

Webアプリケーション開発演習A

MVCモデルを知る

情報システム専攻科2年

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！

MVCモデルを知ろう



目次

ここまでの復習 MVCモデルとは？

A decorative blue header 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 blue area curves downwards towards the right.

目次

ここまでの復習
MVCモデルとは？

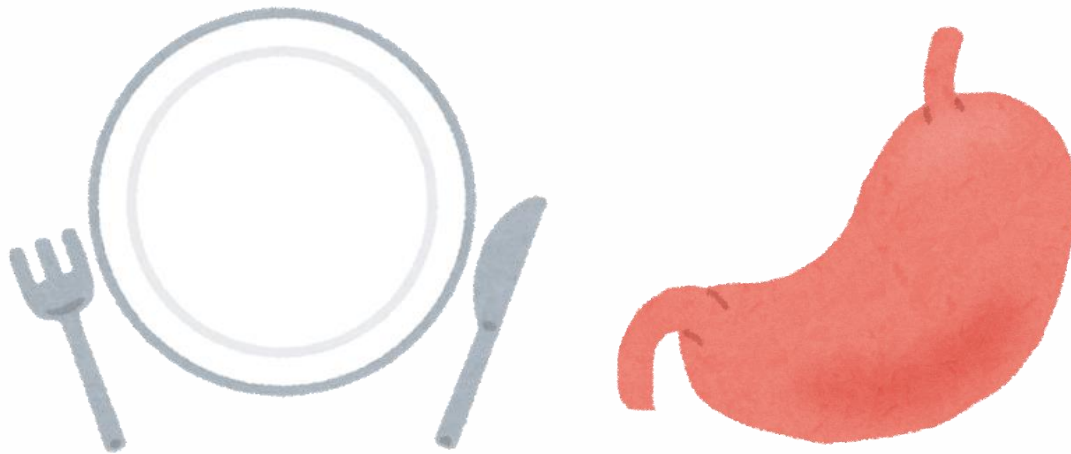
ここまでの復習

**ここまででWeb
ずいぶんアプリケーションっぽいものが
作れるようになっていました！**



ここまでの復習

ここまでをおさらいしてみましょう！



ここまでの復習

サーブレットの決まりごとが3つありました！

- **HttpServlet**から派生させる
- URLとサーブレットクラスの関連付けは
@WebServletアノテーションで行う
- サーブレットで実行されるメソッドは
POST通信時はdoPost
GET通信時はdoGet

ここまでの復習

**サーブレットからJSPへ処理を
転送するときは……**

```
RequestDispatcher dispatcher =  
    request.getRequestDispatcher("WEB-INF/jsp/hello.jsp");  
dispatcher.forward(request, response);
```


ここまでの復習

JSP→サーブレットの情報の受け渡しは...

JSP側(渡すほう)

```
<form action="q1page2" method="GET">↵  
  <input type="text" name="message">↵  
  <input type="submit" value="send">↵  
</form>↵
```

サーブレット側(受け取るほう)

```
//パラメータ取得↵  
String message = request.getParameter("message");↵
```

ここまでの復習

サーブレット → **JSP** の情報の受け渡しは...

サーブレット側 (渡すほう)

```
request.setAttribute("message", message);
```

JSP 側 (受け取るほう)

```
// リクエストからメッセージを取得する
```

```
String message = (String) request.getAttribute("message");
```

ここまでの復習

スコープについて

リクエストスコープ... **1回のリクエスト・レスポンスの間のみ**
存在する情報の置き場

セッションスコープ... **一度作られると、削除されるまで存在**
する情報の置き場

ここまでの復習

セッションに関する実装は3ステップです

STEP1.セッションを取得

STEP2.セッションに値を入れる

STEP3.セッションから値を取得する

ここまでの復習

ポイント

STEP1.セッションを取得

//セッションを取得する↓

```
HttpSession session = request.getSession(true);↓
```

セッションは request.getSessionで取得する！！

ここまでの復習

ポイント

STEP2.セッションに値を入れる

//セッションに値をセットする↓

```
session.setAttribute("loginName", "nishino");
```

getSessionで取得した
HttpSessionのインスタンス

箱の名前

入りたい値

セッションへは setAttributeでセットする！！

ポイント

ここまでの復習

STEP3.セッションから値を取得する

getSessionで取得した
HttpSessionのインスタンス？

箱の名前

```
String loginName = (String)session.getAttribute("loginName");
```

実はJSPでは、書かなくても
裏で勝手に getSessionしてくれています。
なので、getSessionを書かずにいきなり
sessionという名前でインスタンスを使えます。
これを「**暗黙オブジェクト**」といいます。

セッションからはgetAttributeで取得する！

ここまでの復習

リクエストとセッションの違いのまとめ

	設定・取得	範囲 (スコープ)	用途
セッション	設定: setAttribute 取得: getAttribute	1度セットすると 複数ページで参照可能	ログインした人の情報や買い物かごの中身など複数のページから参照したい情報を扱いたいとき
リクエスト	設定: setAttribute 取得: getAttribute	1回のリクエスト でのみ有効	JSP→サーブレット サーブレット→JSP など1ページの内で情報をやりとりするとき



目次

ここまでの復習
MVCモデルとは？

A decorative blue header with a white curved border. It contains several light blue geometric shapes: circles, squares, and diamonds, some of which are nested or overlapping.

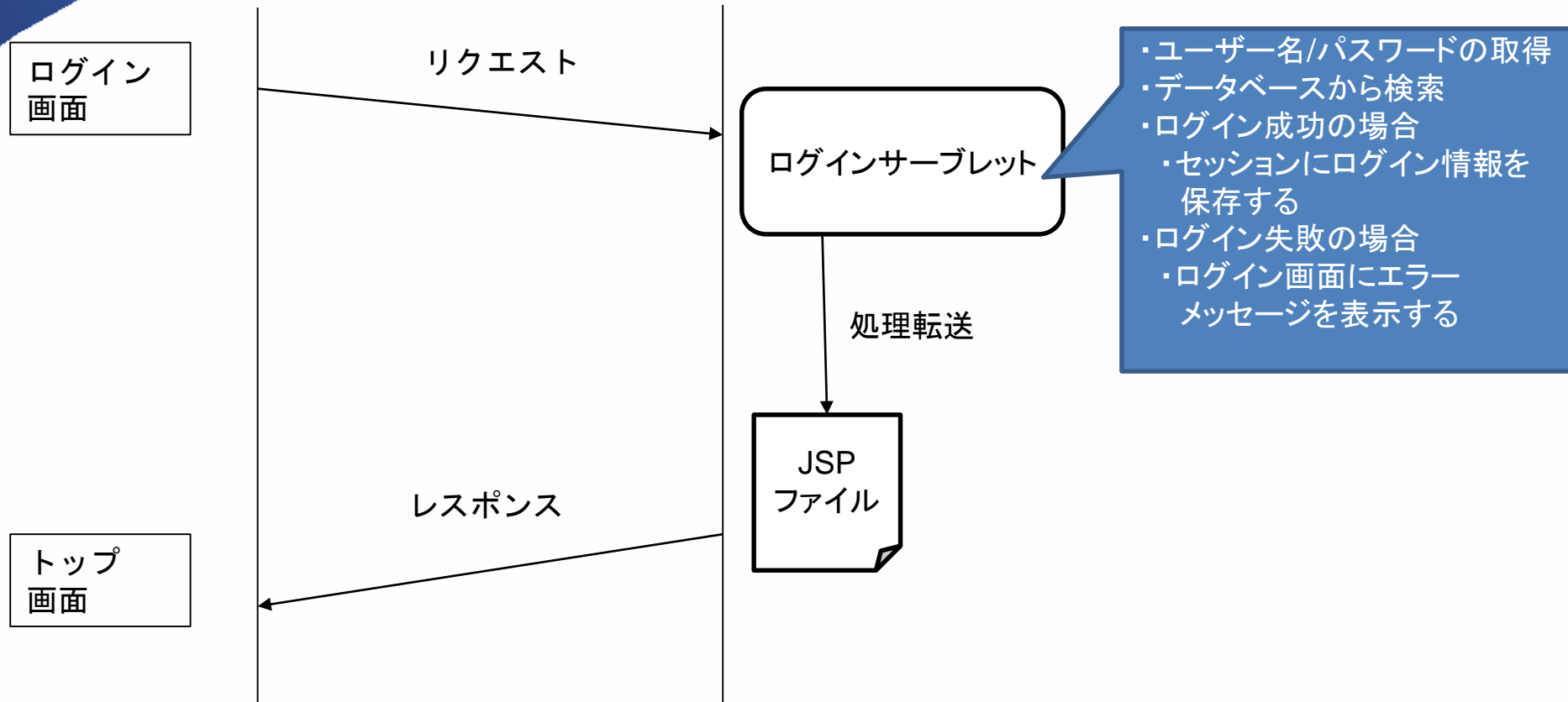
MVCモデルとは？

複雑な処理を考えてみよう！

例えば、ログインの処理を考える

MVCモデルとは？

ログイン処理



MVCモデルとは？

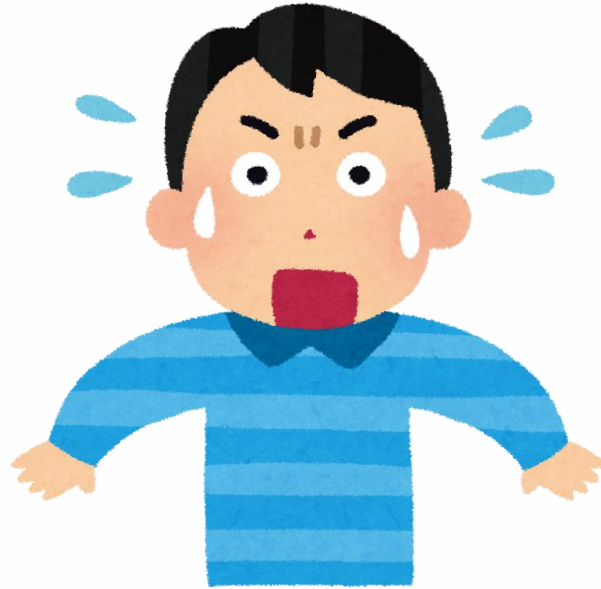
私は以前こういいました。

サーブレットは、**計算や複雑な処理**をする
JSPはサーブレットで**処理した結果**を表示する

サーブレットとJSPの**役割を明確に**わけ
ることで、ソースコードの**メンテナンス性**が
向上して、**効率の良いプログラム**を
作ることが出来る！

MVCモデルとは？

ちょっと待って！



**サーブレットは複雑な処理をやる役割だけど
それじゃあ、サーブレットのソースコードが
肥大しないか？？？**

MVCモデルとは？

ログインサーブレットでやる処理

ログインサーブレット

- ・ユーザー名/パスワードの取得
- ・データベースから検索
- ・ログイン成功の場合
 - ・セッションにログイン情報を保存する
- ・ログイン失敗の場合
 - ・ログイン画面にエラーメッセージを表示する

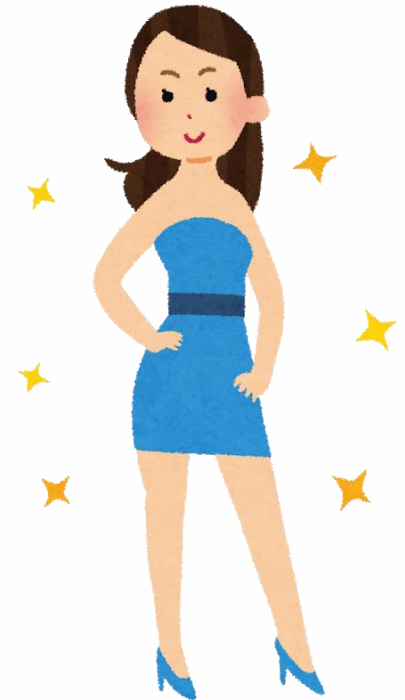
これだけの処理を書くとコードが長くなる！

MVCモデルとは？

ここで登場するのが...

MVCモデル

という**考え方**です



MVCモデルとは？

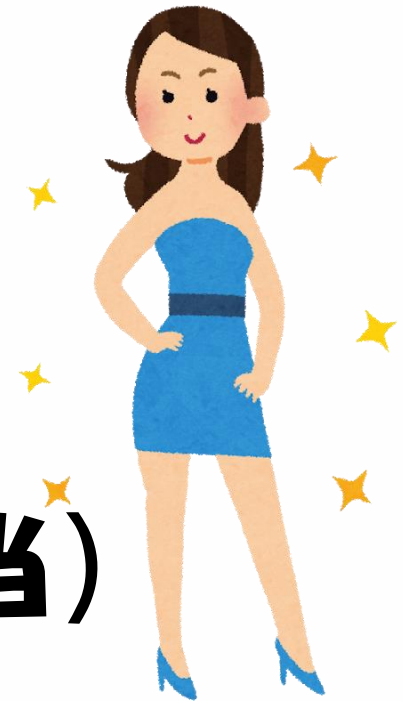
MVCモデル

教科書P.164,165

M・・・Model(処理担当)

V・・・View(表示担当)

C・・・Controller(割振り担当)



やってみよう!

演習

MVCモデルのプログラムを 作ってみよう

<https://github.com/nishino-naoyuki/2019Web-MVCサンプル1.pdf>

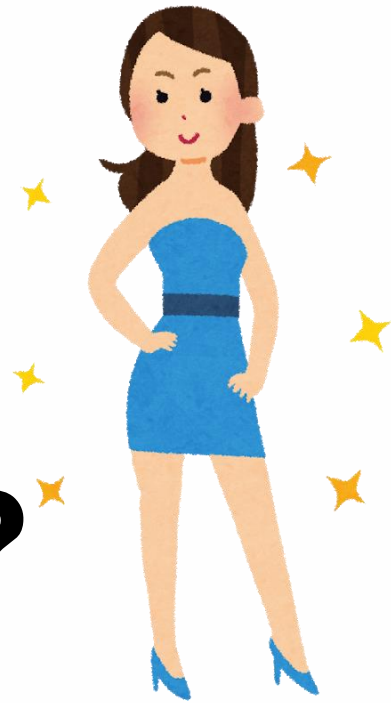


MVCモデルとは？

サンプルの解説

どのように
MVCモデル
が組みまれていたか？

を解説します



MVCモデルとは？

今回のサンプルでは

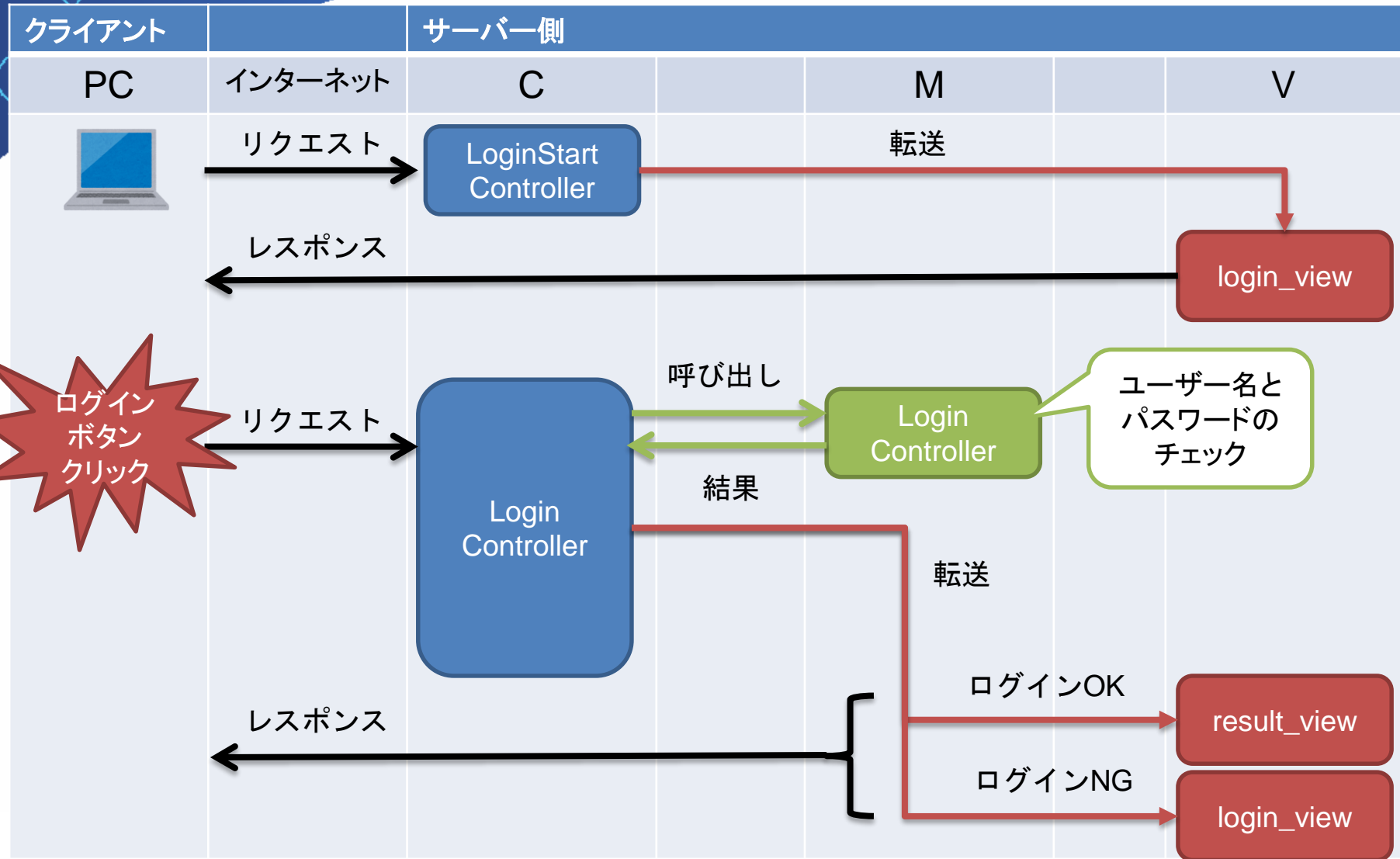
M・・・LoginModelクラス(処理をする)

V・・・login_view.jsp、result_view.jsp

**C・・・LoginStartController、LoginController
(仕事の割振り)**

MVCモデルとは？

サンプルの動き



MVCモデルとは？

M・・・ Javaクラス(主たる処理をする)

V・・・JSP(結果を表示する)

C・・・サーブレット(処理を振り分ける)

まとめ

今日の目標！

MVCモデルを知ろう

- **MVCモデル**とは、メンテナンス性が高いソースコードを組む為の**考え方**！
M(モデル)、V(ビュー)、C(コントローラ)の3つの役割に沿ってソースコードを組んでいく。

M(モデル)・・・Javaクラス(主たる処理を書く)

V(ビュー)・・・JSP(結果の表示(画面))

C(コントローラ)・・・サーブレット(処理の振り分け)

- 通常はリクエストを**C**が受け、**M**に処理を依頼し結果を**V**で表示する。