JSP by Example

Welcome to the JavaServer Pages™ technology, the cross-platform method of generating dynamic content for the Web.

If you have reached this learn-by-example trail, you are probably new to the technology. You might be a Web developer or enterprise developer who wants to use JavaServer Pages to develop dynamic Web applications. The steps in this trail contain a series of topics and sample code that teach you how to write JavaServer Pages applications. Each step illustrates a group of related principles.

We recommend that you read the *JSP Technical FAQ* first and follow the instructions for installing and configuring your JSP reference implementation. After that, start with *Tutorial 1* or jump ahead to any other tutorial, depending on what interests you.
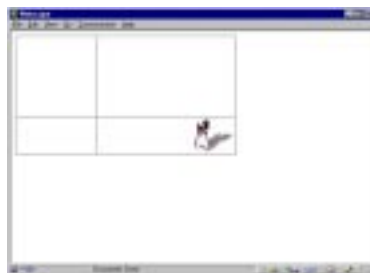
Now turn to the first page, and let's get started.

The Very Beginning

So you want to get started developing JSP applications. FIGURE 1-1 shows what is perhaps the simplest JSP application one could write. CODE EXAMPLE 1-1 and CODE EXAMPLE 1-2 list its code.

FIGURE 1-1   Duke Says Hello

CODE EXAMPLE 1-1   The Duke Banner *(dukebanner.html)*


```
<table border="0" width="400" cellspacing="0"
        cellpadding="0">

<tr>
<td height="150" width="150">   </td>
<td width="250">   </td>
</tr>

<tr>
<td width="150">   </td>
<td align="right" width="250">
    <img src="duke.waving.gif"> </td>
</tr>

</table>
<br>
```

```
<%@ page info="a hello world example" %>

<html>
<head><title>Hello, World</title></head>
<body bgcolor="#ffffff" background="background.gif">

<%@ include file="dukebanner.html" %>

<table>
<tr>
<td width=150>   </td>
<td width=250 align=right> <h1>Hello, World!</h1> </td>
</tr>
</table>

</body>
</html>
```

The Page Directive

The `page` directive is a JSP tag that you will use in almost every JSP source file you write. In *helloworld.jsp*, it's the line that looks like this:

```
<%@ page info="a hello world example" %>
```

The `page` directive gives instructions to the JSP engine that apply to the entire JSP source file. In this example, `page` specifies an informative comment that will become part of the compiled JSP file. In other cases, `page` might specify the scripting language used in the JSP source file, packages the source file would import, or the error page called if an error or exception occurs.

You can use the `page` directive anywhere in the JSP file, but it's good coding style to place it at the top of the file. Because it's a JSP tag, you can even place it before the opening `<html>` tag.

The Include Directive

The `include` directive inserts the contents of another file in the main JSP file, where the directive is located. It's useful for including copyright information, scripting language files, or anything you might want to reuse in other applications. In this example, the included file is an HTML table that creates a graphic banner.

You can see the content of the included file by viewing the page source of the main JSP file while you are running *Hello, World*. The included file does not contain `<html>` or `<body>` tags, because these tags would conflict with the same tags in the calling JSP file.

A Note About the JSP Tags

As you use the examples in this chapter, remember that the JSP tags are case sensitive. If, for example, you type `<jsp:usebean>` instead of `<jsp:useBean>`, your tag will not be recognized, and the JSP 1.0 reference implementation will throw an exception. Some of the attributes on the tags take class names, package names, pathnames or other case-sensitive values as well.

If you have any doubts about the correct spelling or syntax of any JSP tag, see the *JavaServer Pages Syntax Card.*

How To Run the Example

The instructions given here use a UNIX-style pathname. If you are working on Windows, use the same pathname with the proper separator.

1 Create the directory (or folder) `../jswdk-1.0/examples/jsp/tutorial/helloworld`.

2 Place the following files in the `../tutorial/hello` directory (or folder): `background.gif`, `duke.waving.gif`, `dukebanner.html`, and `helloworld.jsp`.

3 From the command line, start the Sun JSP reference implementation:
```
cd ../jswdk-1.0
startserver
```

4 Open a Web browser and go to
`http://yourMachineName:8080/examples/jsp/tutorial/helloworld/helloworld.jsp`