

# ウィークエンド・シャッフル ーJSP入門ー

岐阜経済大学 経営学部 経営情報学科 井戸 伸彦

来歴:

0.0版 2002年11月5日

1.0版 2004年2月14日:全面見直し

1.1版 2006年11月26日:(1.10,1.11)

## スライドの構成

- (1) JSP
- (2) HTML
- (3) JSPでプログラミング
- (4) フォーム
- (5) JSPでのインタラクティブ Webページ
- (6) 例題1:カード作成の拡張
- (7) JSPについての整理
- (8) 例題2:簡単な掲示板
- (9) 課題:掲示板の拡張

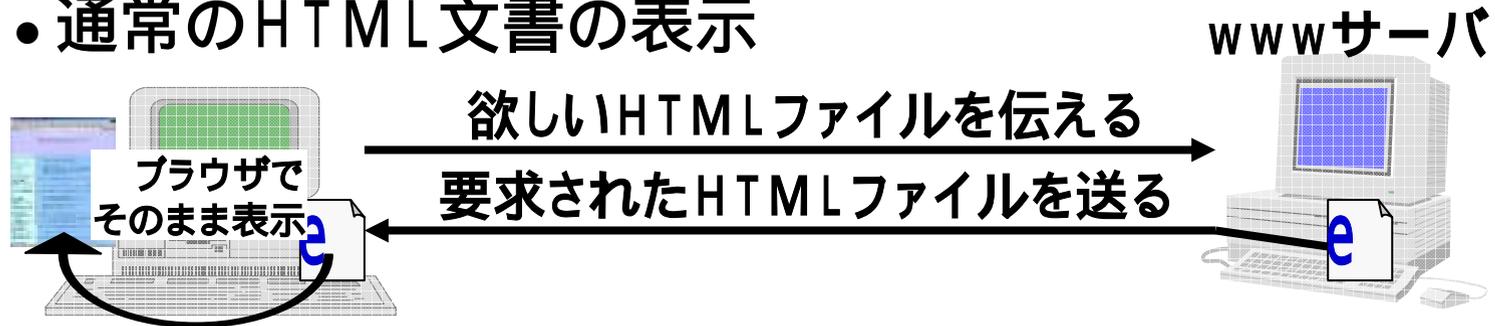
# ( 0 ) 前提

- 本講座では、JSPを用いてインタラクティブなWebページを作成する方法について説明します。
- 次のスライドと関連があります。下記のスライドの内容を一部重複して掲載しているところがあります。
  - Web関係
    - ◆「ヘンタイ良い子のWeb講座」
    - ◆「ツアラトストラ書くWeb - 速習HTML入門 - 」
    - ◆「シャボン玉HTML - フォーム入門 - 」
  - Java関係
    - ◆「シャバドゥビ、ジャバ - Java覚書 - 」
    - ◆「ドゥビドゥバ、ジャバ - 直感Javaのオブジェクト - 」
  - eclipse関係
    - ◆「月に吠える - eclipseによるJavaアプリケーション作成 - 」
    - ◆「ただ一疋の青い猫のかげ - eclipseによるJavaサーブレット作成 - 」
- セキュリティについては、次の段階で学びます。本講座のプログラムには、安全でないものが含まれています。
- 本資料にては、文字コードの指定を“Shift\_JIS”としています。これが適当でない環境においては、別の指定を行ってください。

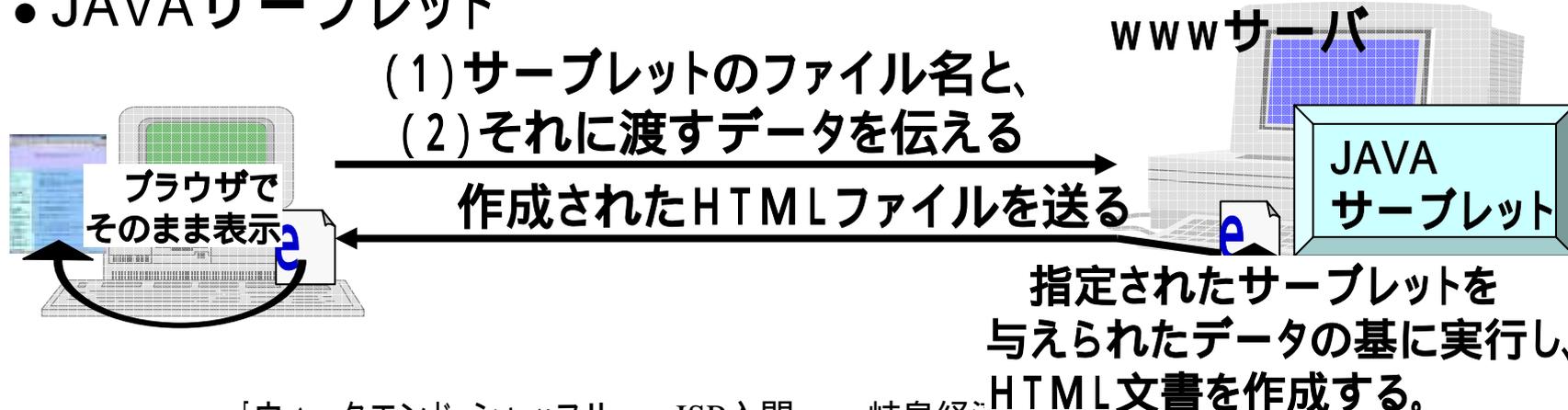
# (1) インタラクティブなWebページのしくみ

## ■JAVAサーブレット

- HTML文書がサーバ側に用意されているのではなく、HTML文書を作成するプログラムが用意されています。これにより送られてきたデータに基づくWebページが提供されます。
- 通常のHTML文書の表示



- JAVAサーブレット



# (1.1) HTML文書を作るプログラム

- 文書を作成するプログラムといっても、難しく考える必要はありません。左下図のHTML文書は、右下図のJavaサーブレットにより作成されます。すなわち、print文を連ねて作成するHTML文書を出力する訳です。

```
<html>
<head>
<title>hello</title>
</head>
<body>
hello, world.
</body>
</html>
```

< HTML >

```
out.println("<html>¥n");
out.println("<head>¥n");
out.println("<title>");
out.println("hello");
out.println("</title>¥n");
out.println("</head>¥n");
out.println("<body>¥n");
out.println("hello, world.¥n");
out.println("</body>¥n");
out.println("</html>¥n");
```

< JAVAサーブレット >

- このようにprint文を繰り返すやり方は、いささかうんざりするものです。

# (1.2) JSP

- JSP (Java Server Pages) では、直接HTMLを書くことができます。プログラムの部分だけをJavaで書いて、これをHTMLの間に埋め込みます。

普通にHTMLを書ける

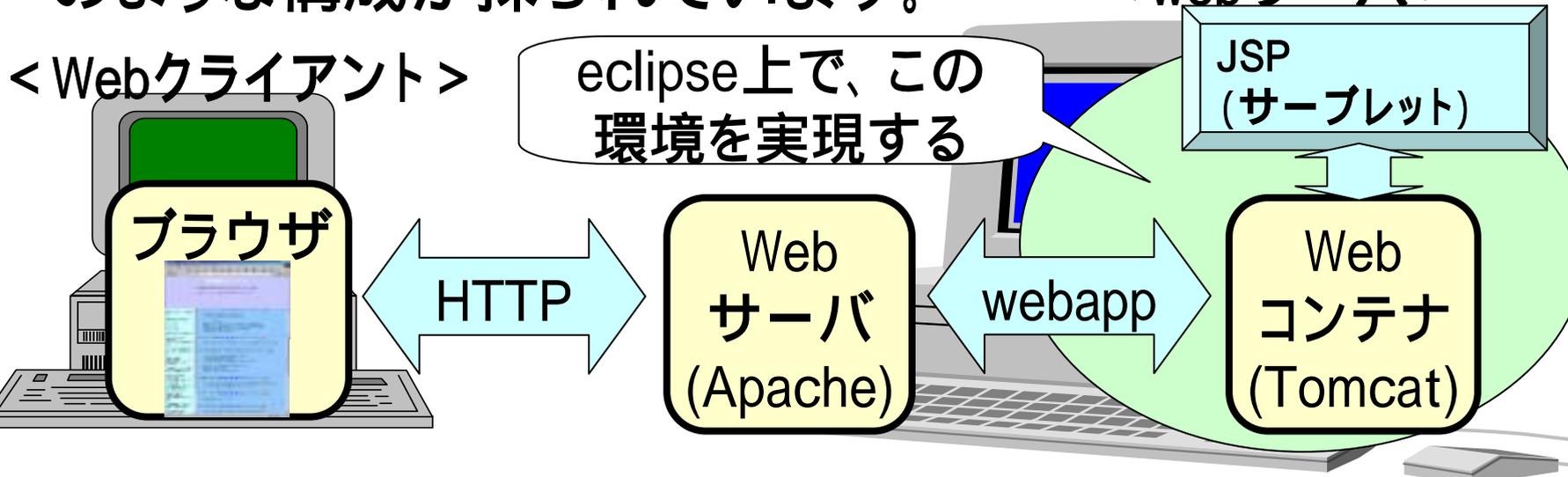
```
<html>
<head>
<title>hello</title>
</head>
<body bgcolor="#AAAAAA" text="#FFFFFF">
<% for(int i=3;i<7;i++){ %>
<p><font size=<%= i%>>メリークリスマス！ </font></p>
<% } %>
<p><font color="green">願わくば、 </font></p>
```

埋め込んだ  
Javaプログラム

< J S P >

# (1.3) サーバの構成

- JSP (Javaサーブレット) を動かす Webサーバでは、次のような構成が採られています。



- Javaサーブレットは、コンパイルしておく必要がありますが、JSPの場合は、ファイルが変化すると自動でコンパイルが行われます。

- 本スライドは、eclipse上の開発環境を前提としています。

# ( 1 . 4 ) eclipse

- eclipseは、Javaに対応したIDE (Interated Development Environment)ソフトウェアです。
- IDEとは、コンパイラやエディタ、デバッグツールなど、プログラムの開発に必要なツールを統合したものです。
- 岐阜経済大学内のWindows PCでは、次のようにして起動できます。

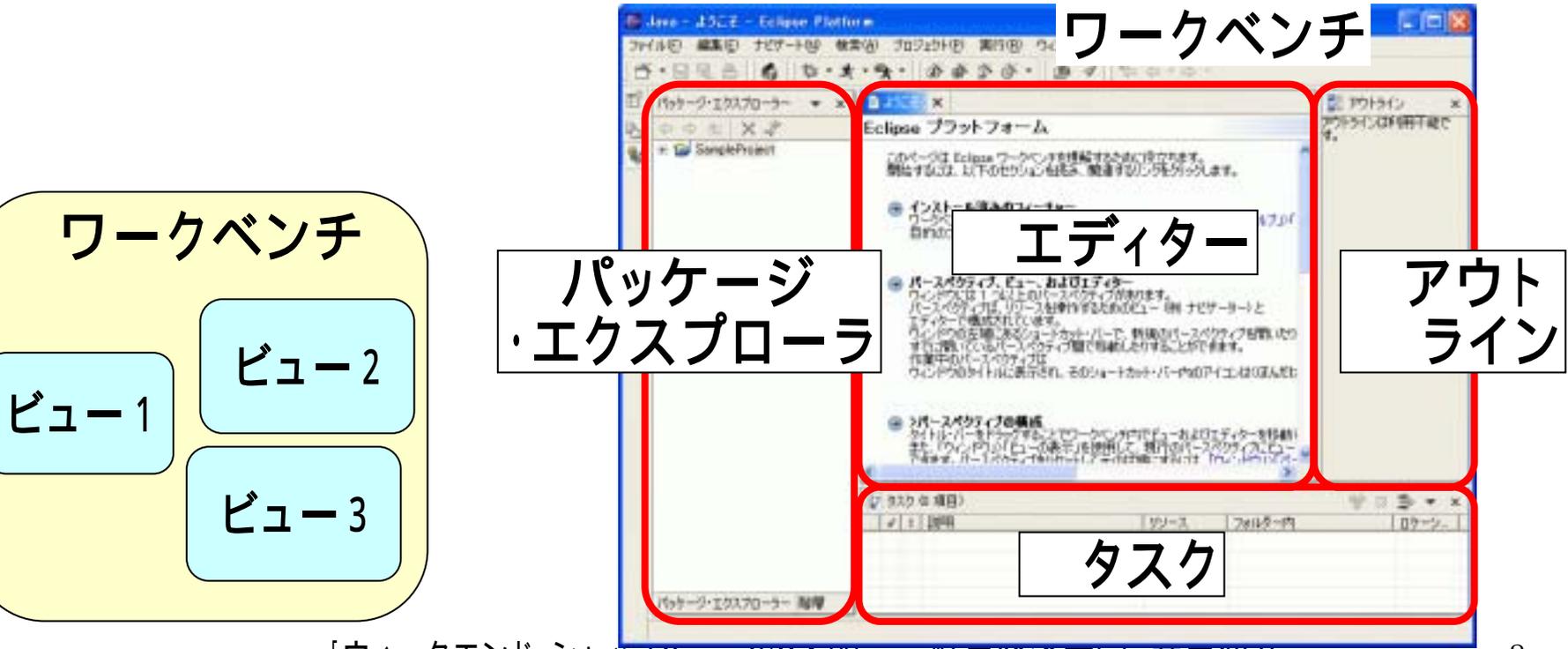
[スタート]-[すべてのプログラム]-[アプリケーション]-[eclipse]



全部まとめて、わかりやすく、簡単な操作により、Javaプログラムを作る！

# ( 1 . 5 ) eclipseのビュー

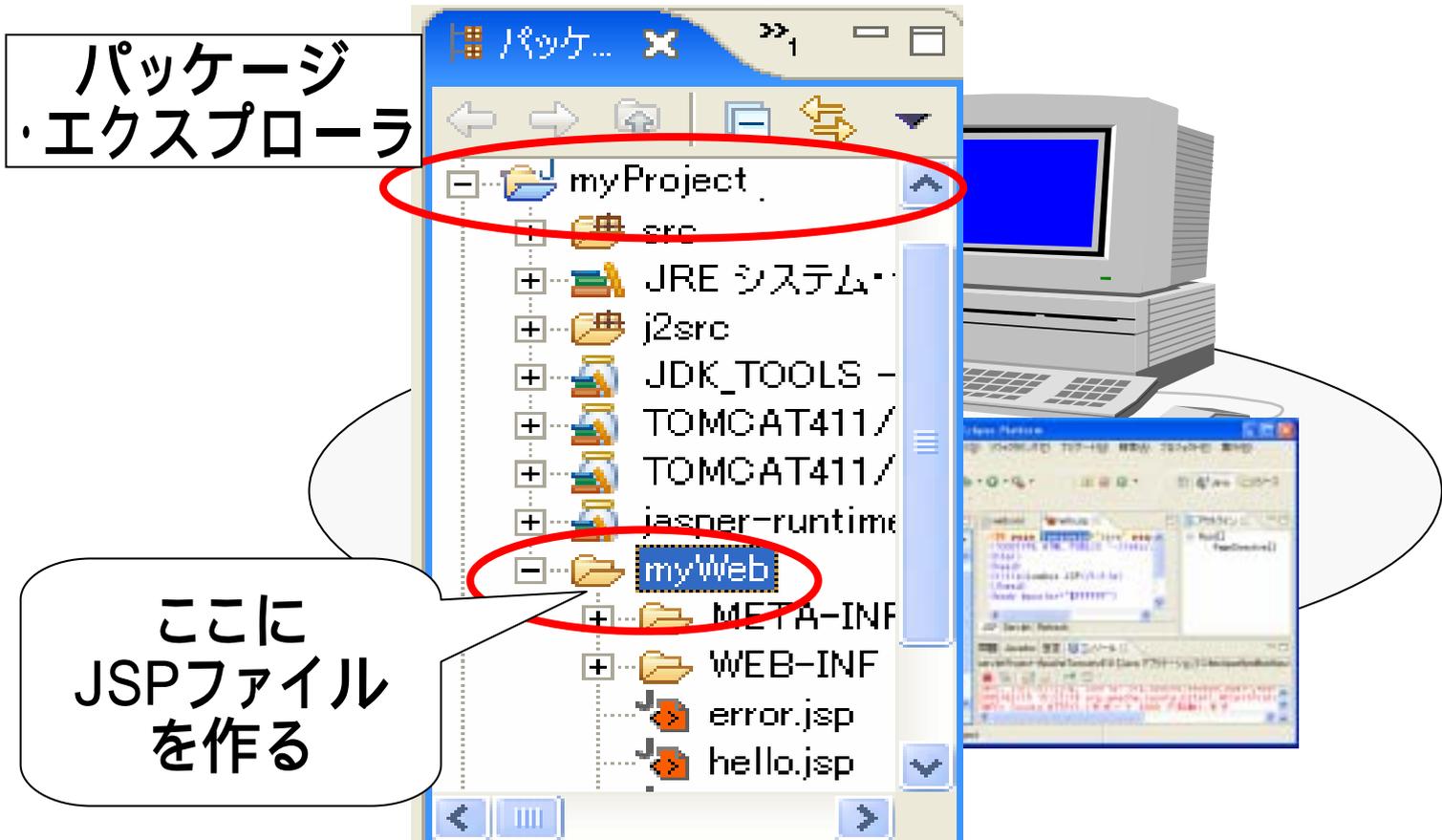
- eclipseを使う際に表示されるウィンドウのことを、ワークベンチと言います。ワークベンチの中の分割されたそれぞれの区画を、ビュー(view)と言います。
- このスライドを用いて行う作業では、「パッケージ・エクスプローラ」で作成するファイルの選択を行い、「エディター」にてファイルの編集を行います。



# (1.6) JSPプログラムの置き場所

■「パッケージ・エクスプローラ」では、作成するプログラムのおき場所を決めます。本スライドでは、場所にJSPファイルを置きます。

- JSPファイルの置き場所: [myProject]-[myWeb]



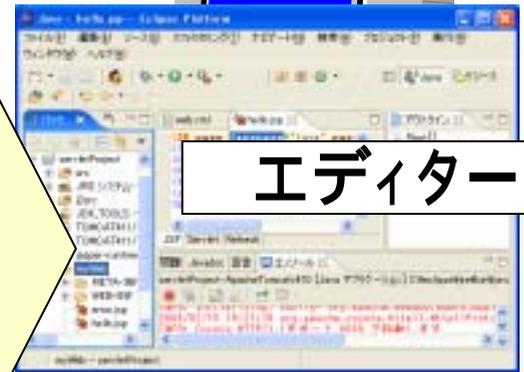
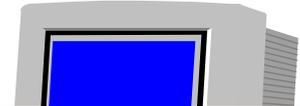
# (1.7) ブラウザでの表示

■前スライドの場所に作成したJSPを、「エディター」にて編集し、ブラウザで表示します。その表示のURLは、次のとおりです。

- URL: `http://localhost:8080/myApp/hello.jsp`

< Webクライアント >

< webサーバ >

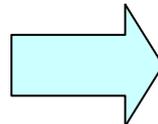


エディター

ブラウザ `http://localhost:8080/myApp/hello.jsp` でアクセス

■ここでの作業では、使用しているPCが、WebサーバとWebクライアントとの二役をこなします。

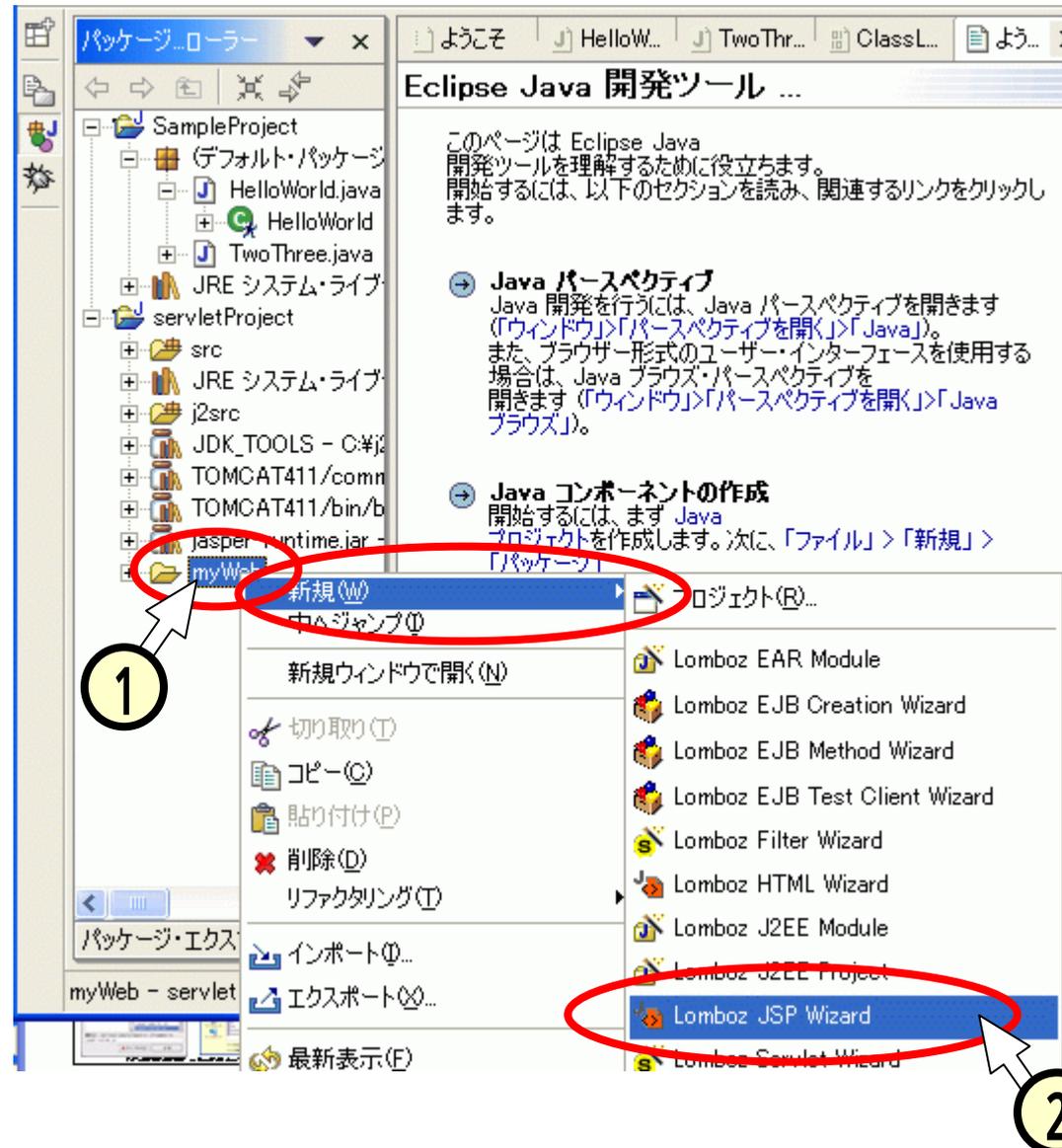
全部同じマシンで作業！



eclipse

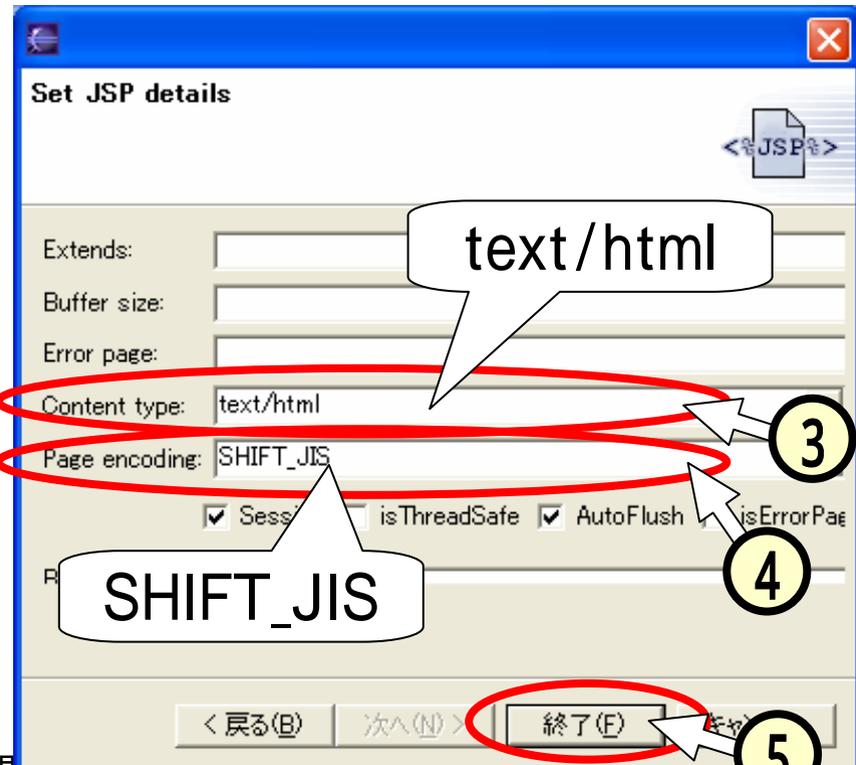
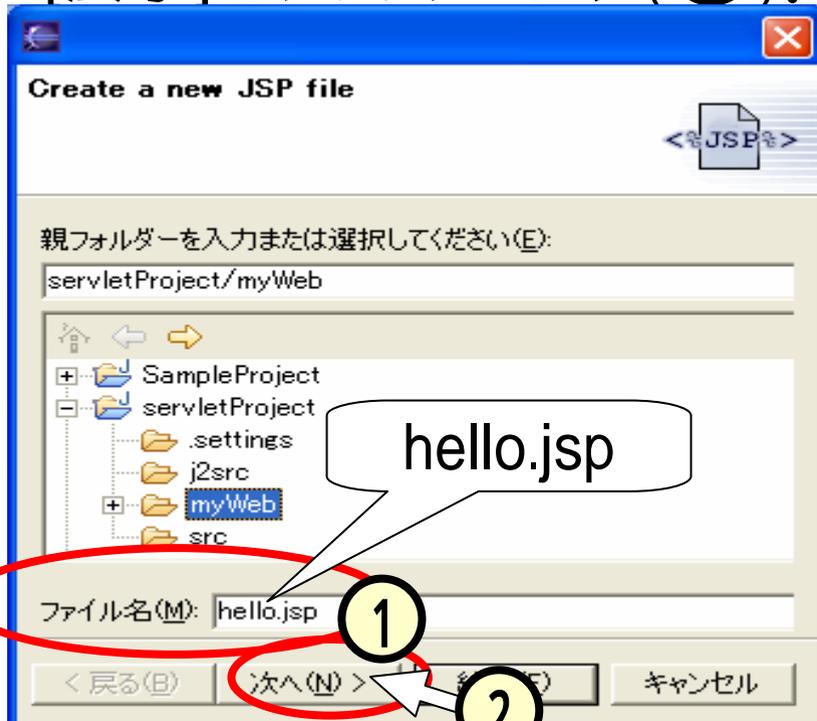
# (1.8.1) JSPの作成

- “Hello!”と表示する JSPを作成します。
- 「パッケージ・エクスプローラー」中の “myWeb” (①) を右クリックし、コンテキストメニューから [新規]- [Lomboz JSP Wizard] をクリック (②) します。



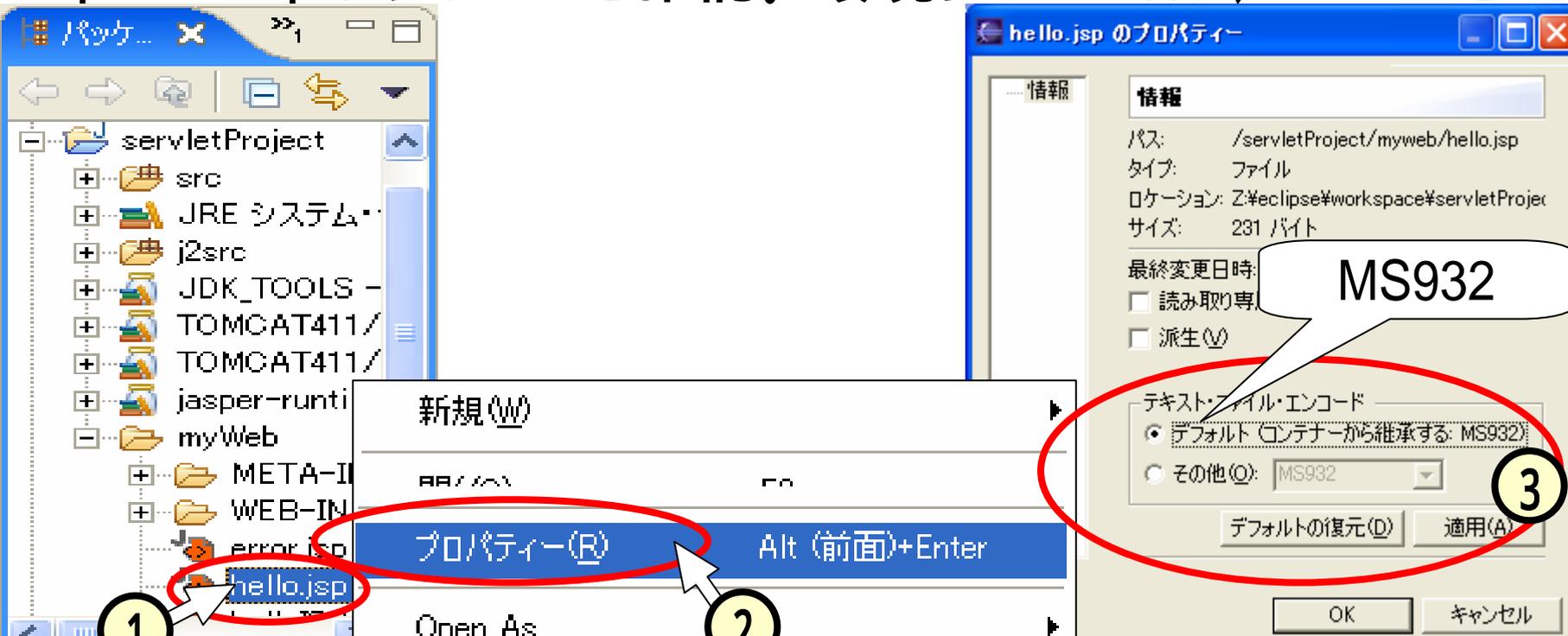
# (1.8.2) ウィザード

- 「Create a new JSP file」ウィンドウにて、ファイル名に“hello.jsp”を入力(①)し、[次へ]をクリック(②)します。
- 「Set JSP details」ウィンドウにて、次の通り入力します。
  - [Content type] “text/html”(③)
  - [Page encoding] “SHIFT\_JIS”(④)、環境により読替えてください)
- [終了]をクリックします(⑤)。



# (1.8.3) 文字エンコードの設定

- パッケージエクスプローラ中の“hello.jsp”の表示を右クリック(①)し、ポップアップメニューから[プロパティ]をクリック(②)します。
- 現れたダイアログ中で、文字エンコードが正しいかをチェックしてください(③)、SHIFT\_JISの場合は、[MS932]であることを確認。環境によっては、UTF-8を

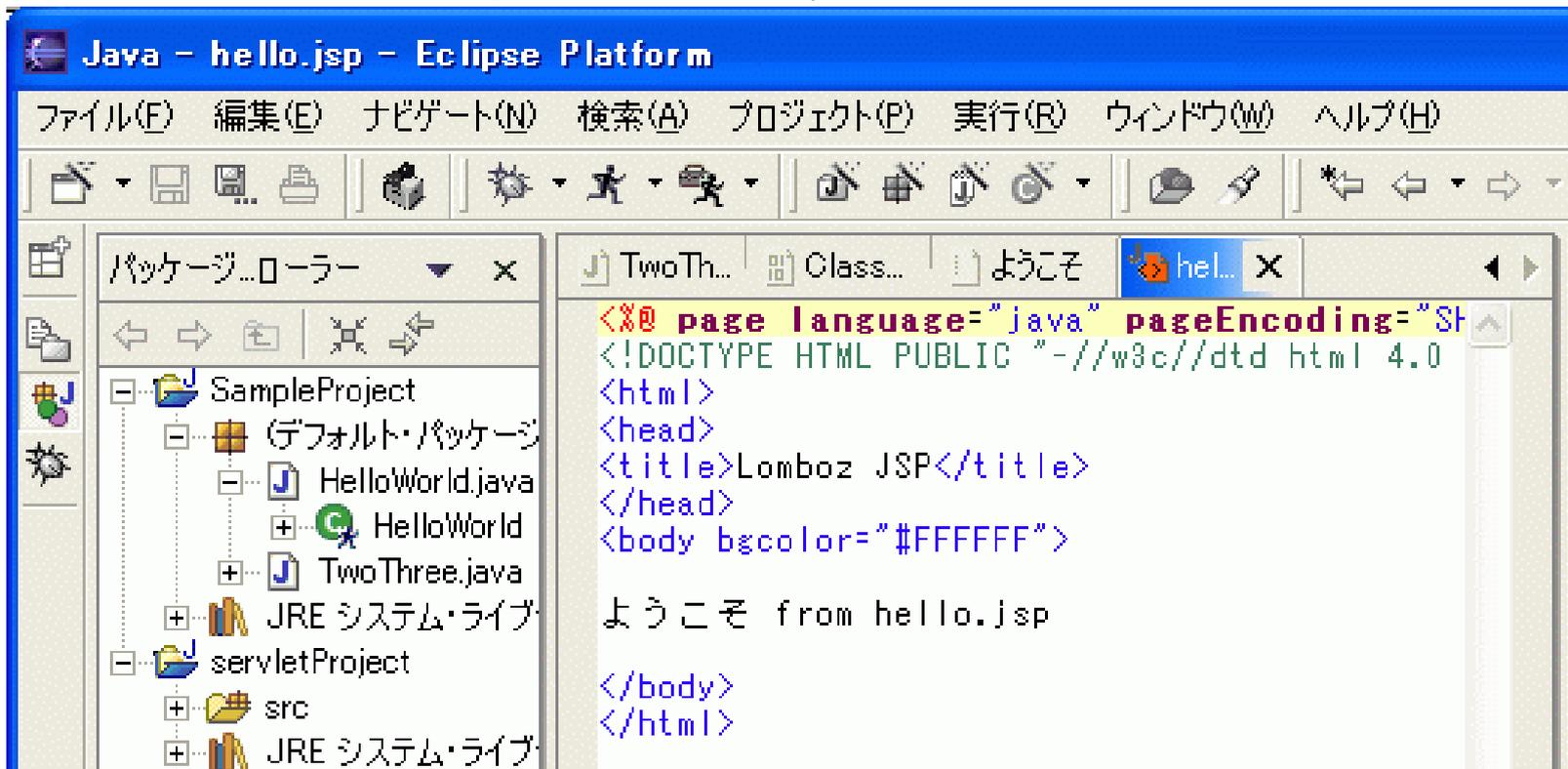


# ( 1 . 8 . 4 ) 編集

■エディターにて、次の書き換えを行います。

Write your content here      ようこそ from hello.jsp

■保存 ([Ctrl]+S押下) します。エラーがある場合は、このタイミングで表示されます。







# (1.10) ファイルの保存

- 以上のような手順で作成したJSPファイルは、使用しているPCのCドライブに保存されます。
- Cドライブで保存すると、誰かが消したりする恐れがあるため、これを個人のネットワークドライブに保存するようにします。下図のように、インポートとエクスポートとを行います。

保存する時、  
エクスポート  
(輸出する)。

読み込む時、  
インポート  
(輸入する)。



エクスポート

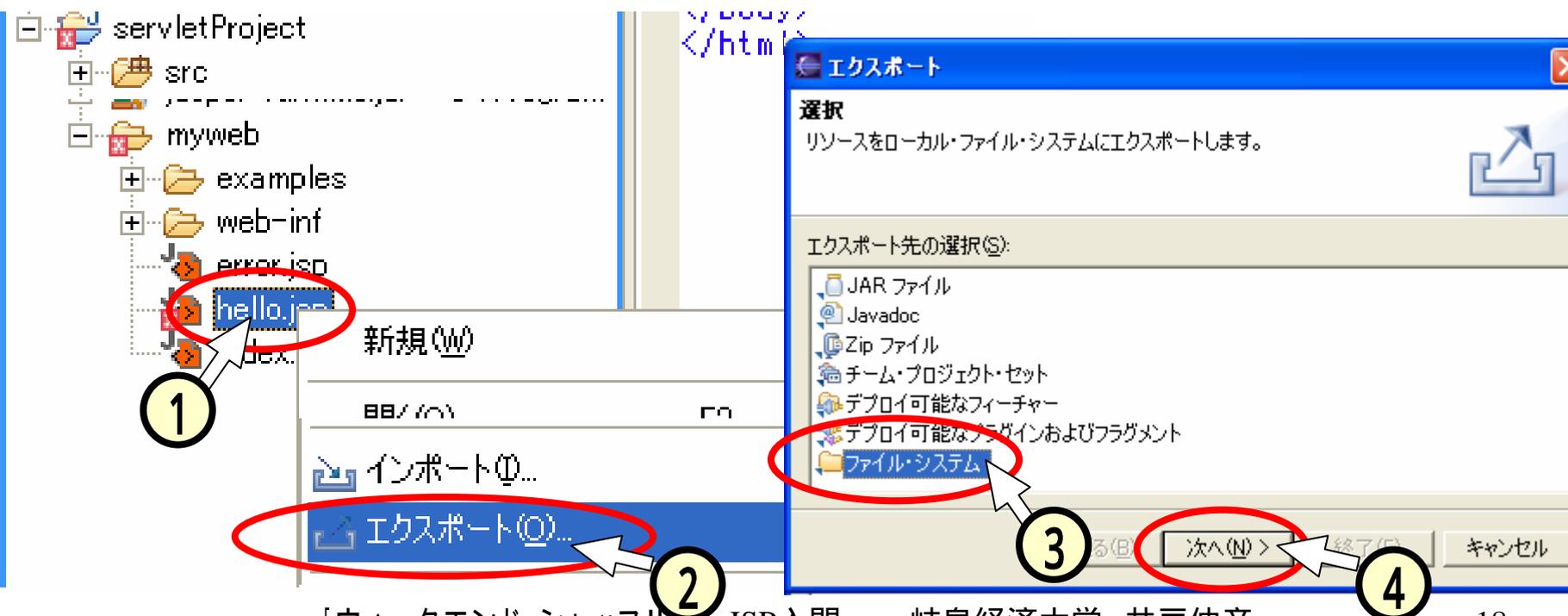
インポート

Cドライブ

ネットワーク  
ドライブ

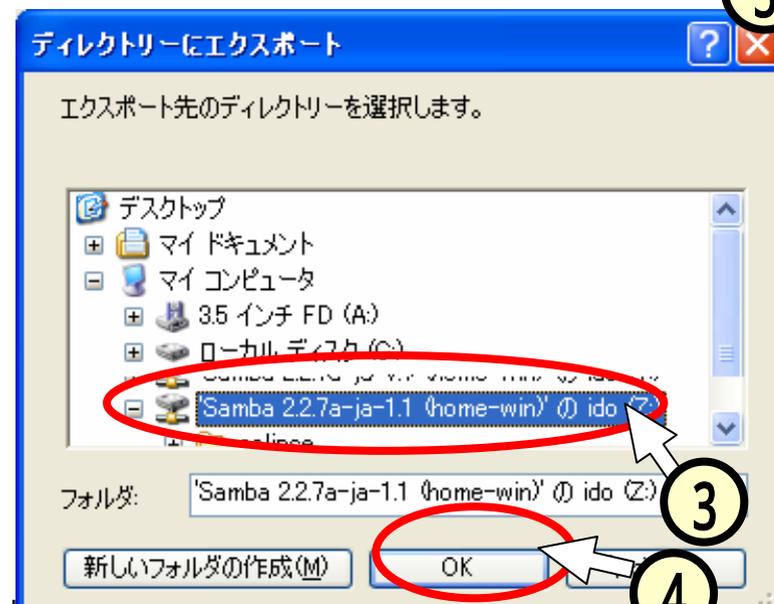
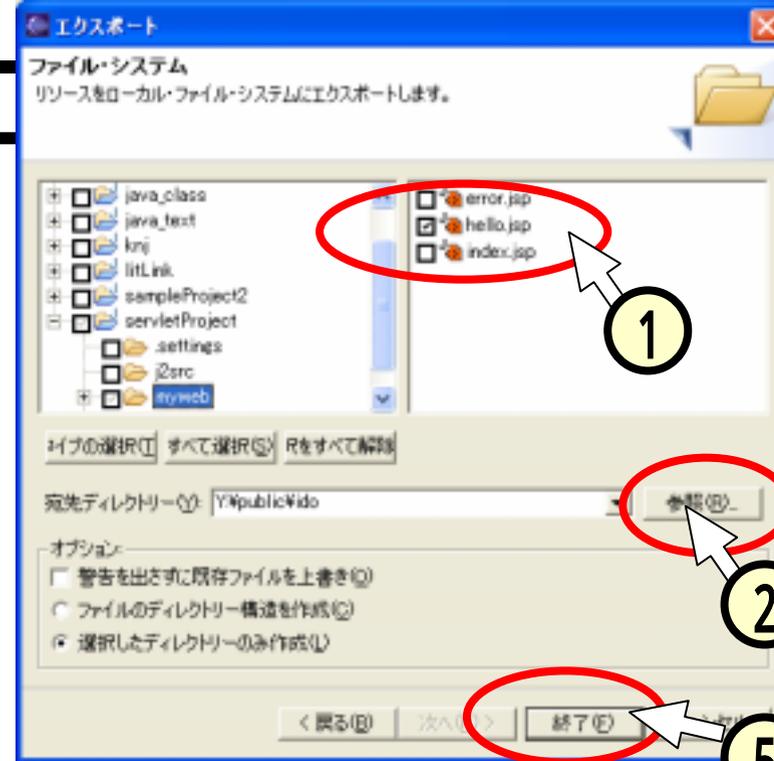
# (1.10.1) エクスポートー1ー

- ネットワークドライブは割り当て済みとします。
- エクスポートするファイルを右クリック(①)して、メニューからエクスポートをクリック(②)します。
- 「エクスポート」画面から、[ファイルシステム]を選択(③)して、[次へ]をクリック(④)します。



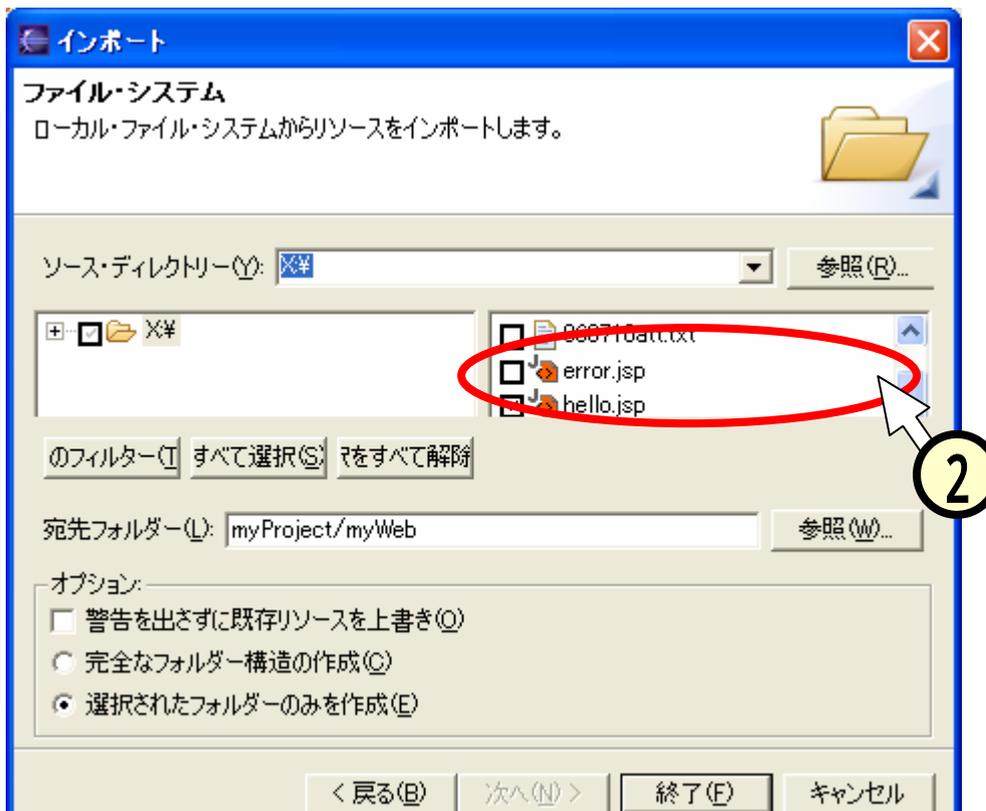
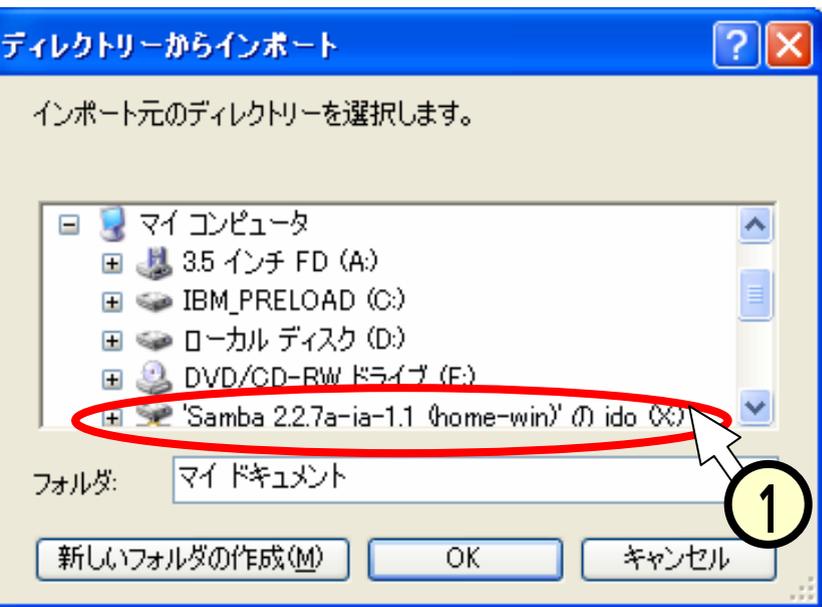
# (1.10.2) エクスポートー 2

- 「エクスポート」の画面の中で、エクスポートするファイルをチェック(①)します。
- 宛先ディレクトリーの[参照]ボタンをクリック(②)し、出てきた「ディレクトリーにエクスポート」の画面にて、自分のネットワークドライブ([マイコンピュータ]の配下)を選択(③)し、[OK]をクリック(④)します。
- 「エクスポート」の画面に戻って、[終了]をクリック(⑤)します。



# (1.10.3) インポート

- スライド(2.8.1) ~ (2.8.3)に記した方法で実施します。
- 但し、インポートするファイルの指定では、自分のネットワークドライブ(①)とそこに保存したファイル(②)を指定することになります。

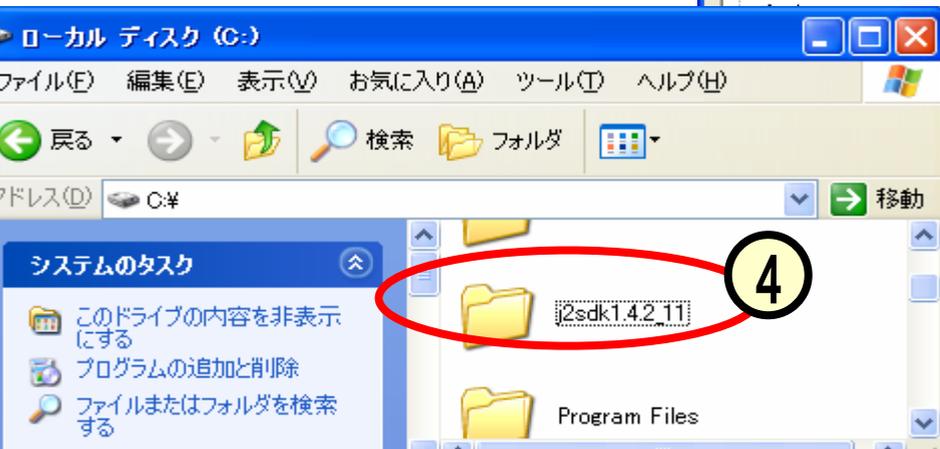
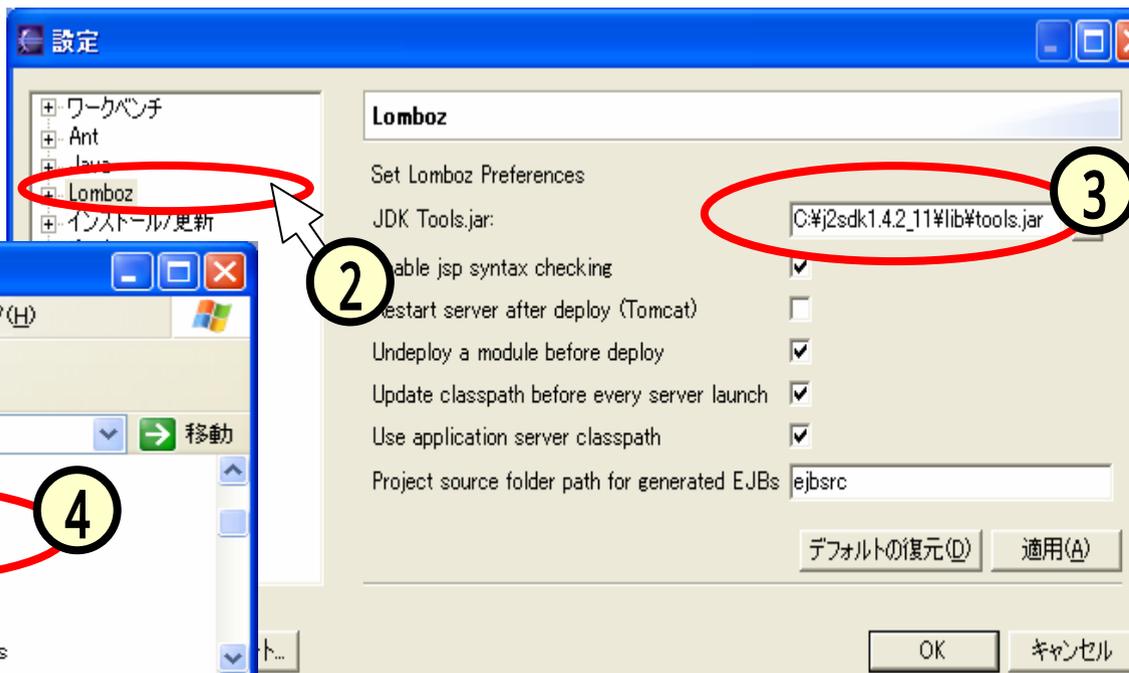


# (1.11) 注意事項

■ tomcatが起動出来ない場合、次の点をチェックしてください。

■ メニューから、[ウインドウ]-[設定]をクリック(①)。

■ [Lomboz]の項目をクリック(②)し、[JDK Tools.jar]欄のj2sdkの版数(③)を、Cドライブのj2sdkのフォルダ(④)と一致させる。



## (2) HTML(Hyper Text Markup Language)

- ここでは、HTMLにてページをつくり、(3)以下のJavaプログラムの利用の準備をします。
- HTMLについては、「(0)前提」に記したとおり、学習済みであることを前提としていますが、ここでは基本的な部分の説明を繰り返しています。
- HTMLをよくご存知の方は飛ばして頂いてOKです。
- 右のようなページを作ります。



# (2.1) HTML文書の基本構造

- HTML文書は、右図のような形をしています。
- ここで現れるタグは、<head></head>のようになっています。<head>を開始タグ、</head>を終了タグと呼び、この2つの間にタグで指定した内容(この場合はヘッダ)を書き込むことになります。
- <html>と<head>については説明しません。eclipseを用いてJSPを作成する際には、自動的に設定されているものを用います。

```
<html>
<head>
  文書の情報を記述
  タイトルを指定
</head>
<body>
  画面に表示する文章
  画像など(これから説明)
</body>
</html>
```

```
<%@ page language="java" pageEncoding="SHIFT_JIS" %>
<!DOCTYPE HTML PUBLIC "-//w3c//dtd/html 4.0 transitional//en">
<html>
<head>
<title>Lomboz JSP</title>
</head>
<body bgcolor="#FFFFFF">

井戸伸彦

</body>
</html>
```

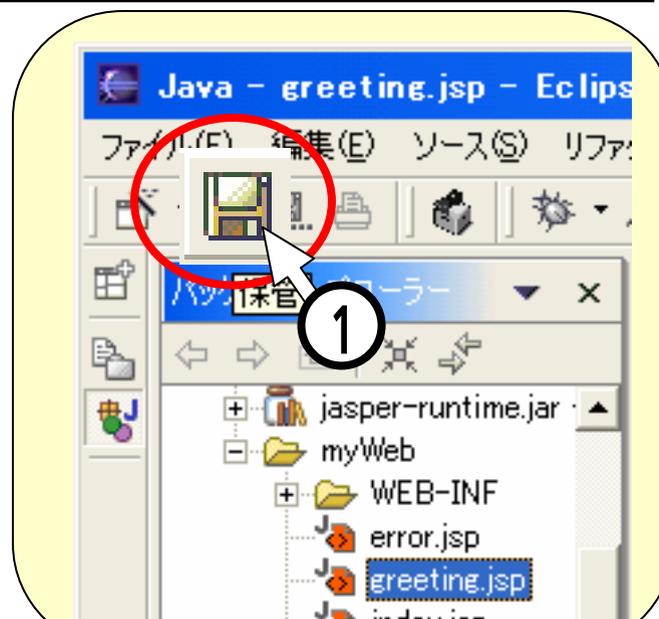
自動的に設定されたものをそのまま用いる。

## (2.2) JSPの保管、ブラウザの更新

■ここからの作業では、hello.jspと同様の方法で次の名前のJSPを編集して行きます。

- greeting.jsp

■編集が終わるたびに、JSPを保存し、上記JSPを表示したブラウザを更新して閲覧します。

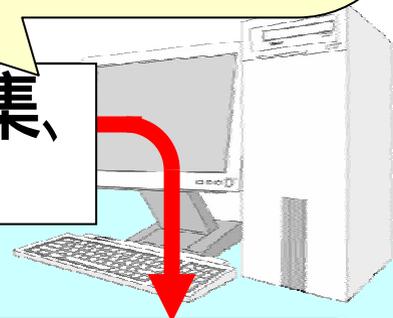


編集、  
保管

更新、閲覧

Webブラウザ

greeting.jsp

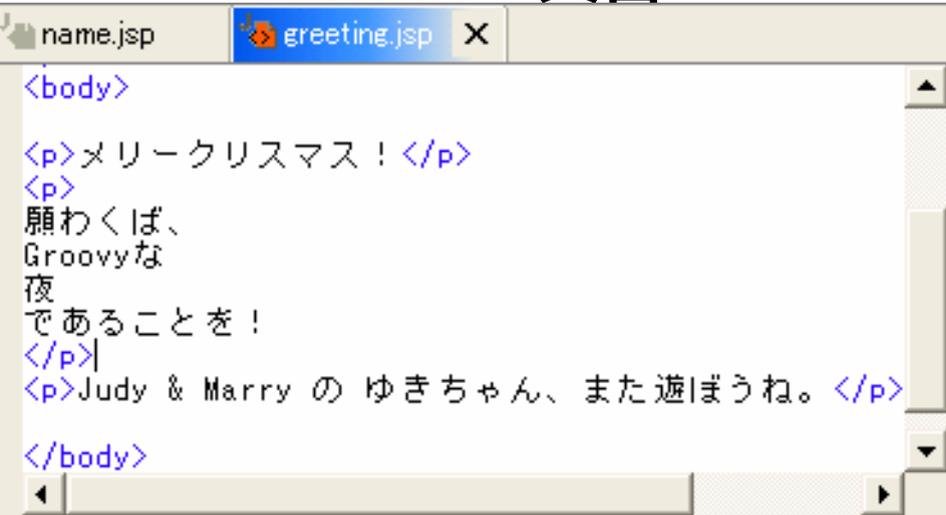


## (2.3) 文章を文節として入力する

- 次のように“<p>”と“</p>”のタグではさんで、文章のかたまり(文節)であることを指定します。
- “<p>”の“p”は、パラグラフ(文節)のpです。

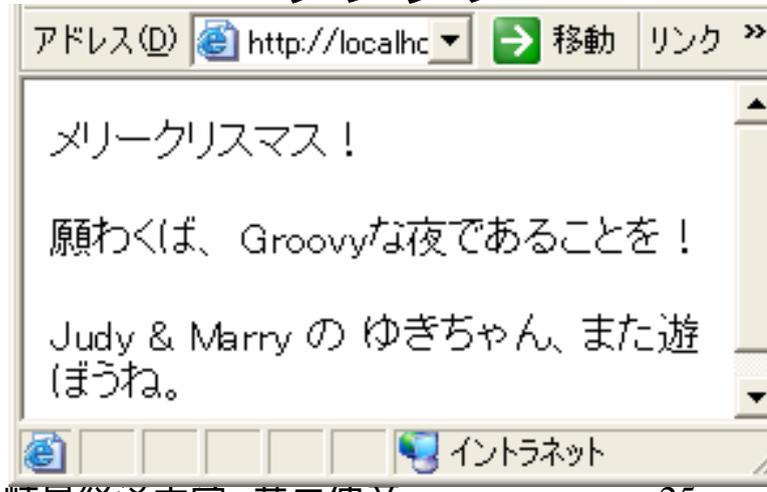
<p>文章のかたまり(文節)</p>

### < HTML文書 >



```
name.jsp  greeting.jsp X
<body>
<p>メリークリスマス！</p>
<p>
願わくば、
Groovyな
夜
であることを！
</p>
<p>Judy & Marry の ゆきちゃん、また遊ぼうね。</p>
</body>
```

### < ブラウザ >



## (2.4) 文字サイズを指定する

- 文字サイズを指定する際には、サイズを変更した部分を、次のようにタグで囲みます。

```
<font size="6">サイズを変更したい部分</font>
```

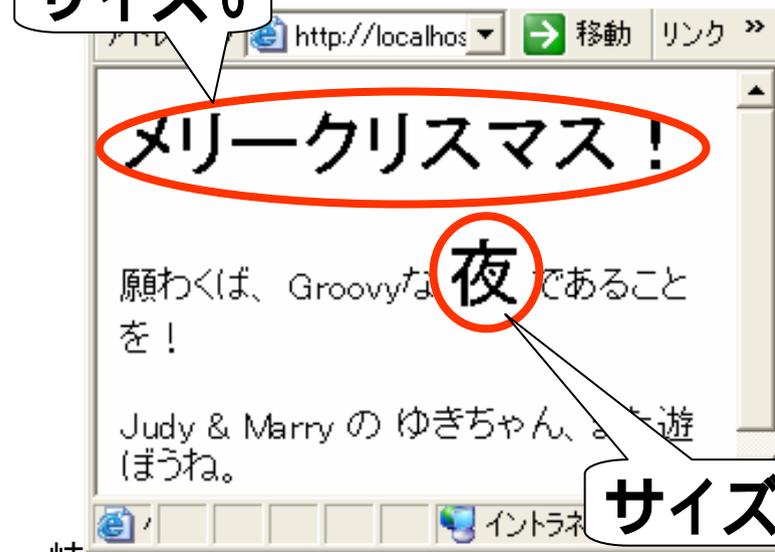
属性名 ← 属性値 (“ ” で囲む)

- サイズ以外にも、属性名 = “属性値” の形式で色も指定できます (次のスライド)。

### < HTML文書 >

```
name.jsp greeting.jsp x
<body>
<p><font size="6">メリークリスマス！</font></p>
<p>
願わくば、
Groovyな
<font size="6">夜</font>
であることせ！
</p>
<p>Judy & Marry の ゆきちゃん、また遊ぼうね。</p>
</body>
```

### サイズ6 < ブラウザ >



## (2.5) 文字の色を指定する

■色の指定も、タグ“<font>”の属性で指定します。

```
<font color="#FF5500">色を変更したい部分</font>
```

赤の成分を“FF”

緑の成分を“55”

青の成分を“00”

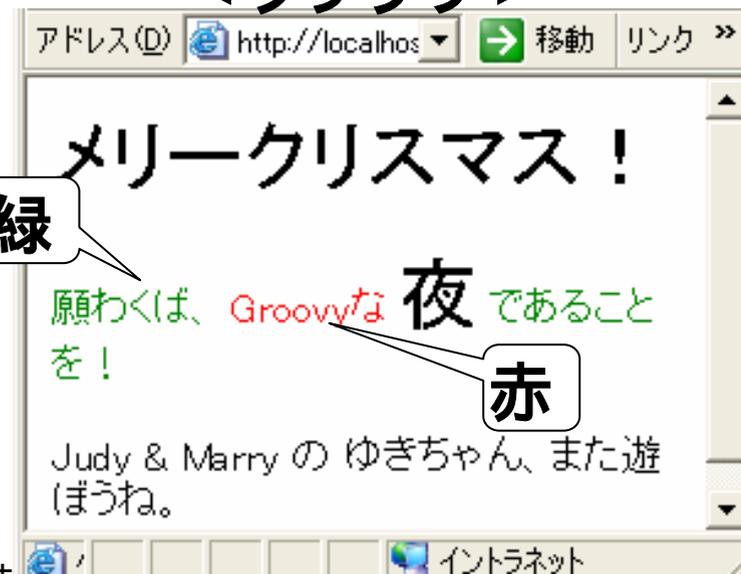
合計すると  
オレンジ

■色見本 (<http://tohoho.wakusei.ne.jp/wwwcolor.htm#Color> など)を見ると、番号と色の対応が判ります。

注: 基本的な色は、色の名前(“white”, “red”)でも指定できます。

<HTML文書>

```
name.jsp greeting.jsp x
<body>
<p><font size="8">メリークリスマス！</font></p>
<p>
<font color="green">願わくば、</font>
<font color="#FF0000">Groovyな</font>
<font size="8">夜</font>
<font color="green">であることを！</font>
</p>
<p>Judy & Marry の ゆきちゃん、また遊ぼうね。</p>
</body>
```



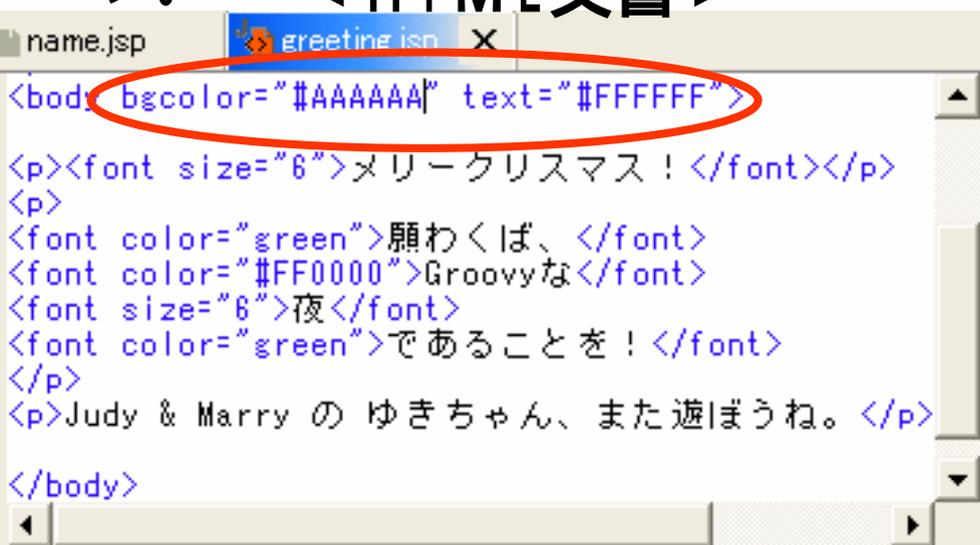
## (2.6) 背景と全体の文字の色とを指定する

- 背景、全体の文字の色は、タグ“<body>” (スライド(2.1)で説明したものです)の属性として指定します。

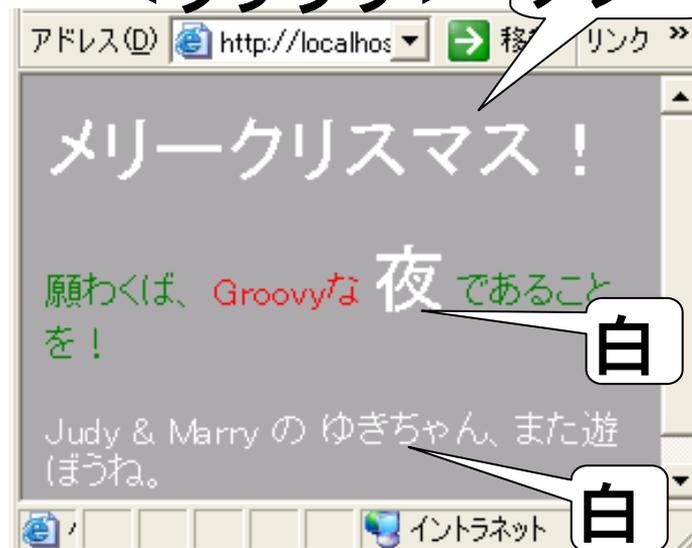
```
<body bgcolor="#888888" text="#FFFFFF">
```

“グレー”の背景に、“白”の字を書く

- タグ“<font>”の属性で指定した色は、そのまま残りま  
す。 <HTML文書>



```
name.jsp  greeting.jsp X
<body bgcolor="#AAAAAA" text="#FFFFFF">
<p><font size="6">メリークリスマス！</font></p>
<p>
<font color="green">願わくば、</font>
<font color="#FF0000">Groovyな</font>
<font size="6">夜</font>
<font color="green">であることを！</font>
</p>
<p>Judy & Marry の ゆきちゃん、また遊ぼうね。</p>
</body>
```



## (2.7) 画像を入れる

- タグ“<img>”を使って、挿入する画像ファイルを指定します。

```

```

- ファイル名は、ファイルが収めてあるディレクトリ(フォルダ)も含めて指定します。フォルダ“sports”の中の“ski.jpg”を指定する場合、“sports/ski.jpg”となります。
- 大きさ、枠など、たくさんの項目が設定できます。

### < HTML文書 >

```
name.jsp  greeting.jsp X
<body bgcolor="#AAAAAA" text="#FFFFFF">
<p><font size="6">メリークリスマス！</font></p>
<p>
<font color="green">願わくば、</font>
<font color="#FF0000">Groovyな</font>
<font size="6">夜</font>
<font color="green">であることを！</font>
</p>

<p>Judy & Marry の 紗ちゃん、また廻らね。</p>
</body>
```

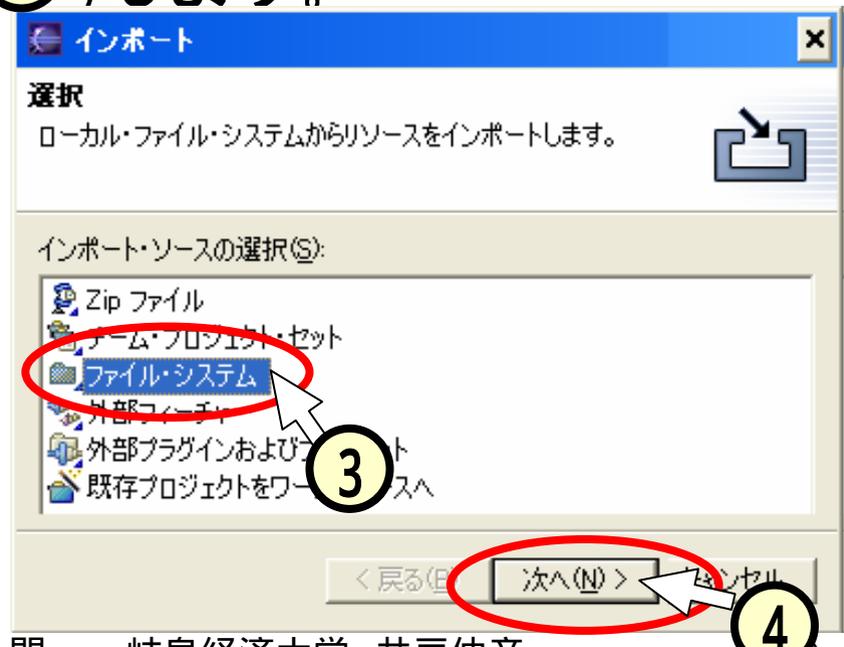
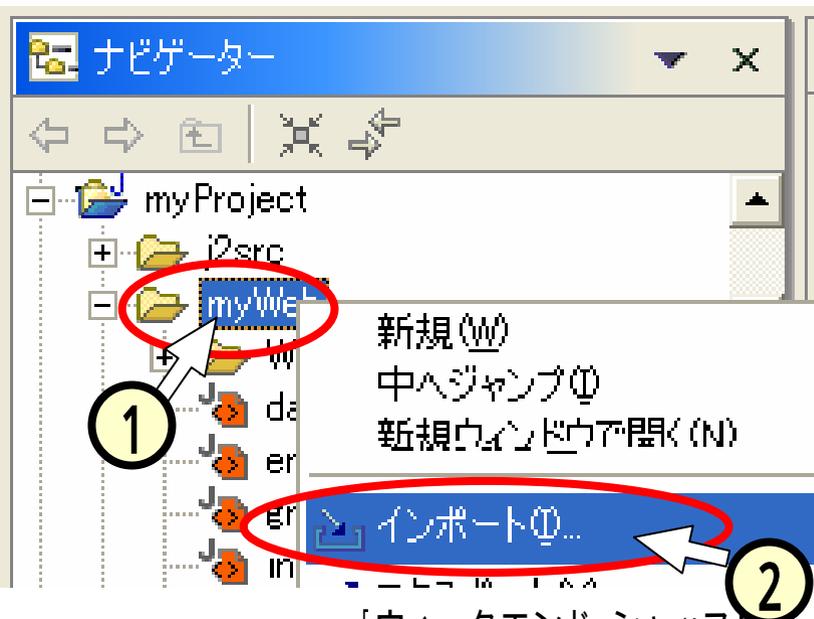
### < ブラウザ >



画像

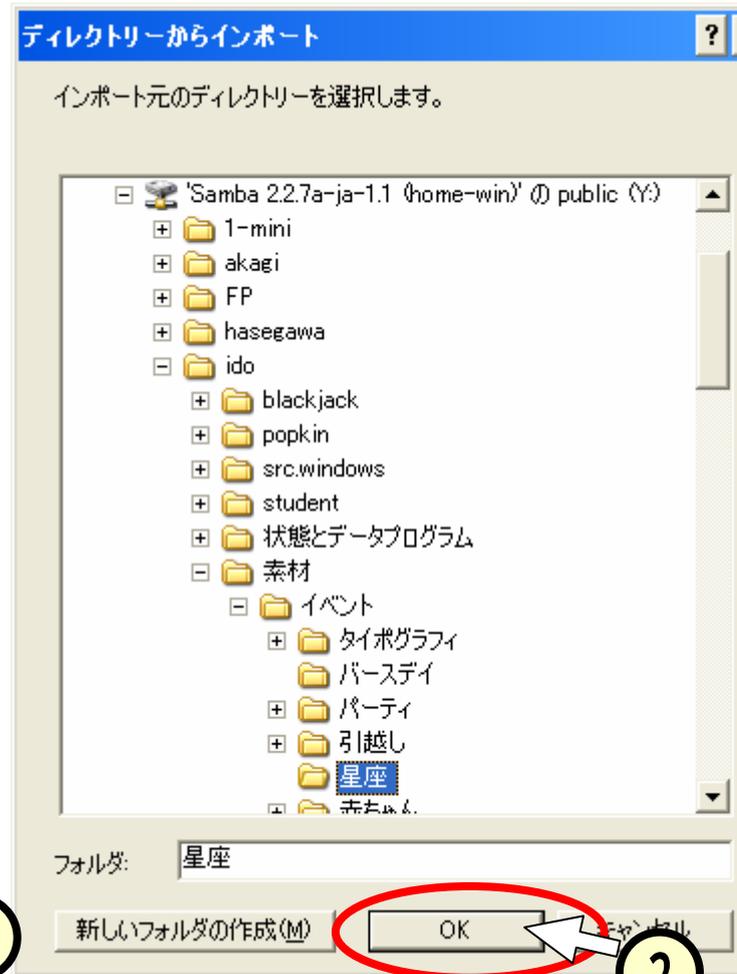
## (2.8.1) インポート - 1 -

- 取り込む画像のファイルは、予めインポート(輸入)します。
- パッケージ・エクスプローラー中の、“myWeb”を右クリック(①)して、[インポート]をクリック(②)します。
- 「選択」のウィンドウにて、[ファイルシステム]をクリック(③)し、[次へ]をクリック(④)します。



# (2.8.2) インポート - 2 -

- 「ソースフォルダー」の右の[ブラウズ]ボタンをクリック(①)します。
- 次のようにフォルダを順に開き、[OK]をクリック(②)します。
  - [マイコンピュータ]  
[‘Samba 2.2.7a-ja-1.1 (samba)’ の public]  
[ido] [素材]  
[イベント] [星座]

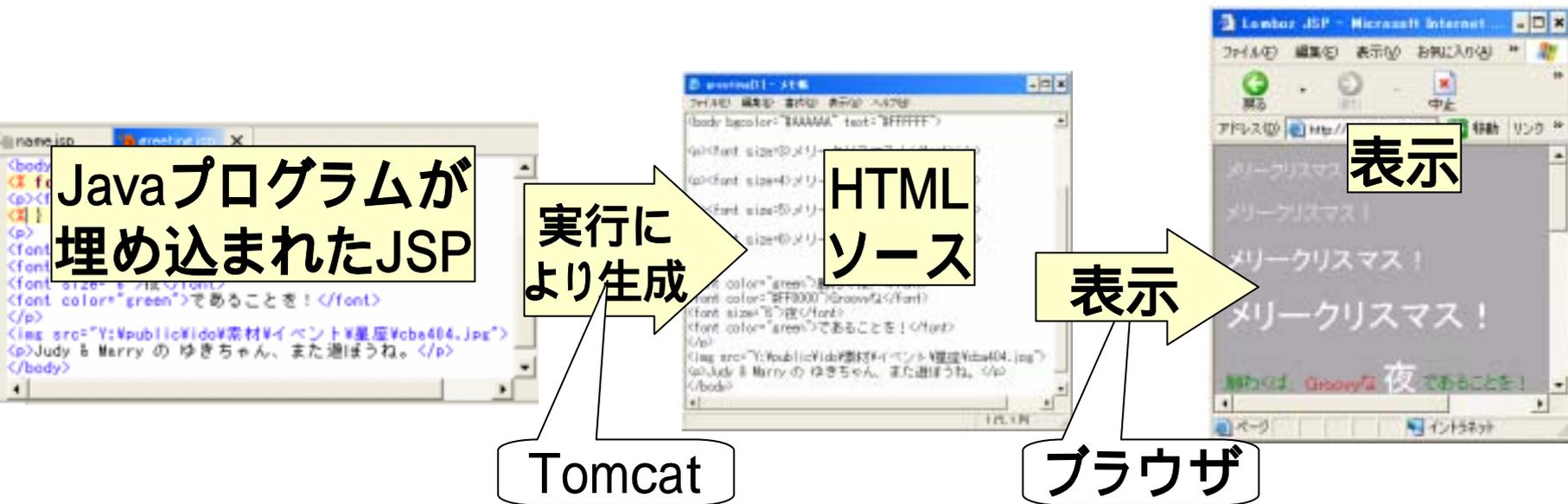




# (3) JSPでプログラミング！

■JSPでのプログラミングは次のような仕掛けにより動きます。

- Javaによるプログラムを、JSPに埋め込む。
- JSPを実行すると、ソースとなるHTMLの文書が出来る。
- HTMLの文書がブラウザ上で実行される。



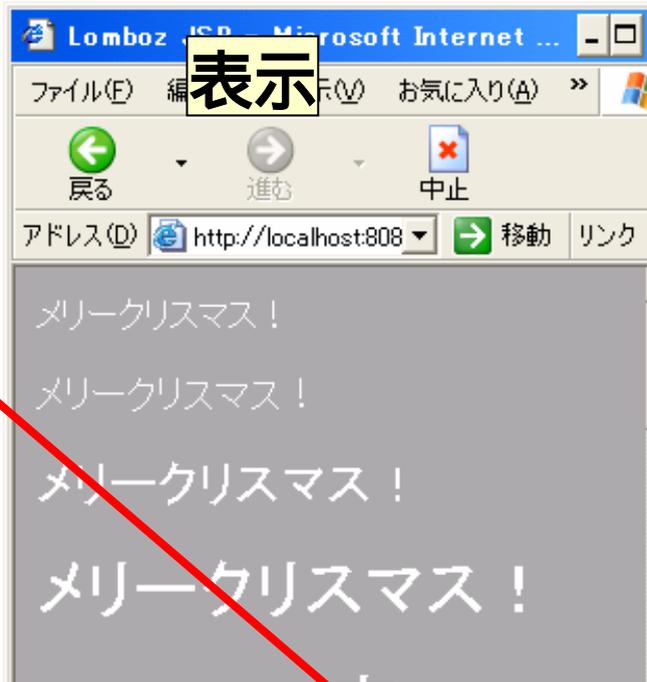
# (3.1) 繰り返し

- 最初にプログラムにより、“メリークリスマス！”の部分  
を、大きさを変えて繰り返し表示してみます。

Javaプログラムが  
埋め込まれたJSP

```
name.jsp
<body bgcolor="#AAAAAA" text="#FFFFFF">
<% for(int i=3;i<7;i++){ %>
<p><font size=<%= i%>>メリークリスマス！</font></p>
<% } %>
<p>
<font color="green">願わくば、</font>
<font color="#FF0000">Groovyな</font>
<font size="6">夜</font>
<font color="green">であることを！</font>
</p>

<p>Judy & Marry の ゆきちゃん、また遊ぼうね。</p>
</body>
```



```
(略)
<% for(int i=3;i<7;i++){ %>
<p><font size=<%= i%>>メリークリスマス！ </font></p>
<% } %>
```

<p>  
(略)

夜であることを！  
イントラネット



# (3.3) JSPとHTMLソースの対応

■両者のHTML部分( )はよく対応しています。

< JSP >

(略)

• `<body bgcolor="#AAAAAA" text="#FFFFFF">`

• `<% for(int i=3;i<7;i++){ %>`

• `<p><font size=<%= i %>>メリークリスマス! </font></p>`

• `<% } %>`

• `<p>`

• `<font color="green">願わくば、 </font>`

(略)

実行により生成

< HTMLのソース >

(略)

• `<body bgcolor="#AAAAAA" text="#FFFFFF">`

• `<p><font size=3>メリークリスマス! </font></p>`

• `<p><font size=4>メリークリスマス! </font></p>`

• `<p><font size=5>メリークリスマス! </font></p>`

• `<p><font size=6>メリークリスマス! </font></p>`

• `<p>`

• `<font color="green">願わくば、 </font>`

(略)

## (3.4) スクリプトレット

- JSP中のJavaプログラムを“スクリプトレット”と呼び、次のように“<%”と“%>”とで囲みます。

```
<% Javaのプログラム(スクリプトレット) %>
```

- スクリプトレットの中身だけ取り出すと、次のようになります。つまり、繰り返しのプログラムを書き、これを“<% %>”で囲って、挟んだHTML部分を繰り返させています。

```
1. for(int i=3;i<7;i++){  
2. ここを繰り返す  
3. }
```

<スクリプトレットの中身>

```
1. <p><font size=3>メリー  
   クリスマス! </font></p>  
2. <p><font size=4>メリー  
   クリスマス! </font></p>  
3. <p><font size=5>メリー  
   クリスマス! </font></p>  
4. <p><font size=6>メリー  
   クリスマス! </font></p>
```

## ( 3 . 5 ) 式 ( Expression )

- 繰り返されるHTMLの部分は、単純なHTML部分ではなく、「式 ( Expression )」が含まれており、次の形をしています。

```
<%= 式(Expression) %>
```

- JSPでの「式」は、スクリプトレットの文脈の中でのその値が、HTMLソースに現れるようになっています。

```
1. <% for(int i=3;i<7;i++){ %>  
2. <p><font size="<%= i %>">メリークリスマス! </font></p>  
3. <% } %>
```

“<%= %>” で囲まれた部分の“i”は、  
「for(int i=3;i<7;i++){」の文脈で解釈され、値が、“3”、“4”、“5”、“6”となる。

```
1. <p><font size=3>メリークリスマス! </font></p>  
2. <p><font size=4>メリークリスマス! </font></p>  
3. <p><font size=5>メリークリスマス! </font></p>  
4. <p><font size=6>メリークリスマス! </font></p>
```

## ( 3 . 6 ) J a v a プログラミングについて

---

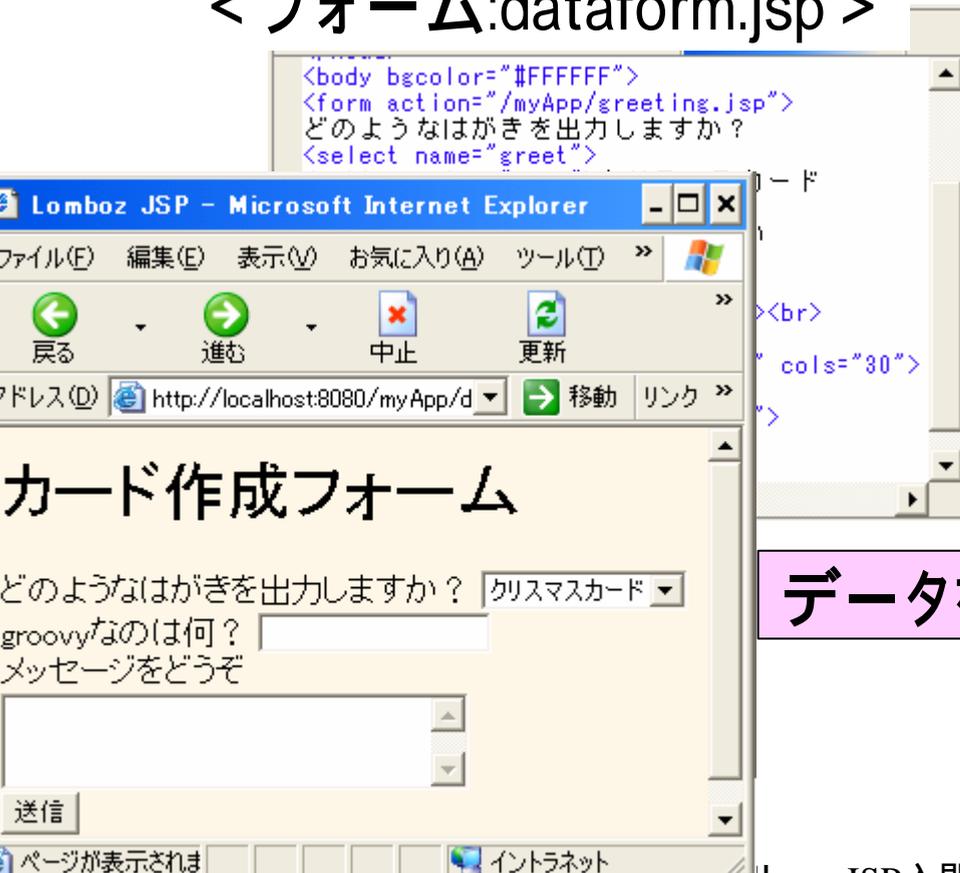
- プログラミングの学習は、ある程度時間を費やして行うべきものであり、この場では扱いません。
- Javaについては、次のスライドにて説明しています。
  - 「シャバドゥビ、ジャバ - Java覚書 - 」  
[http://www.gifu-keizai.ac.jp/ido/doc/java/java\\_text.pdf](http://www.gifu-keizai.ac.jp/ido/doc/java/java_text.pdf)
- Javaについてあまり詳しくない状態で本スライドを実施されている方は、上記資料を参照して雰囲気はわかって頂ければ結構です(あるいは講師が説明します)。

# (4) フォーム

- ここでは、HTMLのフォームを用いて、先ほど作成したJSP”greeting.jsp”へ、データを送るためのページ(dataform.jsp)を作成します。

< フォーム:dataform.jsp >

```
<body bgcolor="#FFFFFF">
<form action="/myApp/greeting.jsp">
  どのようなはがきを出力しますか？
  <select name="greet">
```



カード作成フォーム

どのようなはがきを出力しますか？

groovyなのは何?

メッセージをどうぞ

データを送る



メリークリスマス！

メリークリスマス！

メリークリスマス！

緑色のは、Groovyな夜であることを！

greeting.jsp (JSP)

TAURUS

# (4.1) 作成するフォーム (dataform.jsp)

```
<body bgcolor="##FDF5E6">  
<p><font size="6">カード作成フォーム  
</font></p>  
<form action="/myApp/greeting.jsp">  
  どのようなはがきを出力しますか？  
<select name="greet">  
<option value="xmas">クリスマスカード  
<option value="newyear">年賀状  
<option value="summer">書中見舞い  
</select><br>  
  groovyなのは何？  
<input type="text" name="groovy"><br>  
  メッセージをどうぞ<br>  
<textarea name="message" rows="3"  
  cols="30">  
</textarea><br>  
<input type="submit" value="送信">  
</form>  
</body>
```

Lomboz JSP - Microsoft Internet Explorer

ファイル(F) 編集(E) 表示(V) お気に入り(A) ツール(T) »

戻る 進む 中止 更新

アドレス(D) http://localhost:8080/myApp/d 移動 リンク »

## カード作成フォーム

どのようなはがきを出力しますか？

groovyなのは何？

メッセージをどうぞ

送信

ページが表示されま イントラネット

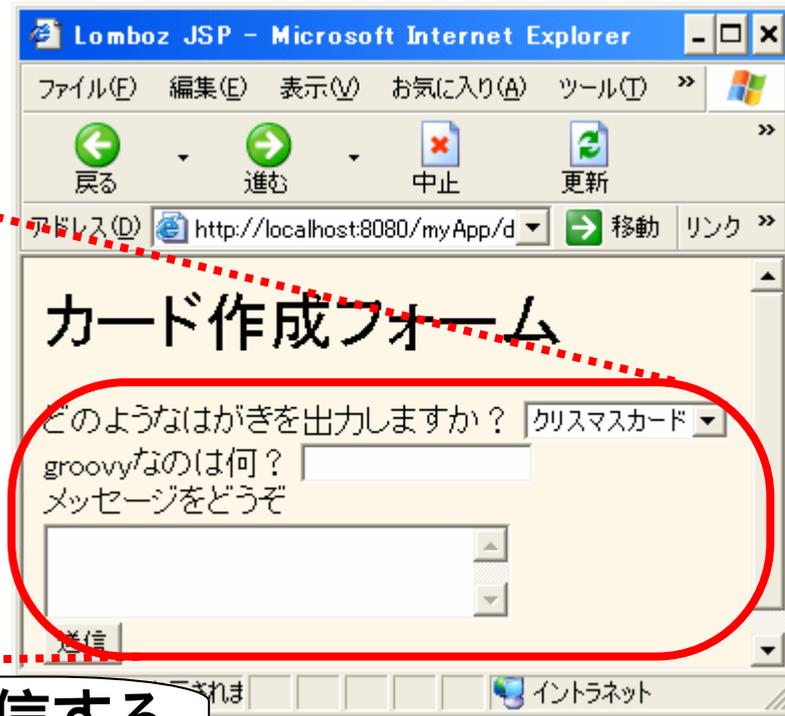
## (4.2) フォーム要素の構成

- フォームについても学習済みであることを前提としていますが、ここでは説明を繰り返します。
- タグ“<form>”と“</form>”に囲まれた部分で、送信するデータを指定する部分を作ります。

ここでは、  
データを送る先のJSP

```
<form action="送信先">  
  HTML要素  
  HTML要素  
  :  
  HTML要素  
</form>
```

入力画面と、それにより送信する  
データについて書く



# (4.3) テキスト・フィールド

■HTML要素:

```
<input type="text" name="...">
```

■属性: name

フィールドに入力されたデータの名前

value

フィールドに予め入れておく文字列

```
<input type="text"  
name="groovy">
```

このように書くと。。

テキストを入力する箱が作成され、  
(ここでは“夜”と入力したとする)

プログラムでは、  
“groovy=夜”を利用して  
処理をする

groovy/何は何

夜

“groovy=夜”とい  
うデータが送られ、

インター  
ネット

groovyWeb

greeting.jsp

<Webサーバ>

# (4.4) 送信ボタン

- HTML要素: `<input type="submit" value="...">`
- 属性: value      ボタンの上のラベルとなる

```
<input type="SUBMIT" value="送信">
```

このように書くと。。。

ボタンが表示され、

クリックすると、



データが送られる

インターネット

<Webサーバ>



# (4.5) テキスト・エリア

■テキスト・エリア: 複数行のテキスト領域

■HTML要素: `<textarea name="..."></textarea>`

■属性: name エリアに入力されたデータの名前

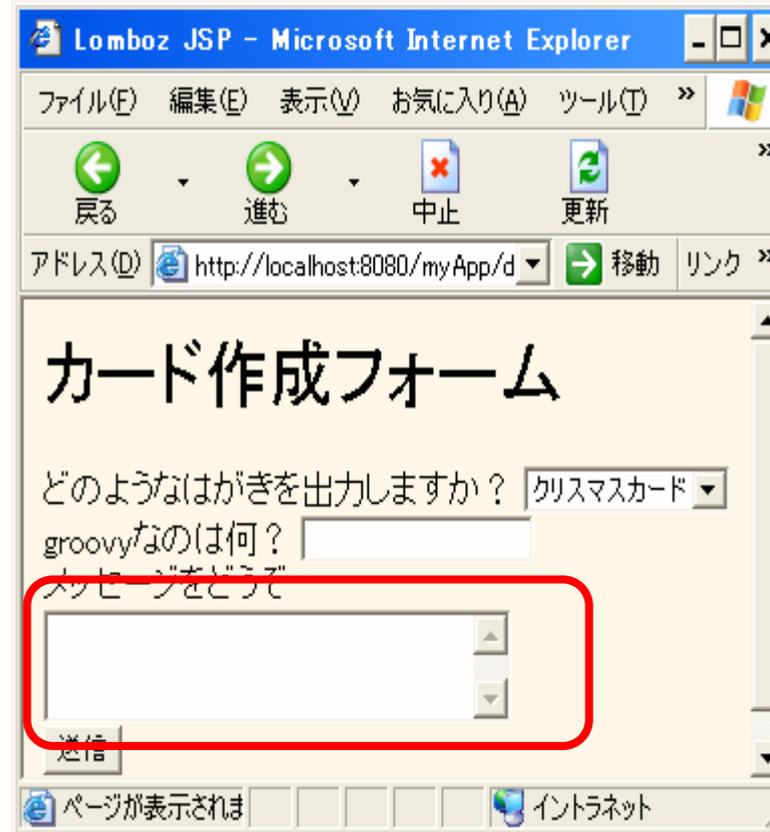
rows 行の数

cols 行の長さ

■動作は、テキストフィールドと同じです。

```
<textarea name="message"
  rows="3" cols="30">
</textarea><br>
```

“<br>”は改行です



# (4.6) コンボ・ボックス

■コンボ・ボックス: 選択項目から、ひとつをリストで選択

■HTML要素 `<select name="..."><option value="...">..</select`

どのようなはがきを出力

```
<select name="greet">
```

```
<option value="xmas">クリスマスカード
```

```
<option value="newyear">年賀状
```

```
<option value="summer">書中見舞い
```

```
</select><br>
```

このように書くと...



選択が可能となり

クリックすると、

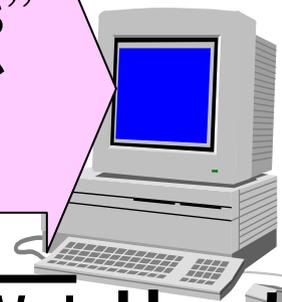
“greet=xmas”  
というデータが  
送られる

コンボボックスが表示され、

インター  
ネット

<Webサーバ>

力しますか？





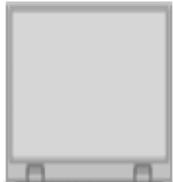
## ( 5 . 2 ) greeting.jsp

```
1. <body bgcolor="#AAAAAA" text="#FFFFFF">
2. <% request.setCharacterEncoding("Shift_JIS");
3.     String dataGreet=request.getParameter("greet");
4.     String dataGroovy=request.getParameter("groovy");
5.     String dataMessage=request.getParameter("message");
6.     String msg="";
7.     if(dataGreet.equals("xmas")){
8.         msg="メリークリスマス！";
9.     }else if(dataGreet.equals("newyear")){
10.        msg="あけましておめでとう！";
11.    }else if(dataGreet.equals("summer")){
12.        msg="暑中お見舞い";
13.    }
14.    for(int i=3;i<7;i++){ %>
15. <p><font size=<%= i%>><%= msg%> ! </font></p>
16. <% } %>
17. <p><font color="green">願わくば、 </font>
18.     <font color="#FF0000">Groovyな</font>
19.     <font size="6"><%= dataGroovy %></font>
20.     <font color="green">であることを！ </font></p>
21. 
22. <p><%= dataMessage%></p>
```

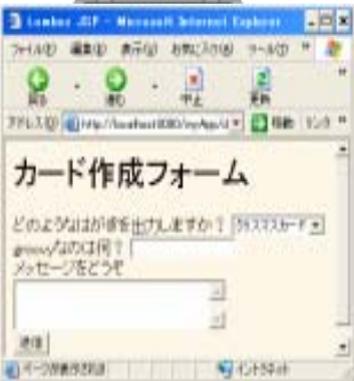
# (5.3) リクエスト・データの読み出し

■JSPでは、次のようなデータの読み出しが行われます。

```
2. <% request.setCharacterEncoding("Shift_JIS");  
3.   String dataGreet=request.getParameter("greet");  
4.   String dataGroovy=request.getParameter("groovy");
```



クライアント  
PC



出力しますか？

">

">クリスマスカード

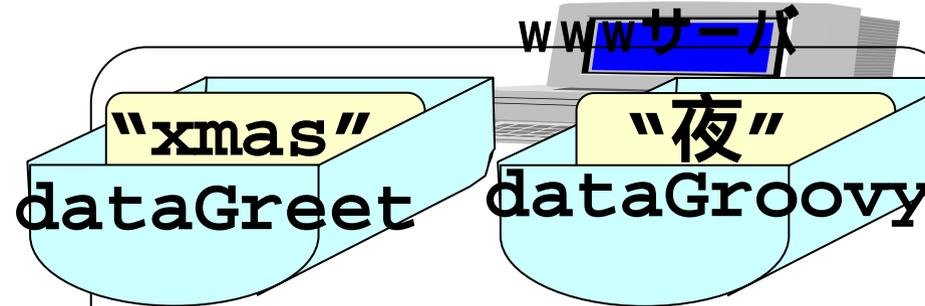
<option value="newyear">年賀状

<option value="summer">書中見舞い

</select><br>

groovyなのは何？

<input type="text" name="groovy"><br>



dataGreet	xmas
dataGroovy	夜

?greet="xmas"  
&groovy="夜"



## (5.4) 分岐による変数msgへの設定

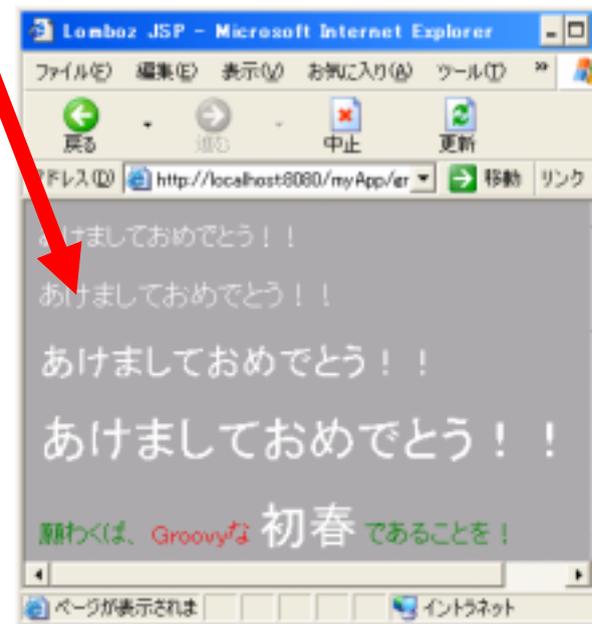
- 変数“dataGreet”に読み取った値を判定して、変数“msg”に“メリークリスマス！”などの文字列を設定します。

```
6.   String msg="";
7.   if(dataGreet.equals("xmas")){
8.       msg="メリークリスマス！";
9.   }else if(dataGreet.equals("newyear")){
10.      msg="あけましておめでとう！";
11.  }else if(dataGreet.equals("summer")){
12.      msg="暑中お見舞い";
13.  }
```

# ( 5 . 5 ) 表示

- 変数msgに設定された値が、繰り返し表示されます。

```
14.    for(int i=3;i<7;i++){ %>  
15.<p><font size=<%= i%>><%= msg%> ! </font></p>  
16.<% } %>
```



# (6) 例題1: カード作成の拡張

- 繰り返しの回数と、画像の選択とを追加します。
- 出来る方は、フォームにて色を指定することをやってみてください。

カード作成フォーム

どのようなはがきを出力しますか?

繰り返しの回数は?

groovyなのは何?

イラストを選択してください。

メッセージをどうぞ



## ( 6 . 2 ) dataform.jsp

```
<body bgcolor="#FDF5E6">
<p><font size="6">カード作成フォーム</font></p>
<form action="/myApp/greeting.jsp">
  どのようなはがきを出力しますか？
  <select name="greet">
    <option value="xmas">クリスマスカード
    <option value="newyear">年賀状
    <option value="summer">書中見舞い
  </select><br>
  繰り返しの回数は？
  <input type="text" name="repeat"><br>
  groovyなのは何？
  <input type="text" name="groovy"><br>
  イラストを選択してください。
  <select name="constellations">
    <option value="0">taurus(おうし座)
    <option value="1">gemini(ふたご座)
    <option value="2">cancer(かに座)
  </select><br>
  メッセージをどうぞ<br>
  <textarea name="message" rows="3" cols="30">
  </textarea><br>
  <input type="submit" value="送信">
</form>
</body>
```

# ( 6 . 3 ) greeting.jsp 一前半一

```
1. <body bgcolor="#AAAAAA" text="#FFFFFF">
2. <% String constellations[] = {
3.     "cba404.jpg",
4.     "cba504.jpg",
5.     "cba604.jpg",
6. };
7.     request.setCharacterEncoding("Shift_JIS");
8.     String dataGreet=request.getParameter("greet");
9.     String dataGroovy=request.getParameter("groovy");
10. String
dataMessage=request.getParameter("message");
11. int
dataRepeat=Integer.parseInt(request.getParameter("rep
eat"));
12. int
itemp=Integer.parseInt(request.getParameter("constell
ations"));
13. String img=constellations[itemp];
```

## ( 6 . 4 ) greeting.jsp ー後半ー

```
1. String msg="";
2. if(dataGreet.equals("xmas")){
3.     msg="メリークリスマス!";
4. }else if(dataGreet.equals("newyear")){
5.     msg="あけましておめでとう!";
6. }else if(dataGreet.equals("summer")){
7.     msg="暑中お見舞い!";
8. }
9. for(int i=3;i<3+dataRepeat;i++){ %>
10.<p><font size=<%= i%>><%= msg%>!</font></p>
11.<% } %>
12.<p>
13.<font color="green">願わくば、</font>
14.<font color="#FF0000">Groovyな</font>
15.<font size="6"><%= dataGroovy %></font>
16.<font color="green">であることを!</font>
17.</p>
18.
19.<p><%= dataMessage%></p>
20.</body>
```

# (7) JSPについての整理

■HTMLの部分(下記例では、下線なしの部分)

■JSPの部分(例には無いが、これら以外にタグ/コメントがある)

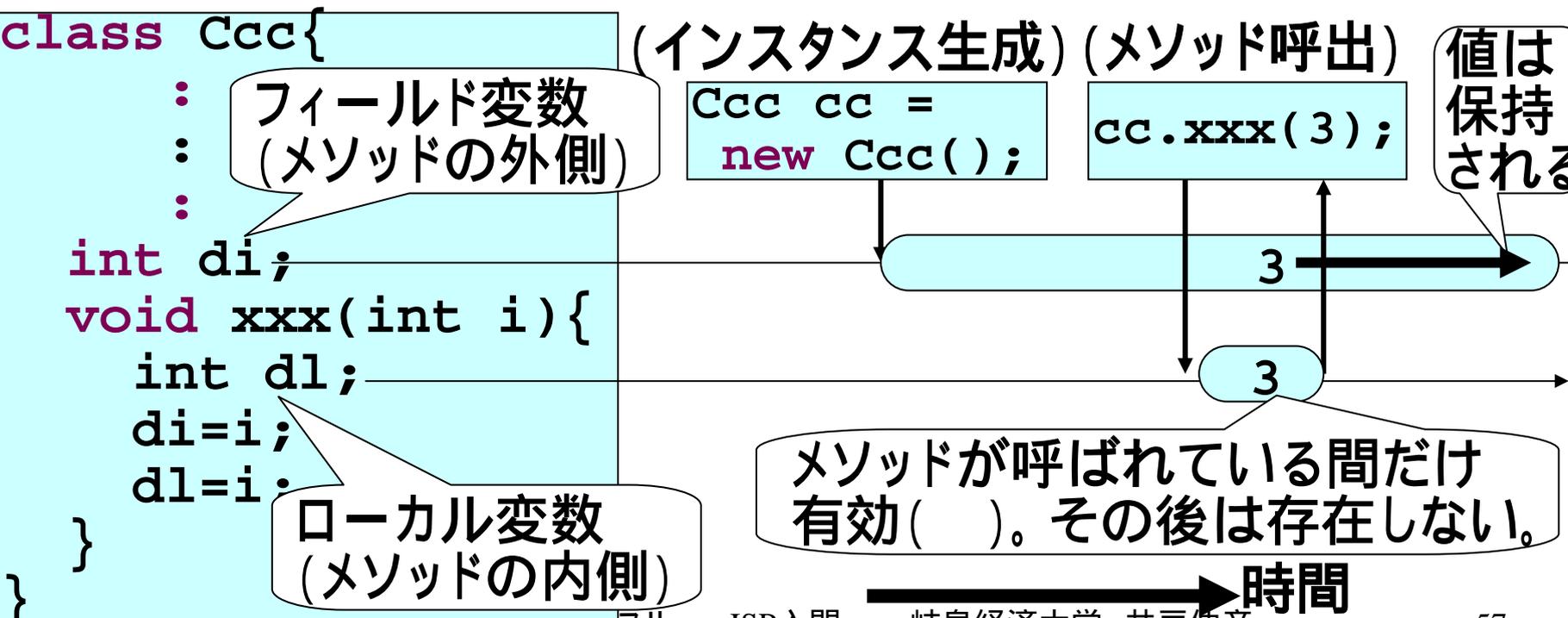
- ① ディレクティブ(指令): ページ単位の設定 <%@ %>
- ② 宣言: プログラムでの変数などの宣言 <%! %>
- ③ スクリプトレット: プログラム(制御構造)の部分 <% %>
- ④ 式(expression): プログラムの式の値 <%= %>

```
1. <@page contentType="text/html; charset=Shift_JIS" %>
2. <! int i; %>
3. <html>
4. <head>
5. <title>hello, world.</title>
6. </head>
7. <body>
8. < for(i=5; i>0; i--){ %>
9. <p><font size="<= i %>">御機嫌よう。 </font></p>
10.< %>
11.</body>
```

# (7.1) 宣言

## ■宣言 (<%! %>)

- JSPが呼び出されて処理を終えるまでしか使わない変数は、スクリプトレットで宣言してOKです。
- ここ(<%! %>)で宣言する変数は、呼び出しが繰り返されても同じ値を保つ変数です。
- フィールド変数だと理解しておけばOKです。



## (7.2) ディレクティブ、コメント

### ■ ディレクティブ (<%@ %>) と タグ (<jsp: ~ ></jsp>)

- 詳細については、別稿に譲ります。
- ひとまず、日本語を使う場合にスライド(2)の1行目のディレクティブを入れる必要があることを覚えておいてください。

### ■ コメント

#### • HTMLのコメント

1. <!-- コメント -->

- ◆ HTML中のコメントは、右のように書きます。
- ◆ このコメントは、ブラウザ側でもソースとして見ることが出来ます。

#### • JAVAのコメント

- ◆ スクリプトレット中のコメントは、JAVAと同じで、右の2つの書き方があります。
- ◆ このコメントは、ブラウザ側には見えません。

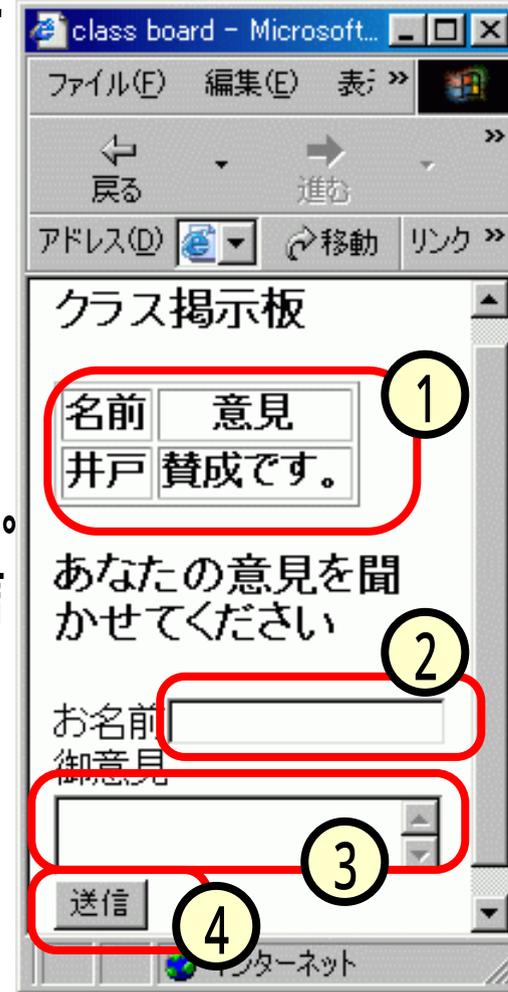
1. <%  
2. // 1行のコメント  
3. /\* 複数行に  
4.   わたるコメント \*/  
5. %>

# ( 8 . 1 . 1 ) 例題 2 : 簡単な掲示板

■右図のようなJSPプログラムを作成してください。

- ① は、テーブルにより表示しています。
- ② は、テキスト・フィールドです  
(シャボン玉(2.4)参照)。
- ③ は、テキスト・エリアです(同(3.1)参照)。
- ② に名前を、③ に、意見を入力して送信ボタン(④)を押すと、①のように表示します。

■ここでは、一人分の意見しか出ませんから、一般でいう掲示板にはなっていません。



## ( 8 . 1 . 2 ) プログラム例

<classBoard.jsp>

```
1. <%@page contentType="text/html; charset= Shift_JIS " %>
2. <% String strName = request.getParameter("yourName");
3.     String strComment =
   request.getParameter("yourComment"); %>
4. <html>
5. <head><title>class board</title></head>
6. <body>
7. <h3>クラス掲示板</h3>
8. <table border="1">
9. <tr><th>名前</th><th>意見</th></tr>
10. <tr><th><%= strName %></th><th><%= strComment%></th></tr>
11. </table><br>
12. <form action="/idoApp/classBoard.jsp">
13. <h3>あなたの意見を聞かせてください</h3>
14. お名前<input type="text" NAME="yourName"><br>
15. 御意見<textarea name="yourComment"></textarea><br>
16. <input type="submit" value="送信">
17. </form>
18. </body>
```

## (8.2) 多人数への拡張

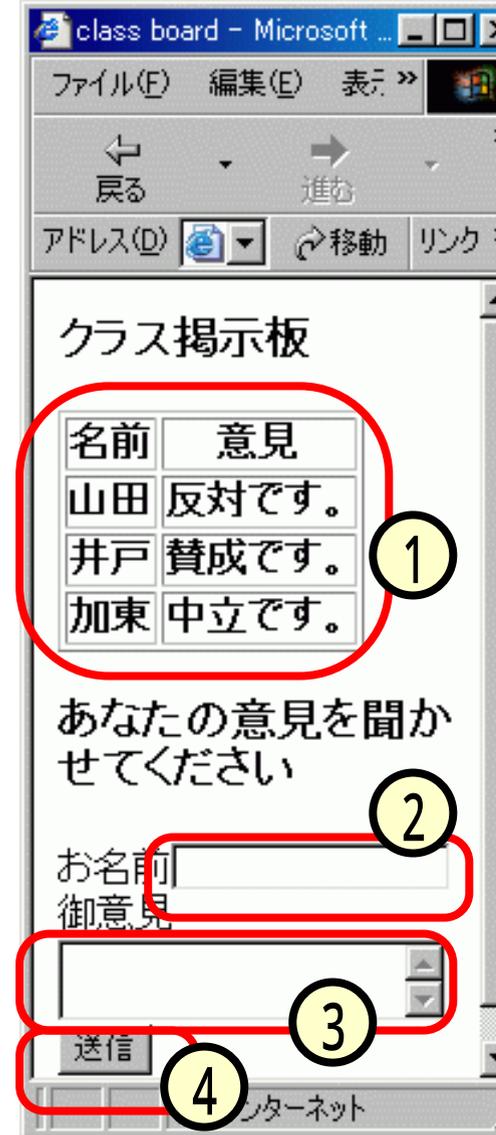
■右図のような掲示板を作ります。

- ② に名前を、③ に意見を入力して、送信ボタン(④)をクリックすると、一覧表(①)に表示されます。
- 同じ名前で再度送信すると、意見の部分が更新されます。

■次のようなデータは、HashMapクラスで実現することとします。

引数"井戸"とすると、"賛成です。"の値を得る

山田	反対です。
井戸	賛成です。
加東	中立です。



# ( 8 . 2 . 1 ) インポート

## ■ インポート

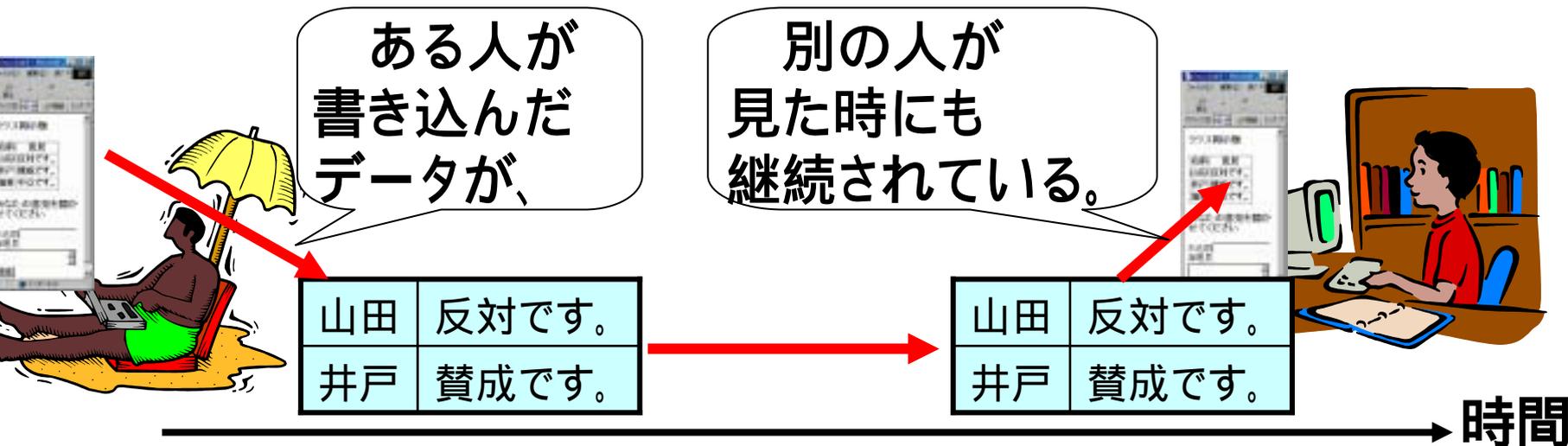
- クラス“HashMap”は、“java.util.\*”パッケージに含まれています。
- Javaでのインポートは、次のようにページ・ディレクティブに記述します。

```
1. <%@page import="java.util.*"  
   contentType="text/html; charset=UTF-8" %>
```

## (8.2.2) 宣言部

- このデータは、JSPが呼び出されて処理を終えても、次に呼び出される時には、前の値を保っていなければならない。
- よって、宣言部(<%!%>)で宣言する必要があります。

```
1. <%! HashMap commentTable = new HashMap(); %>
```



# ( 8 . 2 . 3 ) プログラム例

```
1. <%@page import="java.util.*"
2.     contentType="text/html; charset= Shift_JIS " %>
3. <%! HashMap commentTable = new HashMap(); %>
4. <% String strName = request.getParameter("yourName");
5.     String strComment =
6.     request.getParameter("yourComment");
7.     if((strName!=null)&&(strComment!=null)){
8.         commentTable.put(strName,strComment);
9.     }
10. %>
11. <<中略>>
12. <body>
13. <h3>クラス掲示板</h3>
14. <table border="1">
15. <tr><th>名前</th><th>意見</th></tr>
16. <% Object[] keys=commentTable.keySet().toArray();
17.     for(int i=0;i<keys.length;i++){ %>
18. <tr><th><%= keys[i] %></th><th><%=
19.     commentTable.get(keys[i]) %></th></tr>
20. <% } %>
21. </table><br> <<後略>>
```

# (9) 課題：掲示板の拡張

■右図のような掲示板を作ってください。

## ■動作

- ページにアクセスすると、各メンバーの名前と意見の一覧が、賛成と反対とに分かれて、それぞれととに表示されます。
- 自分の意見を書き込む際には、いままでの掲示板に追加して、の賛否を設定します。

## ■解答例

- 井戸のサイト参照。

The screenshot shows a web form titled "クラス掲示板" (Class Bulletin Board). It is divided into two sections: "賛成側" (Pro side) and "反対側" (Contra side). Each section contains a table with two columns: "名前" (Name) and "意見" (Opinion). In the "賛成側", the name "井戸" (Iido) is listed with the opinion "絶対、賛成。" (Absolutely,赞成). In the "反対側", the names "山田" (Yamada) and "加東" (Kaido) are listed with opinions "絶対、反対。" (Absolutely, 反対) and "なんとなく。" (なんとなく) respectively. Below these sections is a text prompt "あなたの意見を聞かせてください" (Please let us hear your opinion). This is followed by a form with a label "お名前" (Your name), a radio button for "賛否" (Approval/Disapproval), and radio buttons for "賛成" (Pro) and "反対" (Contra). A "御意見" (Your opinion) text area and a "送信" (Send) button are also present. Three red circles with numbers 1, 2, and 3 are overlaid on the image. Circle 1 highlights the "賛成側" table. Circle 2 highlights the "反対側" table. Circle 3 highlights the "賛否" and "賛成/反対" radio buttons.

クラス掲示板

賛成側

名前	意見
井戸	絶対、賛成。

反対側

名前	意見
山田	絶対、反対。
加東	なんとなく。

あなたの意見を聞かせてください

お名前

賛否  賛成  反対

御意見

送信

ページが表示され インターネット

# (10) JAVAサーブレットとCGIとの違い

■ CGIによるWebサイトを作成したことがある方に対して、CGIとサーブレット(JSP)との違いを記しておきます。

## ■ 効率・性能

- 複数回の書き込みがあった際に、CGIではその都度別のプロセスが立ち上がってこれを処理します。JAVAサーブレットでは、同じプロセスが処理します。

## ■ サポート

- CGIがほぼ裸のマシン上で動かなければならないのに対し、JAVAサーブレットでは、Webアプリケーション固有の必要機能がサポートされています。

### < CGI >



・インタフェース等を除くと、裸のマシンを利用することとなる。

### < JAVAサーブレット >



・セッション管理など、様々なサポートされている。