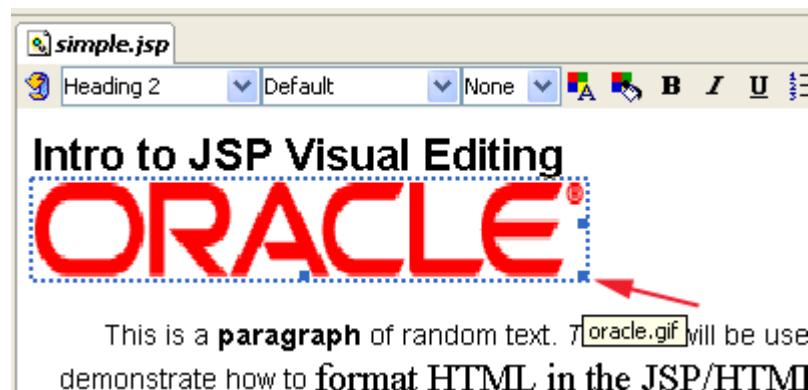
Esta parte del tutorial mostrará como trabajar con imágenes en JDeveloper 11g.

1. Busque y guarde una imagen de Oracle: . Añada una imagen a su página JSP en JDeveloper, busque en el disco la imagen descargada. Conteste afirmativamente a la pregunta de si desea guardar la imagen en el documento raíz (**Document Root**) de su aplicación.

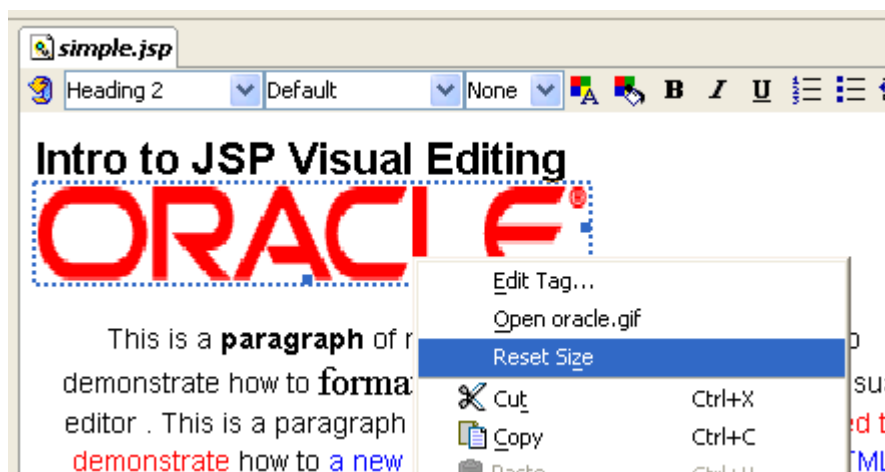


2. Es posible arrastrar y soltar imágenes directamente desde el escritorio de Windows y los navegadores.

Cambie el tamaño de la imagen de Oracle pulsando sobre los puntos de agarre en la parte inferior derecha de la imagen.

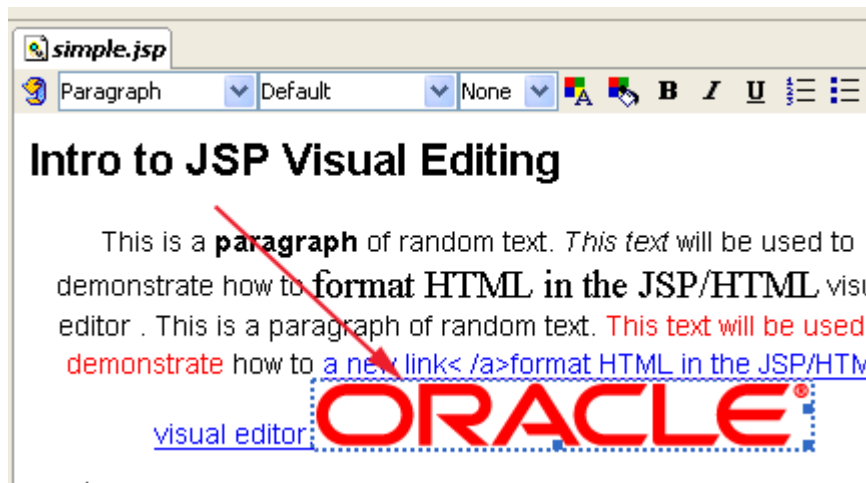


3. Devuelva su tamaño original a la imagen pulsando con el botón derecho sobre la imagen y seleccionando **Reset Size**.



Observe que también puede volver al tamaño anterior utilizando el enlace **Reset Size** en el inspector de propiedades.

4. Mueva la imagen a una posición diferente de la página, pulsándola primero para seleccionarla y arrastrándola luego a la posición deseada.



También puede probar a copiar, pegar y cortar/borrar la imagen en el editor visual.

Paso 3 - Utilizando CSS

A continuación se explica como trabajar con CSS en JDeveloper 11g.

[Aplicando un estilo CSS y personalizándolo](#)
[Añadiendo CSS a la Paleta de Componentes](#)

Breve introducción a CSS

CSS son las siglas de Cascade StyleSheet (Hoja de Estilo en Cascada). Se trata de una especificación sobre los estilos físicos aplicables a un documento HTML, y trata de dar la separación definitiva de la lógica (estructura) y el físico (presentación) del documento.

La especificación de la presentación se realiza en ficheros externos CSS que serán incluidos desde la página HTML cuyo diseño se desee modificar.

Los cambios en el diseño incluyen tamaños, fuentes y colores del texto, así como estilos de las tablas, estilos de los enlaces, etc...

La definición de los diferentes estilos se realiza siguiendo la sintaxis:

```
Etiqueta1, Etiqueta2 {propiedad1:valor;} /* Elementos con el mismo estilo */  
Etiqueta3 {propiedad1:valor;...propiedadN:valor;} /* Varias propiedades */  
.Clase1 {propiedad1:valor;...propiedadN:valor;} /* Define clases */
```

Para incluir una hoja de estilo desde una página HTML hay que incluir la siguiente línea:

```
<LINK REL="stylesheet" TYPE="text/css" HREF="URL_CSS.css">
```

Para hacer referencia a un estilo de clase creado se indica:

```
<etiqueta CLASS="Clase_Definida"> ... </etiqueta>
```

También es posible cambiar el estilo de un elemento según el estado en el que se encuentra. Esto se hace a través de las pseudoclases con la siguiente sintaxis:

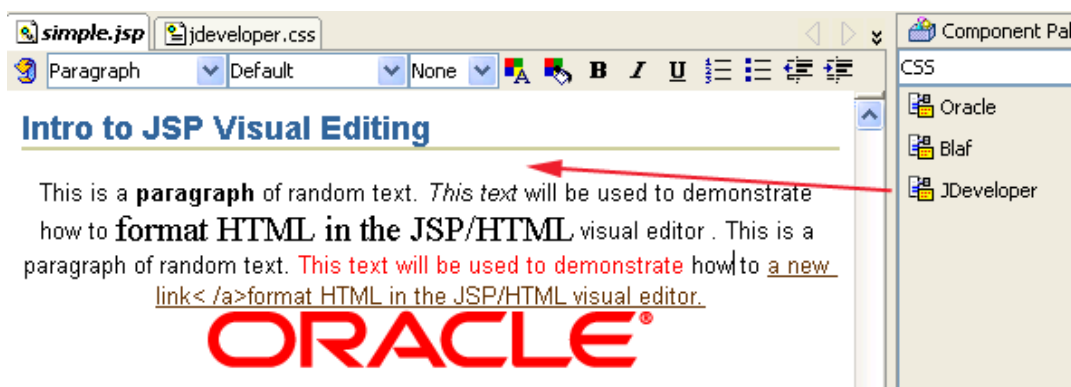
etiqueta:pseudoclase {propiedad1:valor;...;propiedadN:valor}

Algunas de las pseudoclases más importantes para la etiqueta <A> (enlaces), son:

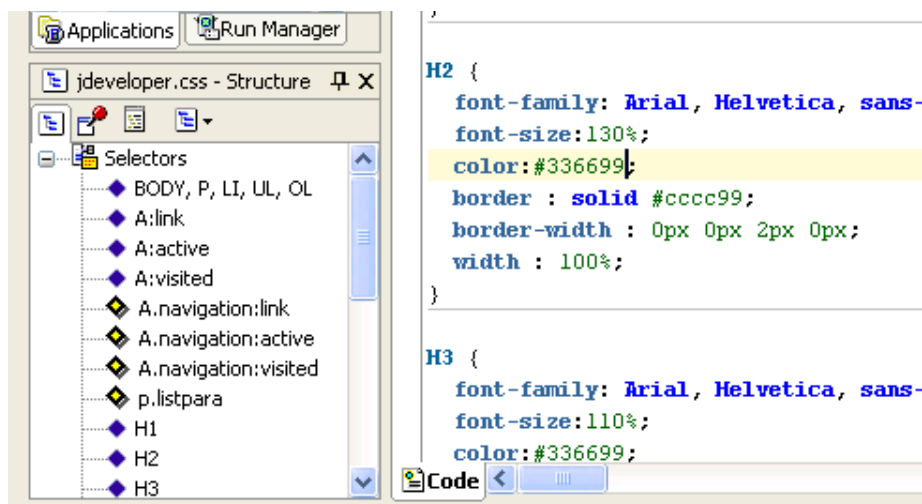
```
A:link { } /* Indica el estilo de un enlace que no ha sido visitado */
A:visited { } /* Indica el estilo de un enlace que ha sido visitado */
A:active { } /* Indica el estilo de un enlace que está siendo pulsado */
A:hover { } /* Indica el estilo de un enlace sobre el que pasa el ratón */
```

Aplicando un estilo CSS y personalizándolo.

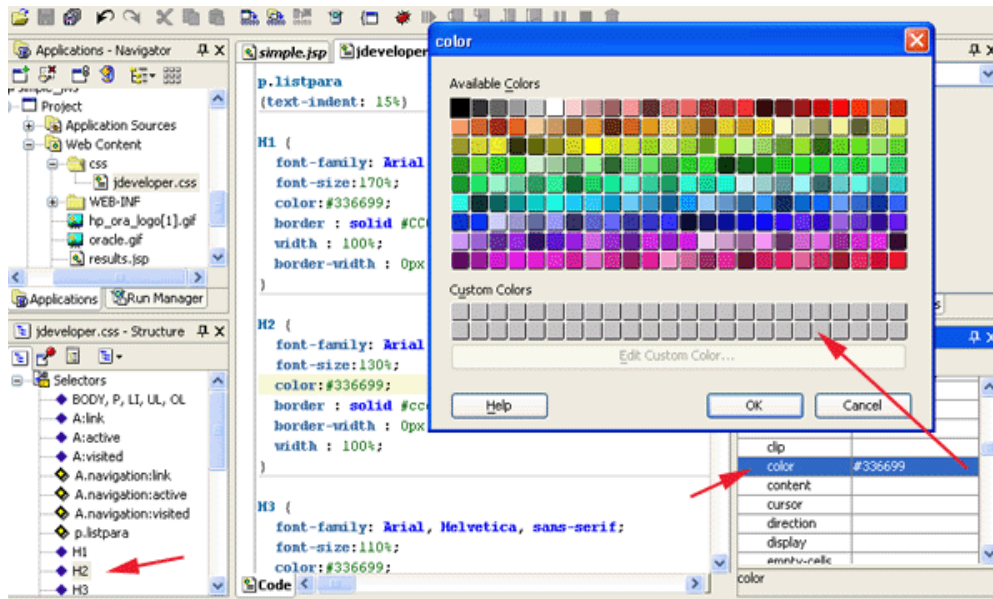
1. Añada un estilo CSS a su página JSP cambiando la Paleta de Componentes a **CSS** y arrastrando el estilo **JDeveloper** a la página. Observe como cambia la apariencia de la página. Cambie a la Vista de Código y compruebe los cambios efectuados al código.



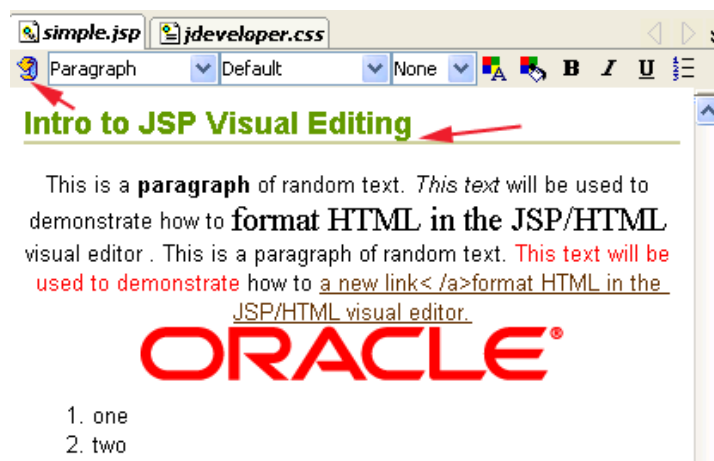
2. En el navegador, se ha añadido un nuevo fichero **css/ jdeveloper.css**. Edite este fichero haciendo doble clic sobre él. Verá el color resaltado en el editor de código y la estructura CSS en el Panel de Estructura.



3. Cambie el **color** del atributo **H2** en el fichero CSS. En el Panel de Estructura pulse sobre **H2** y actualice el valor de **color** en el Inspector de Propiedades. Use el selector de color para seleccionar un color diferente.



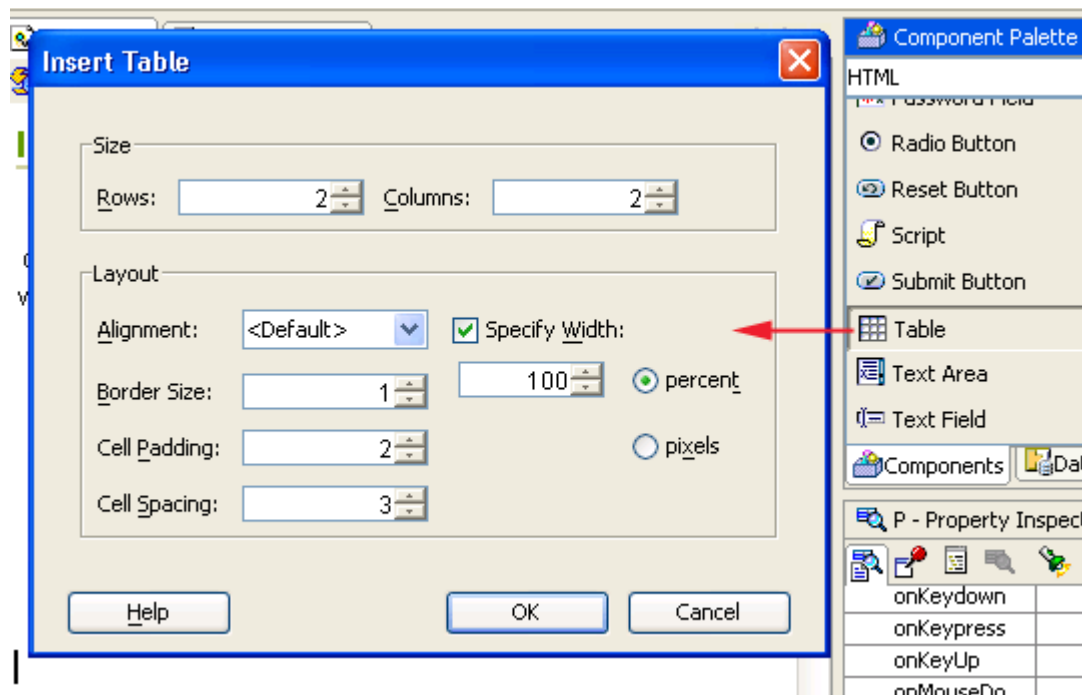
4. De vuelta al editor visual JSP pulse en el botón **refresh** para ver los cambios.



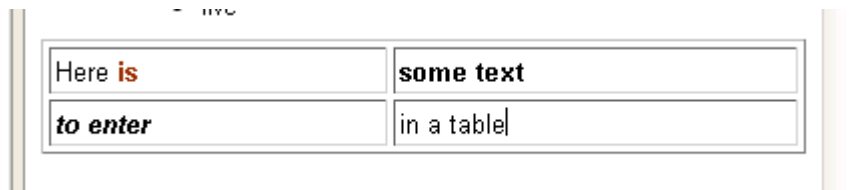
Paso 4 – Tablas HTML

Estos pasos explican como trabajar con tablas HTML en el editor visual JSP/HTML.

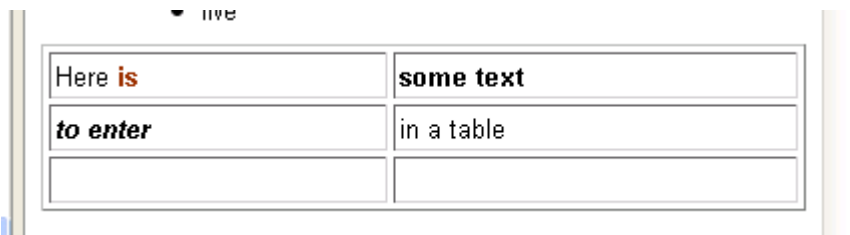
1. Añada una tabla HTML a la página JSP arrastrando el elemento **table** de la página **HTML** de la Paleta de Componentes. En el diálogo acepte la configuración por defecto y pulse **OK**.



2. Introduzca contenido HTML en las celdas.

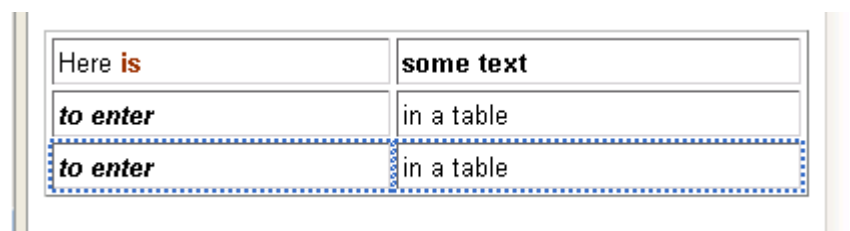


3. Añada una nueva fila pulsando sobre la celda inferior derecha y pulsando el Tabulador. Una nueva fila aparecerá al final de la tabla.

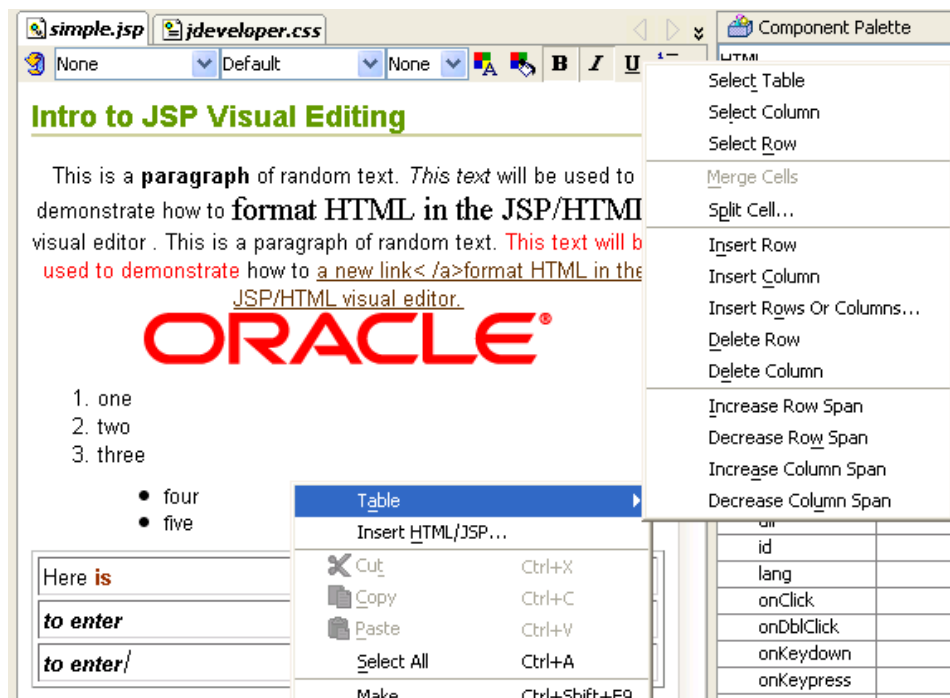


4. Copie el contenido de la segunda fila seleccionando la fila completa, pulsando Ctrl-C para copiar el contenido y luego haciendo clic en la fila vacía y pulsando Ctrl-V.

Nota: para seleccionar una fila completa, puede arrastrar el ratón a través de la fila o puede situar el cursor en la fila y pulsando el botón derecho del ratón seleccionar: **Table-> Select Row**.



- Experimente con las opciones de los menús contextuales. Inserte y borre columnas y filas. Divida y una celdas etc...

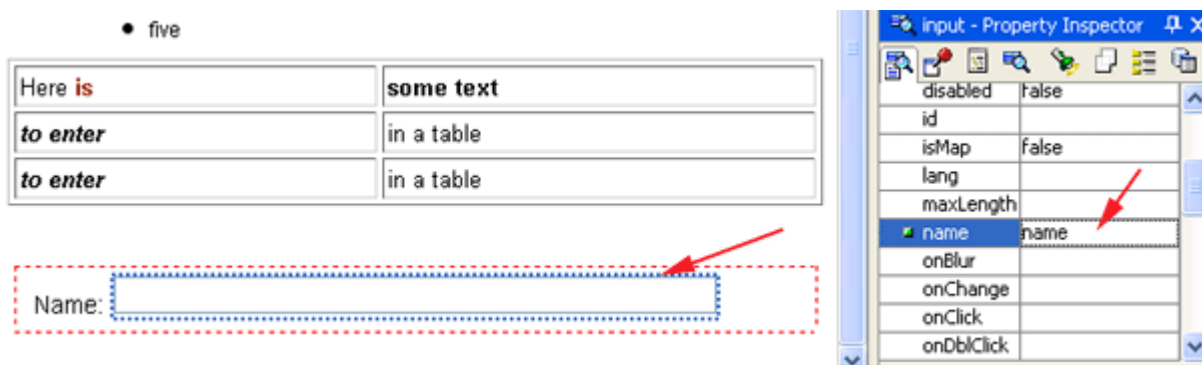


Paso 5 – Formularios HTML

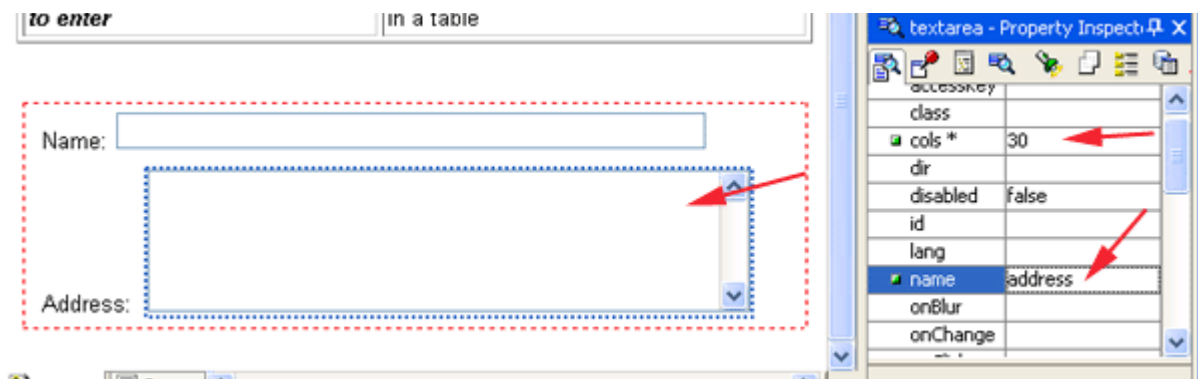
A continuación veremos como trabajar con elementos de formulario HTML.

- En la misma página, añada un campo de texto arrastrando el elemento **Text Field** de la página **HTML** de la paleta de componentes. Observe como JDeveloper ofrece añadir el elemento del formulario. Acepte para continuar. Utilizando el inspector de propiedades ponga el atributo **size** a **30** y el atributo **name** a **name**.

Comprobará que el elemento de formulario se muestra con una línea roja punteada alrededor del campo de texto. Añada una etiqueta con el nombre a la izquierda del campo añadiendo el texto, **Name:**.



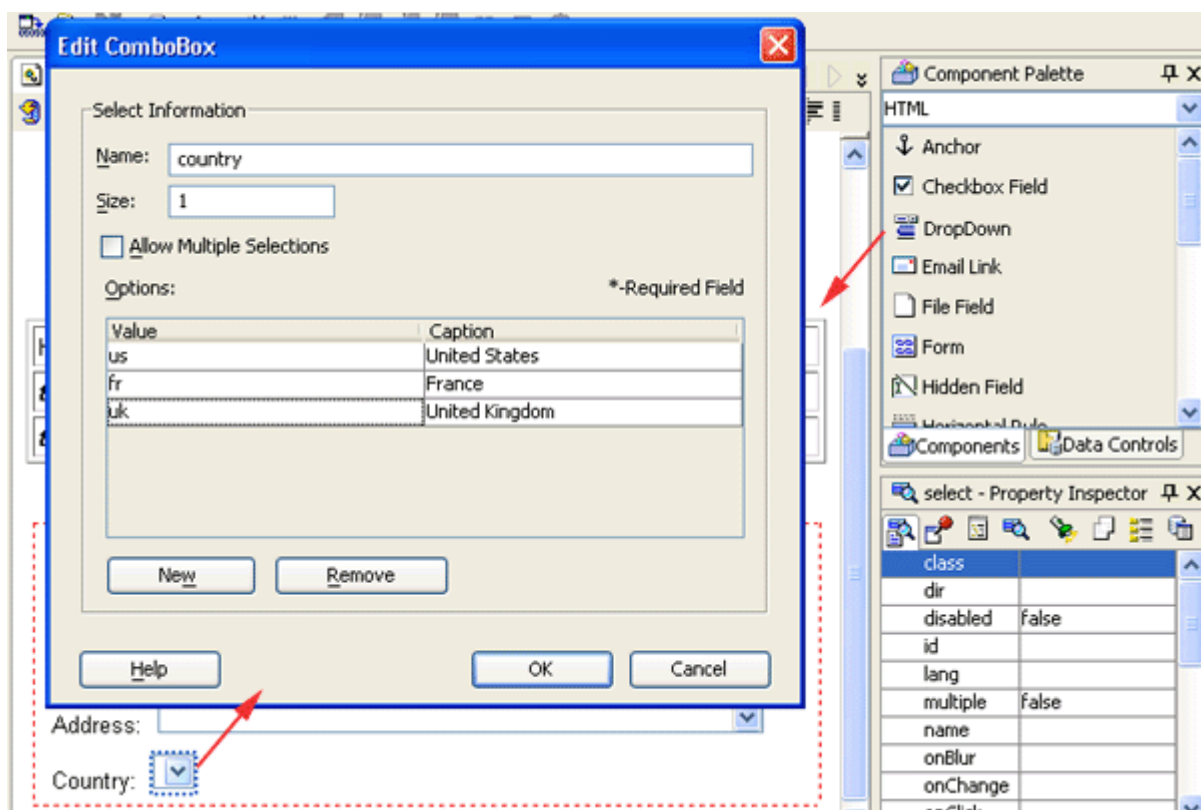
- Coloque el cursor a la derecha del campo de texto e introduzca un nuevo párrafo (retorno de carro). En la siguiente línea añada el texto **Address:** y un elemento **Text Area** de la Paleta de Componentes. Use el Inspector de Propiedades para poner el atributo **cols** a **30** y ponga **name** a **address**.



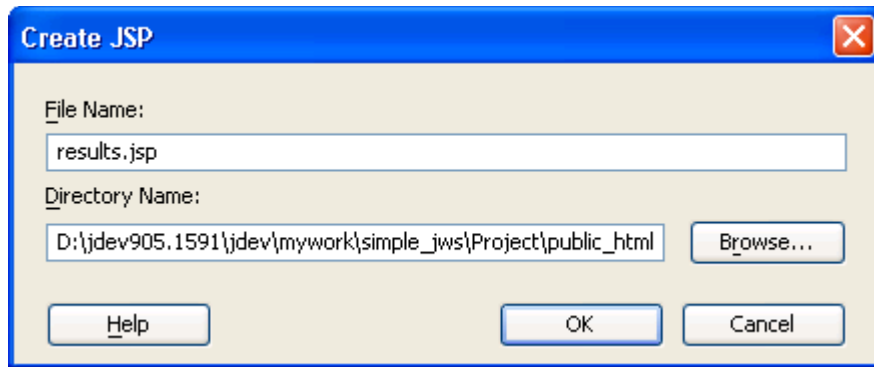
3. Introduzca una nueva línea bajo el área de texto y añada una lista de selección arrastrando el elemento **ComboBox** a la página. Una vez que aparece la lista desplegable, haga doble clic sobre ella para invocar al editor de listas (ComboBox editor).

Ponga **name** a **country** y **size** a **1**. (No active la casilla **Allow Multiple Selections**)

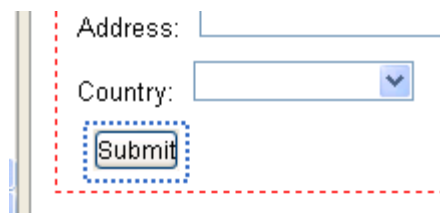
Pulse el botón **New** para añadir nuevas opciones **Options**: en la lista. Edite los campos **Value** y **Caption** indicando **us** y **United States** respectivamente. Añada otras opciones de países pulsando en el botón **New** y editando los valores y etiquetas. (Puede usar: **es** y **España**, **fr** y **France**, etc...)



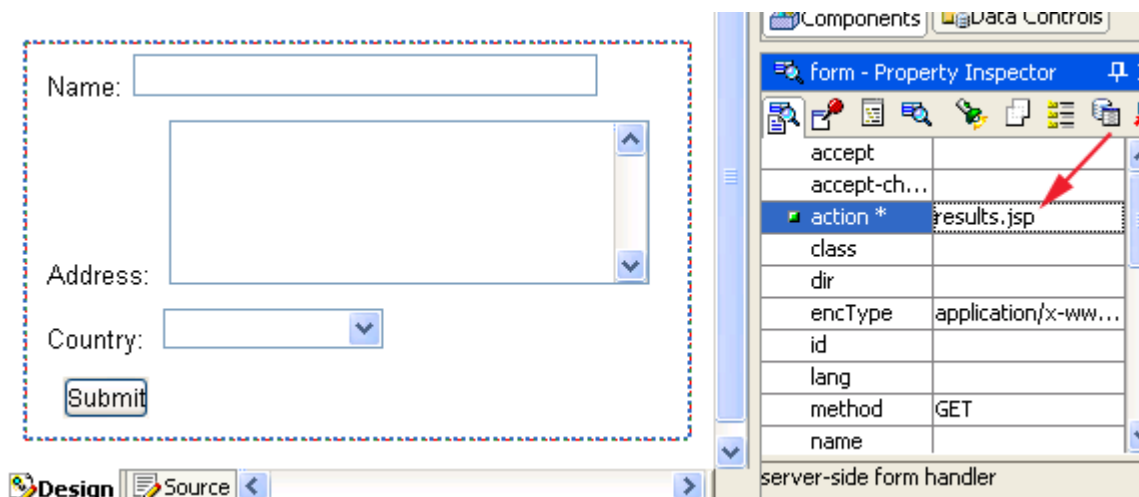
4. Antes de añadir un botón de envío y definir la acción del formulario, vamos a crear una página JSP que recogerá los datos enviados por nuestro formulario. Del mismo modo en que se hizo en el paso 1, cree una nueva página JSP **results.jsp**. Terminaremos esta página después, en el Paso 6.



- De vuelta a la página original donde se encuentra el formulario, añada un botón de envío HTML **Submit Button** y colóquelo dentro del formulario cerca de la lista de selección del país..



- Ahora rellenaremos el atributo **action** del formulario. Para seleccionar el formulario en el editor visual, pulse dentro del formulario y con el botón derecho pulsado seleccione **Form -> Select Form Tag**. Una vez el formulario ha sido seleccionado, podrá introducir el atributo **action** con valor **results.jsp** en el Inspector de Propiedades. En caso de que obtenga el valor **results.jsp** de una lista desplegable, debe eliminar el carácter "/" que se añade al nombre de acción **results.jsp**.

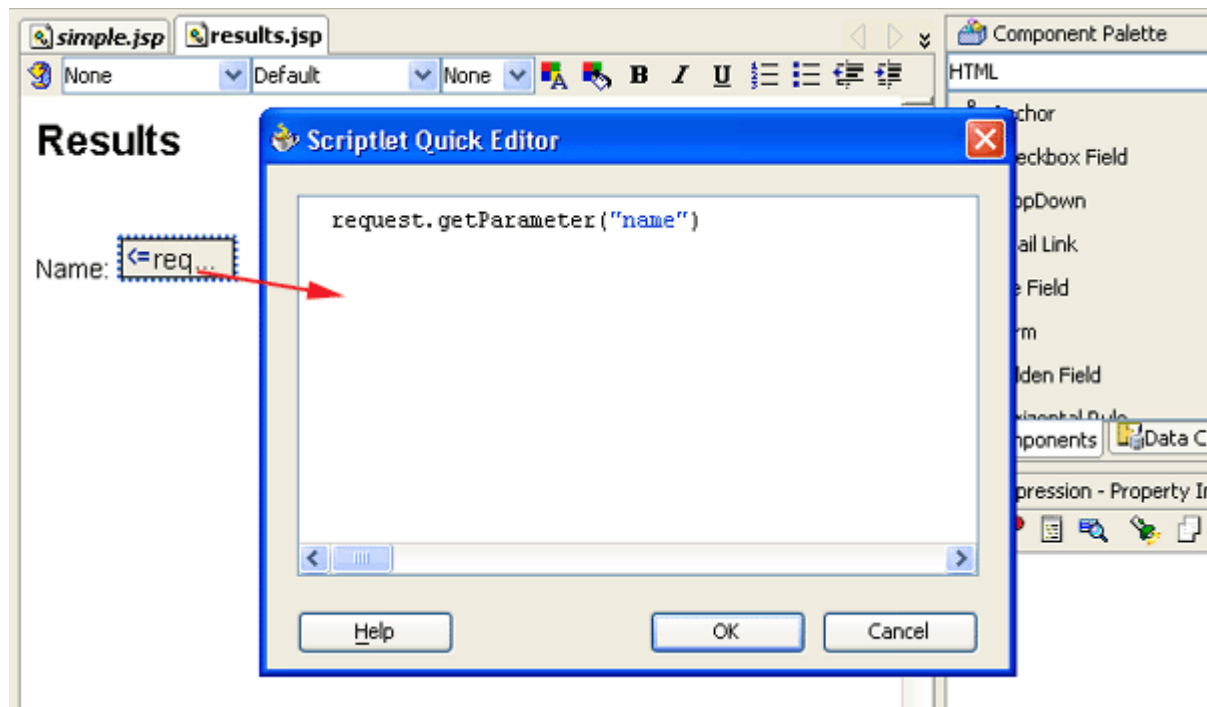


Paso 6- Etiquetas JSP

Estos pasos finales muestran como trabajar con etiquetas básicas JSP en el editor visual.

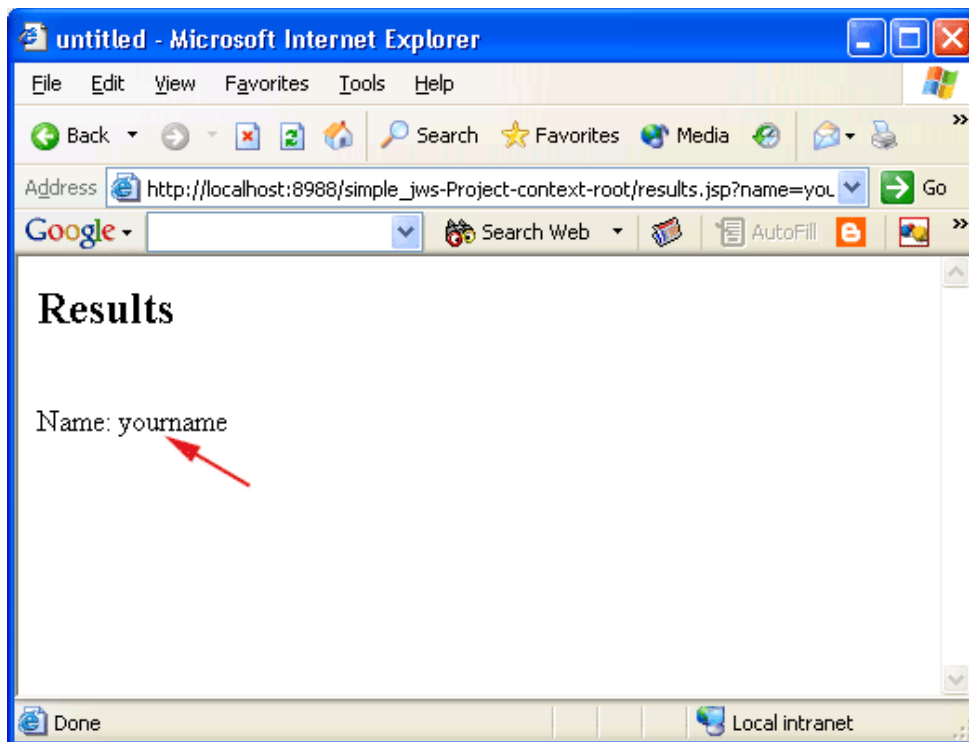
- En la página nueva **results.jsp**, añadiremos una expresión JSP para mostrar el parámetro **name** pasado al formulario por la página anterior. En la Paleta de Componentes (**Component Palette**) seleccione la página **JSP**. Arrastre a la página el elemento **Expression**. Debería ver un icono representando de forma visual la expresión JSP. Haga doble clic en el icono para editar su valor.

Introduzca: **request.getParameter("name")** en el diálogo del editor. Pulse **OK** para continuar. Antes de probar la página puede añadir texto antes de la expresión, tal como: **Name parameter is:.**
Guarde la página **results.jsp**.



En general todas las acciones estándar JSP y las etiquetas personalizadas se representan como icono en el editor visual. Una vez insertadas en una página puede utilizar el Inspector de Propiedades para editar sus atributos, o bien hacer doble clic sobre el icono.

2. Pruebe el formulario y la página de resultados ejecutando la página original **simple.jsp**, haciendo clic con el botón derecho sobre la página y seleccionando **Run**. Cuando el formulario aparezca introduzca su nombre en el campo **Name:** y pulse Enviar (**submit**). La página **results.jsp** aparecerá mostrando el valor indicado para el nombre.

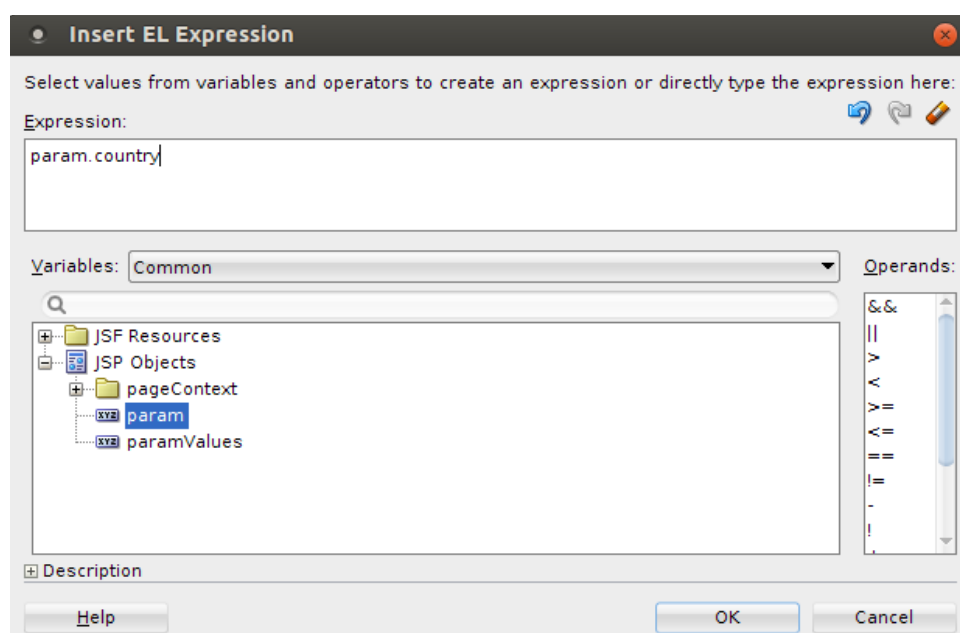


En el siguiente paso utilizaremos una librería de etiquetas en lugar de mostrar el valor de una petición.

3. En la página **results.jsp** añadimos una etiqueta predefinida (**custom tag**) en la librería estándar JSP (JavaServer Pages Standard Tag Library (JSTL)) que mostrará otros valores de los parámetros enviados por nuestro formulario.

Introduzca un salto de carro tras la expresión JSP y añada una nueva etiqueta **Country**:

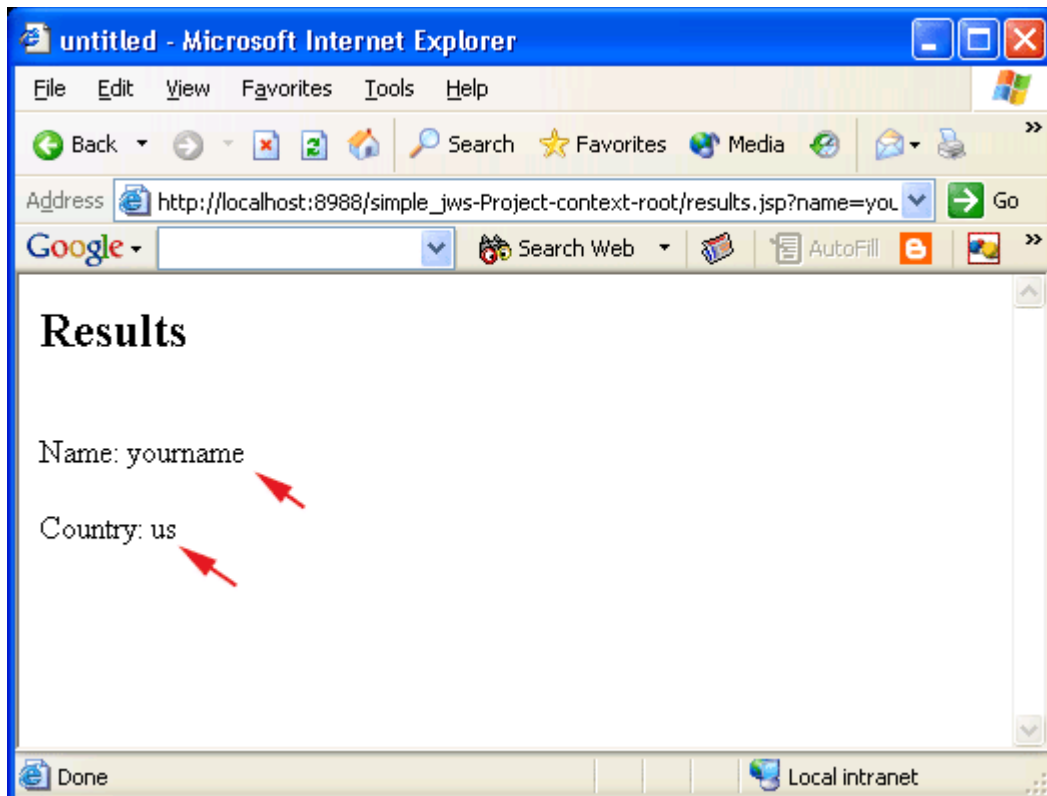
Seleccione en la Paleta de Componentes **EL Expression** y arrastrela a la página tras el texto **Country**:. Observe cómo puede seleccionar de entre los distintos objetos a los que tiene acceso la página. Pulsamos sobre **param** y completamos la expresión para acceder al parámetro del país indicando **param.country** y aceptamos. Podemos observar que el código que se genera es el siguiente: **`\${param.country}**. Cuando se muestre la página se indicará el valor del parámetro *country*.



Añada otras etiquetas y muestre el parámetro **address**.
Guarde **results.jsp**.

Expression Language (EL) permite mostrar los datos de parámetros. EL proporciona una forma sencilla y compacta de mostrar e interactuar con cualquier dato de una aplicación Web incluyendo parámetros de formularios. Los usuarios pueden introducir expresiones EL directamente en la página JSP y pueden mostrarlas en tiempo de ejecución.

Vuelva a ejecutar la página del formulario original **simple.jsp**, introduzca valores en el formulario y observe cómo la página **results.jsp** muestra los parámetros pasados.



Resumen

Se ha introducido el entorno de edición visual JSP/HTML de JDeveloper 11g a través de ejercicios básicos que cubren los diferentes aspectos de la edición de páginas HTML/JSP.

Ejercicios

1. Cree una página personal utilizando el interfaz de diseño web de JDeveloper. Incluya su nombre, fotografía, correo electrónico y una tabla con su horario para este cuatrimestre. Utilice CSS para dar formato.
2. Cree un formulario similar al que vería al registrarse en una red social. Cree una página JSP que recupere todos los datos enviados por el formulario y los muestre formateados usando CSS.
3. Revise la parte referente a sintaxis de las páginas JSP de este documento y compruebe cómo podría implementar de forma sencilla un contador de visitas en una página JSP.

4. En el ejemplo realizado en el tutorial, modifique **results.jsp** para que muestre un mensaje de error a aquellos usuarios que provengan de fuera de la comunidad europea.
5. Cree una página con un formulario que envíe el número de veces que se debe repetir una imagen. Otra página JSP debe recoger dicho número y mostrar las imágenes solicitadas.
6. Cree una página JSP que envíe un formulario indicando el tamaño de una tabla a crear (filas y columnas), otra página JSP debe recibir la información y generar la tabla.
7. Cree con JSP un sencillo sistema para realizar y guardar comentarios en una página, al estilo de un libro de firmas.
8. Tome la página personal que realizó en el primer ejercicio y añada un formulario para seleccionar el estilo CSS mediante una lista. La página debe modificar su estilo tal y como se indique en el formulario.

Enlaces de interés

Especificaciones HTML – <http://www.w3.org/MarkUp/>

Especificación JSP y otra información - http://docs.oracle.com/cd/E13222_01/wls/docs70/jsp/index.html

Bibliografía

<http://www.oracle.com/technetwork/testcontent/intro-to-visual-jsp-093381.html>
<http://otn.oracle.com>

Un tutorial de JSP
<http://geneura.ugr.es/~jmerelo/JSP/>

Manual de CSS
<http://www.webestilo.com/css/>

Este documento es una adaptación de la documentación oficial disponible de JDeveloper.
El documento presenta algunas palabras en inglés para facilitar la localización de los elementos en la interfaz JDeveloper.
La documentación original puede encontrarse en <http://www.oracle.com/technology/obe/obe9051jdev/index.htm>
