

# Webアプリケーション開発演習A

## サーブレット

情報システム専攻科2年

○ **基本的なHTMLを学びました**  
・ **タグの基本的な使い方**

○ **JSPを学びました**  
・ **<% %>の部分はjavaのコードとして扱われ  
サーバー側で実行される。**

○ **実際にサーバーにファイルをアップロードした**  
・ **WinSCP、FileZilaをつかってサーバーにファイルを  
アップロードしまして、ブラウザから見ました**

A blue header area at the top left of the slide, featuring various white geometric shapes like circles, squares, and diamonds, some of which are nested or overlapping. The shapes are scattered across the blue background, which curves downwards towards the right.

**今日の目標！**

**サーフレットを理解する！**



# 目次

**サーフレットとは？**  
**サーフレットの例**



# 目次

## サーフレットとは？

### サーフレットの例

# サーフレットとは？

教科書P.60

サーフレット・・・**java**を用いて実装された  
サーバーサイドの**プログラム**

サーフ・レットとは？

どゆこと??




# サーフレットとは？

ざっくり言うと……

JSP……**HTML+Java**で記述されたもの  
サーフレット……**Java**だけで記述されたもの



A blue header area with a white curved line separating it from the main content. The blue area contains various white geometric shapes: circles, squares, and diamonds, some of which are nested or overlapping.

**サーフレットとは？**

**ん？サーフレットはなぜ必要なんだ？**

**JSPはHTML+Javaなんだから  
別にJSPで全て事足りるのでは？**

# サーフレットとは？


**理由を理解する為に  
まだ、教えていませんが  
例えば、DBに接続して値を取得する  
JSPがあったとします。  
ソースを見てみましょう！**

```

1 <? page language="java" contentType="text/html; charset=UTF-8"
2 pageEncoding="UTF-8"%>
3 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
4 <html>
5 <head>
6 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
7 <title>Insert title here</title>
8 </head>
9 <body>
10 <table>
11
12 Connection con = null;
13
14 InitialContext ctx;
15 try {
16     ctx = new InitialContext();
17
18     DataSource ds =
19         (DataSource)ctx.lookup(AppSettingProperty.getInstance().getDBString());
20
21     // MySQLに接続
22     con = ds.getConnection();
23
24     PreparedStatement ps = null;
25     ResultSet rs = null;
26     TaskTblEntity taskDetail = null;
27
28     try {
29         // ステートメント生成
30         StringBuilder sb = new StringBuilder(TASK_DETAIL_SQL);
31         if( RoleId.STUDENT.equals(roleId)){
32             sb.append(TASK_DETAIL_WHERE_SQL);
33         }
34
35         ps = con.prepareStatement(sb.toString());
36
37         ps.setInt(TASK_DETAIL_SQL_USER_IDX, studentId);
38         ps.setInt(TASK_DETAIL_SQL_COURSE_IDX, courseId);
39         ps.setInt(TASK_DETAIL_SQL_TASKID_IDX, taskId);
40
41         // SQLを実行
42         rs = ps.executeQuery();
43
44         //値を取り出す
45         while(rs.next()){
46
47             <tr>
48                 Integer testCaseId = fixInt(rs.getInt("TESTCASE_ID"),rs.isNull());
49                 if(testCaseId != null ){
50
51                     <td><%=rs.getInt("ALLMOST_OF_MARKS") %></td>
52                     <td><%=rs.getString("OUTPUT_FILE_NAME") %></td>
53                     <td><%=rs.getString("INPUT_FILE_NAME") %></td>
54                 }
55             </tr>
56         }
57     }
58 }
59 </table>
60
61 }
62
63 } catch (NamingException e) {
64     throw new DBConnectException(e);
65 } catch (SQLException e) {
66     throw new DBConnectException(e);
67 } catch (AsoLearningSystemErrException e) {
68     throw new DBConnectException(e);
69 }finally{
70
71     if( con != null ){
72         try {
73             con.close();
74             con = null;
75         } catch (SQLException e) {
76             logger.info("closeに失敗しました : ",e);
77         }
78     }
79 }
80 </table>
81 </body>
82 </html>

```





サーフレットとは？

**複雑な処理をJSPに書くと  
見にくくなる！**

**じゃあ、複雑な処理はどこに書くの？**

# サーフレットとは？

## サーフレットでしょ！



正確には、「サーフレット＝複雑な処理をかくところ」ではないのですが  
今のところは、**JSPには複雑な処理を書かないほうが良い**  
と覚えておきましょう。



# 目次

サーフレットとは？  
**サーフレットの例**

やってみよう!

# サーフレットの例

**では、早速サーフレットを使った  
プログラムを書いてみましょう！**

**<https://github.com/nishino-naoyuki/2019Web>  
・サーフレットサンプル.pdf**



# サーフシートの例

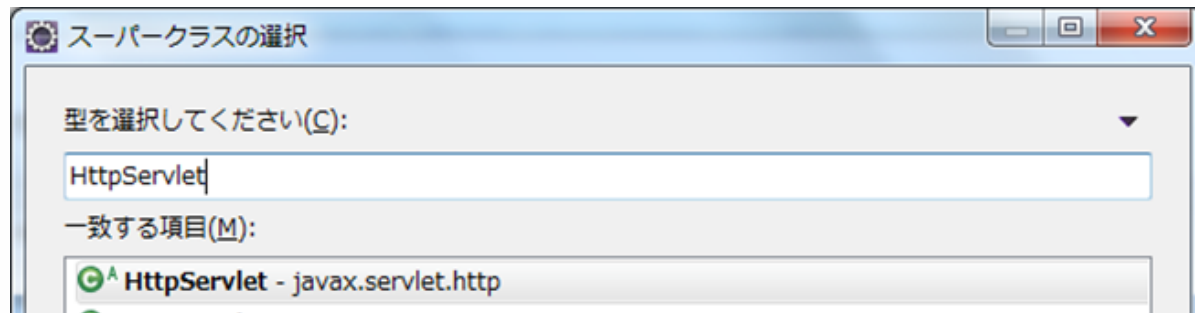
**新しいことが何個出てきました！**



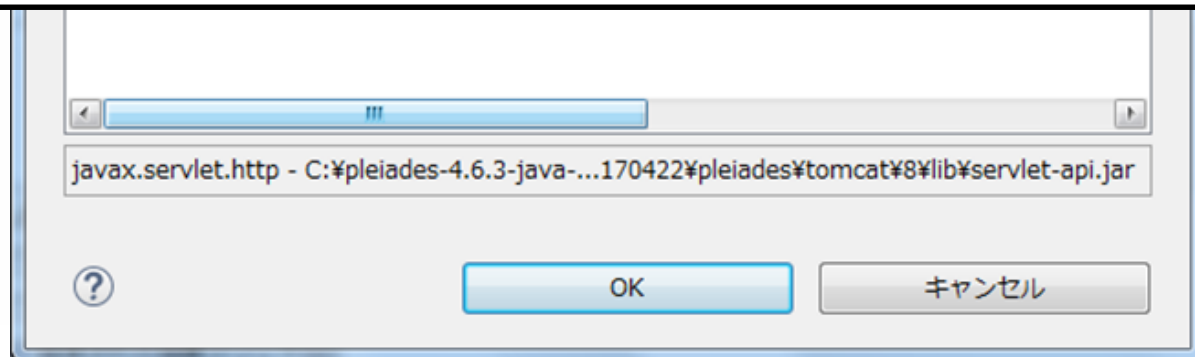


# サーブレットの例

## 1つ目....

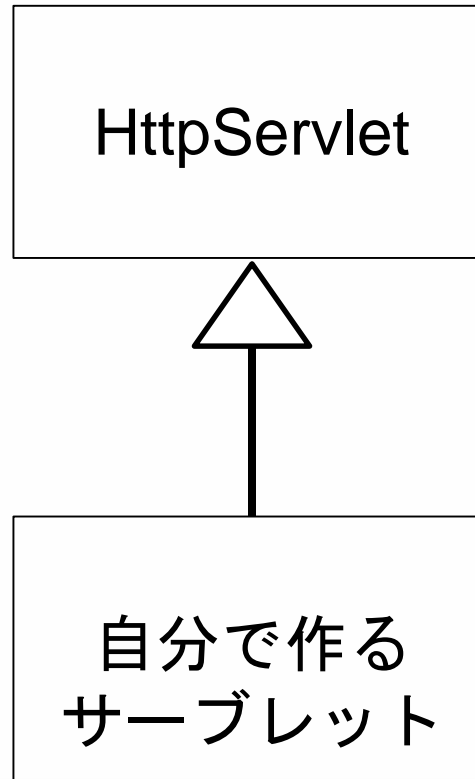


HelloServletを作成する時に、HttpServletクラス  
をスーパークラスにしました！



重要!

# サーブレットの例



**サーブレットは必ずHttpServletから  
派生させなければならない!**

# サーブレットの例

## 2つ目....

```
15 /**  
16  * @author nishino  
17  *  
18  */  
19 @WebServlet("/hello")  
20 public class HelloServlet extends HttpServlet {  
21  
22     @Override  
23     protected void doGet(HttpServletRequest request, HttpServletResponse response)  
24         throws ServletException, IOException {  
25         response.setContentType("text/html; charset=UTF-8");  
26         PrintWriter out = response.getWriter();  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36     }  
37  
38 }  
39
```

@WebServlet("/hello")  
ってなに??????

# サーブレットの例

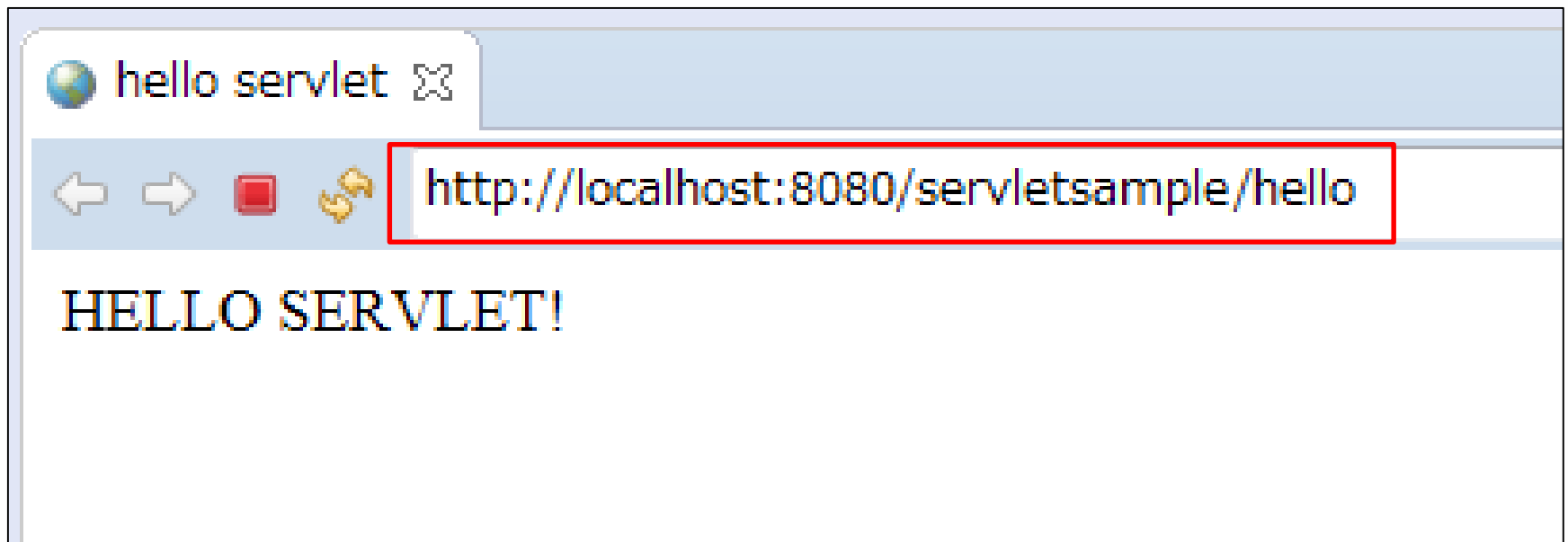
**@が付いているこれは  
アノテーション  
と言います。**

**アノテーションにはいろいろな種類がありますが  
@WebServletアノテーションは  
サーブレットとURLを結びつける  
アノテーションです**

**教科書P.89**

# サーブレットの例

## 実行した時のURLを再確認



**重要!**

# サーブレットの例

**http://localhost:8080/servletsample/hello**

**localhost:8080**・・・サーバー名(この場合「自分のPC」)

**servletsample**・・・Webアプリ名(プロジェクト名と一致)

**hello**・・・実行するサーブレット(@WebServletで指定)

**Webアプリ名の後に来る文字列が@WebServletの対象となる**

# サーブレットの例

ちなみに……

URLのhelloをhello2に変更すると……



# サーブレットの例

## 3つ目....

```
15 /**  
16  * @author nishino  
17  *  
18  */  
19 @WebServlet("/hello")  
20 public class HelloServlet extends HttpServlet {  
21  
22     @Override  
23     protected void doGet(HttpServletRequest request, HttpServletResponse response)  
24         throws ServletException, IOException {  
25         response.setContentType("text/html; charset=UTF-8");  
26         PrintWriter out = response.getWriter();  
27     }  
28  
29 }
```

doGetメソッドって何やねん！



# サーブレットの例

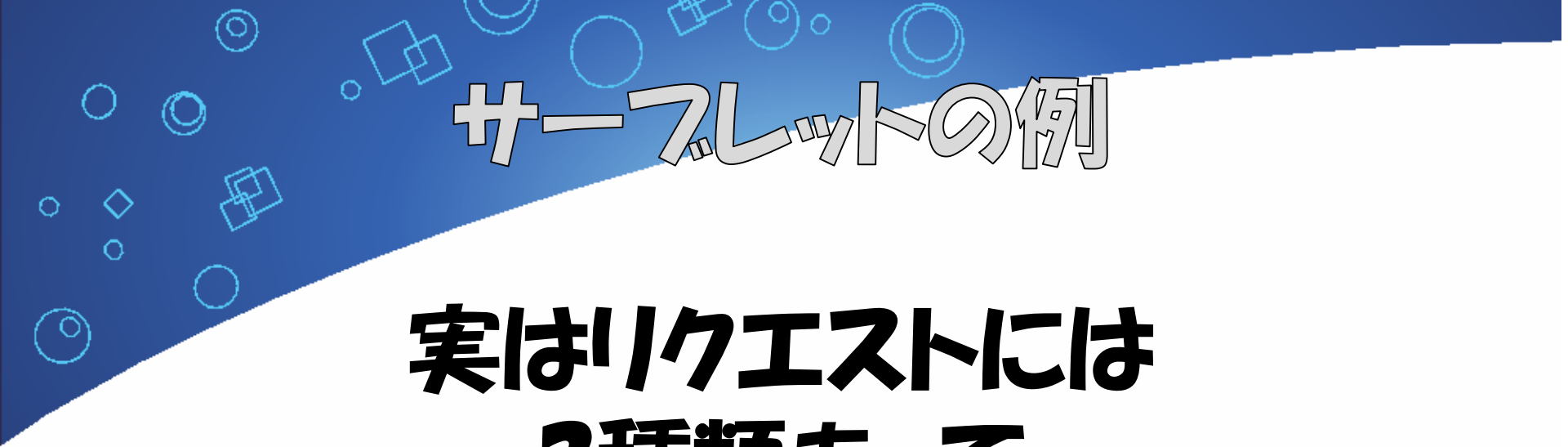
リクエスト

`http://localhost:8080/servletsample/hello`



自分のPCのTomcat

あ、あれ？HelloServletの  
どのメソッドを実行すつと??



# サーブレットの例

実はリクエストには  
2種類あって  
**GET通信**  
**POST通信**  
の2種類があります

# サーブレットの例

リクエスト

`http://localhost:8080/servletsample/hello`

GET通信



自分のPCのTomcat

あ、GET通信だから  
doGetメソッドを呼び出して  
OKかー

# サーブレットの例

**じゃあ、POST通信の時は??**

**→doGetと同じ引数を持つ  
doPostメソッドが存在し  
doPostが呼ばれる!**

# サーフレットの例

**じゃあ、どんなときにGET通信で  
どんなときにPOST通信なの？**



# サーブレットの例

## ざっくり言うと...

### ○POST通信

formタグのmethod属性でPOSTを指定した時

```
<form action="login2.jsp" method="POST">
```

### ○GET通信

それ以外(formでPOSTを指定する以外)のとき

- ・ブラウザのアドレスバーにURL直打ち
- ・Aタグで画面遷移するとき
- ・formタグのmethodでGETを指定した時

**重要!**

# サーブレットの例

## まとめると・・・

### ○POST通信

サーブレットのdoPostメソッドが呼ばれる

### ○GET通信

サーブレットのdoGetメソッドが呼ばれる

# サーブレットの例

ちなみに・・・

よく見るとここに聞いたことがある  
キーワードがありますねえ

```
21  
22 @Override  
23 protected void doGet(HttpServletRequest request, HttpServletResponse response)  
24     throws ServletException, IOException {
```

オーバーライド・・・親クラスのメソッドを子クラスで  
定義して、動きを変更すること

実はdoGetというメソッドはHttpServletが持っていて自  
作のサーブレットでオーバーライドすることで動作を変  
更しているのです！



やってみよう!

# サーブレットの例

以上のことを踏まえて  
教科書のサンプルを作ってみよう

教科書P.94 リスト3-4



**まとめ**

**今日の目標！**

# **サーブレットを理解する**

- **JSPは「HTML + Java」なのに対してサーブレットはJavaオンリー**
- **サーブレットの決まりごと**
  - **HttpServletクラスから派生する**
  - **URLとサーブレットの関連付けは@WebServletで行う**
  - **POST通信の時はdoPost、GET通信の時はdoGetが呼ばれる**