

式言語

式言語(Expression Language)はJSP2.0から導入された技術です。変数やJavaBeansクラスのプロパティ値など、指定された式の出力を行うための言語です。式言語を略してELと言ったりしますが、ELを使えば型の意識が少なくなります。JSTLなどの拡張タグライブラリと用いれば、プログラムの知識が少ない人でJSPに記述することができます。

`$(Object.property)`

というようにオブジェクトからプロパティを簡単に取得できるのが特徴です。

表1. 式言語で使える暗黙オブジェクト

オブジェクト	概要
pageScope	pageスコープの属性を取得します。
requestScope	requestスコープの属性を取得します。
applicationScope	applicationスコープの属性を取得します。
sessionScope	sessionスコープの属性を取得します。
header	Httpヘッダを取得します。(HttpServletRequest#getHeader)
headersValues	複数の値を持つHttpヘッダを取得します。(HttpServletRequest#getHeaders)
param	リクエストパラメータを取得します。(HttpServletRequest#getParameter)
paramValues	複数の値を持つリクエストパラメータを取得します。(HttpServletRequest#request.getParameterValues)
initParam	アプリケーションの初期化パラメータを取得。
cookie	クッキー値を取得します。(HttpServletRequestCookie#getCookies)
pageContext	pageContextを参照します。

表2. 式言語で使える演算子

演算子	エイリアス	概要
<	lt	より小さい
>	gt	より大きい
<=	le	以下
>=	ge	以上
==	eq	等しい
!=	ne	等しくない
+	なし	加算
-	なし	減算
*	なし	積算
/	div	除算
%	mod	剰余
&&	and	論理積
	or	論理和
!	not	否定
empty	なし	nullまたは空文字列
a ? b : c	なし	aがtrueの場合bを実行、falseの場合はcを実行。

タグファイル

Java言語の知識が無くてもJSPの文法でカスタムタグを作ることができる仕組みです。

一番のメリットは処理やレイアウトをタグファイルに持たせて、表示はフラグメントで、呼び出し側から渡せることです。

JSPフラグメント

JSPコードの断片(JSPフラグメント)をカスタムタグの属性の値として扱うことができます。

通常の属性と大きく異なる点として、そのコードの断片は評価されず、そのままの形でカスタムタグに渡されるということです。フラグメントを受け取る属性はattributeディレクティブのfragment属性をtrueにすることで宣言できます。(例1)

渡されたフラグメントは、タグファイルの中で<jsp:invoke>タグによって評価され、その結果が出力されます。(例2)

例1 : <%@ attribute name="sample" fragment="true" %>

例2 : <jsp:invoke fragment="sample" var="[スコープ内に格納されるオブジェクト名]" scope="[オブジェクトのスコープ]" />

Simple Tag

JSP2.0からはSimpleTagSupportクラスを継承することで記述できるようになりました。

それによってJSPのスクリプト部分をJavaクラスに記述し、タグディスクリプタ(*-taglib.tld)で紐付きを記述します。

JSP上はシンプルなタグとして扱えますが、JavaクラスにHTML記述を書くことになるので汎用的なもの以外を記述すると複雑になるため、使用するのには少し抵抗があるかもしれません。