

# 上級プログラミング2(第7回)

---

工学部 情報工学科  
木村昌臣  
中島 毅

# 今日のテーマ

---

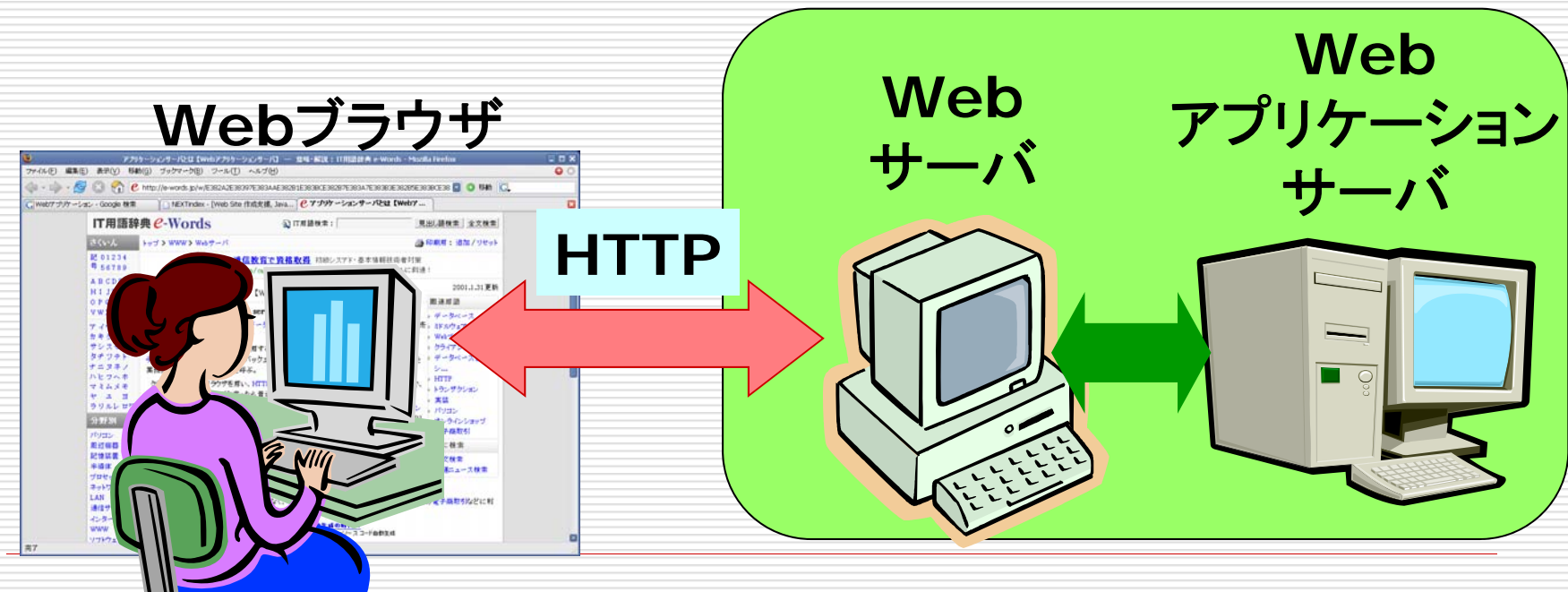
- Webアプリケーションとは
    - Webアプリケーションとはなにか
    - Webアプリケーションの仕組み
    - 三層アプリケーション
    - サーブレット、JSP、JavaBeans
    - MVCモデル
  - Webアプリケーションの環境構築
    - Webサーバ(Apache)
    - Webアプリケーションサーバ(Tomcat)
-

# Webアプリケーションとは

---

# Webアプリケーションとはなにか

- 実際の処理を行うサーバとのデータのやり取りやユーザインタフェースとしてWeb技術を利用したもの



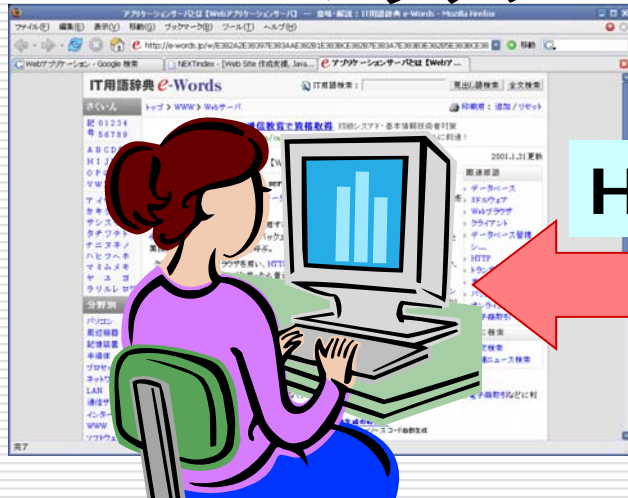
# Webアプリケーションの仕組み

ユーザ  
インタフェース

HTTP  
-アプリケーション間  
インタフェース

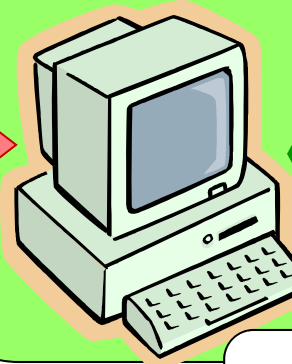
アプリケーション  
が動作  
(Javaプログラム等)

Webブラウザ



HTTP

Web  
サーバ



Web  
アプリケーション  
サーバ



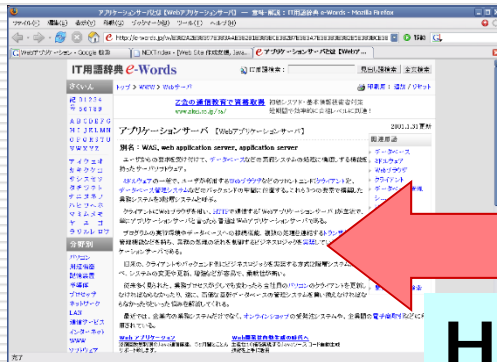
接続には専用の  
モジュールが必要

# 三層アプリケーション

第一層

プレゼンテーション

Webブラウザ



第二層

ビジネスロジック

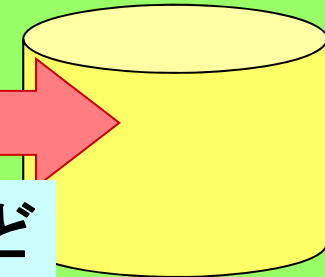
Webサーバ/  
Webアプリケーション  
サーバ



HTTP

第三層

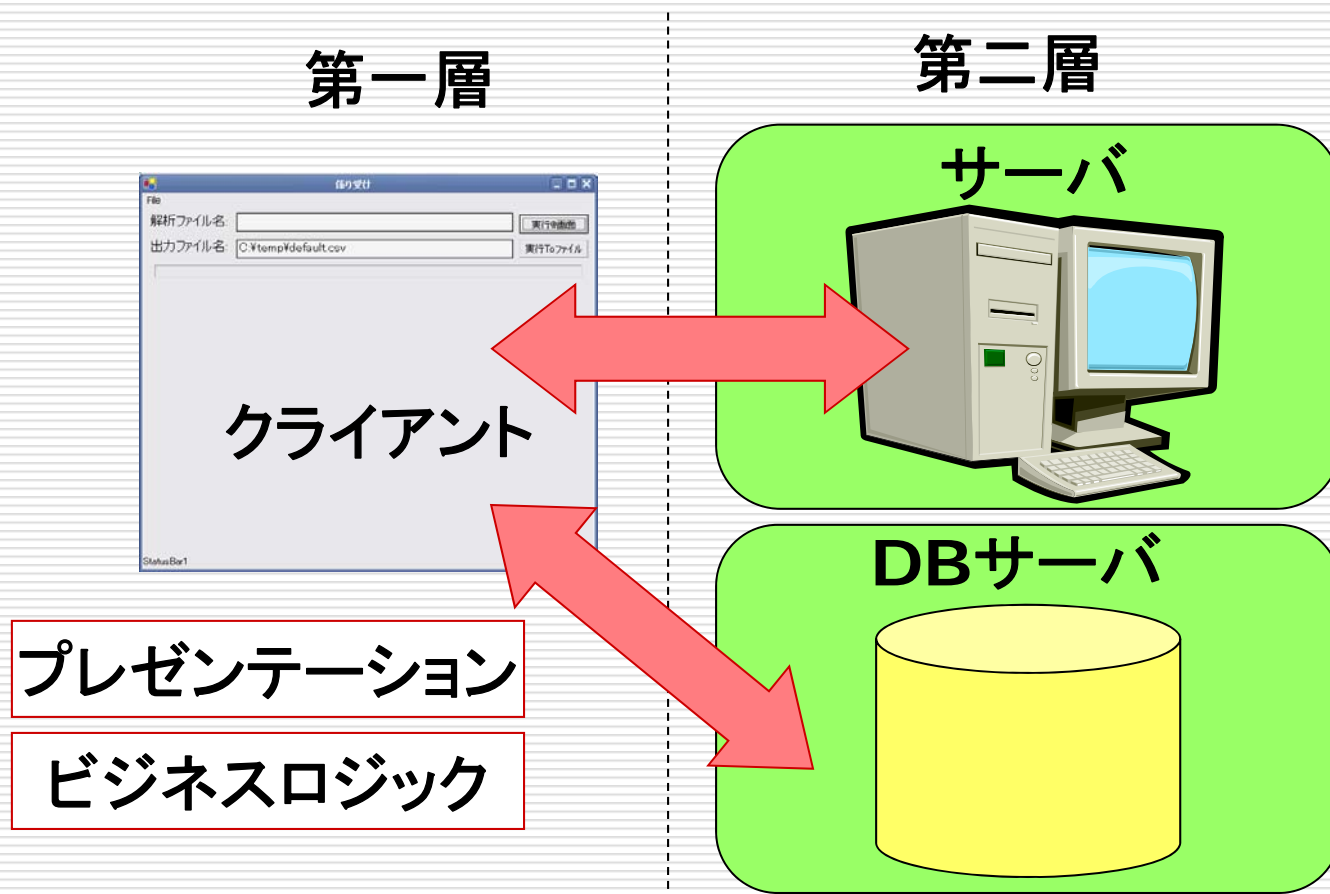
DBサーバ



JDBCなど

# 【参考】従来のクライアント・サーバアプリケーション（二層アプリケーション）

---



# クライアント・サーバモデルと比べて 三層アプリケーションモデルが優れている点

---

- 表示(プレゼンテーション)とデータ処理(ビジネスロジック)の分離
    - ブラウザさえあればクライアントをインストールする必要がない → クライアントの更新が不要
    - データベースなどの基幹システムに対してクライアントから直接アクセスすることがない(アクセスはWebアプリケーションサーバのみ) → セキュリティが向上
-



# サーブレット(Servlet)

---

- Webアプリケーションサーバで動作する  
Javaプログラム → 動的にWebページを作る
  - HttpServletクラスを継承して作る
    - 次の三つのパッケージをインポートする必要あり
      - java.io.\*
      - javax.servlet.\*
      - javax.servlet.http.\*
    - HttpServletRequestで入力、  
HttpServletResponseで出力を行う
-

# JSP

---

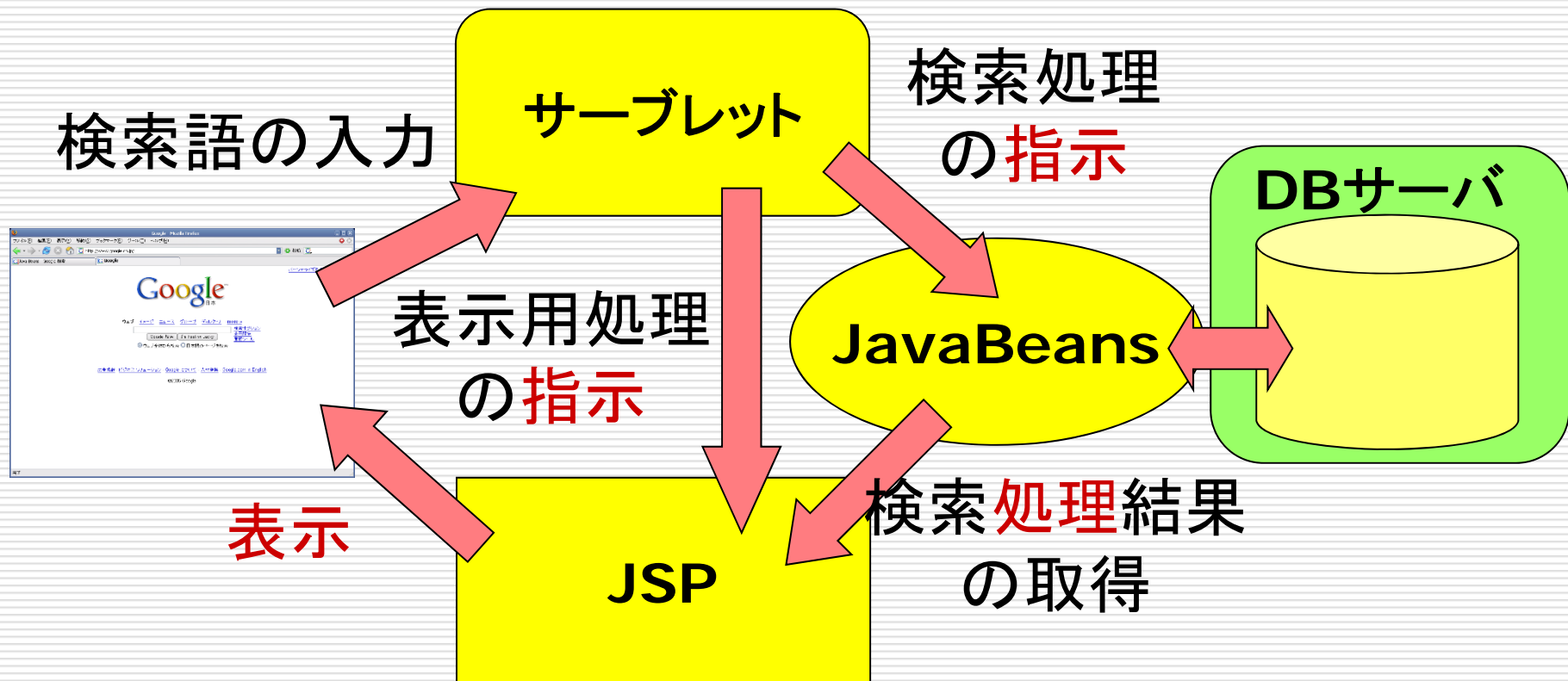
- Java Server Page
  - HTMLに Javaのプログラム処理を付加したもの
    - サーブレットは最終的にHTML形式で出力  
→ページのレイアウトの調整を詳細に行うことは困難
    - JSPでは, Webページのレイアウトを決めて(HTMLベース)、必要な処理(Javaプログラム)を付加する
-

# JavaBeans

---

- いろいろなWebアプリケーションで共通に使われるJavaクラス群をモジュール化してまとめたもの
  - データ接続用など
  - クラスのプロパティ(変数)にアクセスするときには、  
直接、変数にアクセスするのではなく、メソッド経由で  
アクセスする
    - `set変数名()`・・・変数に値のセット
    - `get変数名()`・・・変数の値を取得

## それぞれの役割(検索エンジンを例として)



※ただし、JavaBeansやJSPの役割をサーブレットだけで実現することもある

# MVCモデル

リクエストに応じて適切な  
処理を指示する

**C**ontrol

サーブレット



JavaBeans

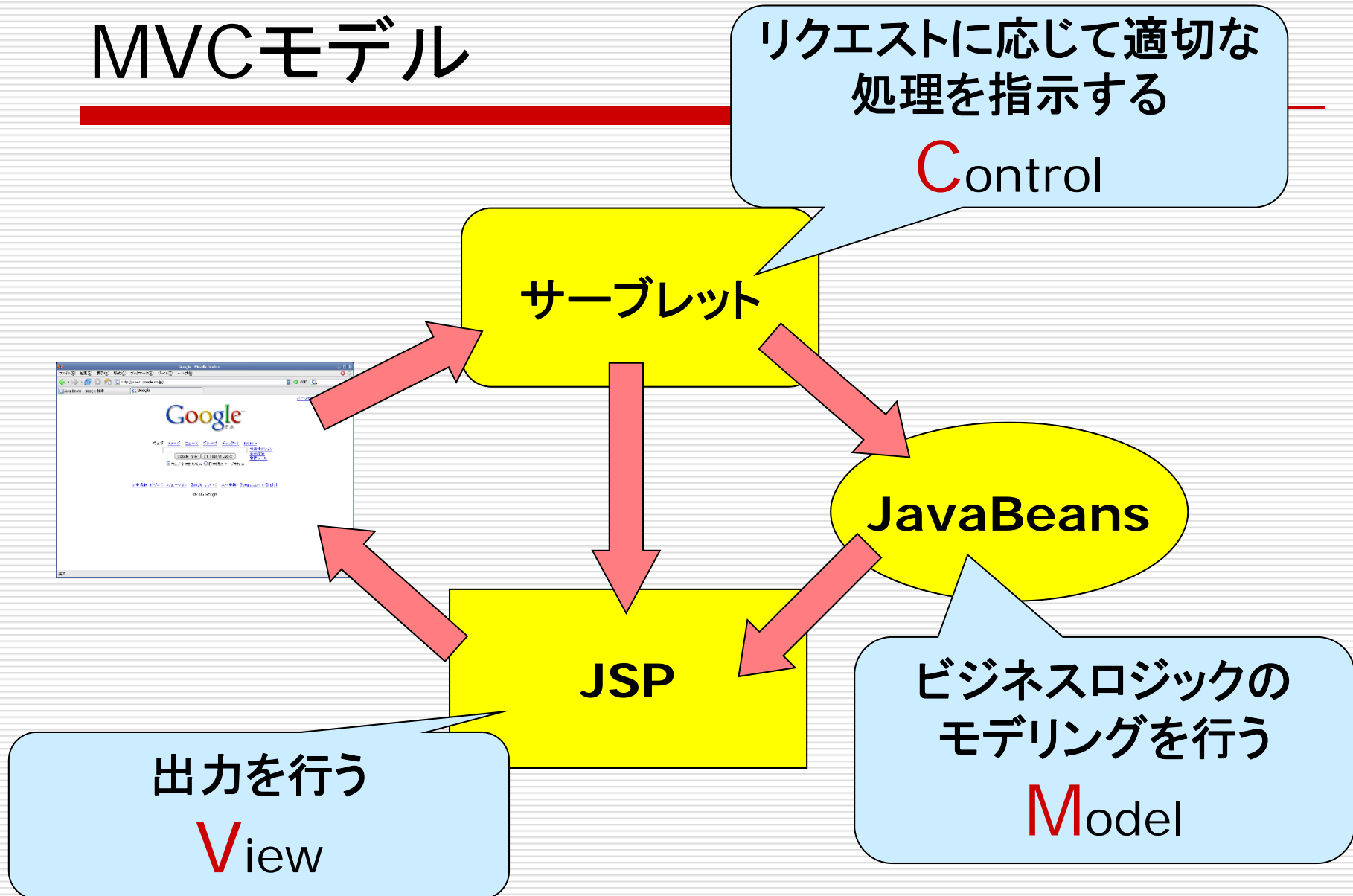
JSP

ビジネスロジックの  
モデリングを行う

**M**odel

出力を行う

**V**iew



# MVCモデルにのっとなったWebアプリケーションを設計する利点

---

- 規模が大きいアプリケーションに向いている
    - 役割別に分類されているため、異なる役割の人が開発に参画しやすい
      - M(モデル): 共通部品開発者
      - V(ビュー): 画面デザイナー
      - C(コントロール): アプリケーション開発者
-

# Webアプリケーションの作り方の変化

---

## □ 従来のWebアプリケーションの困難

- 従来のWebアプリケーションでは、出力に対してWebページ 1つが対応

- 出力には、ページ遷移が必要 → 使いづらい

- Webブラウザの「戻る」ボタン押下 → 別途対応が必要  
(処理の流れが断ち切られる)

## □ Ajax (Asynchronous JavaScript + XML)

## □ Websocket

- WebブラウザとWebサーバ間のページ遷移なしの通信を利用したプログラムを実現
-

# Single Page Application (SPA)

---

- HTML5で高度な処理をJavaScriptで行うことができるようになったことが背景
  - HTMLからページ遷移なしにHTTPでサーバとデータのやり取りが可能(Ajax)
  - 描画方法が豊かになったので、いろいろなUI部品が簡単に使用可能
- ひとつのWebページのみでアプリケーションの全ての画面を実現
  - Webアプリケーションとネイティブアプリケーションのいいとこ取り



# SPAの例

ブラウザ上で  
Excelが動いていることに  
注意

The screenshot shows the Excel Online interface in a web browser. The address bar shows the URL: <https://onedrive.live.com/edit.aspx?cid=592B05F7655>. The ribbon is set to 'Font', and a dropdown menu is open showing options like 'Font', 'Double Underline', 'Underline', 'Italic', and 'Font Color'. The spreadsheet contains a table titled 'Team Members' with columns 'Name', 'Age', 'Fingers', and an unlabeled column with numerical values. The 'Tell Me' search bar is visible on the right side of the ribbon, with a prompt: 'Tell Excel what you want to do and it will pull up the perfect commands for you - no more hunting in the Ribbon!'. Below this, a 'Chart' section is expanded, showing options like 'Insert Pie Chart', 'Insert Bar Chart', and 'Insert Area Chart'.

|    | Name      | Age | Fingers |     |
|----|-----------|-----|---------|-----|
| 5  | Dan       | 23  | 9.5     | 180 |
| 6  | Daniel    | 26  | 10      | 159 |
| 7  | Lorissa   | 22  | 10      | 149 |
| 8  | Guy       | 26  | 10      | 159 |
| 9  | Prash     | 29  | 11      | 156 |
| 10 | Renu      | 26  | 10      | 171 |
| 11 | Andrew    | 21  | 10      | 127 |
| 12 | Cathy     | 21  | 10      | 198 |
| 13 | Alexander | 23  | 10      | 174 |

<https://blogs.office.com/2014/04/14/more-office-online-commenting-printing-tell-me-and-more/>

## 以降、この講義では

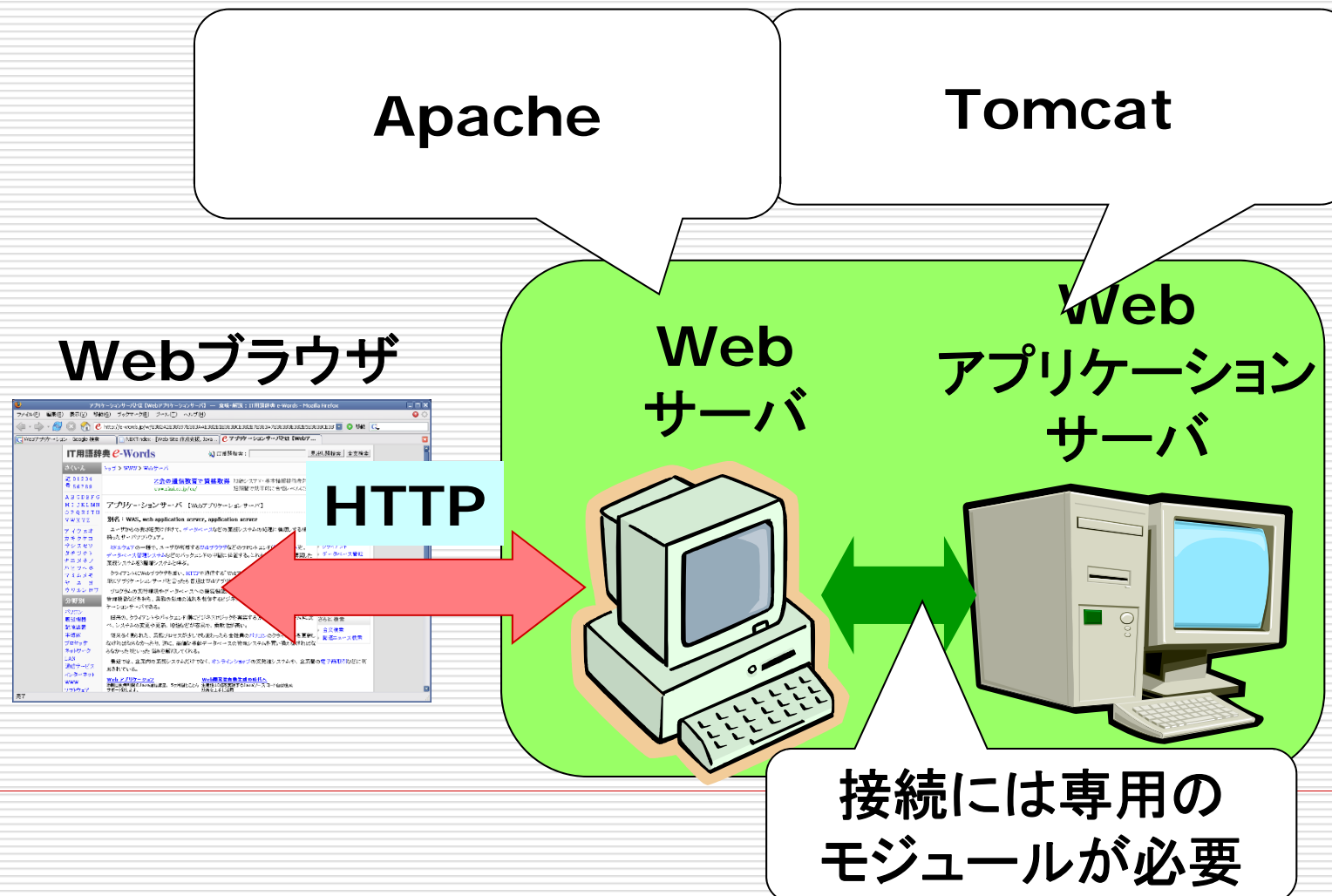
---

- Javaだけで話を閉じる
  - サーブレット、JSP、BeansによるMVCモデルに基づき説明
  - SPAではJavaScriptの活用が不可欠・・・
-

# Webアプリケーションの環境構築

---

# Webアプリケーションの実行環境例



# Webアプリケーションの実行環境

---

## □ Webサーバ

- Apache (Apache Software foundation)
- IIS (Microsoft)
- IHS (IBM)

## □ Web アプリケーションサーバ

- Tomcat (Jakarta Apache Project)
- WebSphere Application Server (IBM)

など

---

# Apacheの入手

---

## Downloading Apache for Windows

The Apache HTTP Server Project itself does not provide binary releases of software, only source code. Individual

If you cannot compile the Apache HTTP Server yourself, you can obtain a binary package from numerous binary

Popular options for deploying Apache httpd, and, optionally, PHP and MySQL, on Microsoft Windows, include:

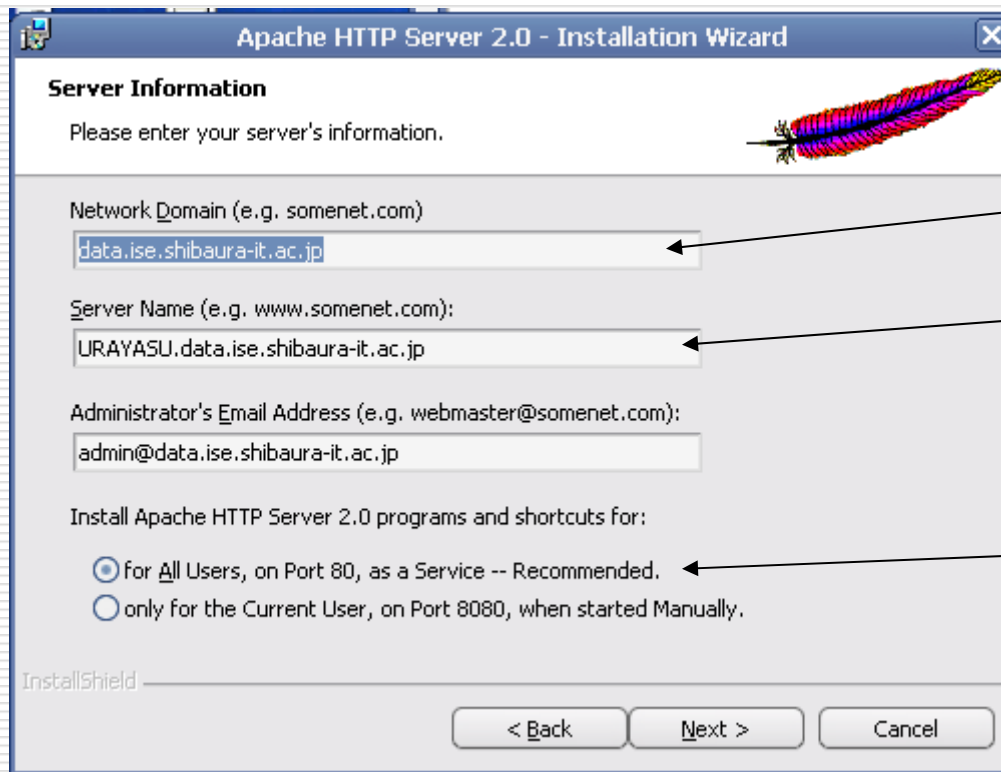
- [ApacheHaus](#)
- [Apache Lounge](#)
- [BitNami WAMP Stack](#)
- [WampServer](#)
- [XAMPP](#)

[http://httpd.apache.org/docs/current/  
platform/windows.html#down](http://httpd.apache.org/docs/current/platform/windows.html#down)  
のリンクからダウンロード

# Apacheのインストール

---

## □ インストーラを実行(デフォルトでよい)



Apache HTTP Server 2.0 - Installation Wizard

**Server Information**  
Please enter your server's information.

Network Domain (e.g. somenet.com):  
data.ise.shibaura-it.ac.jp

Server Name (e.g. www.somenet.com):  
URAYASU.data.ise.shibaura-it.ac.jp

Administrator's Email Address (e.g. webmaster@somenet.com):  
admin@data.ise.shibaura-it.ac.jp

Install Apache HTTP Server 2.0 programs and shortcuts for:

☒ for All Users, on Port 80, as a Service -- Recommended.

☐ only for the Current User, on Port 8080, when started Manually.

InstallShield

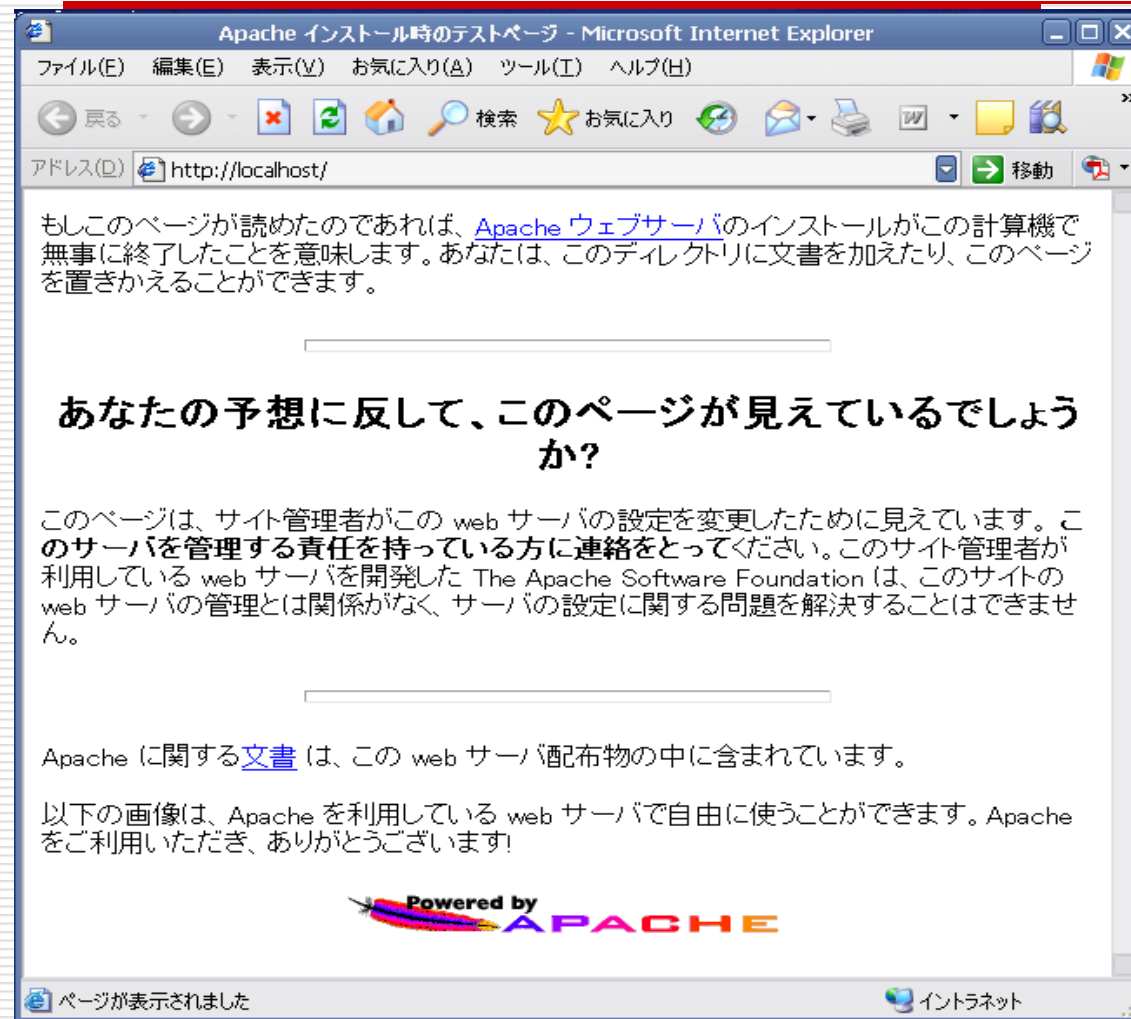
< Back   Next >   Cancel

ネットワークドメイン

サーバ名

サービスとして実行

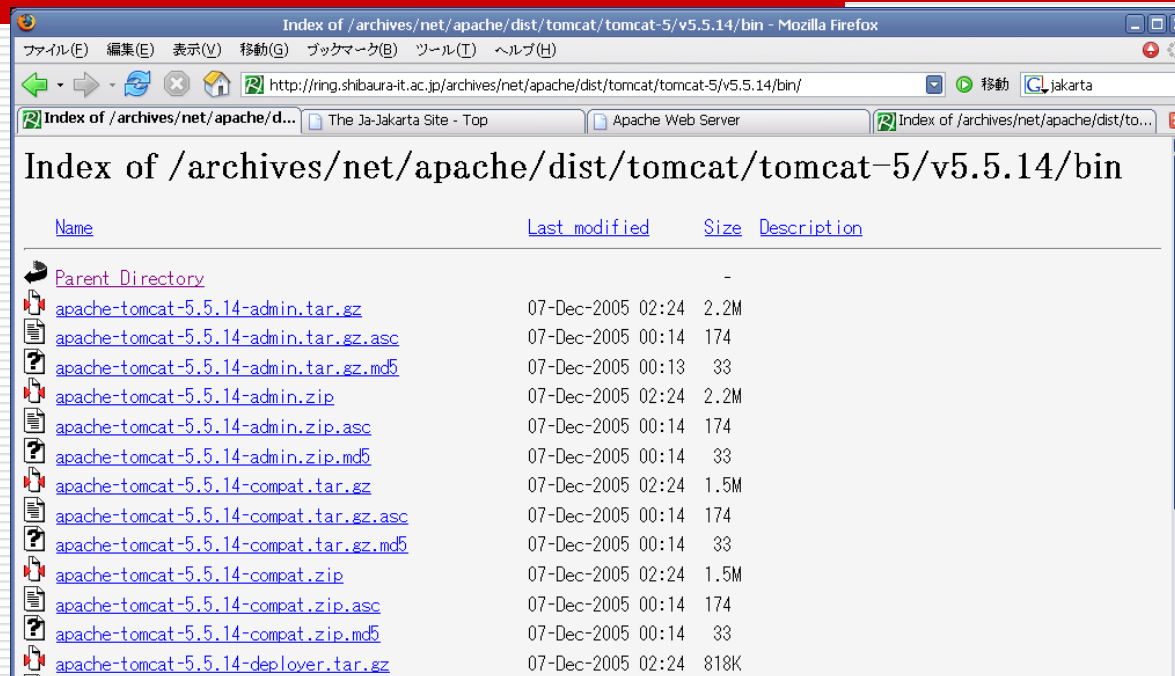
# 正しくインストールできると、この画面が現れる



ブラウザで  
**http://localhost/**  
にアクセス



# Tomcatの入手



<http://ring.shibaura-it.ac.jp/archives/net/apache/dist/tomcat/tomcat-9/v9.0.21/bin/>  
からダウンロード

# 環境変数JAVA\_HOMEを設定

---

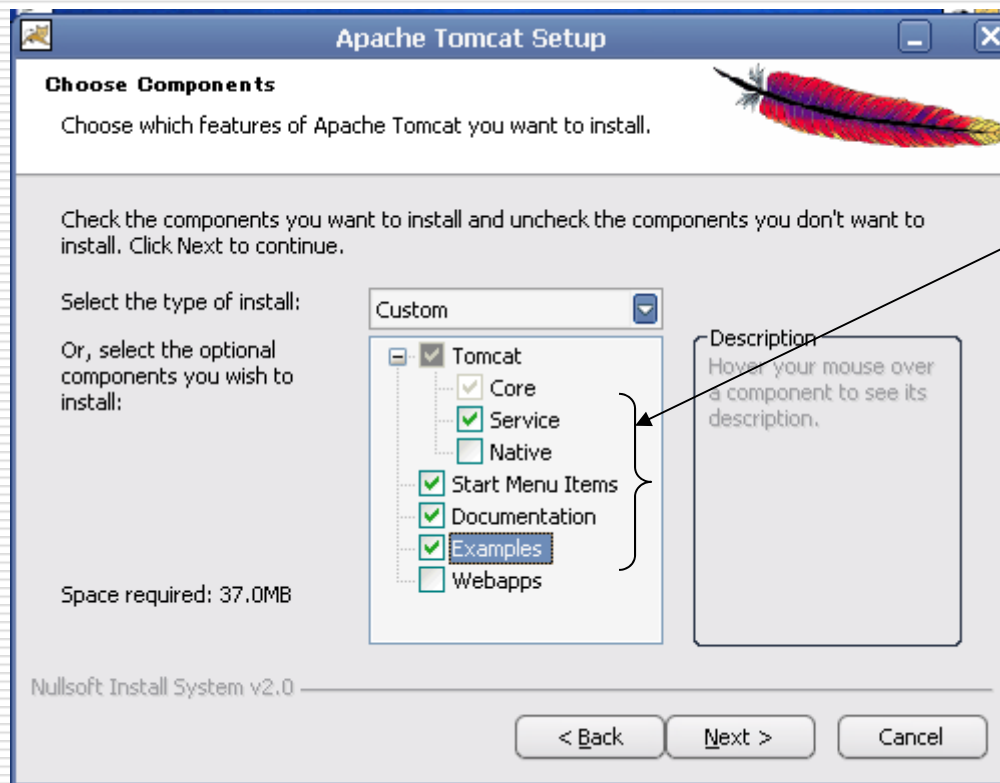
- ❑ 「マイコンピュータ」を右クリックし、「プロパティ」→「詳細設定」タブ→「環境変数」ボタンを押す
- ❑ 「システム環境変数」の「新規ボタン」を押して、環境変数JAVA\_HOMEにjdkのインストールディレクトリを設定する



# Tomcatのインストール(1)

---

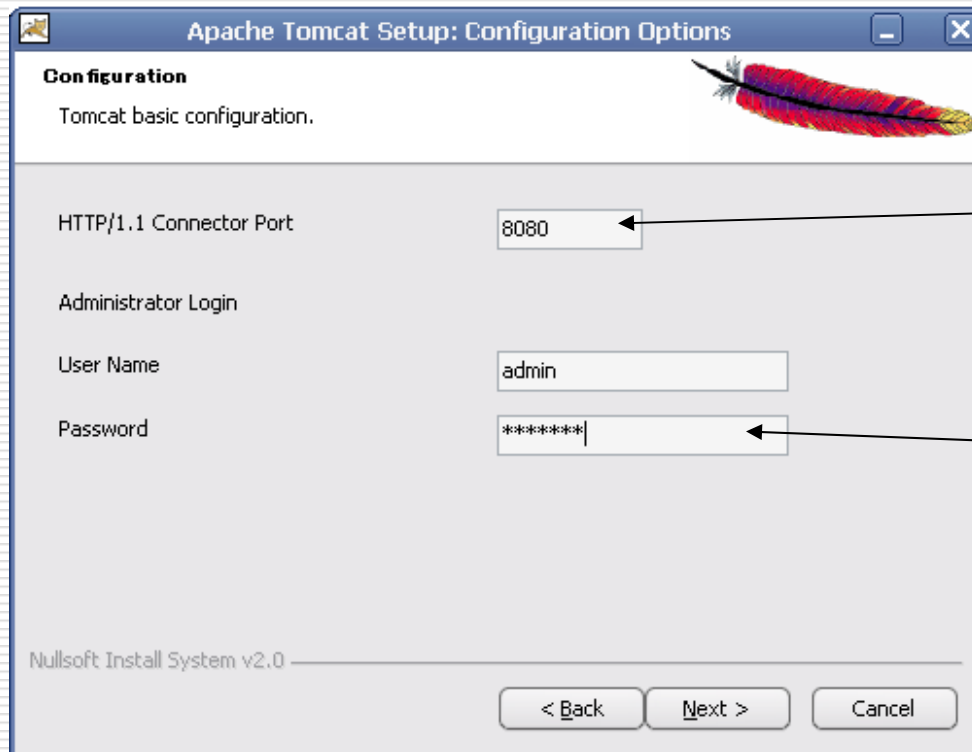
□ 基本的にデフォルトでOK



**Service  
Start Menu Items  
Documentation  
Examples  
にチェック**

# Tomcatのインストール(2)

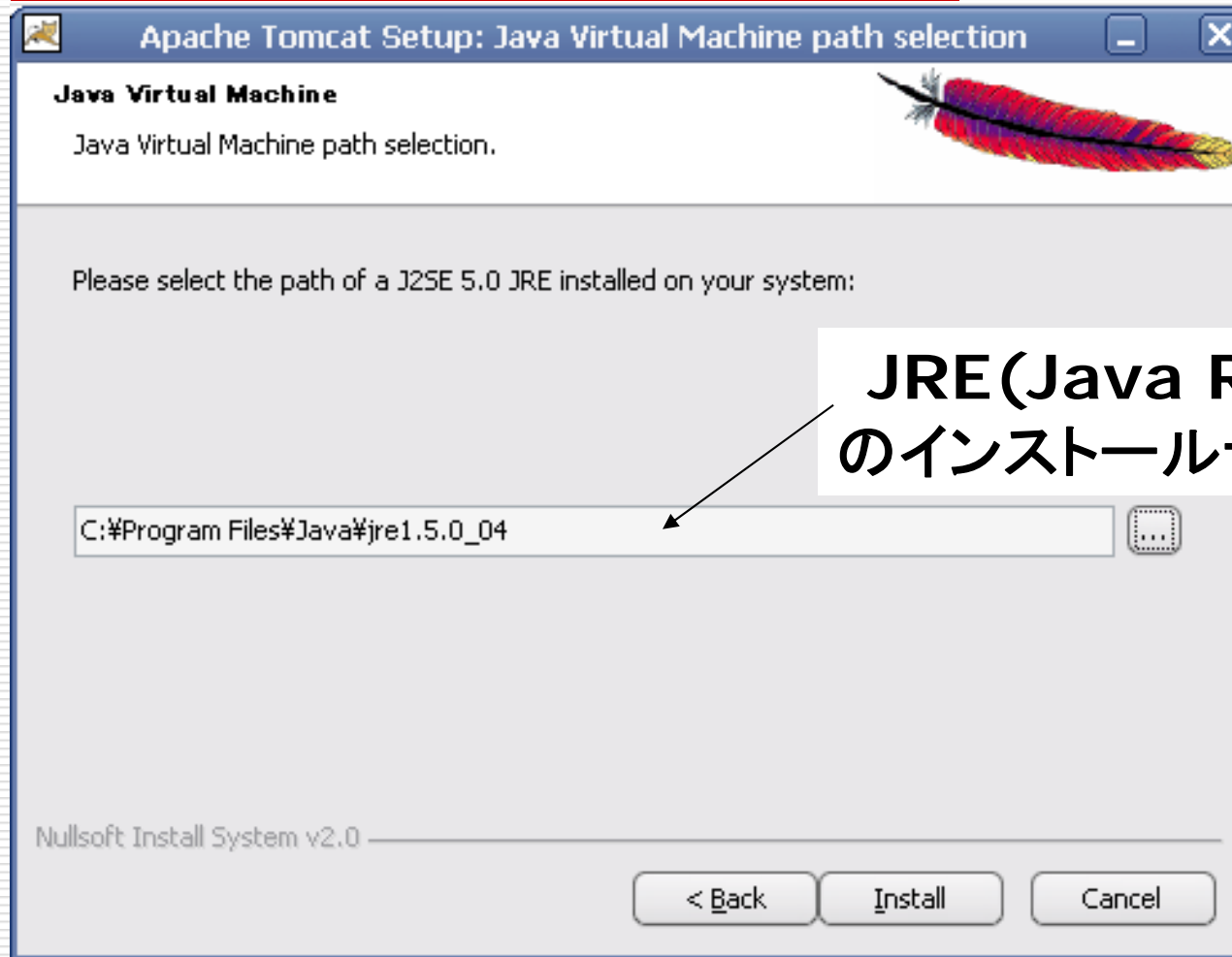
---



Tomcatの  
使用ポートは8080

管理パスワードを  
つけておく

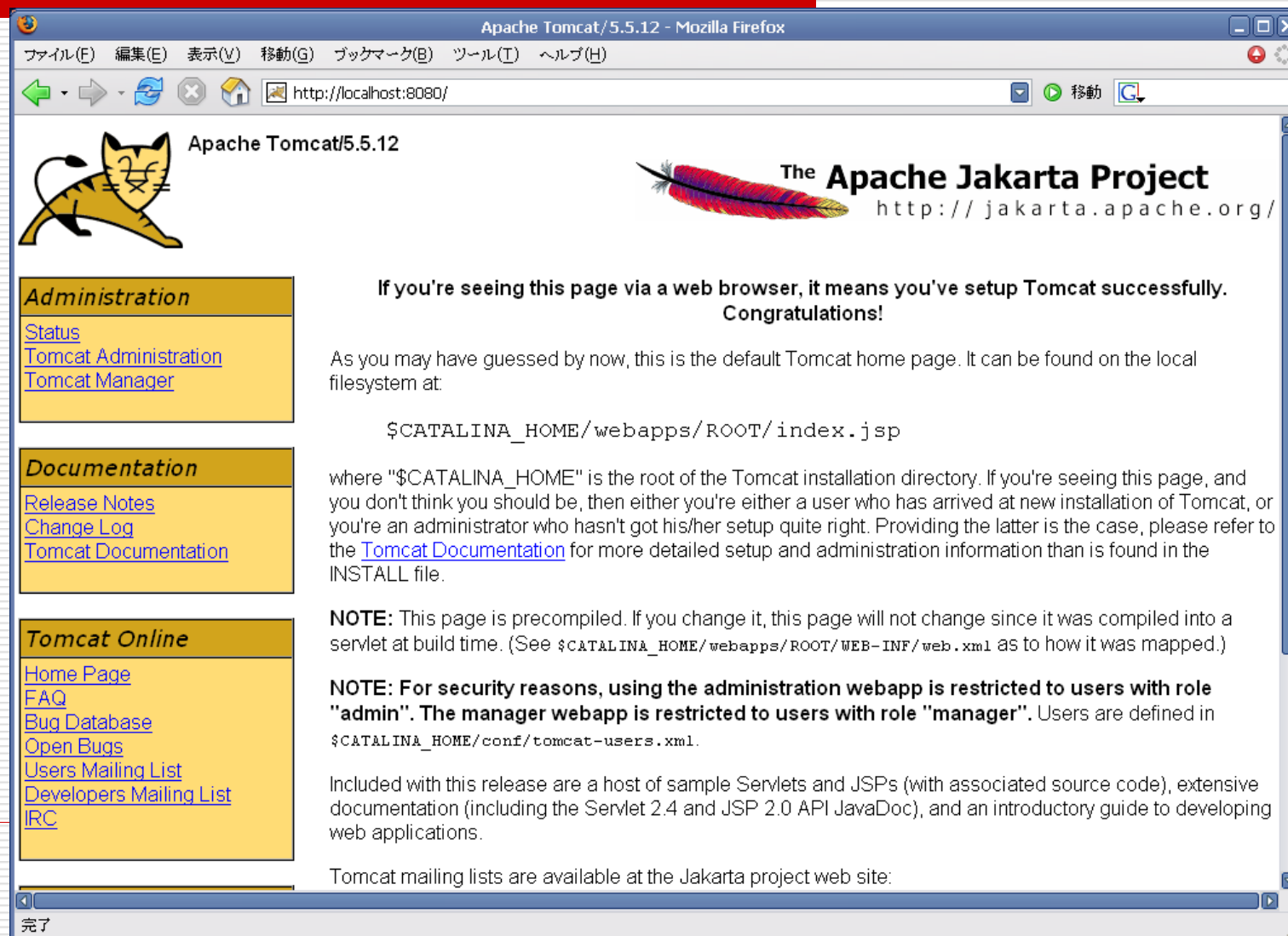
# Tomcatのインストール(2)



JRE (Java Runtime環境)  
のインストールディレクトリを指定

# 成功すると

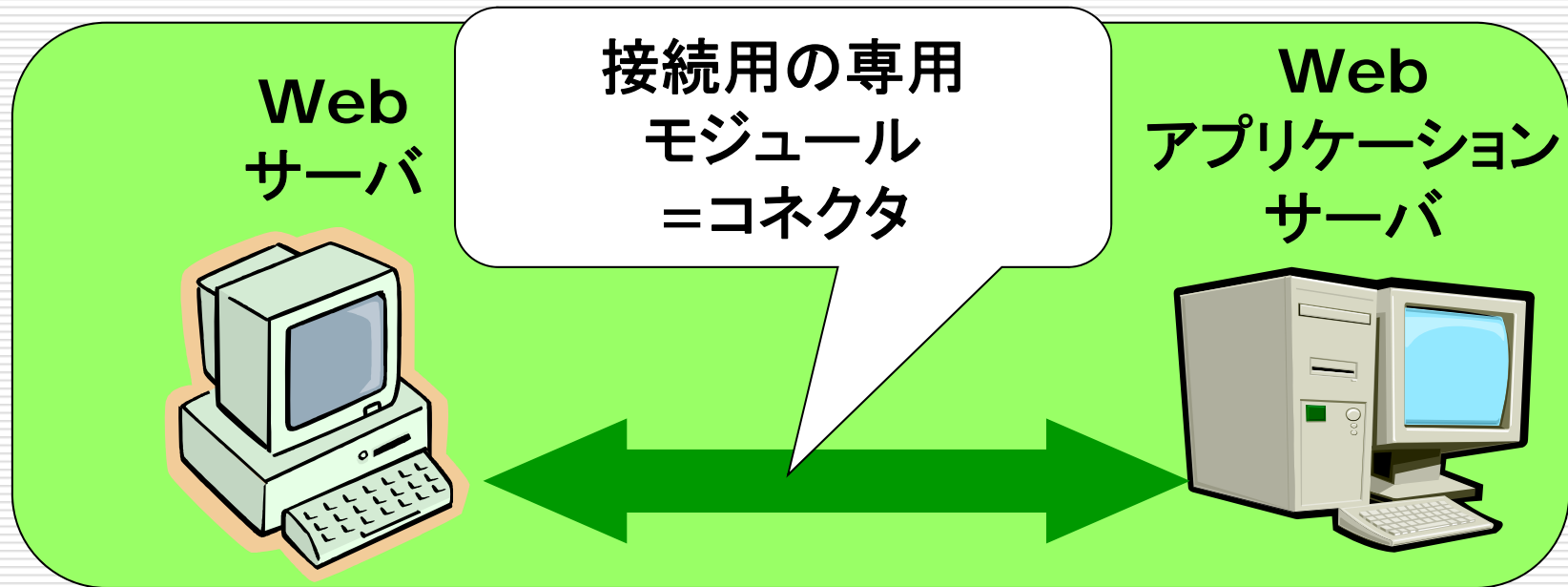
# http://localhost:8080/で以下の画面が表示される



# コネクタ

proxy\_module, proxy\_ajp\_module

- proxy\_module, proxy\_ajp\_moduleというモジュールを使ってWebサーバとWebアプリケーションサーバを連携させる



# Apache側の設定: httpd.conf (1)

---

- httpd.confはapacheの重要な設定ファイル
    - 複数のファイルに分かれている場合もあり
  - コネクタとして使うモジュールについての以下の行の先頭文字#を削除
    - LoadModule proxy\_module ...
    - LoadModule proxy\_ajp\_module ...
-



## Apache側の設定: httpd.conf (2)

---

- ファイルの最後に転送先となるwebアプリケーションサーバへの接続情報を記入

```
<Location>
```

```
    ProxyPass ajp://localhost:8009/
```

```
</Location>
```

---

動作確認(サンプルプログラムを動かしてみる)

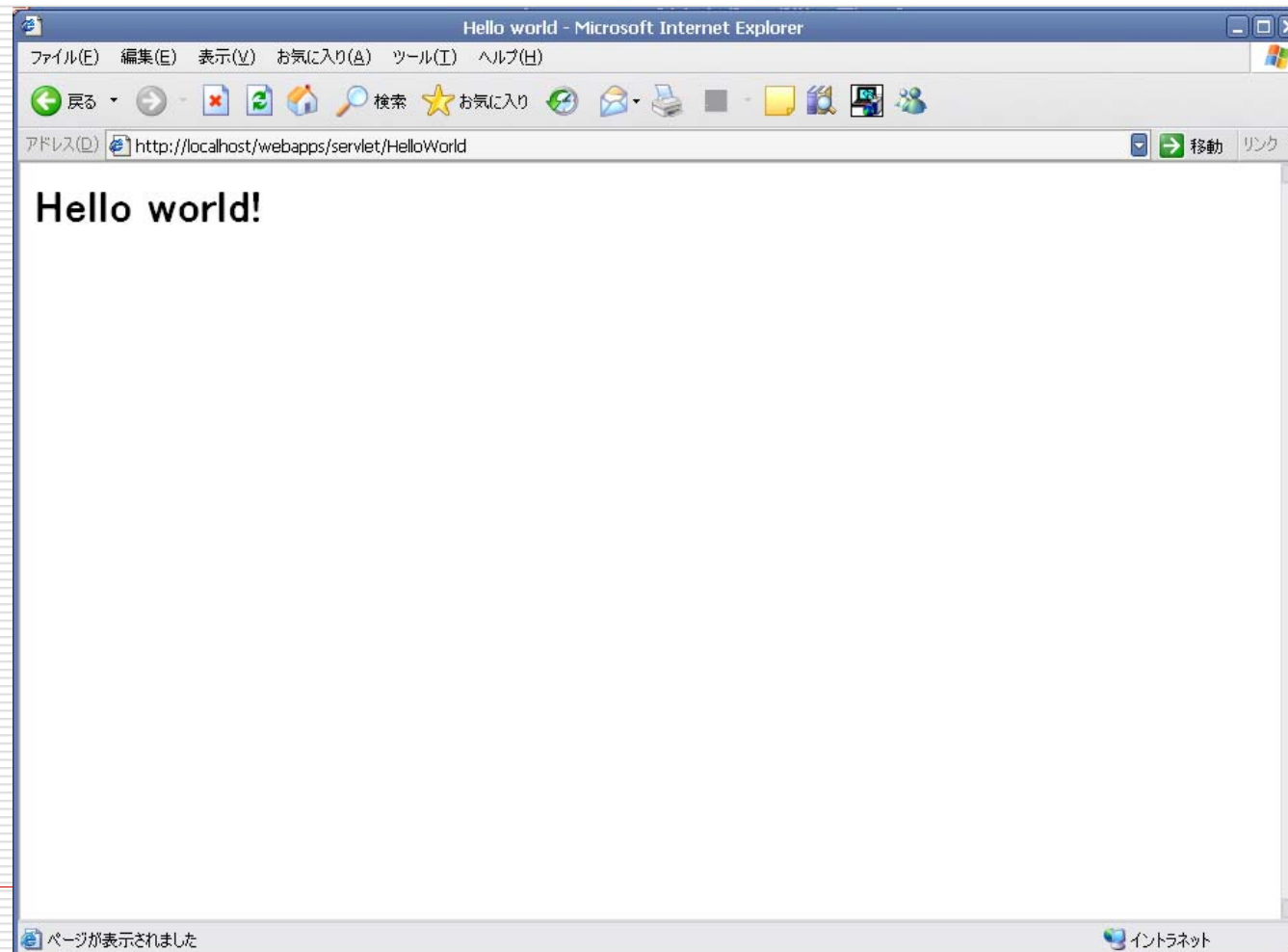
---

```
import java.io.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class HelloWorld extends HttpServlet {
    public void doGet(HttpServletRequest request,
                      HttpServletResponse response)
        throws IOException, ServletException
    {
        PrintWriter out = response.getWriter();
        out.println("<html><head>");
        out.println("<title>Hello world</title></head>");
        out.println("<body><h1> Hello world! </h1>");
        out.println("</body>");
        out.println("</html>");
    }
}
```

# 実行結果

---



## 参考URL

---

- <http://programnet.hp.infoseek.co.jp/jsp&servlet/tomcat.html>
  - <http://www.hellohiro.com/servletwebxml.htm>
  - [http://www.castle.flnet.org/pw/?Java%2FTomcat5.5%2Fmod\\_jk](http://www.castle.flnet.org/pw/?Java%2FTomcat5.5%2Fmod_jk)
-