

付録 A JSP の作成

A.1 JSP とは

Servlet では、Java のプログラムの中に文字列定数などの形で HTML が埋め込まれているのに対し、JSP では HTML の中に Java のプログラムが埋め込まれるような形になっている。(ここでは、話を簡単にするために出力形式を HTML と仮定しているが、他のテキスト形式を出力する JSP も可能である。)

JSP は、自動的に Servlet に変換されて実行されるので、できることは Servlet と違いはない。出力すべき HTML が多くて、プログラム部分が少ない場合は、JSP の方が便利というだけのことである。

ファイル MyDate.jsp

```

1 <%@ page contentType="text/html;_charset=Windows-31J"
2     import="java.util.Calendar" %>
3 <%! String[] youbi = {"日", "月", "火", "水", "木", "金", "土"}; %>
4 <% Calendar cal = Calendar.getInstance(); %>
5 <html><head></head><body>
6     <%= cal.get(Calendar.YEAR) %>年
7     <%= cal.get(Calendar.MONTH)+1 %>月
8     <%= cal.get(Calendar.DAY_OF_MONTH) %>日
9     <%= youbi[cal.get(Calendar.DAY_OF_WEEK)] %>曜日
10    <%= cal.get(Calendar.HOUR_OF_DAY) %>時
11    <%= cal.get(Calendar.MINUTE) %>分
12    <%= cal.get(Calendar.SECOND) %>秒
13 </body></html>

```

この例では、Servlet の MyDate.java とほぼ同じ（空白の入り方だけが違う）ページを出力する。

A.2 JSP の主なタグ

ここでは JSP のよく使われるタグをいくつか紹介する。

ページディレクティブ: `<%@ page ~ %>` `<%@ page` と `%>` の間にはページディレクティブという JSP ページ全体に影響する属性を指定する。

まず、`contentType` という属性を指定するのが一般的である。これは、文字通り出力する内容のタイプである。(静的なコンテンツなら `.html`, `.txt` のようなファイルの拡張子で `contentType` を推測することが可能だが、Servlet や JSP はこのように明示的に指定する必要がある。)

また、import という属性でインポートする Java のクラスあるいは、java.util.* のような形式でパッケージを指定する。クラスやパッケージが複数ある場合はコンマで区切って並べる。

スクリプトレット: <% ~ %> <% と %>の間には、文や変数の宣言など、Java のメソッドの中にかくことができる任意のコードの断片を書くことができる。

ここに書くものは文法的に完全な文である必要はない。例えば次のような書き方も可能である。

ファイル ScriptletTest.jsp

```
1 <%@ page language="java"
2     contentType="text/html;_charset=windows-31j" %>
3 <html>
4 <head>
5 <meta http-equiv="Content-Type"
6     content="text/html;_charset=windows-31j">
7 <title>スクリプトレットテスト</title>
8 </head>
9 <body>
10 <h1>スクリプトレットテスト</h1>
11 <%
12     double r = Math.random();
13     if ( r < 0.5 ) {
14 %>
15         <span style='color:red'>小吉</span>
16 <% } else { %>
17         <span style='color:green'>大吉</span>
18 <% } %>
19 </body>
20 </html>
```

また、ここで宣言された変数は、Servlet のメソッド内の局所変数に翻訳される。つまり JSP ページが表示されるたびに初期化される。

式: <%= ~ %> <%=と%>の間には Java の式を書くことができる。String 型に変換されて出力中に挿入される。

宣言: <%! ~ %> <%! と %>の間には変数やメソッドの宣言を書くことができる。特に、ここで宣言された変数は Servlet のフィールド (インスタンス変数) に翻訳される。つまり、Tomcat などの Servlet コンテナが起動している間は (timeout しない限り) ずっと値を保持し続ける変数になる。

コメント: <%-- ~ --%> <%--と--%>の間はコメントである。この間の文字は出力に現れない。

ファイル CommentTest.jsp

```

1 <%@ contentType="text/html; charset=windows-31j" %>
2 <html>
3 <head><title>コメントのテスト</title></head>
4 <body>
5 <h2>コメントのテスト</h2>
6 <!-- このコメントは出力に含まれません。 --%>
7 <!-- このコメントはどうでしょう。 -->
8 </body>
9 </html>

```

インクルードディレクティブ: `<%@ include ~ %>` `<%@ include file = " ~ " %>`という形で他のファイルを読み込む。file 属性の値 (~ の部分) は相対 URL である。

A.3 暗黙オブジェクト

式やスクリプトレットを簡単にするために、8 つの変数が前もって定義されている。そのうち、よく使われると思われる 6 つを次の表で紹介する。

変数名	クラス	備考 (Servlet との対応)
request	HttpServletRequest	doGet や doPost の第 1 引数に相当
response	HttpServletResponse	doGet や doPost の第 2 引数に相当
out	PrintWriter	response.getWriter() に相当
session	HttpSession	request.getSession() に相当
config	ServletConfig	getServletConfig() に相当
application	ServletContext	getServletContext() に相当

この中のクラスや備考は厳密な説明ではない。例えば out は本当は JspWriter というクラスのオブジェクトで、PrintWriter とは若干の違いがある。ほとんどの場合 PrintWriter クラスに相当すると考えて良い、ということである。

A.4 フォワードとリダイレクト

フォワードとは、Servlet (あるいは JSP) から、別の Servlet や JSP にリクエストを転送する処理である。これにより Servlet がリクエストの前処理を行って、他の Servlet や JSP が表示を行うなどの分業が可能になる。

Servlet から他の JSP や Servlet にフォワード (転送) するには、まず HttpServletRequest から getRequestDispatcher というメソッドの引数にフォワード先のパスを指定して RequestDispatcher オブジェクトを取り出す。このパスは / から始まっている場合は、現在の Web アプリケーションのコンテキストルートからの相対パスと解釈される。そうでない場合は、Servlet の URL から

の相対パスである。次に RequestDispatcher オブジェクトに対して request と response を引数として forward メソッドを呼び出す。

フォワードに似た概念にリダイレクトがある。リダイレクトは、いったんブラウザ側に転送先の URL を送り、ブラウザがこの URL に改めてリクエストを送る、という違いがある。HttpServletResponse クラスの sendRedirect というメソッドを用いる。sendRedirect の引数は、転送先の URL である。

```
response.sendRedirect(URLString);
```

なお、詳しく説明しないが、同じ HttpServletResponse クラスの encodeRedirectURL メソッドと併用する場合もある。

```
response.sendRedirect(response.encodeRedirectURL(URLString));
```

フォワードの特徴は

- 応答が早い
- 転送するページの間で request スコープのデータを共有することができる

など、であり、リダイレクトの特徴は

- 異なる Web アプリケーションや異なるホストへも転送できる
- ブラウザーが転送先の URL を知ることができる

である。必要に応じて使い分ける必要がある。(例えば転送先のページをブックマークの対象として欲しいなら、フォワードではなくリダイレクトを使う必要がある。)

キーワード:

<%@ page ~ %>タグ, <% ~ %>タグ, <%= ~ %>タグ, <%! ~ %>タグ, <%-- ~ --%>タグ, <%@ include ~ %>タグ, 暗黙オブジェクト, forward メソッド, sendRedirect メソッド,