



# JSP クイックリファレンス カード

## 基本的な構文

### 規定のスクリプト記述言語

JSP ページの既定のスクリプト記述言語は Java です。JSP ページで JavaScript が使用可能になるよう設定するには、次の行を挿入します。

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

### 空白文字の使用

テンプレート コードに含まれる空白文字は、JSP に入力されているとおりにクライアントに返されます。

### 属性値の引用

JSP 要素に対する属性値はすべて、一重引用符、または二重引用符を使って、引用する必要があります。以下に例を示します。

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

### JSP に関するコメントの書き込み

JSP コメントが、JSP ページの出力の一部として、クライアントに出力されることはありません。

```
<!-- コメント文字列 ... -->
```

### クライアントへのコメント出力

HTML コメントがクライアントに出力されます。

```
<!-- コメント -->
```

## ディレクティブ

### page

ページ全体にわたる属性を定義します。

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

属性とその既定値は、次のとおりです。

```
attribute = language="java" | session="true"
| contentType=text/html;charset="ISO-8859-1"
| import="package(s)" | buffer="8kb"
| autoflush="true" | isThreadSafe="true"
| info="text_string" | errorPage="relativeURL"
| isErrorpage="true" | extends="class_name"
```

*value* = 一重引用符、または二重引用符で囲まれた文字列リテラル。

### include

JSP ページにテキストを挿入します。

```
<%@ include file = "path" ... %>
```

### taglib

JSP ページによって使用されるカスタム タグ ライブラリを定義します。

```
<%@ taglib uri="tagLibraryURI"
prefix="tagPrefix" %>
```

taglib ディレクティブの後に次の構文を使用して、カスタム タグを参照します。

```
<tagPrefix:tagName>
```

```
...
```

```
</tagPrefix:tagName>
```

## スクリプト記述要素

### 宣言

変数など、ページ全体で使用される定義を行います。

```
<%! declaration %>
```

例 :

```
<%! private String foo = null;
public String getFoo() {return this.foo;} %>
```

### スクリプトレット

スクリプト コードのブロックが 1 つ含まれています。JSP ページには、スクリプト コードのブロックを複数含めることができます。

```
<% script code %>
```

例 :

```
<% String greeting =
request.getParameter("Greeting");
out.println(greeting); %>
```

### 式

ページ出力をクライアントへ送信する前にサーバーで評価されるステートメントを定義します。

```
<%= expression %>
```

例 :

```
<%= myVar1 %>
```

## アクション

### jsp:include

ある JSP ページを別の JSP ページから呼び出します。アクションが完了すると、呼び出し先のページから呼び出し元のページに制御が戻されます。

```
<jsp:include page="path" flush="true"/>
```

```
<jsp:include page="path" flush="true">
<jsp:param name="paramName"
value="paramValue" /> ...
</jsp:include>
```

### jsp:forward

ある JSP ページから別の JSP ページを呼び出します。この呼び出しにより、呼び出し元ページの実行が終了されます。

```
<jsp:forward page="path" />
```

```
<jsp:forward page="path">
```

```
<jsp:param name="paramName"
value="paramValue" /> ...
</jsp:forward>
```

## jsp:plugin

クライアントブラウザでアプレットを起動できるようになります。

```
<jsp:plugin
type="bean|applet"
code="objectCode"
codebase="objectCodebase"
{ align="alignment" }
{ archive="archiveList" }
{ height="height" }
{ hspace="hspace" }
{ jreversion="jreversion" }
{ name="componentName" }
{ vspace="vspace" }
{ width="width" }
{ nspluginurl="url" }
{ iepluginurl="url" } >
{ <jsp:params>
{ <jsp:param name="paramName"
value="paramValue" /> }+
</jsp:params> }
{ <jsp:fallback> arbitrary_text
</jsp:fallback> } >
</jsp:plugin>
```

角かっこ ({} ) で囲まれた要素はオプションです。

## jsp:useBean

Java bean のインスタンスを定義します。

```
<jsp:useBean id="name"
scope="page|request|session|application"
typeSpec />

<jsp:useBean id="name"
scope="page|request|session|application"
typeSpec >
body
</jsp:useBean>
```

*typespec* には次のいずれか 1 つが入ります。

```
class="className" |
class="className" type="typeName" |
beanName="beanName" type="typeName" |
type="typeName"
```

## jsp:setProperty

bean にある 1 つ以上のプロパティ値を設定します。

```
<jsp:setProperty name="beanName" prop_expr />

prop_expr は次のいずれかのフォームになります。

property="*" |
property="propertyName" |
property="propertyName" param="parameterName" |
property="propertyName" value="propertyValue"
```

## jsp:getProperty

bean プロパティの値を文字列として、out オブジェクトに書き込みます。

```
<jsp:getProperty name="name"
property="propertyName" />
```

## JSP オブジェクト

オブジェクトに使用可能なメソッドについては、対応する Java オブジェクトタイプを参照してください。

### application

サーブレット設定オブジェクトから取得したサーブレット コンテキスト。

**Java type:** javax.servlet.ServletContext

### config

JSP ページの ServletConfig オブジェクト。

**Java type:** javax.servlet.ServletConfig

### exception

エラー ページが表示される原因となる未処理の例外。

**Java type:** java.lang.Throwable

### out

JSP ページの出力ストリームに書き込みを行うオブジェクト。

**Java type:** javax.servlet.jsp.JspWriter

## pageContext

JSP に対するページ コンテキスト。

**Java type:** javax.servlet.jsp.PageContext

## request

クライアント要求。

**Java type:** javax.servlet.HttpServletRequest

## response

クライアントへの応答。

**Java type:** javax.servlet.HttpServletResponse

## session

要求元クライアントのために作成されたセッション オブジェクト。

**Java type:** javax.servlet.http.HttpSession

## コンタクト先

### Allaire の Web サイト

Allaire の主な Web サイトは、次のとおりです。

[www.allaire.com](http://www.allaire.com)

JRun 開発者センター

[www.allaire.com/developer/jrunreferencedesk/](http://www.allaire.com/developer/jrunreferencedesk/)

JRun 開発者フォーラム

[forums.allaire.com/jrunconf](http://forums.allaire.com/jrunconf)

### (株) アイ・ティ・フロンティア シリウス事業部

JRun のページ

[cfusion.sirius.co.jp/jrun/](http://cfusion.sirius.co.jp/jrun/)

JRun は Allaire Corporation の商標です。その他の商標は、各社の商標です。  
© 2001 Allaire Corporation. All rights reserved.  
部品番号 :AA-JJQRF-RK