

4-2-3.暗黙オブジェクト

■ 暗黙オブジェクト

はじめに

暗黙オブジェクトとは、「何の宣言もせずにJSP内で使用できるオブジェクト」です。

例えば、javaでは特に指定を行わず`System.out.println()`と記述することで値の出力が可能ですが、jspでも同じ様に、特別な宣言を行わず`out.print()`と記述すれば値の出力が行えます。

暗黙オブジェクトの種類

項目名	説明
request	クライアントからのリクエストを取得
response	クライアントへのレスポンスを設定
out	JSPの実行結果をクライアントへの出力
pageContext	JSPのオブジェクトを管理
session	セッション情報を管理
application	アプリケーションデータを管理
page	JSPページ自身を表す
config	JSPページのパラメータを設定
exception	例外発生時のエラー情報を取得

では、先程説明した `out` の使用例を見てみましょう。

サンプルコード↓

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@ page import="java.util.* , java.text.SimpleDateFormat" %>
<html>
<head>
<title>sample_1</title>
</head>
<body>
<div>
<% Date date = new Date();
SimpleDateFormat sdf = new SimpleDateFormat("yyyy/MM/dd");
String formatDate = sdf.format(date);
out.print(formatDate); %>
</div>
</body>
</html>
```

実行結果

2019/02/10

`out.print`を使用することにより、画面へ年月日を出力しています。

折角なのでもう一つ、`request` のサンプルも見てみましょう。

サンプルコード(Request.java)↓

※色々書いてありますが、今回はdoGet()メソッド内のみ注目すれば大丈夫です。

```
import java.io.IOException;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

/**
 * Servlet implementation class Request
 */
@WebServlet("/Request")
public class Request extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public Request() {
        super();
        // TODO Auto-generated constructor stub
    }

    /**
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        // TODO Auto-generated method stub

        request.setAttribute("sample", "Sample.Inc");

        String view = "/request.jsp";
        RequestDispatcher dispatcher = request.getRequestDispatcher(view);

        dispatcher.forward(request, response);
    }

    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        // TODO Auto-generated method stub
    }
}
```

サンプルコード(request.jsp)↓

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>requestSample</title>
</head>
<body>
<%= request.getAttribute("sample")%>
</body>
</html>
```

Request.javaのdoGet()メソッド内に

```
request.setAttribute("sample", "Sample.Inc");
```

と記述されているのが確認できます。

第一引数の"sample"をkeyとしてjsp側で呼び出すことによって、

第二引数に指定した値"Sample.Inc"が画面に出力されます。

request.jsp内で

```
<%= request.getAttribute("sample")%>
```

と指定されていますね。

```
request.setAttribute
request.getAttribute
```

による、servret→jspへの値の受け渡しは現場で目にする機会も多く、この考え方は開発を行う上で

非常に応用が効きます。

形は違えど同様の処理を書く機会は多いと思いますので、今の内に処理の流れをしっかりと理解しておきましょう。

まとめ

ここでは`out.print`を用いて簡単な例を紹介しましたが、アプリケーションを開発する上で
`out` や `request` 以外にも `response` といった様々な暗黙オブジェクトを使用する機会もあるかと思います。
とはいっても暗黙オブジェクトの使用法は上記の通りなので、まずは暗黙オブジェクトの種類を覚える所から始めましょう。

課題

提出課題はありませんので、一通り学習が終わったら次の章に進んで下さい。

最終更新日時: 2022年 08月 28日(日曜日) 08:13