

Java



盛岡情報ビジネス&デザイン専門学校

★☆☆★本日の内容★☆☆★

1. スクリプトレット
2. スクリプト式
3. Pageディレクティブ
4. aタグを使った画面遷移



◆JSPの書き方

基本的にはHTMLと同じく<body>タグの中に中身のWebページを記述していきます。

それに加えて、Javaのコードを記述して条件分岐や繰り返し処理を書くこともできます。

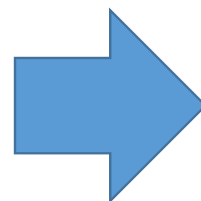


◆スクリプトレット

JSPの中にJavaのコードを埋め込むために使用します。

<% ~ %>で囲んで使用する。

ex) `<% for(int i = 0; i < 10 ; i++){ %>`
`<p> こんにちは。 </p>`
`<% } %>`



`<p> こんにちは。 </p>`
`<p> こんにちは。 </p>`
`<p> こんにちは。 </p>`
`<p> こんにちは。 </p>`
`<p> こんにちは。 </p>`
`<p> こんにちは。 </p>`
`<p> こんにちは。 </p>`
`<p> こんにちは。 </p>`
`<p> こんにちは。 </p>`
`<p> こんにちは。 </p>`



◆スクリプト式

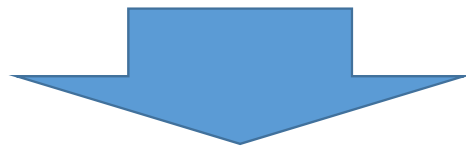
変数やメソッドの戻り値などを直接出力することができます。

<%= 変数名やメソッド() %>で使用する。

※HTMLへのSystem.out.println()のようなものと覚えましょう。

ex) `<% int x = 20; %>`

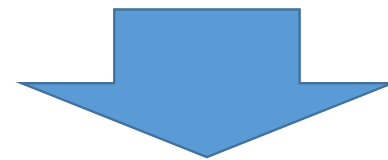
`<p> 年齢は <%= x %>です。 </p>`



「年齢は20です。」と表示されます。

`<% Human h = new Human("高橋",30); %>`

`<p> 年齢は<%= h.getAge() %>です。 %>`



「年齢は30です。」と表示されます。



◆JSPの書き方

- Pageディレクティブ

文字コードの設定やクラスのインポートなど各種設定ができます。

<%@ 設定内容 %>と記述します。

ex) **<%@ page contentType="text/html; charset=UTF-8" %>**

<%@ page import="java.util.ArrayList" %>

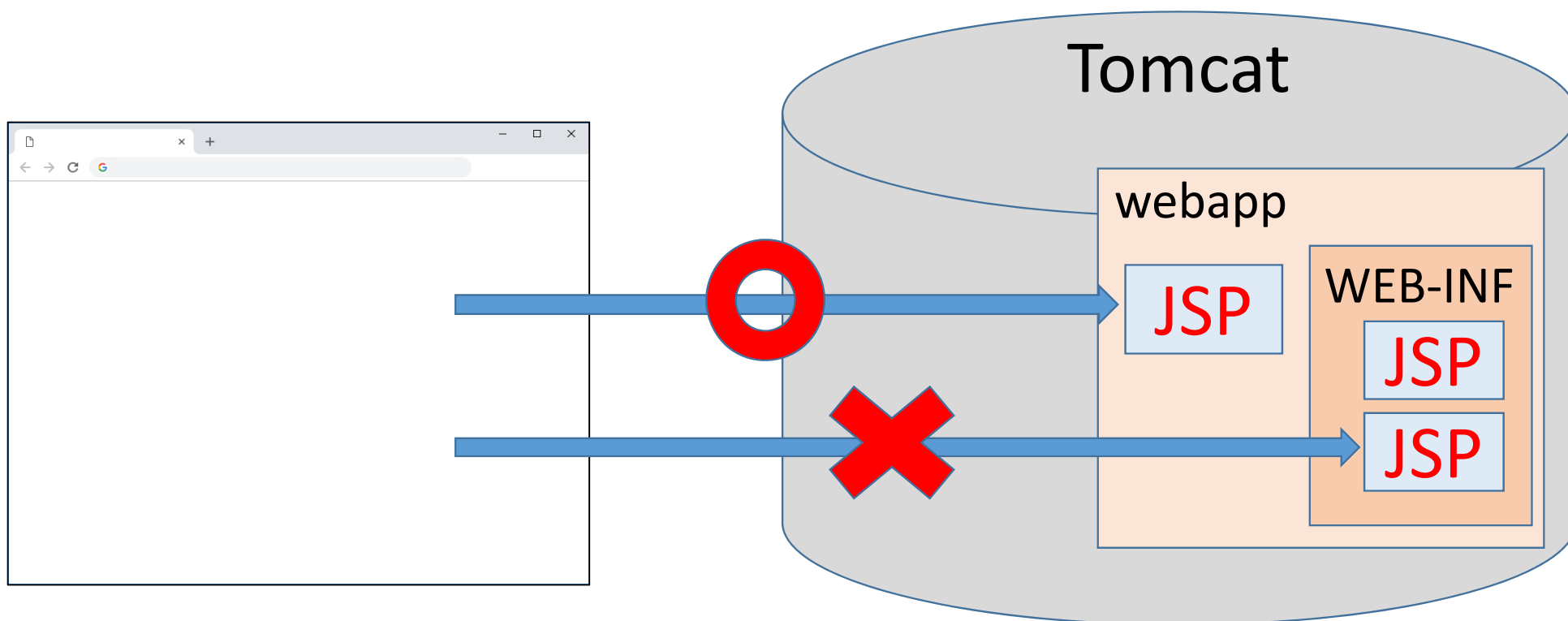
※ **自作のクラス**をインポートする際は**必ずパッケージに属す**必要がある。
つまり、デフォルトパッケージはNGです。



◆WEB-INFのルール

JSPファイルを作成したWEB-INFというフォルダ配下は「**外部からファイルが非公開になる**」という特殊なルールが設定されています。

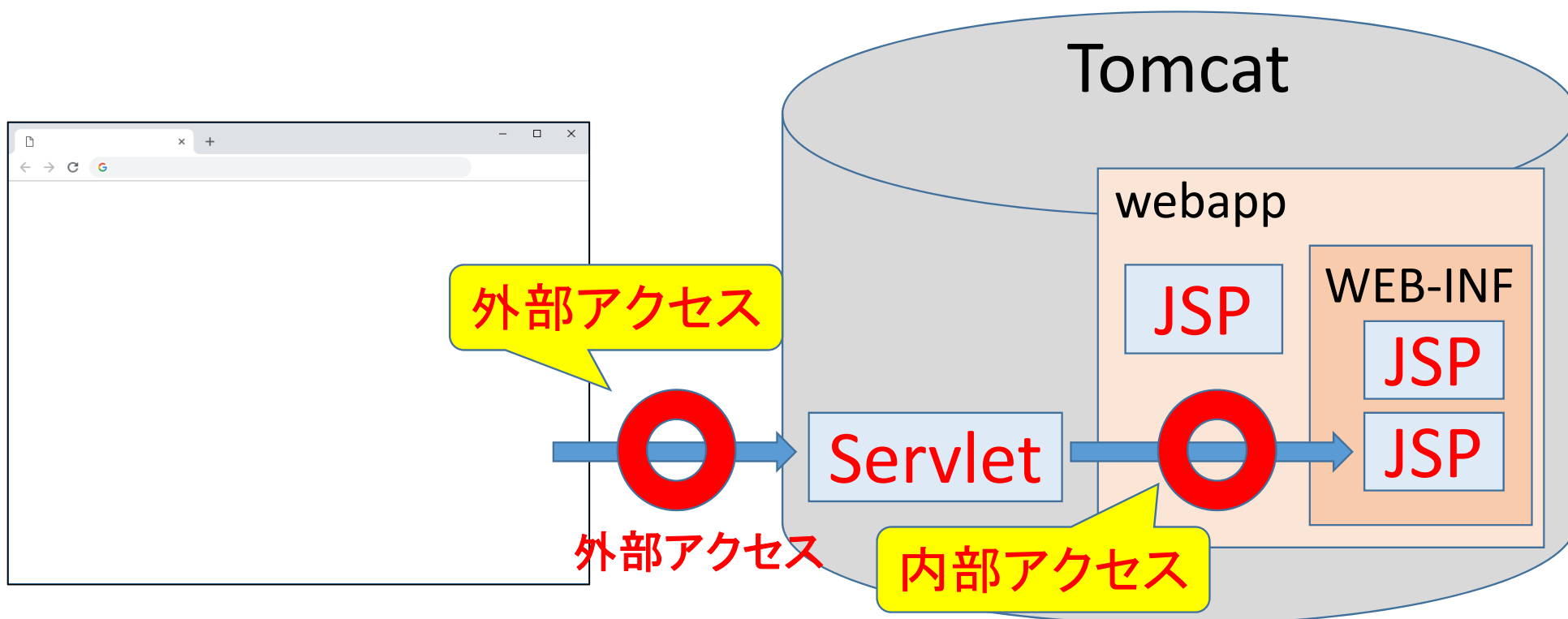
なお、webapp直下はそのルールは適用されませんのでブラウザから直接URLパターンにjspファイル名を記述することでアクセスできます。



◆WEB-INFのルール

外部から非公開のJSPファイルへのアクセスは公開されている**サーブレット経由でアクセス**します。

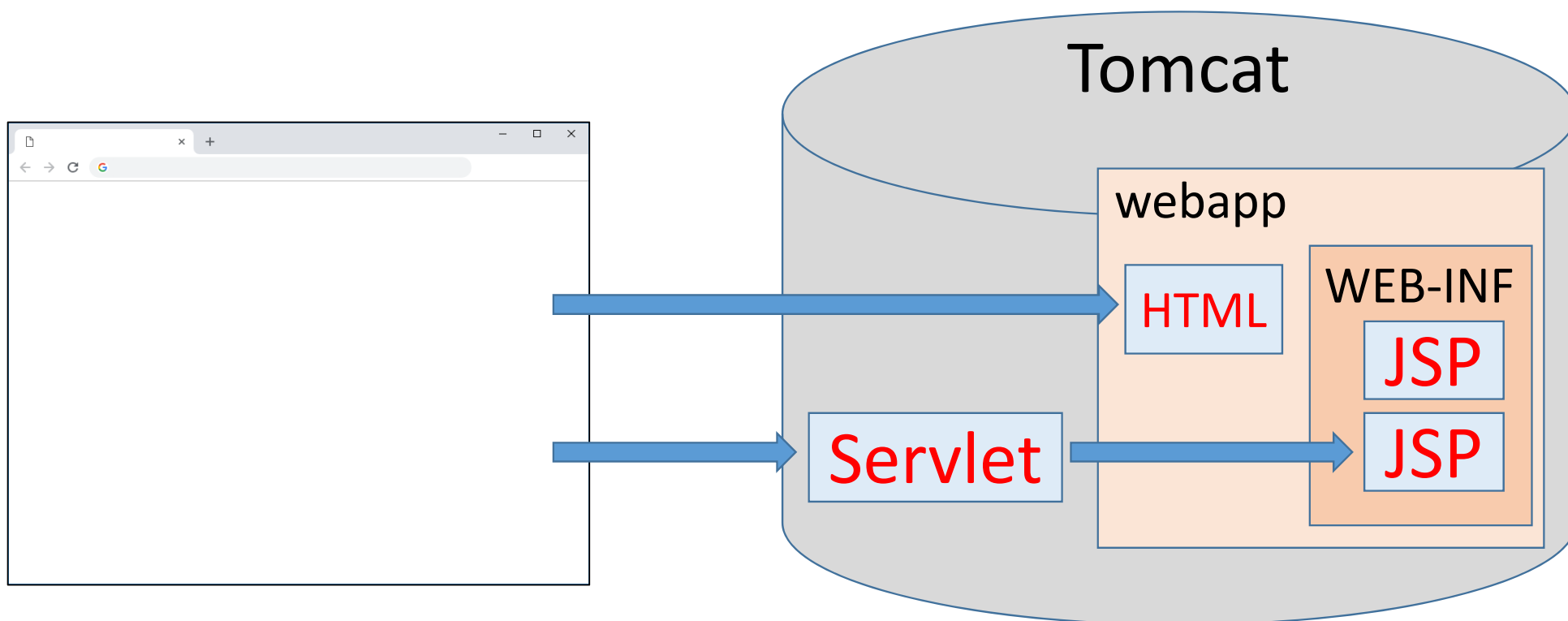
※ServletからJSPへのアクセスはTomcat内部からのアクセスのため、参照が可能です。



◆WEB-INFのルール

通常は静的なページ(HTML)などを webapp 配下に格納します。

動的なページ(JSP)は内部に記述されたコードを隠ぺいするためにWEB-INF配下に格納して、Servlet 経由でアクセスします。



◆EclipseでのJSP文法エラーについて

EclipseではJSPファイルに文法エラーが無い場合でもエラーメッセージである赤×マークが表示されることがあります。

その場合は「Ctrl + a」で全選択し、
「切り取り」→「保存」→「貼り付け」→「保存」と操作することで表示が消えます。

困ったときは試してみましょう！

(これでエラーが消えない場合は本当にエラーの可能性が高いので、良く見て考えよう！)



◆画面遷移

<a>タグを使用して画面遷移することができます。

【記述方法】

` 表示文字列 `



実際に書くとこんな感じ

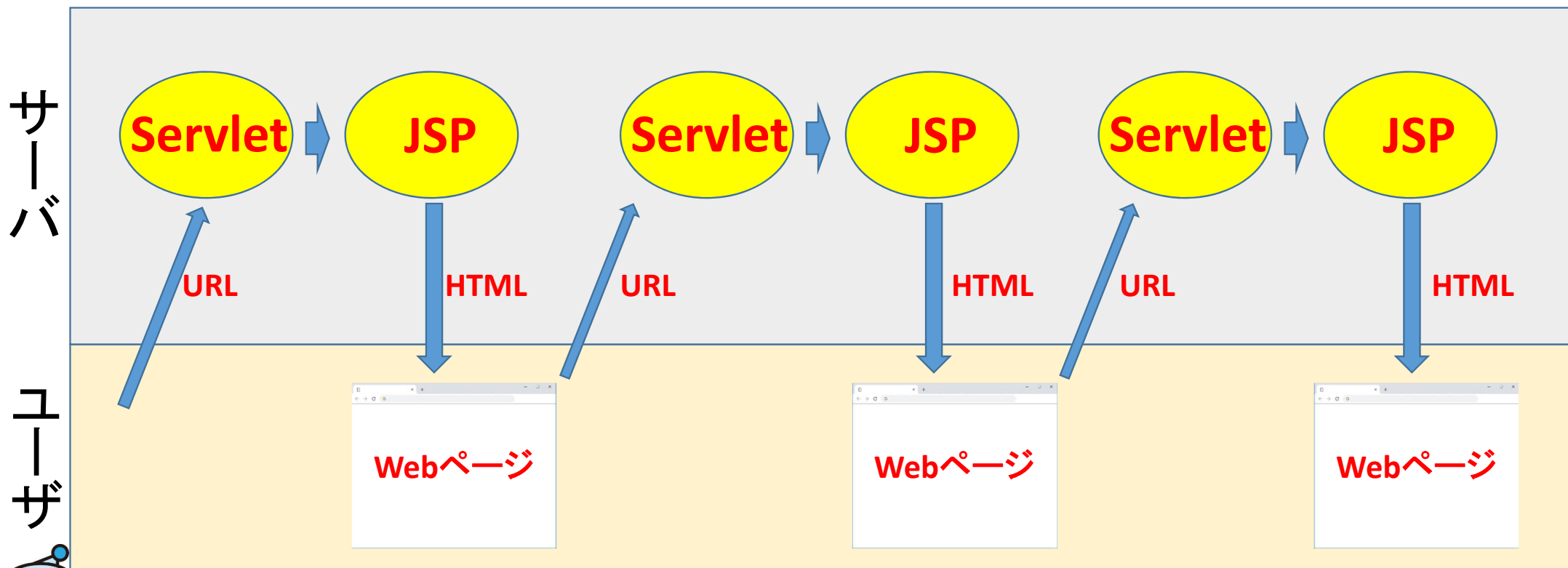
```
<a href = “top”> こちらをクリックしてね </a>
```

【復習】URLパターンとは@WebServletアノテーションで指定されているパターンのことです。デフォルトはクラス名。



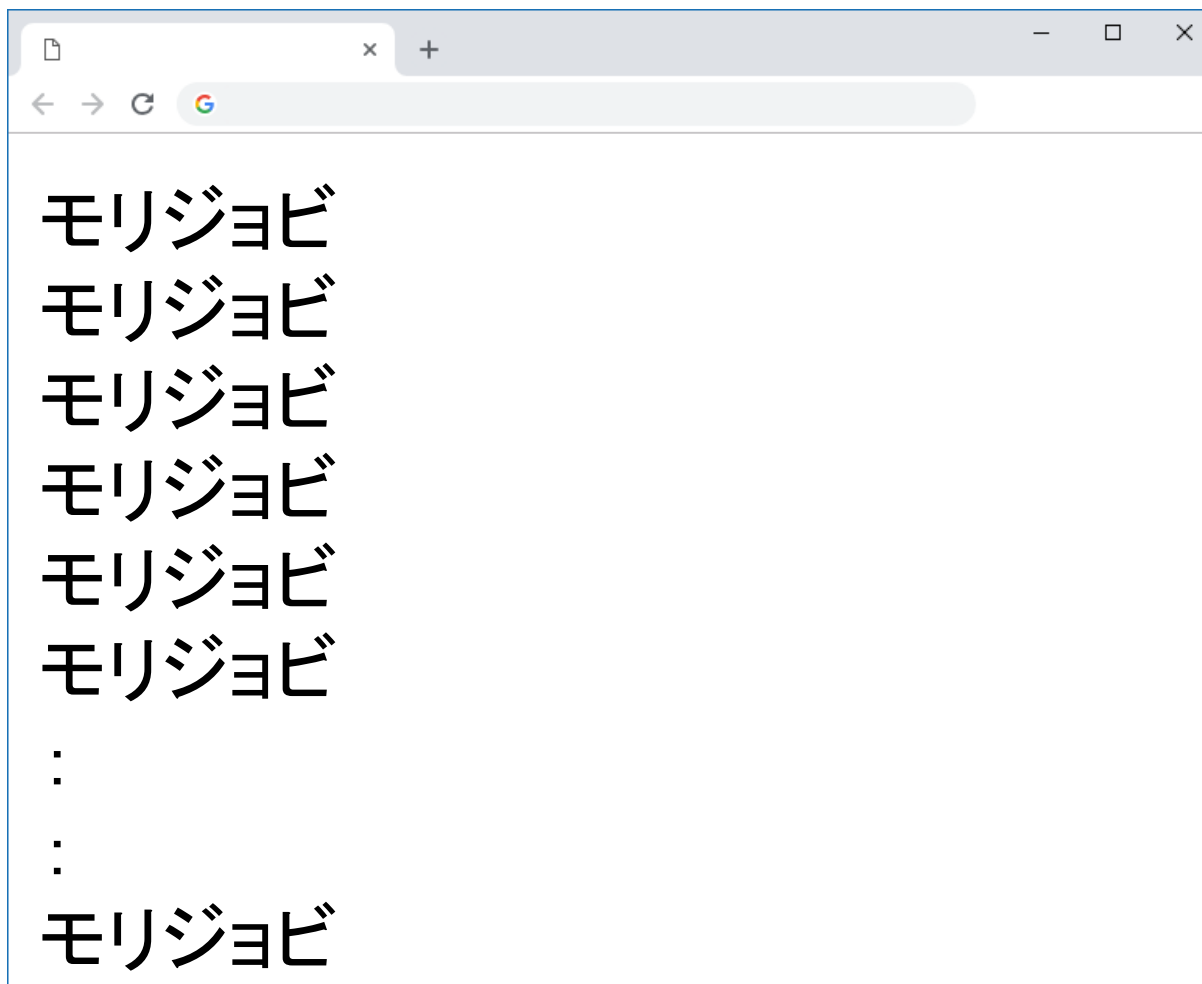
◆ServletとJSPを使ったWebアプリのイメージ

Webアプリが動いているイメージは下記の通りとなります。



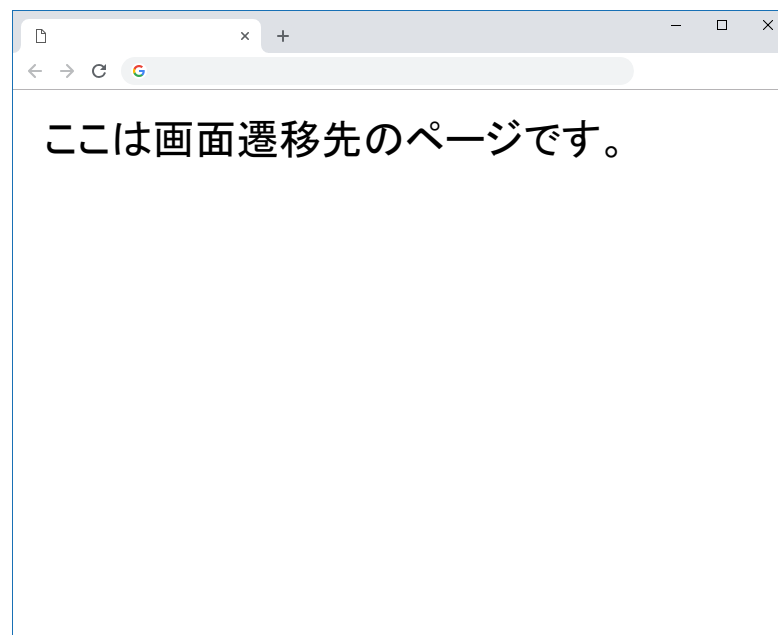
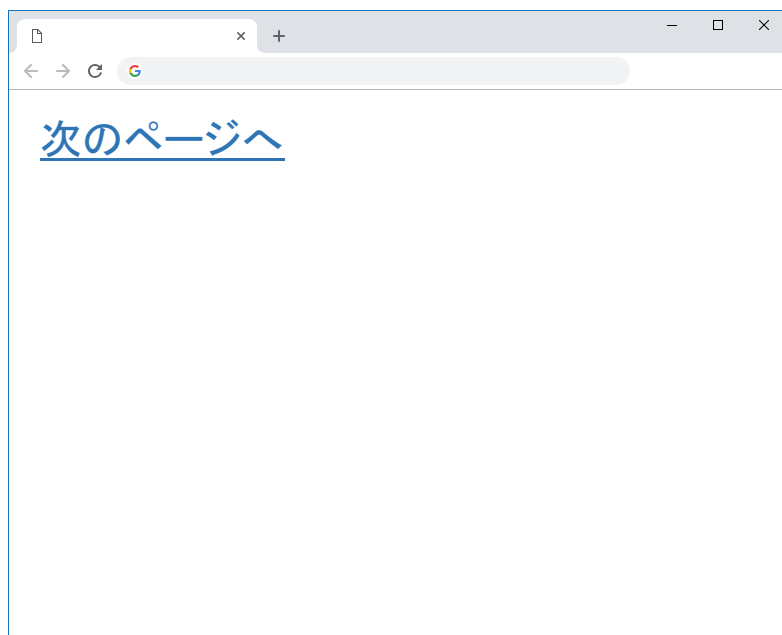
◆練習問題 (必須課題)

問1: ブラウザに「任意の文字列」を1000回表示するアプリケーションを作成せよ。



◆練習問題 (必須課題)

問2: ブラウザに自分の名前を表示するアプリケーションを作成せよ。



◆チャレンジ問題

問3: ブラウザに「任意の文字列」を乱数で10～99回表示するアプリケーションを作成せよ。

ブラウザを更新するたびに表示する数が変わること。

