

JSP - THE PAGE DIRECTIVE

http://www.tutorialspoint.com/jsp/page_directive.htm

Copyright © tutorialspoint.com

The **page** directive is used to provide instructions to the container that pertain to the current JSP page. You may code page directives anywhere in your JSP page. By convention, page directives are coded at the top of the JSP page.

Following is the basic syntax of page directive:

```
<%@ page attribute="value" %>
```

You can write XML equivalent of the above syntax as follows:

```
<jsp:directive.page attribute="value" />
```

Attributes:

Following is the list of attributes associated with page directive:

Attribute	Purpose
buffer	Specifies a buffering model for the output stream.
autoFlush	Controls the behavior of the servlet output buffer.
contentType	Defines the character encoding scheme.
errorPage	Defines the URL of another JSP that reports on Java unchecked runtime exceptions.
isErrorPage	Indicates if this JSP page is a URL specified by another JSP page's errorPage attribute.
extends	Specifies a superclass that the generated servlet must extend
import	Specifies a list of packages or classes for use in the JSP as the Java import statement does for Java classes.
info	Defines a string that can be accessed with the servlet's <code>getServletInfo()</code> method.
isThreadSafe	Defines the threading model for the generated servlet.
language	Defines the programming language used in the JSP page.
session	Specifies whether or not the JSP page participates in HTTP sessions
isELIgnored	Specifies whether or not EL expression within the JSP page will be ignored.
isScriptingEnabled	Determines if scripting elements are allowed for use.

The buffer Attribute:

The **buffer** attribute specifies buffering characteristics for the server output response object.

You may code a value of "none" to specify no buffering so that all servlet output is immediately directed to the response object or you may code a maximum buffer size in kilobytes, which directs the servlet to write to the buffer before writing to the response object.

To direct the servlet to write output directly to the response output object, use the following:

```
<%@ page buffer="none" %>
```

Use the following to direct the servlet to write output to a buffer of size not less than 8 kilobytes:

```
<%@ page buffer="8kb" %>
```

The autoFlush Attribute:

The **autoFlush** attribute specifies whether buffered output should be flushed automatically when the buffer is filled, or whether an exception should be raised to indicate buffer overflow.

A value of true (default) indicates automatic buffer flushing and a value of false throws an exception.

The following directive causes the servlet to throw an exception when the servlet's output buffer is full:

```
<%@ page autoFlush="false" %>
```

This directive causes the servlet to flush the output buffer when full:

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

Usually, the buffer and autoFlush attributes are coded on a single page directive as follows:

```
<%@ page buffer="16kb" autoFlush="true" %>
```

The contentType Attribute:

The contentType attribute sets the character encoding for the JSP page and for the generated response page. The default content type is text/html, which is the standard content type for HTML pages.

If you want to write out XML from your JSP, use the following page directive:

```
<%@ page contentType="text/xml" %>
```

The following statement directs the browser to render the generated page as HTML:

```
<%@ page contentType="text/html" %>
```

The following directive sets the content type as a Microsoft Word document:

```
<%@ page contentType="application/msword" %>
```

You can also specify the character encoding for the response. For example, if you wanted to specify that the resulting page that is returned to the browser uses ISO Latin 1, you would use the following page directive:

```
<%@ page contentType="text/html:charset=ISO-8859-1" %>
```

The errorPage Attribute:

The errorPage attribute tells the JSP engine which page to display if there is an error while the current page runs. The value of the errorPage attribute is a relative URL.

The following directive displays MyErrorPage.jsp when all uncaught exceptions are thrown:

```
<%@ page errorPage="MyErrorPage.jsp" %>
```

The isErrorPage Attribute:

The `isErrorPage` attribute indicates that the current JSP can be used as the error page for another JSP.

The value of `isErrorPage` is either `true` or `false`. The default value of the `isErrorPage` attribute is `false`.

For example, the `handleError.jsp` sets the `isErrorPage` option to `true` because it is supposed to handle errors:

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

The extends Attribute:

The `extends` attribute specifies a superclass that the generated servlet must extend.

For example, the following directive directs the JSP translator to generate the servlet such that the servlet extends *somePackage.SomeClass*:

```
<%@ page extends="somePackage.SomeClass" %>
```

The import Attribute:

The `import` attribute serves the same function as, and behaves like, the Java `import` statement. The value for the `import` option is the name of the package you want to import.

To import `java.sql.*`, use the following page directive:

```
<%@ page import="java.sql.*" %>
```

To import multiple packages you can specify them separated by comma as follows:

```
<%@ page import="java.sql.*,java.util.*" %>
```

By default, a container automatically imports `java.lang.*`, `javax.servlet.*`, `javax.servlet.jsp.*`, and `javax.servlet.http.*`.

The info Attribute:

The `info` attribute lets you provide a description of the JSP. The following is a coding example:

```
<%@ page info="This JSP Page Written By ZARA" %>
```

The isThreadSafe Attribute:

The `isThreadSafe` option marks a page as being thread-safe. By default, all JSPs are considered thread-safe. If you set the `isThreadSafe` option to `false`, the JSP engine makes sure that only one thread at a time is executing your JSP.

The following page directive sets the `isThreadSafe` option to `false`:

```
<%@ page isThreadSafe="false" %>
```

The language Attribute:

The `language` attribute indicates the programming language used in scripting the JSP page.

For example, because you usually use Java as the scripting language, your `language` option looks like this:

```
<%@ page language="java" %>
```

The session Attribute:

The `session` attribute indicates whether or not the JSP page uses HTTP sessions. A value of `true` means that the JSP page has access to a builtin **session** object and a value of `false` means that the JSP page cannot access the builtin session object.

Following directive allows the JSP page to use any of the builtin object session methods such as `session.getCreationTime()` or `session.getLastAccessTime()`:

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

The isELIgnored Attribute:

The `isELIgnored` option gives you the ability to disable the evaluation of Expression Language (EL) expressions which has been introduced in JSP 2.0

The default value of the attribute is `true`, meaning that expressions, `${...}`, are evaluated as dictated by the JSP specification. If the attribute is set to `false`, then expressions are not evaluated but rather treated as static text.

Following directive set an expressions not to be evaluated:

```
<%@ page isELIgnored="false" %>
```

The isScriptingEnabled Attribute:

The `isScriptingEnabled` attribute determines if scripting elements are allowed for use.

The default value (`true`) enables scriptlets, expressions, and declarations. If the attribute's value is set to `false`, a translation-time error will be raised if the JSP uses any scriptlets, expressions (non-EL), or declarations.

You can set this value to `false` if you want to restrict usage of scriptlets, expressions (non-EL), or declarations:

```
<%@ page isScriptingEnabled="false" %>
```