

Webアプリケーション開発演習A

セッション

情報システム専攻科2年

チェック ✓

今日の目標！

Webアプリケーションを
サーバーにアップロードする！

- Webアプリケーションをサーバーにアップする
アップする方法はJSPのときと同じ！
warファイルをつかって、**webapps**フォルダに
アップロード！
- サーブレットのときは**Tomcatの再起動**が必要



今日の目標！

**リクエストスコープ
セッションスコープを理解する**



目次

リクエストスコープ
セッションスコープ



目次

リクエストスコープ
セッションスコープ

リクエストスコーフ

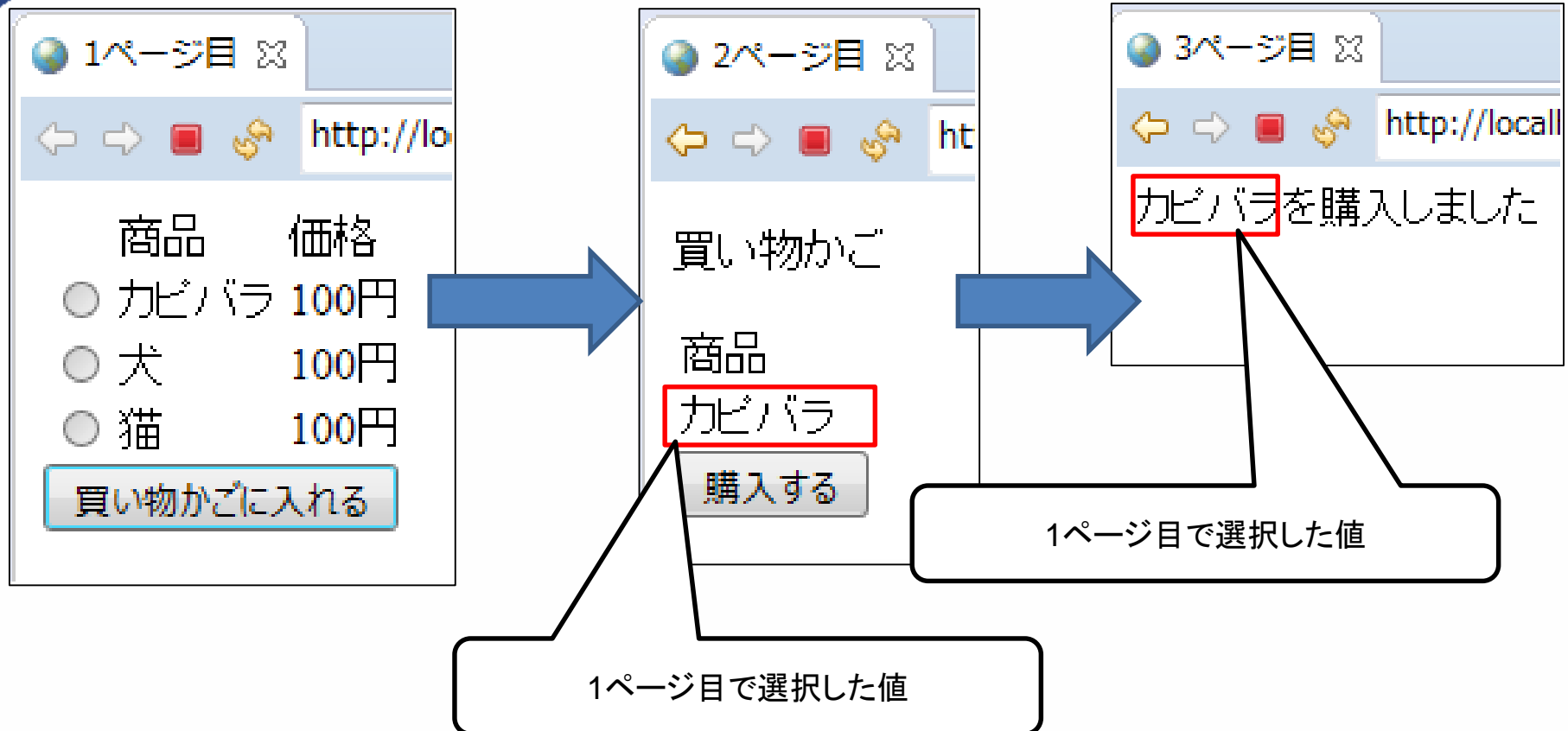
**今日は、ものすごく
重要なことを学びます！**

しっかり聞いておいてください！



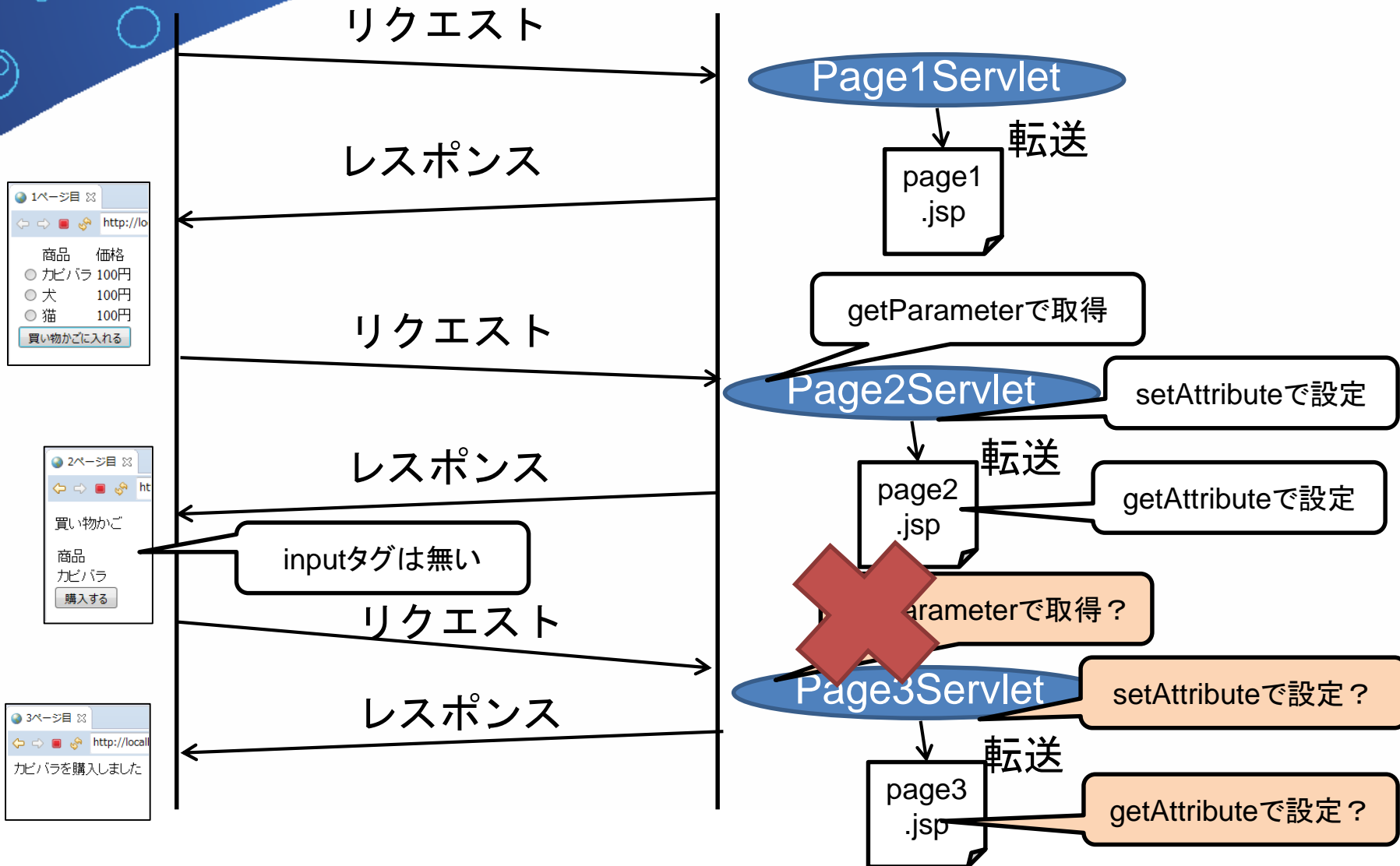
リクエストスコープ

次のような画面遷移を考えましょう



リクエストスコープ

PC



リクエストスコーフ

実は

**request.setAttributeや
formタグで送信されて
箱に入れた値は...**

**レスポンスを返すと
破棄されてしまうんです！**



A decorative blue header at the top of the slide, featuring various white geometric shapes like circles, squares, and diamonds, some of which are nested or overlapping. The header has a wavy bottom edge.

リクエストスコーフ

どういふことか説明します・・・

リクエストスコーフ

そもそも
`request.setAttribute`や
`formタグ`で送信
をすると箱に値が入ると言いましたが
箱は「**リクエストスコーフ**」という
場所に置かれます

リクエストスコープ

リクエスト

http://localhost:8080/servletsample/page2

リクエスト

http://localhost:8080/servletsample/page1

西野



Page2S

転送

西野

西野

リクエストスコープ

1ページ目で入所された文字は「西野」です

JSPが値を取り出す

サーブレットが値を
JSPへ値を送る

page2.js

リクエストスコーフ

さきほどの買い物かごの
例を図で表します

リクエストスコーフがいつ作られて
いつ破棄されるかに注目して
見ていてください。

リクエスト

http://localhost:8080/servletsample/**page3**

リクエスト

http://localhost:8080/servletsample/**page1**

カピ

リクエスト

レスポンス

Page3Servlet

転送

page3.jsp

カピ

カピ

リクエストスコープ

「カピ」が**リクエストスコープ**に
ないから取れない！



A decorative blue header at the top of the page, featuring various white geometric shapes such as circles, squares, and diamonds, some of which are nested or overlapping. The shapes are scattered across the blue background, which curves downwards from the top left towards the center.

リクエストスコーフ

教科書P.196

教科書P.209



目次

リクエストスコープ
セッションスコープ

セッションスコープ

前の章で、**リクエストスコープ**
を学びました。

同時に、リクエストスコープでは
出来ないことがあることも知りました。

セッションスコープ

リクエストスコープでは出来ないこと

AZCAFE

西野直幸(管理者) ▼

AZCAFE

西野直幸(管理者) ▼

ログインした人の名前はどの画面でも表示されている！

AZCAFE

西野直幸(管理者) ▼

課題一覧 あなたへの課題の一覧です

セッションスコープ

つまり

**画面をまたいでのデータの取得
は、リクエストスコープでは出来ない！**

画面をまたぐ = リクエストをまたぐ

重要!

セッションスコープ

もう少し噛み砕いて言うと...

色々な画面で**共通して使いたい値**は
リクエストスコープでは実現不可能！！



セッションスコーフ

では、ログインした人の名前のように
色々な画面で使いたい値がある場合は
どうすれば良いか？

ここで登場するのが

セッションスコーフ！


やってみよう!

演習

**リクエストスコープとの違いを
意識しながらサンプルを作ってみよう!**

**<https://github.com/nishino-naoyuki/2018Web>
・セッションサンプル1.pdf**





セッションスコープ

実装方法を振り返ってみよう

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

ポイント

セッションスコープ

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！！

ポイント

セッションスコープ

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

入れたい値

セッションへは setAttributeでセットする！！

ポイント

セッションスコープ

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！

セッションスコープ

リクエスト

<http://localhost:8080/servletsample/page3>

リクエスト

<http://localhost:8080/servletsample/page1>



リクエスト

レスポンス



Page3Servlet

転送

page3.jsp

西

西野

セッションスコープ

やってみよう!

演習

もうひとつサンプルを作ってみよう

<https://github.com/nishino-naoyuki/2018Web>
・セッションサンプル2.pdf



まとめ

今日の目標！

リクエストスコープ セッションスコープを理解する

- 様々な画面で**共通で使いたい値**は
リクエストスコープでは実現不可能
→セッションスコープでなら実現可能

• セッションを使用するのは3STEP

1. セッションの取得 `HttpSession session = request.getSession(true);`
2. セッションへのセット `session.setAttribute("loginName", "nishino");`
3. セッションからの取得 `(String)session.getAttribute("loginName");`