プログラミング　Java 3

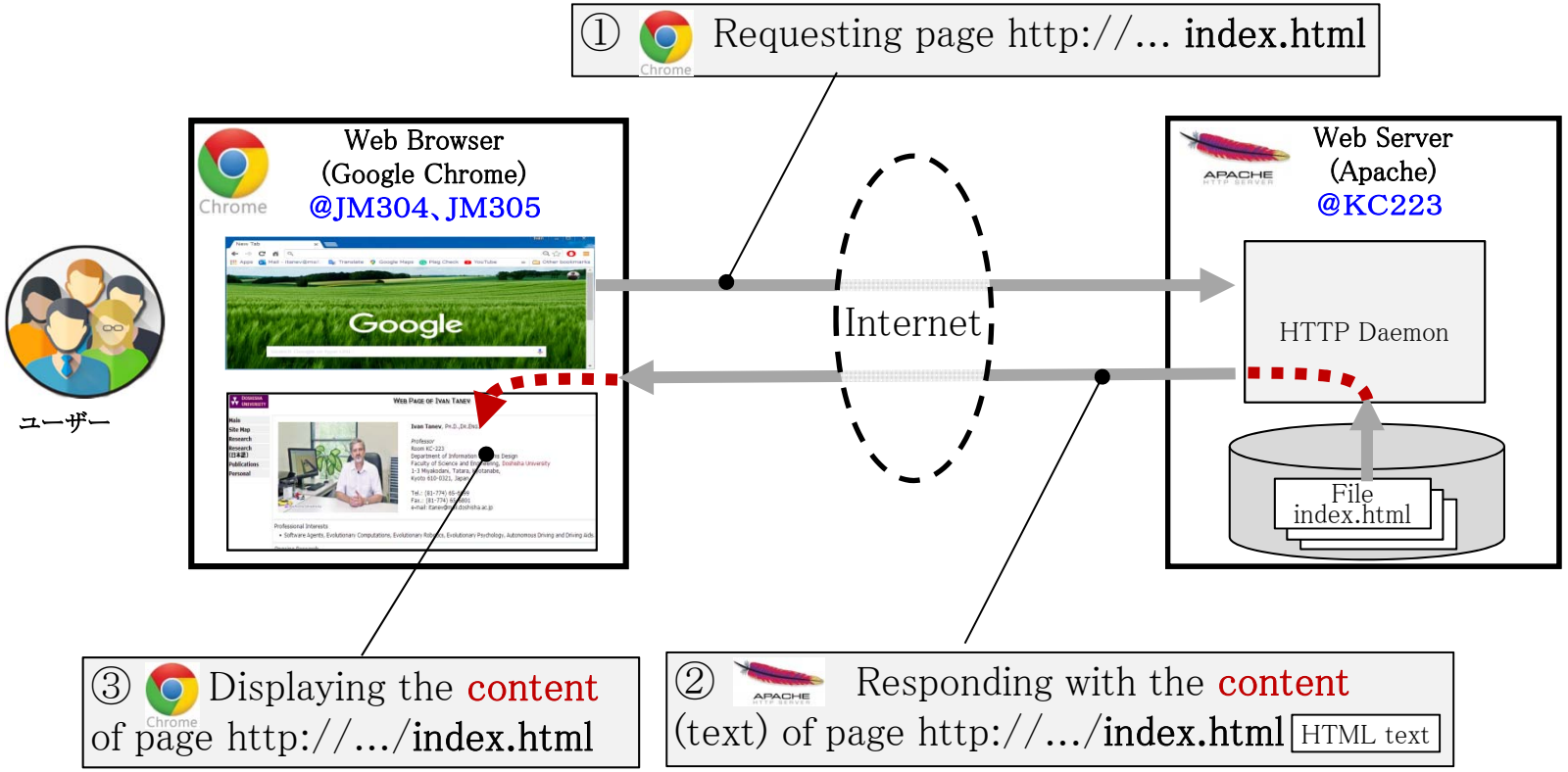
第2回:
Web Formおよびサーブレット入門

Ivan　Tanev

講義の構造

1. スタティック(静的な)Webコンテンツ
2. ダイナミック (動的)Webコンテンツ
3. ダイナミックWebコンテンツとサーブレット
4. Webフォーム
5. 演習

1. スタティック(静的な)Webコンテンツ



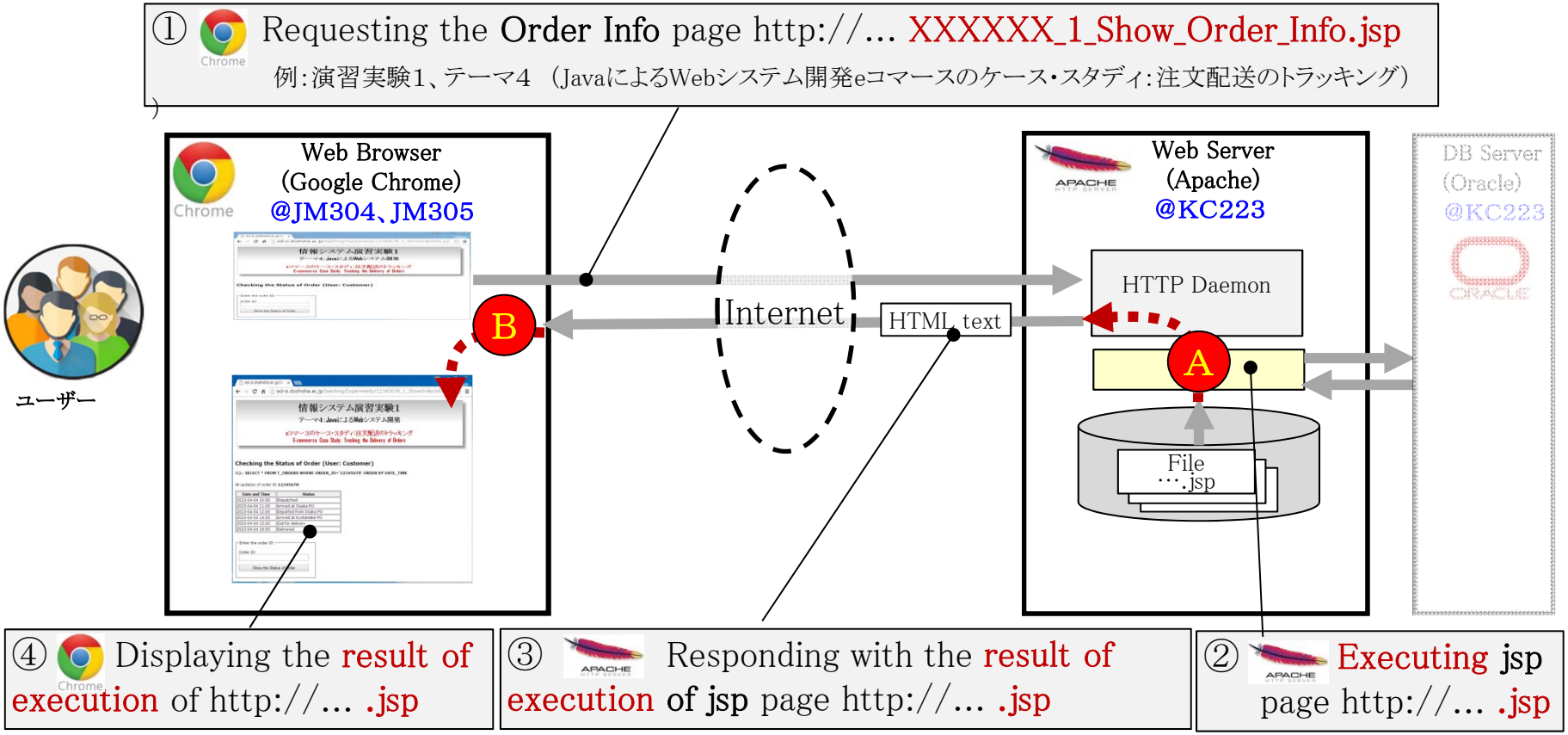
Q: なぜ「静的なWebコンテンツ」と呼ばれているのですか？

A: (a) Web server上に保存された内容と②の時に通信された内容は同じですので。

(b) ②の時に通信された内容と③の時に表示された内容は同じですので。

(a)&(b) => ③の時に表示された内容とWeb server上に保存された内容は同じです.

3. ダイナミック (動的)Webコンテンツ



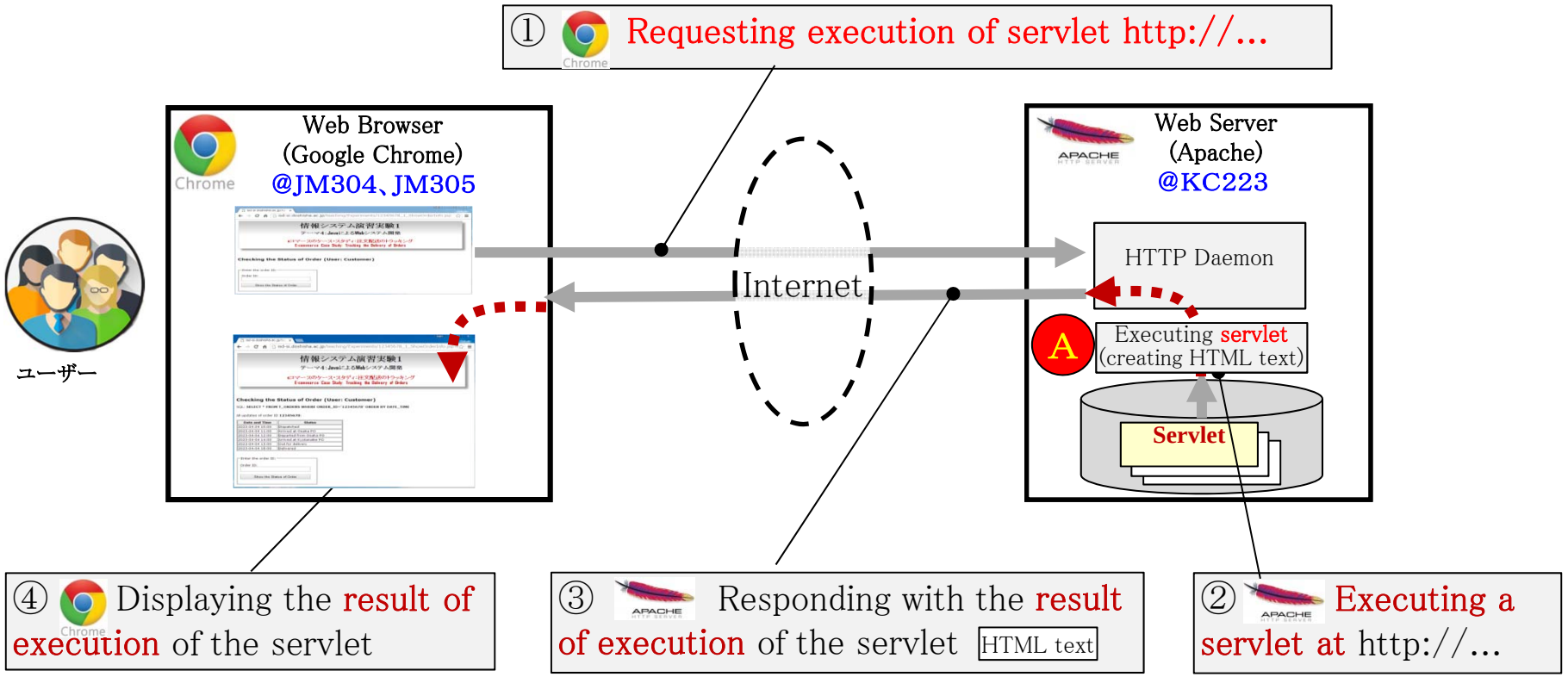
Q: Aのデータ処理はどのように実現されていますか？

A: **サーバサイドスクリプト**:Web server上に**実行されたコード(script)**です. 例えば、Common Gateway Interface (CGI), Java servlet, Java Server Pages (jsp), Active Server Pages (asp), Server Side Java Scriptなど。

Q: Bのデータ処理はどのように実現されていますか？

A: **クライアントサイドスクリプト**:Web browser上に**実行されたコード(script)**です. 例えば、Java Script, ActiveX, Visual Basic Scriptなど。

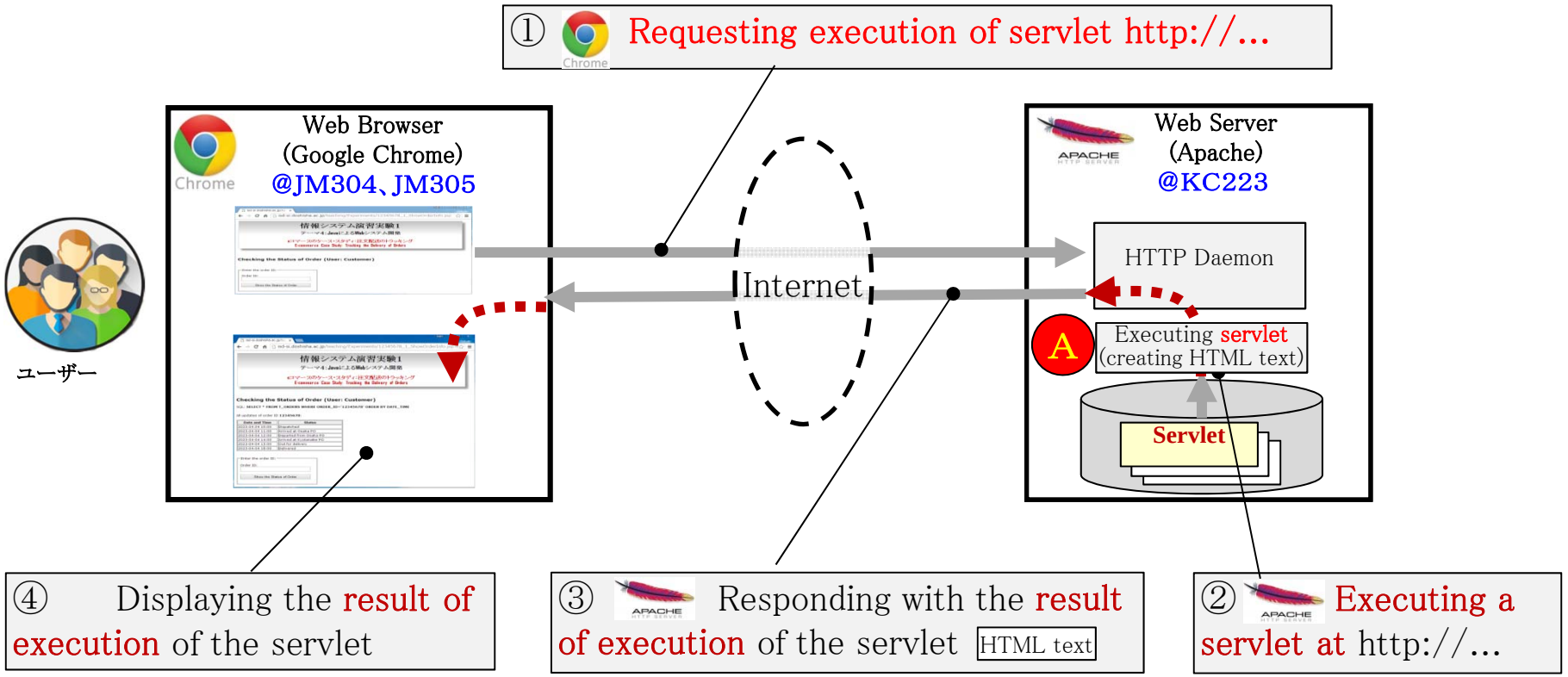
3. ダイナミック (動的)Webコンテンツとサーブレット



Q: A のデータ処理はどのように実現されていますか？

A: サーバサイドスクリプト:Web server上に実行されたコード(script)です. 例えば、Java servlet.

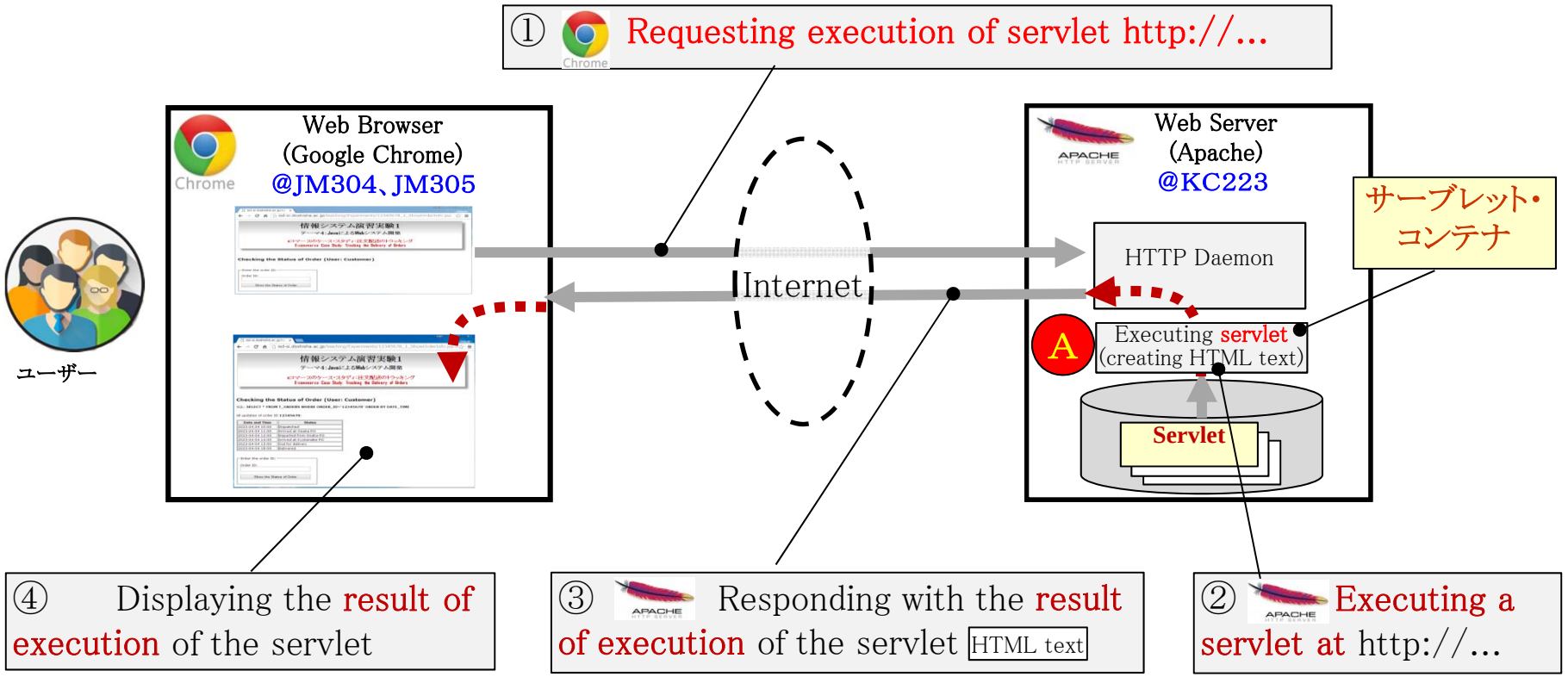
3. ダイナミック (動的)Webコンテンツとサーブレット



Q: サーブレットの**主な特徴**は何ですか?

A: 次のスライドに参考。

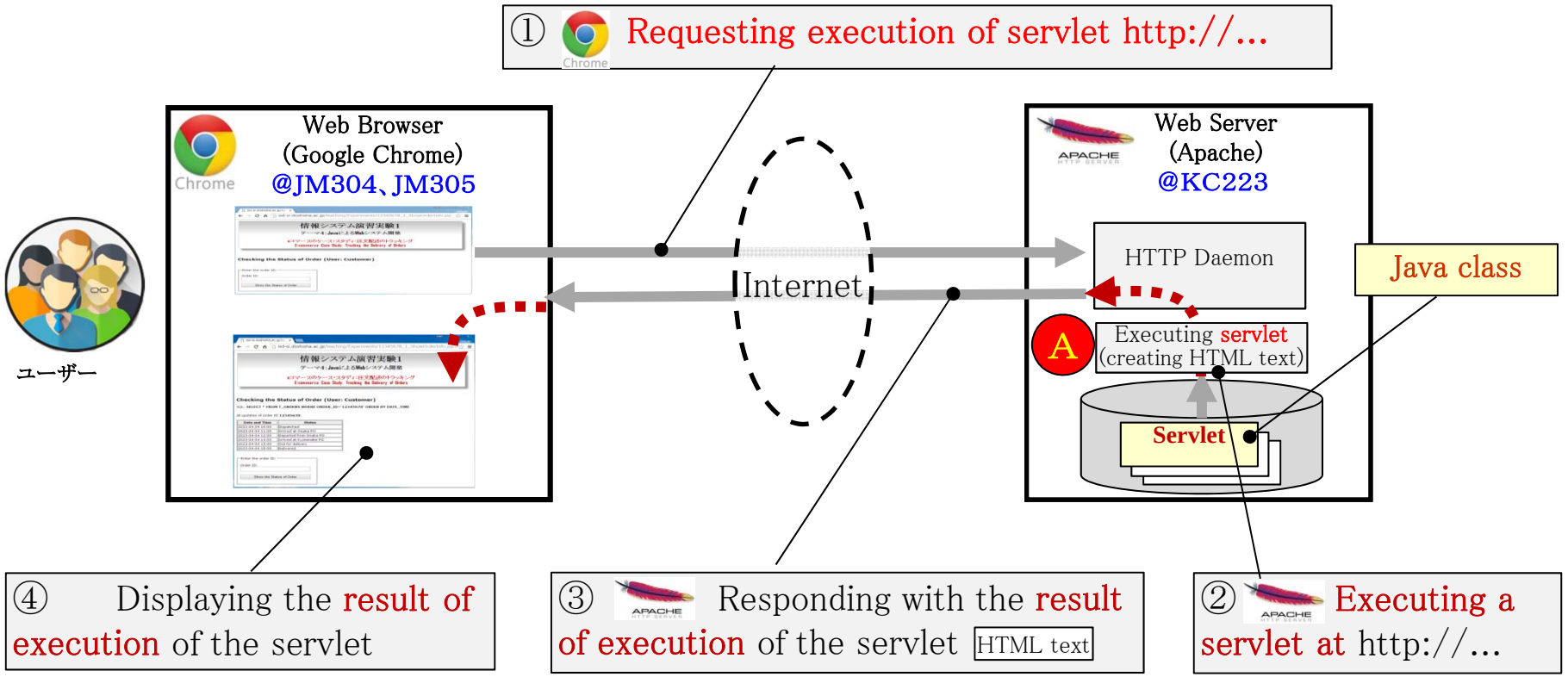
3. ダイナミック (動的)Webコンテンツとサーブレット



サーブレットの特徴 (1/2)

- Sun Microsystemsで開発さています。
- サーブレットはサーバサイドスクリプトである。
- 実行されている所は:
  サーブレット・コンテナおよびサーブレット・エンジンと呼ばれています。
- このスクリプトはJava言語で開発されています。

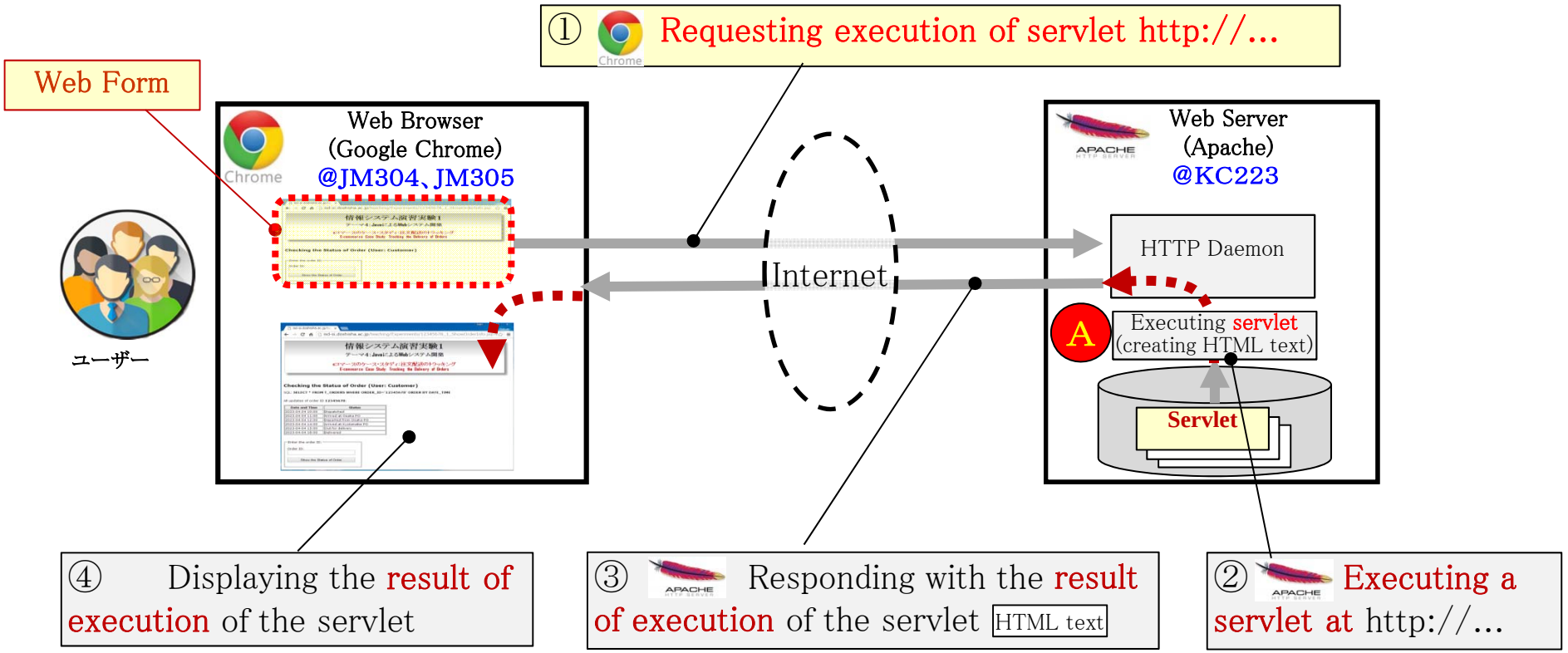
3. ダイナミック (動的)Webコンテンツとサーブレット



サーブレットの特徴 (2/2)

- Java言語:サーブレットはwrite once, run anywhereのJava classです。
- マルチスレード
- サーブレットの実行されたダイナミックな結果は:
  - ユーザインプットとWebサーバのダイナミックな状態が関係ある
  - ユーザインプットとDBのダイナミックな状態が関係ある

3. ダイナミック (動的)Webコンテンツとサーブレット



Q: Webブラウザからサーブレットを**呼び出す方法**は？

A: Web Formからサーブレットを**呼び出す**(次のスライドに参考)。

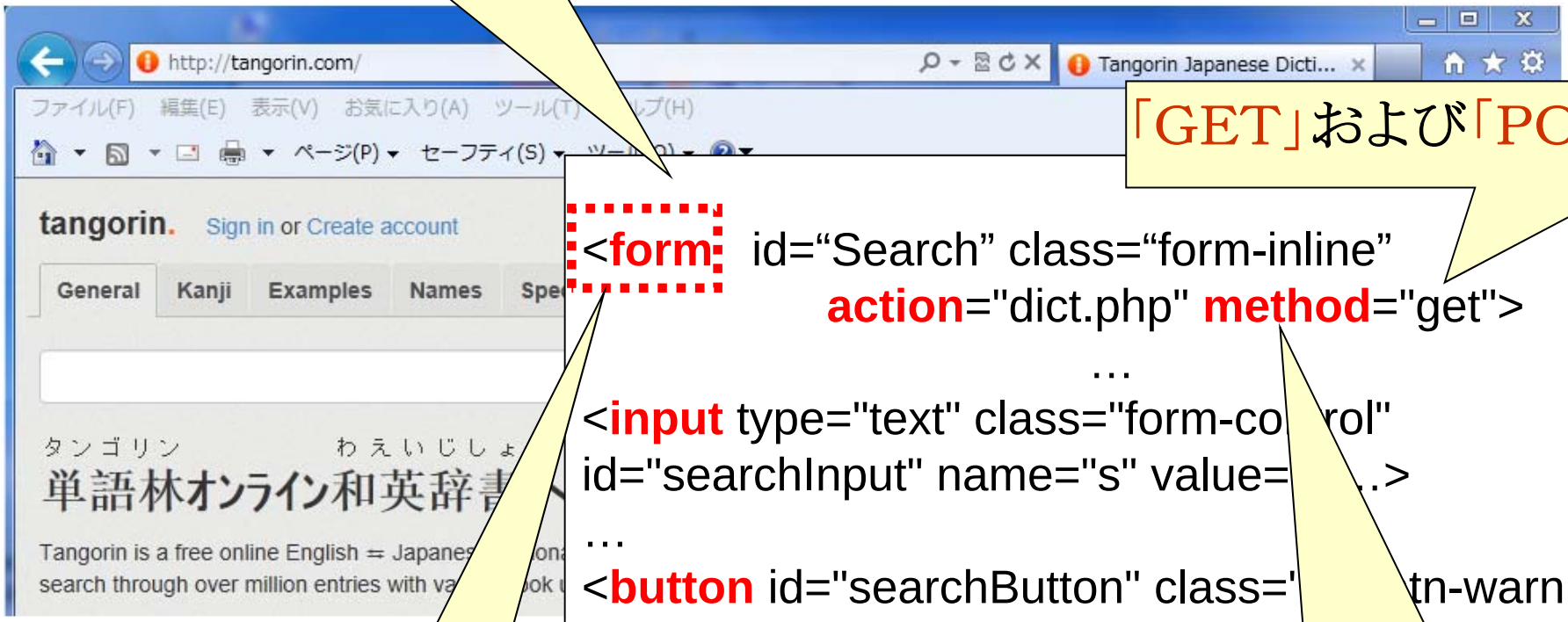
4. Webフォーム

http://tangorin.com/

3 「表示」から「ソース(C)」を開いて下さい。

4. Webフォーム

ページのソース



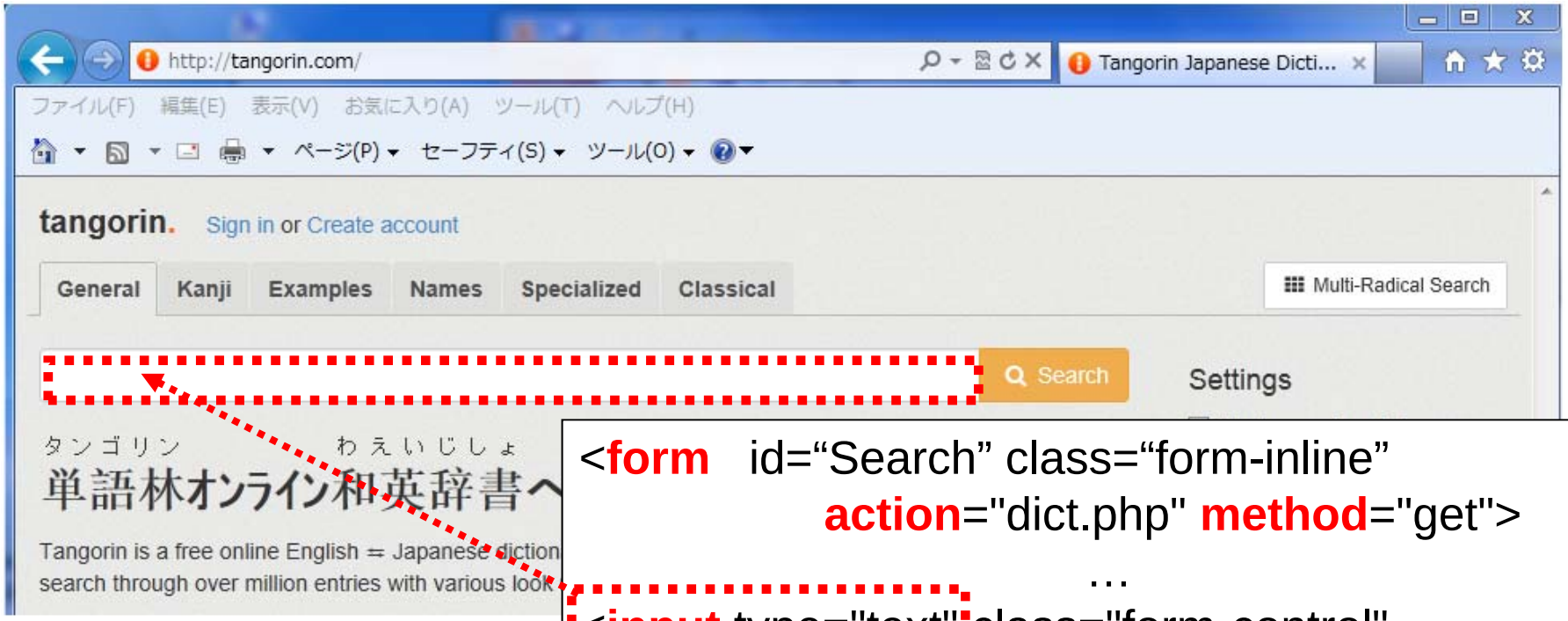

```
<form id="Search" class="form-inline"
      action="dict.php" method="get">
      …
<input type="text" class="form-control"
id="searchInput" name="s" value="…>
…
<button id="searchButton" class="…btn-warning"
tabindex="3" type="submit">
…
</form>
```

「GET」および「POST」

「FORM」はHTMLタッグ名である

「METHOD」はHTMLパラメター名である

4. Webフォーム

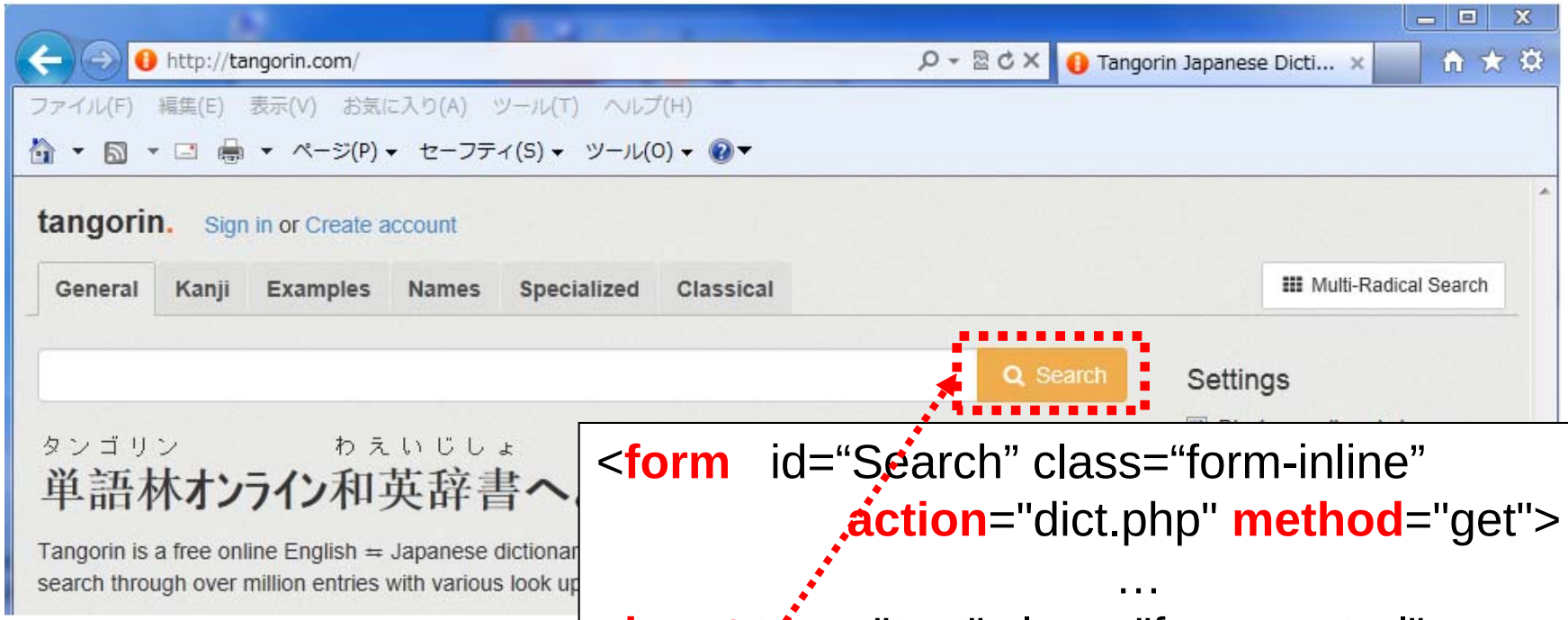



```
<form   id="Search" class="form-inline"
                action="dict.php" method="get">
                                …
<input type="text" class="form-control"
id="searchInput" name="s" value="" …>
…
<button id="searchBut…  …ss="btn btn-warning"
tabindex="3" type="…
…

</form>
```

「INPUT」はHTMLタッグ名である

「TYPE」,「ID 」はHTMLの「INPUT」タッグのパラメター名である

4. Webフォーム




```
<form   id="Search" class="form-inline"
        action="dict.php" method="get">
                …
<input type="text" class="form-control"
id="searchInput" name="s" value="" …>
…
<button id="searchButton" class="btn btn-warning"
tabindex="3" type="submit">
…
</form>
```

「BUTTON」はHTMLタッグ名である

「TYPE」,「ID」はHTMLの「BUTTON」タッグのパラメター名である

4. Webフォーム

「ACTION」はHTMLパラメター名である

```
<form   id="Search" class="form-inline"
        action="dict.php" method="get">
        …
<input type="text" class="form-control"
id="searchInput" name="s" value="" …>
…
<button id="searchButton" class="btn btn-warning"
tabindex="3" type="submit">
…
</form>
```

「searchButton」ボタンを押すとformの「searchInput」のデータはこの「dict.php」のスクリプトへ送られています。このスクリプトはServer Sideで実行されています。

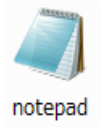
4. Webフォーム


```
<form   id="Search" class="form-inline"
                action="dict.php" method="get">
                                …
<input type="text" class="form-control"
id="searchInput" name="s" value="" …>
…
<button id="searchButton" class="btn btn-warning"
tabindex="3" type="submit">
…
</form>
```

Webフォームには下記の特徴が定義されています：

- 何時（何のイベントの時に）サーバサイドスクリプトを実行するか？ **A: <button id="a" … type="submit">**
- 何のユーザー入力されたパラメターはスクリプトにおくるか？ **A: < input id= "b"…>**
- 何のサーバサイドスクリプトを実行するか？ **A: <form action="c"…>**

5. 演習

A. Servletを実行するため、Webフォームを作り出す
（ 

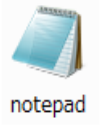
Notepadで）

B. WebフォームをWebサーバにUpload
（ 

FTP Clientで）

C. Webフォームのアクセス
（ 

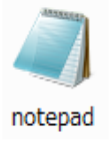
Google Chromeで）

D. サーブレットのソース（java）を確認
（ 

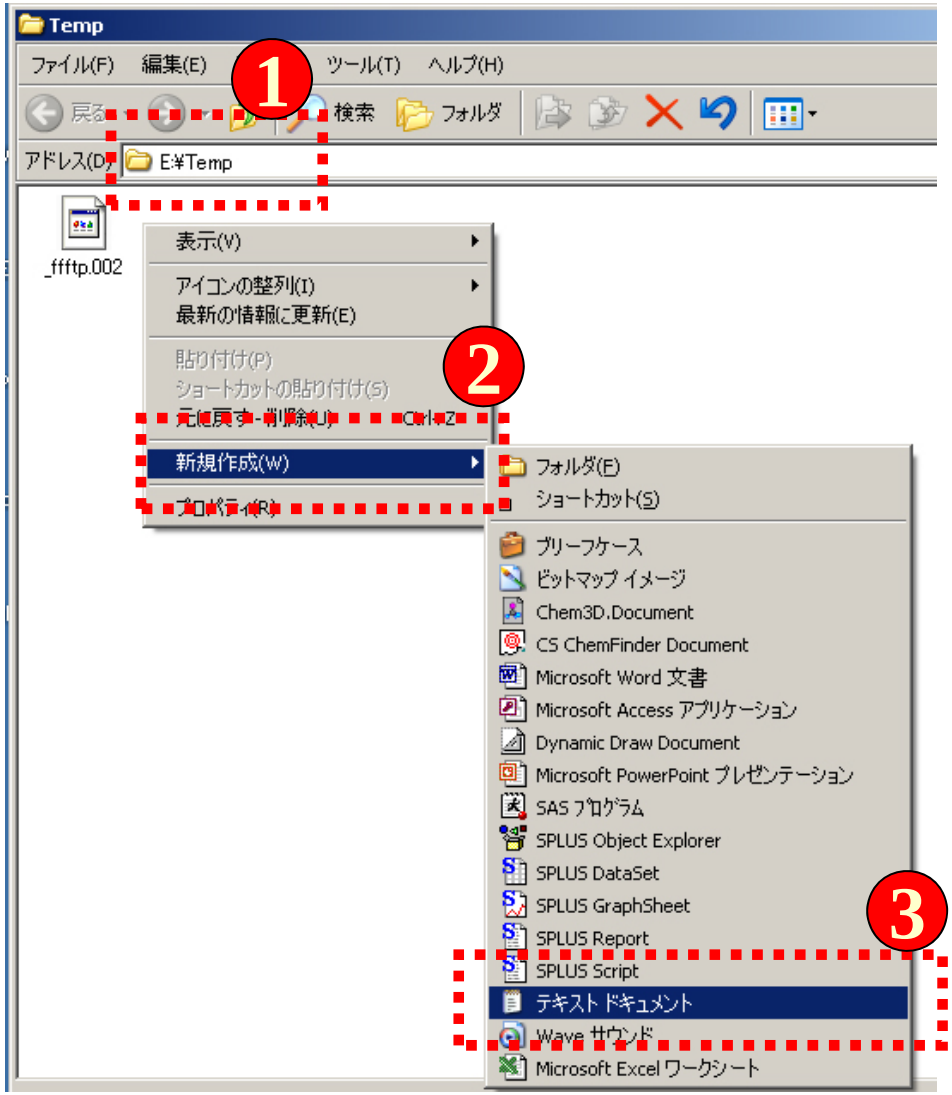
Notepadで）

5. 演習

A. Webフォームを作り出す

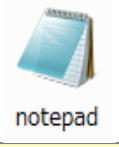


Notepadで



5. 演習

A. Webフォームを作り出す



Notepadで

Webフォームのソース:

Lecture_2_Form - メモ帳

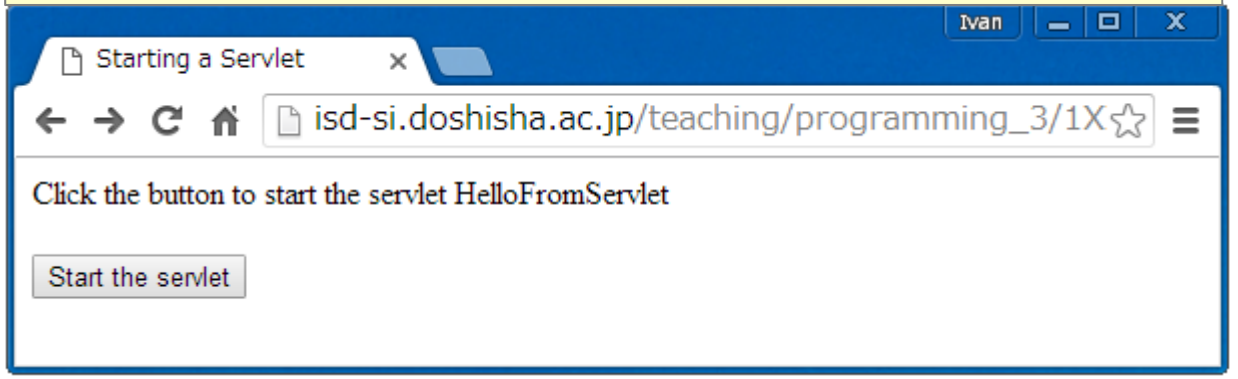
ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)

```
<HTML>
<HEAD><TITLE>Starting a Servlet</TITLE></HEAD>
<BODY>
    <FORM ACTION = "/servlet/HelloFromServlet" METHOD = "get">
       <P><LABEL>Click the button to start the servlet HelloFromServlet<BR><BR>
          <INPUT TYPE = "submit" VALUE = "Start the servlet" />
       </LABEL></P>
    </FORM>
</BODY>
</HTML>
```

Webフォームには下記の特徴が定義されています:

- 何時(何のイベントの時に)サーバサイドスクリプトを実行するか？ **A: <input type="submit" …>**
- 何のユーザー入力されたパラメターはスクリプトにおくるか？ **A: < input id= " "…>**
- 何のサーバサイドスクリプトを実行するか？ **A: <form action=" "…>**

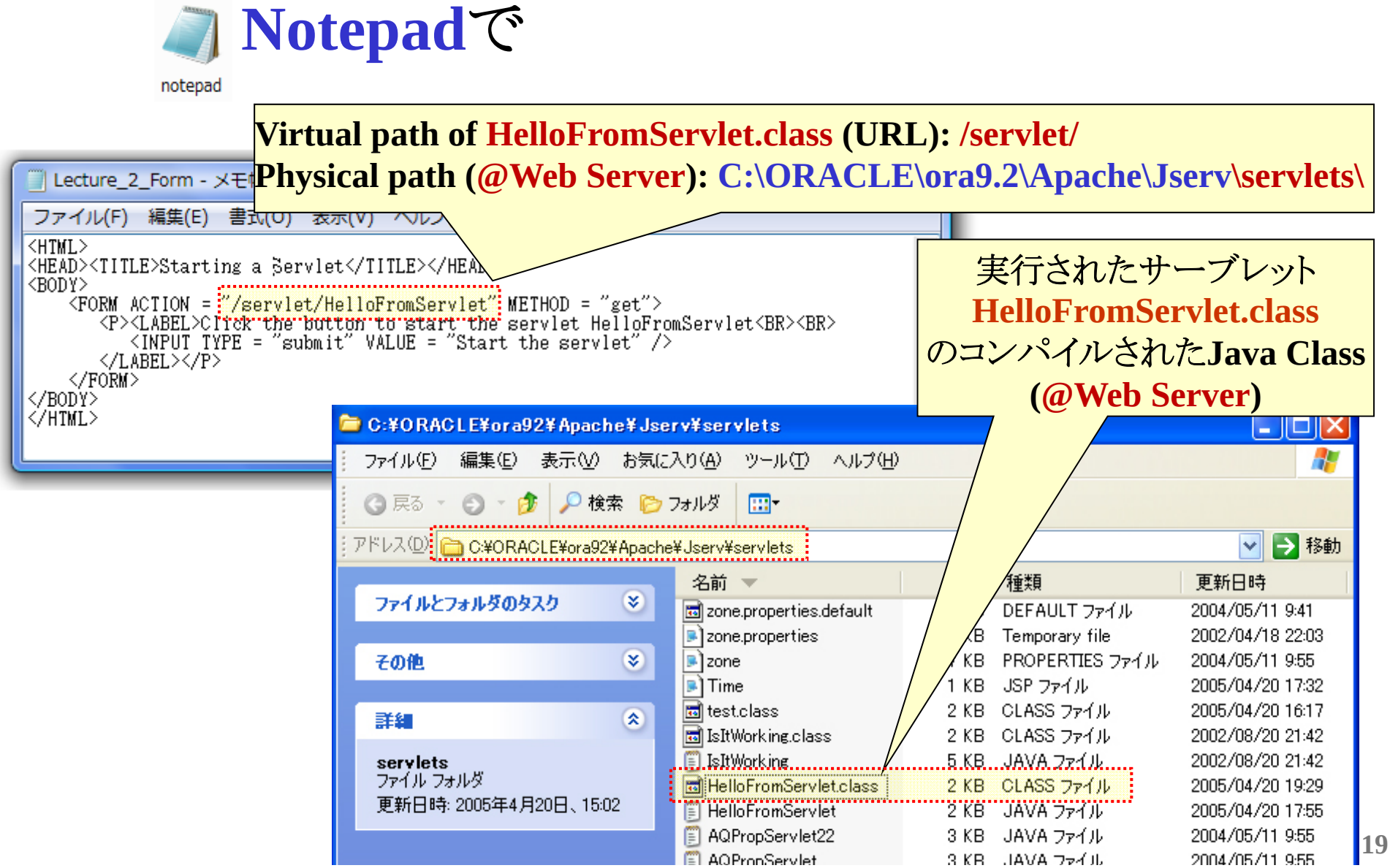
WebフォームのWeb-browser上の表現:



5. 演習

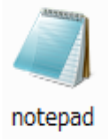
A. Webフォームを作り出す

Notepadで

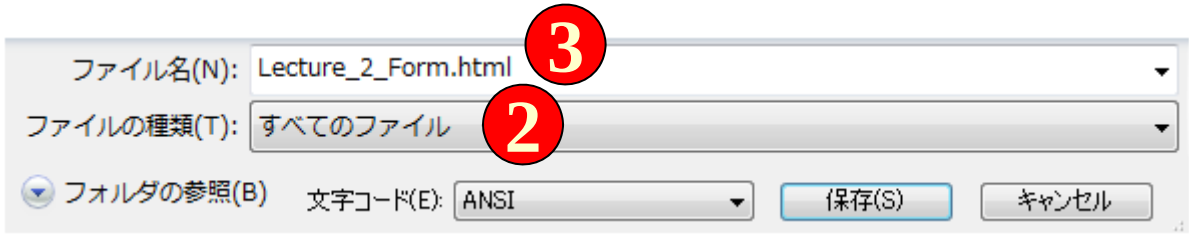


5. 演習

A. Webフォームを作り出す



Notepadで

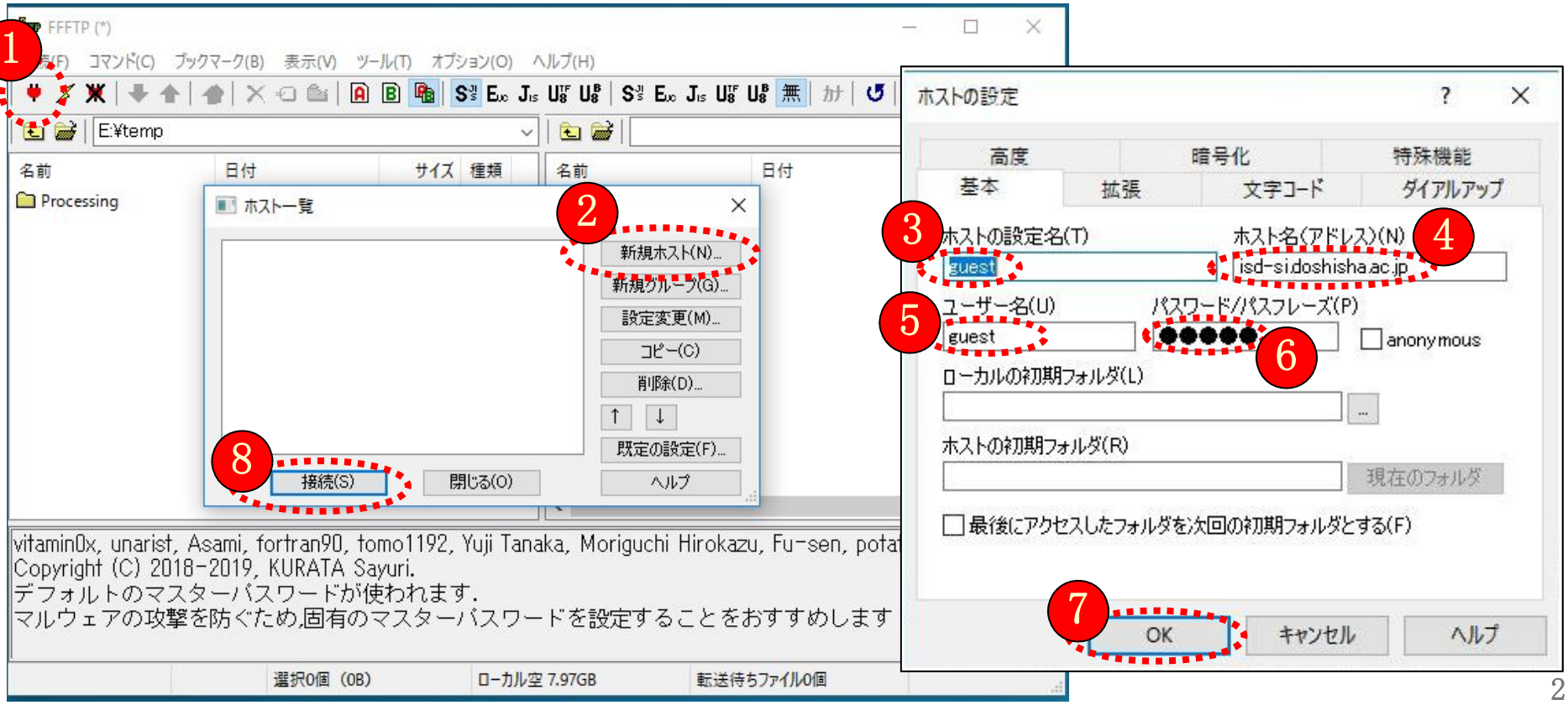


5. 演習

B. WebフォームをWebサーバにUpload

FTP Clientで

「ホストの設定」「基本」のページでFTPサーバーのパラメターを記入。ホスト名④は「isd-si.doshisha.ac.jp」で、ユーザ名⑤とパスワード⑥は「guest」です。

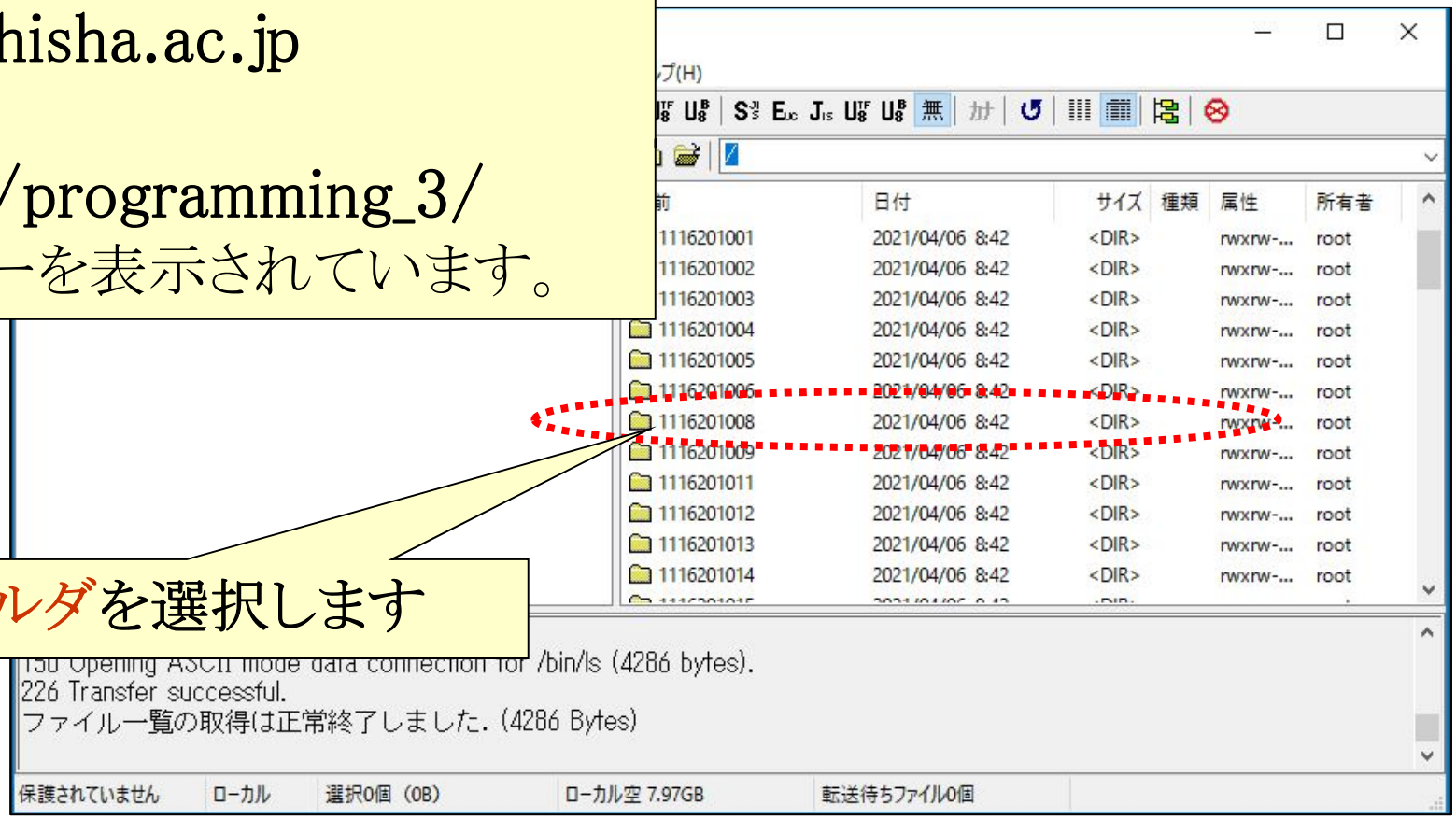


5. 演習

B. WebフォームをWebサーバにUpload

FFFTP FTP Clientで

接続してから
isd-si.doshisha.ac.jp
のサーバー
/teaching/programming_3/
のフォルダーを表示されています。



! 自分のフォルダを選択します

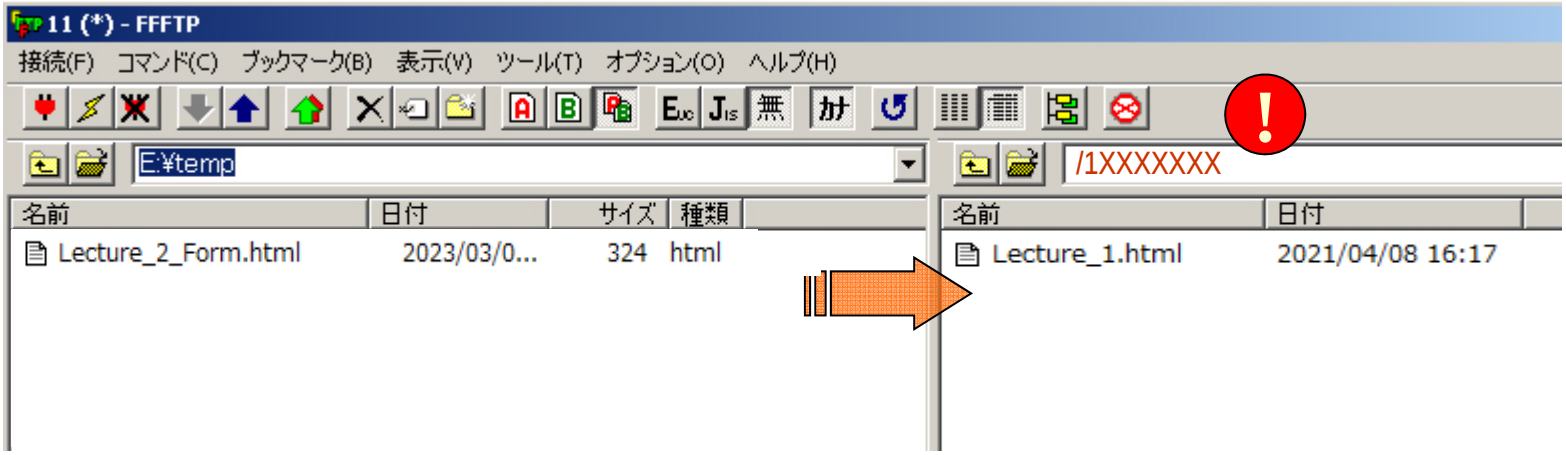
5. 演習

B. WebフォームをWebサーバにUpload

FTP Clientで

Webフォームファイル
E:¥temp¥Lecture_2_Form.html
Webサーバのフォルダー
/1XXXXXXXX
にUploadします。

注意： 1XXXXXXX =自分の学籍番号



5. 演習

C. Webフォームのアクセス

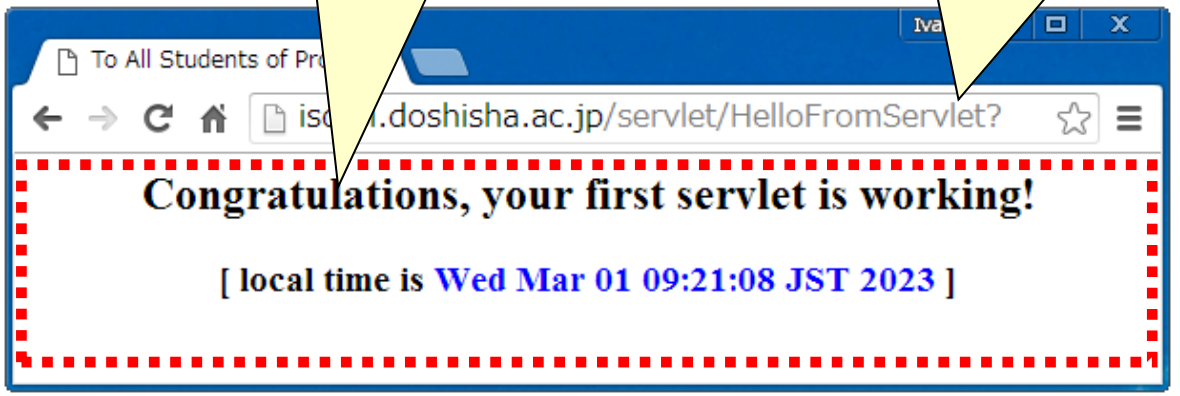
Google Chrome, or Internet Explorerで

1 URL: http://isd-si.doshisha.ac.jp/teaching/programming_3/**1XXXXXXX**/Lecture_2_Form.html !

注意： 1XXXXXXX =自分の学籍番号

サーブレットの実行結果

サーブレット名,パラメタなし



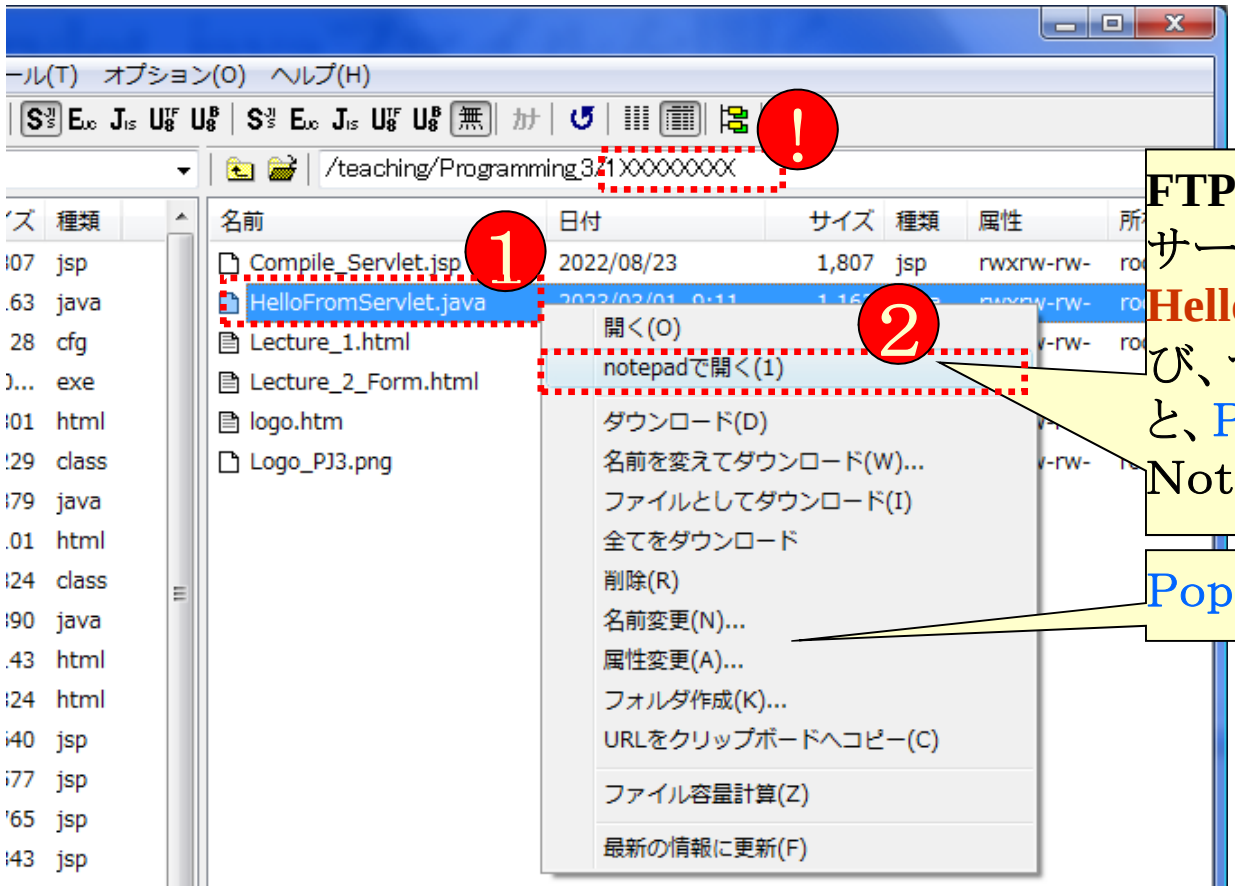
5. 演習

D. サーブレットのソース(java)を確認

FTP Client の右のパネルに表示された HelloFromServlet.javaファイルを開く（ Notepadで)。

notepad

注意： 1XXXXXXX =自分の学籍番号



FTPクライアント
サーブレットソースファイル **HelloFromServlet.java**を選び、マウスの右ボタンを押すと、Pop－upメニュが出ます。Notepadで開きます。

Pop－upメニュ

5. 演習

D. サーブレットのソース(java)を確認

サーブレット**HelloFromServlet**のJavaソース

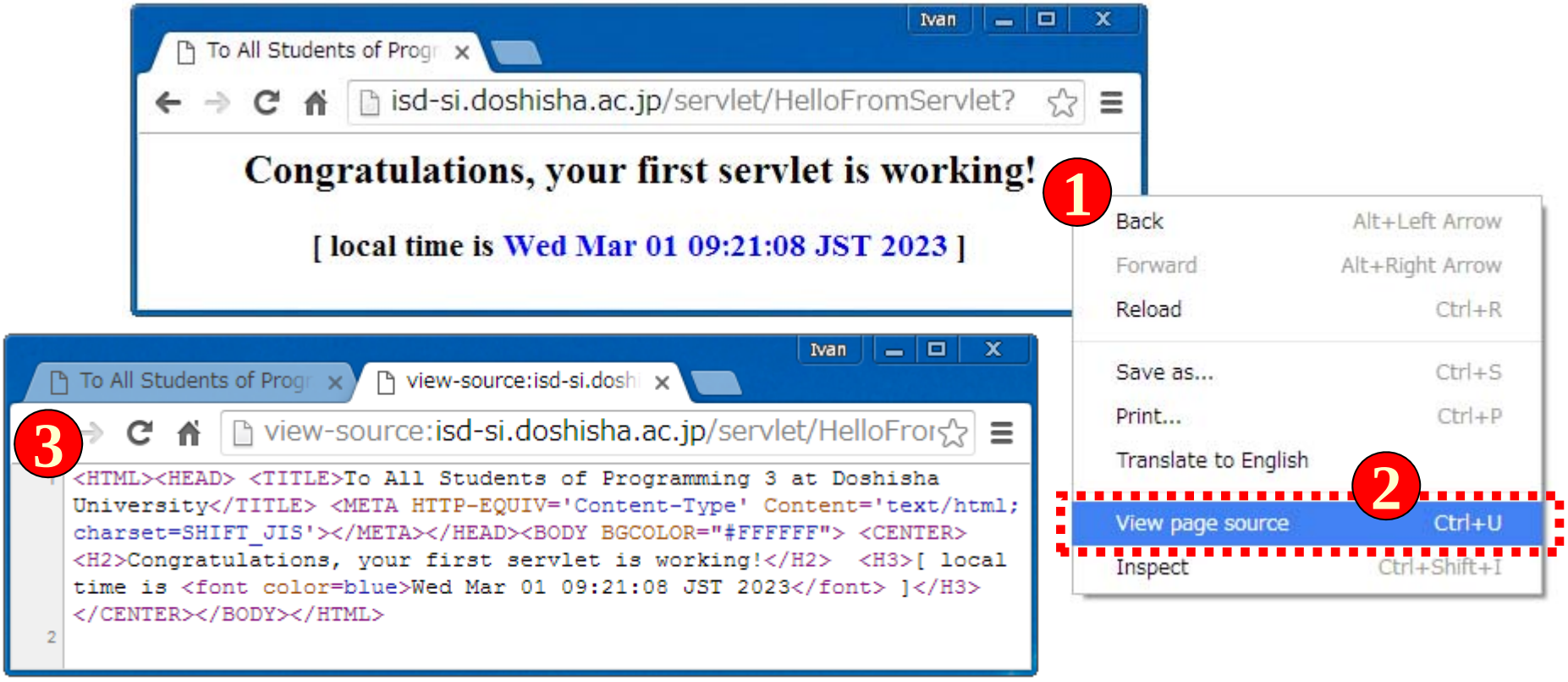
```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class HelloFromServlet extends HttpServlet {

    public void service (HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException
    {
        // set content type and other response header fields first
        response.setContentType("text/html");

        // get the communication channel with the requesting client
        PrintWriter out = response.getWriter();

        // write the data
        out.println("<HTML>"
            + "<HEAD>"
            + " <TITLE>To All Students of Programming 3 at Doshisha University</TITLE>"
            + " <META HTTP-EQUIV='Content-Type' Content='text/html; charset=SHIFT_JIS'></META>"
            + "</HEAD>"
            + "<BODY BGCOLOR=\"#FFFFFF\">"
            + " <CENTER>"
            + "  <H2>Congratulations, your first servlet is working!</H2>"
            + "  <H3>[ local time is <font color=blue>"
            + new java.util.Date()
            + "</font> ]</H3>"
            + " </CENTER>"
            + "</BODY>"
            + "</HTML>");
    }
}
```

5. 演習

D. サーブレットのソース(java)を確認

サーブレットの実行結果を検査します:

1. **Web Briowser**の画面にマウスの右ボタンをクリックします、
2. **Pop-up**メニュから:ソースを示す、
3. **Page**のソース(サーブレット**HelloFromServlet**の実行結果)を確認します。

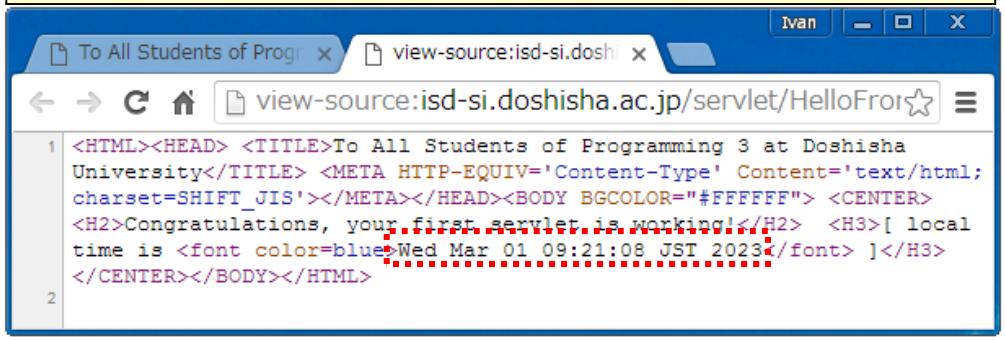


5. 演習
D. サーブレットのソース(java)を確認

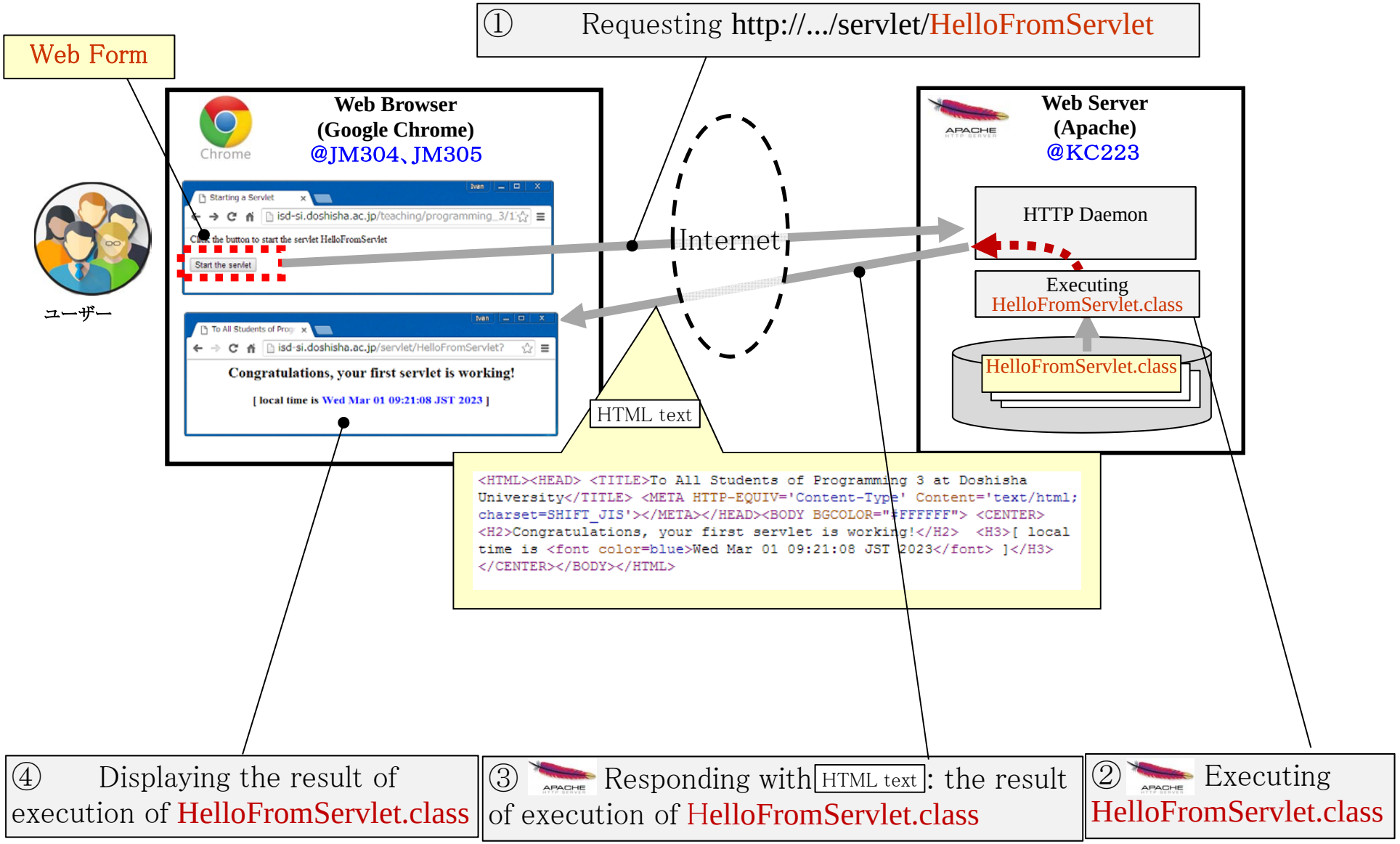
サーブレットHelloFromServletのJavaソース

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

public class HelloFromServlet extends HttpServlet {

    public void service (HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException
    {
        // set content type and other response header fields first
        response.setContentType("text/html");

        // get the communication channel with the requesting client
        PrintWriter out = response.getWriter();

        // write the data
        out.println("<HTML>"
            + "<HEAD>"
            + " <TITLE>To All Students of Programming 3 at Doshisha University</TITLE>"
            + " <META HTTP-EQUIV='Content-Type' Content='text/html; charset=SHIFT_JIS'></META>"
            + "</HEAD>"
            + "<BODY BGCOLOR=\"#FFFFFF\">"
            + " <CENTER>"
            + "  <H2>Congratulations, your first servlet is working!</H2>"
            + "  <H3>[ local time is <font color=blue>"
            + new java.util.Date()
            + "</font> ]</H3>"
            + " </CENTER>"
            + "</BODY>"
            + "</HTML>");
    }
}
```

サーブレットHelloFromServletの実行結果



5. 演習 (まとめ)


① Requesting http://.../servlet/HelloFromServlet
Web Form
Web Browser
(Google Chrome)
@JM304、JM305
Chrome
Starting a Servlet
isd-si.doshisha.ac.jp/teaching/programming_3/1/
Click the button to start the servlet HelloFromServlet
Start the servlet
To All Students of Prog
isd-si.doshisha.ac.jp/servlet/HelloFromServlet?
Congratulations, your first servlet is working!
[local time is Wed Mar 01 09:21:08 JST 2023]
ユーザー
Internet
HTML text
<HTML><HEAD> <TITLE>To All Students of Programming 3 at Doshisha University</TITLE> <META HTTP-EQUIV='Content-Type' Content='text/html; charset=SHIFT_JIS'></META></HEAD><BODY BGCOLOR="#FFFFFF"> <CENTER> <H2>Congratulations, your first servlet is working!</H2> <H3>[local time is Wed Mar 01 09:21:08 JST 2023]</H3> </CENTER></BODY></HTML>
Web Server
(Apache)
@KC223
APACHE
HTTP Daemon
Executing
HelloFromServlet.class
HelloFromServlet.class
④ Displaying the result of execution of HelloFromServlet.class
③ Responding with HTML text: the result of execution of HelloFromServlet.class
② Executing HelloFromServlet.class

3. 演習

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

public class HelloFromServlet extends HttpServlet {

    public void service (HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException
    {
        // set content type and other response header fields first
        response.setContentType("text/html");

        // get the communication channel with the requesting client
        PrintWriter out = response.getWriter();

        // write the data
        out.println("<HTML>"
            + "<HEAD>"
            + " <TITLE>To All Students of Programming 3 at Doshisha University</TITLE>"
            + " <META HTTP-EQUIV='Content-Type' Content='text/html; charset=SHIFT_JIS'></META>"
            + "</HEAD>"
            + "<BODY BGCOLOR=\"#FFFFFF\">"
            + " <CENTER>"
            + "  <H2>Congratulations, your first servlet is working!</H2>"
            + "  <H3>[ local time is <font color=blue>"
            + new java.util.Date()
            + "</font> ]</H3>"
            + " </CENTER>"
            + "</BODY>"
            + "</HTML>");
    }
}
```

Q: サーブレットの構造？Java Classの開発とコンパイル方法？

A: 第３回の講義に説明します。

講義の内容の理解のチェック・リスト（まとめ）

- ✓ ダイナミックWebコンテンツとサーブレットの関係、
- ✓ サーブレットの特徴、
- ✓ Web Formの特徴、Web Formからサーブレットを呼び出す（実行）方法:
  - 何時（何のイベントの時に）サーバサイドスクリプトを実行するか？ **A: <input type="submit" …>**
  - 何のサーバサイドスクリプトを実行するか？ **A: <form action=" "…>**
  - 何のユーザー入力されたパラメターはスサーブレットに通信？ **A: < input id= " "…>**
- ✓ Web Formの  作成、  Upload、  アクセス など。

次の授業:

Lecture #3: サーブレットの構造、Java Classの開発とコンパイル方法など。

Lecture #4: 何のユーザー入力されたパラメターはサーブレットに通信？(A: < input id= " "…〉)

The End