

Lesson 1



Web アプリケーションとは



Lesson のゴール

- ① インターネットの仕組みを理解する
- ② Web アプリケーションとは何かを理解する

[キーワード]

web ブラウザ、web サーバ、リクエスト、レスポンス、
web アプリケーション（サーバサイド/クライアントサイド）

▶ 板書

▶ Webアプリケーション

- ① _____ : webページを表示できるソフトウェア
- ② _____ : webページの情報を提供するソフトウェア(コンピュータ)
- ① _____ : webページの情報をサーバに要求すること
- ② _____ : リクエストに対してサーバが情報を返すこと

- web環境で動くプログラムを _____ とい、 _____ (バックエンド)と _____ に分かれる

① → ①

② ←

宛先IPアドレス など
<html> ~ </html>

https:// google.com/ index.html
送信元IPアドレス、ブラウザ情報
画面への入力情報 など

google.com

②

HTML
<html>
:
</html>
index.html

サブレット
JSP



Lesson 2

HTML 入門



Lesson のゴール

- ① HTML とは何か、HTML の書き方を理解する
- ② HTML ファイルを作れるようになる

[キーワード]

HTML、ハイパーテキスト、マークアップ言語、タグ、属性

▶ 板書

▶ HTML (Hyper Text Markup Language)

- **ハイパーテキスト**: 複数の文書を結びつける仕組み (_____ を使える)
- **マークアップ言語**: _____ を使って文章に意味づけをできる言語

```
<title>HTML入門</title> <br />
```

```
<a href="https://google.com">リンク</a>
```

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8"/>
    <title>HTML入門</title>
  </head>
  <body>
    <h1>HTMLを始めよう!</h1>
    <a href="〜">リンク</a>
  </body>
</html>
```

▶ 参考サイト

- とほほの WWW 入門

<https://www.tohoho-web.com/>

▶ HTML

first.html

```
1 | <!DOCTYPE html>
2 | <html>
3 |   <head>
4 |     <meta charset="utf-8"/>
5 |     <title>HTML 入門</title>
6 |   </head>
7 |   <body>
8 |     <h1>HTML を始めよう ! </h1>
9 |     <p><a href="https://google.com/">Google へのリンク</a></p>
10 |   </body>
11 | </html>
```



Lesson 3

フォーム画面の作り方



Lesson のゴール

- ① フォーム画面からデータを送るときの仕組みを理解する
- ② フォーム画面を HTML で作れるようになる

[キーワード]

フォーム、GET 送信、POST 送信、フォーム部品

▶ 板書

▶ フォーム画面

- 入力情報を _____ に送れるページ
- フォーム部品: (name属性の値)=(入力値)を送信

```
<form action = "プログラムのURL" method = "GET/POST">  
  (フォーム部品)  
</form>
```

部品	タグ	入力値
テキストボックス (パスワード)	<code><input type = "text" /></code> <code><input type = "password" /></code>	入力文字列 ↓
ラジオボタン	<code><input type = "radio" /></code>	value属性の値
チェックボックス	<code><input type = "checkbox" /></code>	(複数可)
リストボックス	<code><select></code> , <code><option></code>	↓ (<code><option></code>)
テキストエリア	<code><textarea></code>	入力文字列
送信ボタン	<code><input type = "submit" /></code>	value属性の値
取消ボタン	<code><input type = "reset" /></code>	×

▶ HTML

register.html

```
1 <!DOCTYPE html>  
2 <html>  
3 <head>  
4 <meta charset="utf-8"/>  
5 <title>登録画面</title>  
6 </head>  
7 <body>  
8 <h1>登録情報を入力してください</h1>  
9 <form action="/javaweb/register" method="post">  
10 名前 : <input type="text" name="name" size="10" /><br/>  
11 パスワード : <input type="password" name="pass" size="10" /><br/>  
12 年齢 : <input type="radio" name="age" value="child" />18歳未満  
13 <input type="radio" name="age" value="adult" />18歳以上<br/>  
14 開発経験 : <input type="checkbox" name="lang" value="Java" />Java  
15 <input type="checkbox" name="lang" value="Python" />Python  
16 <input type="checkbox" name="lang" value="JavaScript" />JavaScript<br/>  
17 住所 : <select name="address" size="1">  
18 <option value="北海道">北海道</option>  
19 <option value="東北">東北</option>
```

```
20 |         <option value="関東">関東</option>
21 |         <option value="中部">中部</option>
22 |         <option value="近畿">近畿</option>
23 |         <option value="中国">中国</option>
24 |         <option value="四国">四国</option>
25 |         <option value="九州">九州</option>
26 |         <option value="沖縄">沖縄</option></select><br/>
27 |     ご意見・お問い合わせ : <br/>
28 |     <textarea name="msg" rows="5" cols="50">入力欄</textarea><br/>
29 |     <input type="submit" value="送信" />
30 |     <input type="reset" value="取消" />
31 | </form>
32 | </body>
33 | </html>
```



Lesson 4

Tomcat



Lesson のゴール

- ① Tomcat とは何かを理解する
- ② Tomcat をインストールして実行できる環境を整える

[キーワード]

Tomcat、web コンテナ、環境変数 CATALINA_HOME

▶ 板書

▶ Apache Tomcat

- _____ と _____ の機能をもつ
Java製の無料ソフトウェア
- webコンテナ: _____ と _____ の
実行環境
- インストール手順
- zipファイルを _____ して _____
- _____ の設定
- CLASSPATH: .; (Tomcatフォルダ)¥lib¥servlet-api.jar
- CATALINA_HOME: Tomcatフォルダ

• webアプリケーションのフォルダ構成

webアプリの単位
参照 ○
参照 ×
サーブレットのクラスファイル
設定ファイル

• webアプリの実行手順

- ファイルを格納
- ブラウザで実行
- bin¥startup.bat
- bin¥shutdown.bat

▶ 参考サイト

● Apache Tomcat

<https://tomcat.apache.org/>

Lesson 5



サーブレットの基本



Lesson のゴール

- ① サーブレットとは何かを理解する
- ② サーブレットを作るための4つのポイントを知る

[キーワード]

サーブレット、アノテーション

▶ 板書

▶ サーブレット

- _____ 側で動くJavaプログラム
- ① 4つの _____ をインポート
- ② _____ クラスを継承
- ③ アノテーションで _____ を指定
- ④ _____ / _____ をオーバーライド
 - ① 出力の _____ を指定
 - ② _____ を取得
 - ③ _____ で出力

```
import java.io.*;           // PrintWriter
import javax.servlet.*;    // GenericServlet など
import javax.servlet.http.*; // HttpServlet など
import javax.servlet.annotation.*; // アノテーション機能

@WebServlet("/first");

public class FirstServlet extends HttpServlet {
    public void doGet( -- req, -- res ) throws -- {
        res.setContentType("text/html; charset=utf-8");
        PrintWriter out = res.getWriter();
        out.println("<html>");
        out.println("</html>");
    }
}
* http://localhost:8080/jowweb/first で実行
```

▶ プログラム

FirstServlet.java

```
1 | import java.io.*;
2 | import javax.servlet.*;
3 | import javax.servlet.http.*;
4 | import javax.servlet.annotation.*;
5 |
6 | @WebServlet("/register")
7 | public class RegisterServlet extends HttpServlet {
8 |     public void doPost(HttpServletRequest req, HttpServletResponse res)
9 |         throws IOException, ServletException {
10 |         req.setCharacterEncoding("utf-8");
11 |         res.setContentType("text/html; charset=utf-8");
12 |         PrintWriter out = res.getWriter();
13 |         out.println("<html>");
14 |         out.println("<head>");
15 |         out.println("<title>確認画面</title>");
16 |         out.println("</head>");
17 |         out.println("<body>");
18 |         out.println("<h2>入力情報を確認して登録ボタンを押してください</h2>");
19 |         out.println("氏名 : <strong>" + req.getParameter("name") + "</strong><br/>");
20 |         out.println("パスワード : <strong>" + req.getParameter("pass") + "</strong><br/>");
21 |         String age = req.getParameter("age");
```

```
22 |     if(age.equals("child")){
23 |         age = "18 歳未満";
24 |     } else {
25 |         age = "18 歳以上";
26 |     }
27 |     out.println("年齢 : <strong>" + age + "</strong><br/>");
28 |     String[] langs = req.getParameterValues("lang");
29 |     out.println("開発経験 : <strong>");
30 |     for(int i = 0; i < langs.length; i++){
31 |         out.println(langs[i] + " ");
32 |     }
33 |     out.println("</strong><br/>");
34 |     out.println("住所 : <strong>" + req.getParameter("address") + "</strong><br/>");
35 |     out.println("ご意見・お問い合わせ : <br/><strong>" + req.getParameter("msg") +
    "</strong><br/><br/>");
36 |     out.println("<input type=\"submit\" value=\"登録\" />");
37 |     out.println("<input type=\"reset\" value=\"戻る\" /><br/><br/>");
38 |     out.println("（この画面はサーブレットで出力しています）");
39 |     out.println("</body>");
40 |     out.println("</html>");
41 | }
42 | public void doGet(HttpServletRequest req, HttpServletResponse res)
43 |     throws IOException, ServletException {
44 |     doPost(req, res);
45 | }
46 | }
```




Lesson 6

サーブレット - フォーム連携 -



Lesson のゴール

フォーム画面から送られるデータをサーブレットで受け取るプログラムを書けるようになる

▶ 板書

▶ サーブレットのフォーム連携

① フォームの入力情報は _____ の形でリクエストに格納される

② _____ / _____ の引数 req がもつメソッドから入力値を取得

□ void setCharacterEncoding(String en)
リクエストデータを読み取る文字コードを指定
req.setCharacterEncoding("utf-8");

□ String getParameter(String name)
String[] getParameterValues(String name)
パラメータ名 name の入力値を取得
String n = req.getParameter("name");

name = suga & pass = suga & ...
& lang = Java & lang = Python & ...

名前
パスワード
送信

リクエスト

サーブレット
JSP

```
<form action = "プログラムのURL" method = "get / post ">  
<input type = "text" name = "name" />  
...
```

▶ プログラム

register.html

```
1 <!DOCTYPE html>  
2 <html>  
3 <head>  
4 <meta charset="utf-8"/>  
5 <title>登録画面</title>  
6 </head>  
7 <body>  
8 <h1>登録情報を入力してください</h1>
```

```

9 | <form action="/javaweb/register" method="post">
10 | 名前 : <input type="text" name="name" size="10" /><br/>
11 | パスワード : <input type="password" name="pass" size="10" /><br/>
12 | 年齢 : <input type="radio" name="age" value="child" />18 歳未満
13 |         <input type="radio" name="age" value="adult" />18 歳以上<br/>
14 | 開発経験 : <input type="checkbox" name="lang" value="Java" />Java
15 |         <input type="checkbox" name="lang" value="Python" />Python
16 |         <input type="checkbox" name="lang" value="JavaScript" />JavaScript<br/>
17 | 住所 : <select name="address" size="1">
18 |         <option value="北海道">北海道</option>
19 |         <option value="東北">東北</option>
20 |         <option value="関東">関東</option>
21 |         <option value="中部">中部</option>
22 |         <option value="近畿">近畿</option>
23 |         <option value="中国">中国</option>
24 |         <option value="四国">四国</option>
25 |         <option value="九州">九州</option>
26 |         <option value="沖縄">沖縄</option></select><br/>
27 | ご意見・お問い合わせ : <br/>
28 | <textarea name="msg" rows="5" cols="50">入力欄</textarea><br/>
29 | <input type="submit" value="送信" />
30 | <input type="reset" value="取消" />
31 | </form>
32 | </body>
33 | </html>

```

RegisterServlet.java

```

1 | import java.io.*;
2 | import javax.servlet.*;
3 | import javax.servlet.http.*;
4 | import javax.servlet.annotation.*;
5 |
6 | @WebServlet("/register")
7 | public class RegisterServlet extends HttpServlet {
8 |     public void doPost(HttpServletRequest req, HttpServletResponse res)
9 |         throws IOException, ServletException {
10 |         req.setCharacterEncoding("utf-8");

```

```
11 |     res.setContentType("text/html;charset=utf-8");
12 |     PrintWriter out = res.getWriter();
13 |     out.println("<html>");
14 |     out.println("<head>");
15 |     out.println("<title>確認画面</title>");
16 |     out.println("</head>");
17 |     out.println("<body>");
18 |     out.println("<h2>入力情報を確認して登録ボタンを押してください</h2>");
19 |     out.println("氏名 : <strong>" + req.getParameter("name") + "</strong><br/>");
20 |     out.println("パスワード : <strong>" + req.getParameter("pass") + "</strong><br/>");
21 |     String age = req.getParameter("age");
22 |     if(age.equals("child")){
23 |         age = "18 歳未満";
24 |     } else {
25 |         age = "18 歳以上";
26 |     }
27 |     out.println("年齢 : <strong>" + age + "</strong><br/>");
28 |     String[] langs = req.getParameterValues("lang");
29 |     out.println("開発経験 : <strong>");
30 |     for(int i = 0; i < langs.length; i++){
31 |         out.println(langs[i] + " ");
32 |     }
33 |     out.println("</strong><br/>");
34 |     out.println("住所 : <strong>" + req.getParameter("address") + "</strong><br/>");
35 |     out.println("ご意見・お問い合わせ : <br/><strong>" + req.getParameter("msg") +
    "</strong><br/><br/>");
36 |     out.println("<input type=\"submit\" value=\"登録\" />");
37 |     out.println("<input type=\"reset\" value=\"戻る\" /><br/><br/>");
38 |     out.println("（この画面はサーブレットで出力しています）");
39 |     out.println("</body>");
40 |     out.println("</html>");
41 | }
42 | public void doGet(HttpServletRequest req, HttpServletResponse res)
43 |     throws IOException, ServletException {
44 |     doPost(req, res);
45 | }
46 | }
```



Lesson 7

サーブレット –セッション管理–



Lesson のゴール

- ① セッションの必要性やセッションが動く仕組みを理解する
- ② セッション管理のプログラムを書けるようになる

[キーワード]

セッション、セッションオブジェクト

▶ 板書

▶ セッション管理

• 複数のリクエスト・レスポンスにまたがって _____ と _____ をする仕組み

ログインサーブレット

```
String name = "菅原";
```

注文サーブレット

```
String book = "本A";
```

- req からセッションオブジェクトを取得
 - ① HttpSession getSession(boolean create)
 - 初回時に true → 生成 / false → null
 - セッションオブジェクトのデータを管理
 - ② void setAttribute(String name, Object value)
 - value を name という名前で格納
 - ③ Object getAttribute(String name)
 - name という名前のデータを取得

▶ プログラム

SessionServlet.java

```

1 | import java.util.*;
2 | import java.io.*;
3 | import javax.servlet.*;
4 | import javax.servlet.http.*;
5 | import javax.servlet.annotation.*;
6 |

```

```
7 | @WebServlet("/session")
8 | public class SessionServlet extends HttpServlet {
9 |     public void doPost(HttpServletRequest req, HttpServletResponse res)
10 |         throws IOException, ServletException {
11 |         // セッションオブジェクトの生成 or 取得
12 |         HttpSession session = req.getSession(true);
13 |         // セッションオブジェクトから cart を取得
14 |         ArrayList<String> booklist =
15 |             (ArrayList<String>)session.getAttribute("cart");
16 |
17 |         if(booklist == null){
18 |             booklist = new ArrayList<String>(); // 初回アクセス
19 |         } else {
20 |             req.setCharacterEncoding("utf-8"); // 2回目以降
21 |             String book = req.getParameter("book");
22 |             booklist.add(book);
23 |         }
24 |         session.setAttribute("cart", booklist);
25 |
26 |         res.setContentType("text/html;charset=utf-8");
27 |         PrintWriter out = res.getWriter();
28 |         out.println("<html><head>");
29 |         out.println("<title>注文画面</title>");
30 |         out.println("</head><body>");
31 |         out.println("<h2>商品を選択してください</h2>");
32 |         out.println("<form action=\"/javaweb/session\" method=\"post\">");
33 |         out.println("<select name=\"book\" size=\"1\">");
34 |         out.println("<option value=\"超絶わかる Java\">超絶わかる Java</option>");
35 |         out.println("<option value=\"小学生からはじめる Java\">小学生からはじめる Java</option>");
36 |         out.println("<option value=\"Oracle 試験 100%合格本\">Oracle 試験 100%合格本</option>");
37 |         out.println("</select><input type=\"submit\" value=\"追加\" /></form>");
38 |         out.println("<h2>カートの中身</h2>");
39 |
40 |         for(int i = 0; i < booklist.size(); i++){
41 |             out.println(booklist.get(i) + "<br/>");
42 |         }
43 |     }
```

```

44     out.println("</body></html>");
45     }
46     public void doGet(HttpServletRequest req, HttpServletResponse res)
47         throws IOException, ServletException {
48         doPost(req, res);
49     }
50 }

```



Lesson 8

JSP の基本



Lesson のゴール

- ① JSP とは何かを理解する
- ② JSP を実行するときに裏で動いている仕組みを理解する

[キーワード]

JSP、JSP サブレット

▶ 板書

▶ JSP (Java Server Pages)

- HTML に独自の _____ で Java のコードを埋め込んで動的な web ページを作る技術
- サブレットとの違い

[サブレット]	[JSP]
<pre> out.println("<body>"); out.println("<h1>"); out.println("サブレット"); out.println("</h1>"); </pre>	<pre> <body> <h1> <% out.println("JSP");%> </h1> </pre>
— .java	— .jsp
拡張子	拡張子
ロック制御	デザイン
要	不要
	コンパイル・再起動

• JSP 実行の仕組み

• page ディレクティブタブ: JSP ファイル全体の設定

```

<%@page contentType = "text/html; charset = utf-8" %>

```

▶ プログラム

first.jsp

```
1 <%@page contentType="text/html;charset=utf-8" %>
2 <html>
3   <head>
4     <title>はじめての JSP</title>
5   </head>
6   <body>
7     <h1><% out.println("JSP を始めよう!"); %></h1>
8   </body>
9 </html>
```



Lesson 9

JSP で使えるタグ



Lesson のゴール

JSP 特有のタグ（6 種類）を使えるようになる

[キーワード]

ディレクティブタグ、宣言タグ、式タグ、スクリプトレット、
JSP コメント、アクションタグ

▶ 板書

▶ JSP のタグ

① ディレクティブタグ <%@ ~ %>

サブレット変換時に必要な設定を記述

```
<%@page contentType="text/html;charset=utf-8" %>
```

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

② 宣言タグ <%! 宣言 %>

メンバ変数やメソッドを定義

```
<%! String fortune(){ ~ } %>
```

③ 式タグ <%= 式 %>

出カしたい式を記述 <%= i+j %>

④ スクリプトレット <% Javaのコード %>

Javaのプログラムを記述

```
<% for(int i=0; i<3; i++){ %>
```

⑤ コメント <!-- コメント -->

JSPのコメントを記述

⑥ アクションタグ <タグ名 属性名="値"/>

特定の処理をシンプルに書くためのタグ

```
<jsp:useBean id="counter" scope="session"
class="Counter" />
```

▶ プログラム

fortune.jsp

```
1  <%@page contentType="text/html;charset=utf-8" %>
2  <%!
3      String fortune(){
4          String f;
5          double fn = Math.random(); // 0.0~1.0 の乱数
6          if( fn >= 0.7 ){
7              f = "大吉";
8          } else if( fn >= 0.4 ){
9              f = "中吉";
10         } else if( fn >= 0.1 ){
11             f = "小吉";
12         } else {
13             f = "凶";
14         }
15         return f;
16     }
17 %>
18 <html>
19     <head>
20         <title>占い JSP</title>
21     </head>
22     <body>
23         <h1>JSP で占い ! </h1>
24         あなたの今日と明日の運勢は……<br/>
25         <% for(int i = 0; i < 2; i++){ %>
26             <strong><%= fortune() %></strong><br/>
27         <% } %>
28     </body>
29 </html>
```




Lesson 10

JSP - フォーム連携 -



Lesson のゴール

フォーム画面から送られるデータを受け取る JSP を書けるようになる

▶ 板書

▶ JSP のフォーム連携

① フォームの入力情報は _____ の形でリクエストに格納される

② _____ の request がもつメソッドから入力値を取得

```

name = suga & pass = suga & ...
& lang = Java & lang = Python & ...

```

リクエスト

```

<% request.setCharacterEncoding("utf-8"); %>
名前: <%= request.getParameter("name") %>

```

- 暗黙オブジェクト: JSP サブレット内で _____ で宣言されるオブジェクト

HttpServletRequest request HttpSession session
 HttpServletResponse response JspWriter out など

```

<form action = "プログラムのURL" method = "get / post ">
  <input type = "text" name = "name" />
  ...

```

▶ プログラム

register2.html

```

1 | <!DOCTYPE html>
2 | <html>
3 | <head>
4 |   <meta charset="utf-8"/>
5 |   <title>登録画面</title>
6 | </head>
7 | <body>
8 |   <h1>登録情報を入力してください</h1>

```

```

9 | <form action="/javaweb/register2.jsp" method="post">
10 | 名前 : <input type="text" name="name" size="10" /><br/>
11 | パスワード : <input type="password" name="pass" size="10" /><br/>
12 | 年齢 : <input type="radio" name="age" value="child" />18 歳未満
13 |         <input type="radio" name="age" value="adult" />18 歳以上<br/>
14 | 開発経験 : <input type="checkbox" name="lang" value="Java" />Java
15 |         <input type="checkbox" name="lang" value="Python" />Python
16 |         <input type="checkbox" name="lang" value="JavaScript" />JavaScript<br/>
17 | 住所 : <select name="address" size="1">
18 |         <option value="北海道">北海道</option>
19 |         <option value="東北">東北</option>
20 |         <option value="関東">関東</option>
21 |         <option value="中部">中部</option>
22 |         <option value="近畿">近畿</option>
23 |         <option value="中国">中国</option>
24 |         <option value="四国">四国</option>
25 |         <option value="九州">九州</option>
26 |         <option value="沖縄">沖縄</option></select><br/>
27 | ご意見・お問い合わせ : <br/>
28 | <textarea name="msg" rows="5" cols="50">入力欄</textarea><br/>
29 | <input type="submit" value="送信" />
30 | <input type="reset" value="取消" />
31 | </form>
32 | </body>
33 | </html>

```

register2.jsp

```

1 | <%@page contentType="text/html;charset=utf-8" %>
2 | <%
3 |     request.setCharacterEncoding("utf-8");
4 | %>
5 | <html>
6 |     <head>
7 |         <title>確認画面</title>
8 |     </head>

```

```
9 | <body>
10 |   <h2>入力情報を確認して登録ボタンを押してください</h2>
11 |   名前 : <strong><%= request.getParameter("name") %></strong><br/>
12 |   パスワード : <strong><%= request.getParameter("pass") %></strong><br/>
13 |   <%
14 |     String age = request.getParameter("age");
15 |     if(age.equals("child")){
16 |       age = "18 歳未満";
17 |     } else {
18 |       age = "18 歳以上";
19 |     }
20 |   %>
21 |   年齢 : <strong><%= age %></strong><br/>
22 |   開発経験 : <strong>
23 |   <%
24 |     String[] langs = request.getParameterValues("lang");
25 |     for(int i = 0; i < langs.length; i++){
26 |       out.println(langs[i] + " ");
27 |     }
28 |   %>
29 |   </strong><br/>
30 |   住所 : <strong><%= request.getParameter("address") %></strong><br/>
31 |   ご意見・お問い合わせ : <br/><strong><%= request.getParameter("msg") %></str
32 |   ong><br/><br/>
33 |   <input type="submit" value="登録" />
34 |   <input type="reset" value="戻る" /><br/><br/>
35 |   (この画面は JSP で出力しています)
36 | </body>
   </html>
```



Lesson 11

サーブレットと JSP の連携



Lesson のゴール

- ① なぜサーブレットと JSP を連携して使うのかを理解する
- ② サーブレットと JSP を連携するプログラムを書けるようになる

[キーワード]

フォワード

▶ 板書

▶ サーブレットと JSP の連携

< サーブレット >

```
void doPost(～) ... {
    ...
    out.println("<body>");
    out.println("名前:" +
        req.getParameter("name"));
    String age =
        req.getParameter("age");
    if (age.equals("child")){
        age = "18才未満";
        ...
    }
}
```

< JSP >

```
<% request.
    setCharacterEncoding
    ("utf-8"); %>
<body>
    名前:<% = request.
        getParameter("name")%>
<%
    String age = request.
        getParameter("age");
    ...
%>
```

→ _____ や _____ が難しい

▶ プログラム

register3.html

```

1 | <!DOCTYPE html>
2 | <html>
3 | <head>
4 |   <meta charset="utf-8"/>
5 |   <title>登録画面</title>
6 | </head>
7 | <body>
8 |   <h1>登録情報を入力してください</h1>
```

```

9 | <form action="/javaweb/register3" method="post">
10 | 名前 : <input type="text" name="name" size="10" /><br/>
11 | パスワード : <input type="password" name="pass" size="10" /><br/>
12 | 年齢 : <input type="radio" name="age" value="child" />18 歳未満
13 |         <input type="radio" name="age" value="adult" />18 歳以上<br/>
14 | 開発経験 : <input type="checkbox" name="lang" value="Java" />Java
15 |         <input type="checkbox" name="lang" value="Python" />Python
16 |         <input type="checkbox" name="lang" value="JavaScript" />JavaScript<br/>
17 | 住所 : <select name="address" size="1">
18 |         <option value="北海道">北海道</option>
19 |         <option value="東北">東北</option>
20 |         <option value="関東">関東</option>
21 |         <option value="中部">中部</option>
22 |         <option value="近畿">近畿</option>
23 |         <option value="中国">中国</option>
24 |         <option value="四国">四国</option>
25 |         <option value="九州">九州</option>
26 |         <option value="沖縄">沖縄</option></select><br/>
27 | ご意見・お問い合わせ : <br/>
28 | <textarea name="msg" rows="5" cols="50">入力欄</textarea><br/>
29 | <input type="submit" value="送信" />
30 | <input type="reset" value="取消" />
31 | </form>
32 | </body>
33 | </html>

```

RegisterServlet3.java

```

1 | import java.io.*;
2 | import javax.servlet.*;
3 | import javax.servlet.http.*;
4 | import javax.servlet.annotation.*;
5 |
6 | @WebServlet("/register3")
7 | public class RegisterServlet3 extends HttpServlet {
8 |     public void doPost(HttpServletRequest req, HttpServletResponse res)
9 |         throws IOException, ServletException {
10 |         req.setCharacterEncoding("utf-8");

```

```

11
12     String age = req.getParameter("age");
13     if(age.equals("child")){
14         age = "18 歳未満";
15     } else {
16         age = "18 歳以上";
17     }
18     req.setAttribute("age", age);
19
20     String langs = "";
21     String[] lang = req.getParameterValues("lang");
22     for(int i = 0; i < lang.length; i++){
23         langs = langs + lang[i] + " ";
24     }
25     req.setAttribute("langs", langs);
26
27     RequestDispatcher rd = req.getRequestDispatcher("/register3.jsp");
28     rd.forward(req, res);
29 }
30 public void doGet(HttpServletRequest req, HttpServletResponse res)
31     throws IOException, ServletException {
32     doPost(req, res);
33 }
34 }

```

register3.jsp

```

1 <%@page contentType="text/html;charset=utf-8" %>
2 <html>
3     <head>
4         <title>確認画面</title>
5     </head>
6 <body>
7     <h2>入力情報を確認して登録ボタンを押してください</h2>
8     氏名 : <strong><%= request.getParameter("name") %></strong><br/>
9     パスワード : <strong><%= request.getParameter("pass") %></strong><br/>
10    年齢 : <strong><%= request.getAttribute("age") %></strong><br/>

```

```

11 | 開発経験 : <strong><%= request.getAttribute("langs") %></strong><br/>
12 | 住所 : <strong><%= request.getParameter("address") %></strong><br/>
13 | ご意見・お問い合わせ : <br/><strong><%= request.getParameter("msg") %></str
   | ong><br/><br/>
14 | <input type="submit" value="登録" />
15 | <input type="reset" value="戻る" /><br/><br/>
16 | (この画面はフォワード先の JSP で出力しています)
17 | </body>
18 | </html>

```



Lesson 12

JavaBeans



Lesson のゴール

- ① JavaBeans の設計ルールを理解する
- ② Bean のクラスを作れるようになる

[キーワード]

JavaBeans、Bean

▶ 板書

▶ JavaBeans

- 部品化したクラスを作るための _____
JavaBeans に従って作られたクラスを
_____ という

JavaBeans の設計ルール

- ① クラスを _____ 指定
- ② 引数なしの _____ を定義
- ③ _____ インタフェースを実装
- ④ _____
- ⑤ _____ メソッドを定義

```

package bean;
import java.io.Serializable;
public class RegisterBean implements Serializable{
    private String name;
    public RegisterBean(){
    }
    public void setName(String name){
        this.name = name;
    }
    public String getName(){
        return name;
    }
}

```

▶ プログラム

RegisterBean.java

```
1  package bean;
2
3  import java.io.Serializable;
4
5  public class RegisterBean implements Serializable{
6      private String name;
7      private String age;
8      private String[] langs;
9
10     public RegisterBean(){
11     }
12
13     public void setName(String name){
14         this.name = name;
15     }
16     public void setAge(String age){
17         this.age = age;
18     }
19     public void setLangs(String[] langs){
20         this.langs = langs;
21     }
22
23     public String getName(){
24         return name;
25     }
26     public String getAge(){
27         return age;
28     }
29     public String[] getLangs(){
30         return langs;
31     }
32
33     public String getJpnAge(){
34         String jpnAge;
35         if(age.equals("child")){
36             jpnAge = "18 歳未満";
```



```
37 |     } else {
38 |         jpnAge = "18 歳以上";
39 |     }
40 |     return jpnAge;
41 | }
42 | public String getStrLangs(){
43 |     String strLangs = "";
44 |     for(int i = 0; i < langs.length; i++){
45 |         strLangs = strLangs + langs[i] + " ";
46 |     }
47 |     return strLangs;
48 | }
49 | }
```



Lesson 13

アクションタグとスコープ



Lesson のゴール

- ① アクションタグを使えるようになる
- ② 4 種類のスコープの違いを理解する

[キーワード]

アクションタグ、スコープ

▶ 板書

▶ アクションタグ

- 特定の _____ を実行するための JSP タグ

① useBean タグ： 指定したスコープの Bean オブジェクトを _____ する。なければ新規生成してスコープに _____ する

```
<jsp:useBean id = "参照名"
scope = "有効範囲" class = "クラス名" />
```

page, request, session, application

```
<jsp:useBean id = "cnt"
scope = "session" class = "bean.CounterBean" />
```

② setProperty タグ： _____ メソッドを呼び出す

```
<jsp:setProperty name = "参照名" property = "プロパティ名" value = "値" />
```

```
<jsp:setProperty name = "cnt" property = "count" value = "1" />
```

③ getProperty タグ： _____ メソッドを呼び出す

```
<jsp:getProperty name = "参照名" property = "プロパティ名" />
```

```
<jsp:getProperty name = "cnt" property = "count" />
```

The diagram illustrates a session scope containing a variable named 'cnt'. This variable points to a CounterBean object. The object has a 'count' attribute and two methods: 'setCount(~)' and 'getCount()'.

▶ プログラム

CounterBean.java

```
1 package bean;
2
3 public class CounterBean{
4     private int count;
5
6     public CounterBean(){
7         count = 0;
8     }
9     public void setCount(int count){
10        this.count = count;
11    }
12    public int getCount(){
13        count++;
14        return count;
15    }
16 }
```

counter.jsp

```
1 <%@page contentType="text/html;charset=utf-8" %>
2 <jsp:useBean id="cnt" scope="request" class="bean.CounterBean" />
3 <html>
4   <head>
5     <title>カウンター</title>
6   </head>
7   <body>
8     <h2><jsp:getProperty name="cnt" property="count" />回目のアクセスです</h2>
9   </body>
10  </html>
```



Lesson 14

MVC モデル



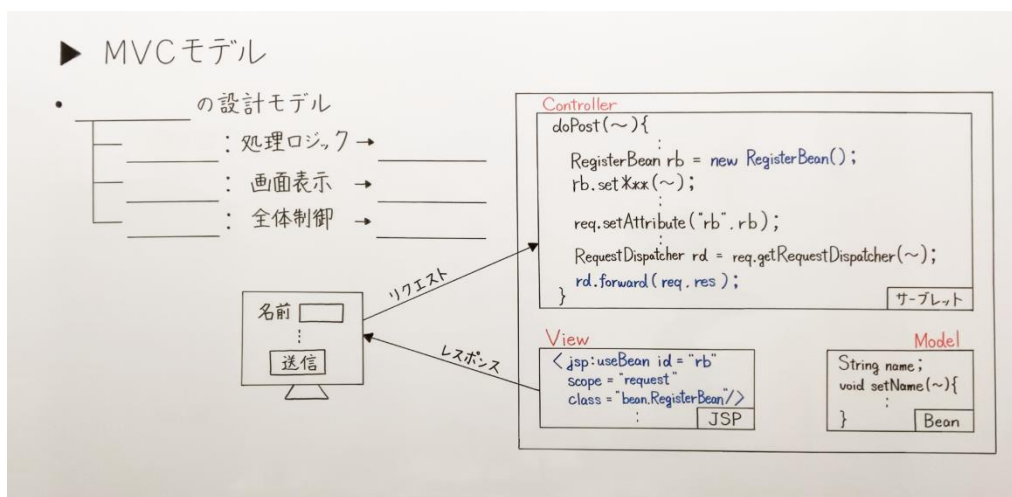
Lesson のゴール

- ① MVC モデルについて理解する
- ② MVC モデルに基づいたプログラムを作れるようになる

[キーワード]

MVC モデル、モデル、ビュー、コントローラ

▶ 板書



▶ プログラム

register4.html

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="utf-8"/>
5   <title>登録画面</title>
6 </head>
7 <body>
8   <h1>登録情報を入力してください</h1>
9   <form action="/javaweb/register4" method="post">
10  名前 : <input type="text" name="name" size="10" /><br/>
11  年齢 : <input type="radio" name="age" value="child" />18 歳未満
12         <input type="radio" name="age" value="adult" />18 歳以上<br/>
13  開発経験 : <input type="checkbox" name="lang" value="Java" />Java
14         <input type="checkbox" name="lang" value="Python" />Python
15         <input type="checkbox" name="lang" value="JavaScript" />JavaScript<br/>
16         <br/>
16   <input type="submit" value="送信" />
17 </form>
18 </body>
19 </html>
```

RegisterServlet4.java

```
1 import bean.RegisterBean;
2 import java.io.*;
3 import javax.servlet.*;
4 import javax.servlet.http.*;
5 import javax.servlet.annotation.*;
6
7 @WebServlet("/register4")
8 public class RegisterServlet4 extends HttpServlet {
9     public void doPost(HttpServletRequest req, HttpServletResponse res)
10         throws IOException, ServletException {
11         // リクエストの文字コード指定
12         req.setCharacterEncoding("utf-8");
```

```

13 |
14 | // 入力情報の取得
15 | String name = req.getParameter("name");
16 | String age = req.getParameter("age");
17 | String[] langs = req.getParameterValues("lang");
18 | res.setContentType("text/html;charset=utf-8");
19 |
20 | // Bean の作成
21 | RegisterBean rb = new RegisterBean();
22 | rb.setName(name);
23 | rb.setAge(age);
24 | rb.setLangs(langs);
25 |
26 | // Bean をリクエストに格納
27 | req.setAttribute("rb", rb);
28 |
29 | // register4.jsp へフォワード
30 | RequestDispatcher rd = req.getRequestDispatcher("/register4.jsp");
31 | rd.forward(req, res);
32 | }
33 | public void doGet(HttpServletRequest req, HttpServletResponse res)
34 |     throws IOException, ServletException {
35 |     doPost(req, res);
36 | }
37 | }

```

RegisterBean.java

```

1 | package bean;
2 |
3 | import java.io.Serializable;
4 |
5 | public class RegisterBean implements Serializable{
6 |     private String name;
7 |     private String age;
8 |     private String[] langs;

```

```
9
10 public RegisterBean(){
11 }
12
13 public void setName(String name){
14     this.name = name;
15 }
16 public void setAge(String age){
17     this.age = age;
18 }
19 public void setLangs(String[] langs){
20     this.langs = langs;
21 }
22
23 public String getName(){
24     return name;
25 }
26 public String getAge(){
27     return age;
28 }
29 public String[] getLangs(){
30     return langs;
31 }
32
33 public String getJpnAge(){
34     String jpnAge;
35     if(age.equals("child")){
36         jpnAge = "18 歳未満";
37     } else {
38         jpnAge = "18 歳以上";
39     }
40     return jpnAge;
41 }
42 public String getStrLangs(){
43     String strLangs = "";
```

```
44     for(int i = 0; i < langs.length; i++){
45         strLangs = strLangs + langs[i] + " ";
46     }
47     return strLangs;
48 }
49 }
```

register4.jsp

```
1  <%@page contentType="text/html;charset=utf-8" %>
2  <jsp:useBean id="rb" scope="request" class="bean.RegisterBean" />
3  <html>
4      <head>
5          <title>確認画面</title>
6      </head>
7      <body>
8          <h2>入力情報を確認して登録ボタンを押してください</h2>
9          氏名 : <strong><jsp:getProperty name="rb" property="name" /></strong><br/>
10         年齢 : <strong><%= rb.getJpnAge() %></strong><br/>
11         開発経験 : <strong><%= rb.getStrLangs() %></strong><br/><br/>
12         <input type="submit" value="登録" />
13         <input type="reset" value="戻る" />
14     </body>
15 </html>
```



＼フリーラーニング（無料で学べる場）を広げたい！／
チャンネル登録や拡散よろしくお願いします！



せかちゃん

