

**JSP**

# Overview

- ◆ Introduction to Jsp
- ◆ JSP Architecture
- ◆ Creating a Simple Jsp
- ◆ JSP Scripting Elements
- ◆ JSP Directives
- ◆ JSP Implicit Objects
- ◆ JSP Action Tags
- ◆ JSP Expression Language
- ◆<sub>2</sub> JSTL

# Introduction to JSP

- ◆ JSP is another technology to create web application
- ◆ A JSP page contains HTML code and JSP tags
- ◆ The jsp pages are easier to maintain than servlet because we can separate designing and development
- ◆ It provides some additional features such as Expression Language, Custom Tags

# Problem in Servlet

- ◆ The development of servlet is a time consuming process .
- ◆ Servlet code needs to be updated and recompiled if we have to change the look and feel of the application

# Introduction to JSP

◆ How JSP overcomes the problem of servlet ?

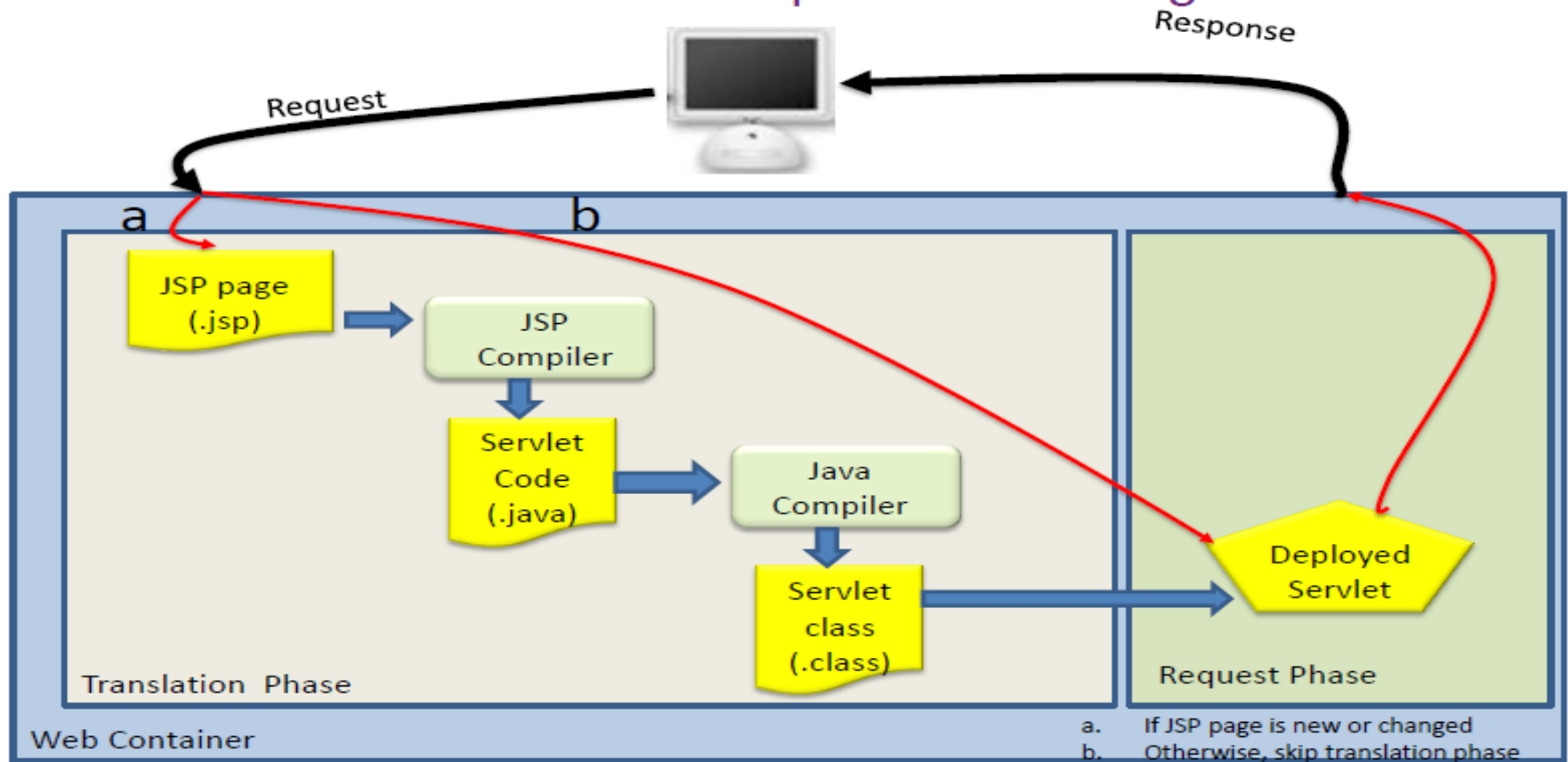
- JSP provides many tags and implicit objects so it saves our time.
- JSP is easy to maintain
- The JSP is internally converted into servlet
- In JSP pages, custom tags and JSP Expression Language provide code reusability

# Life Cycle of a JSP Page

- ◆ JSP pages follows these phases:
  - Translation of JSP Page
  - Compilation of JSP Page
  - Classloading (class file is loaded by the classloader)
  - Instantiation (Object of the Generated Servlet is created)
  - Initialization ( `jspInit()` method is invoked by the container)
  - Request processing ( `_jspService()` method is invoked by the container)
  - Destroy ( `jspDestroy()` method is invoked by the container)

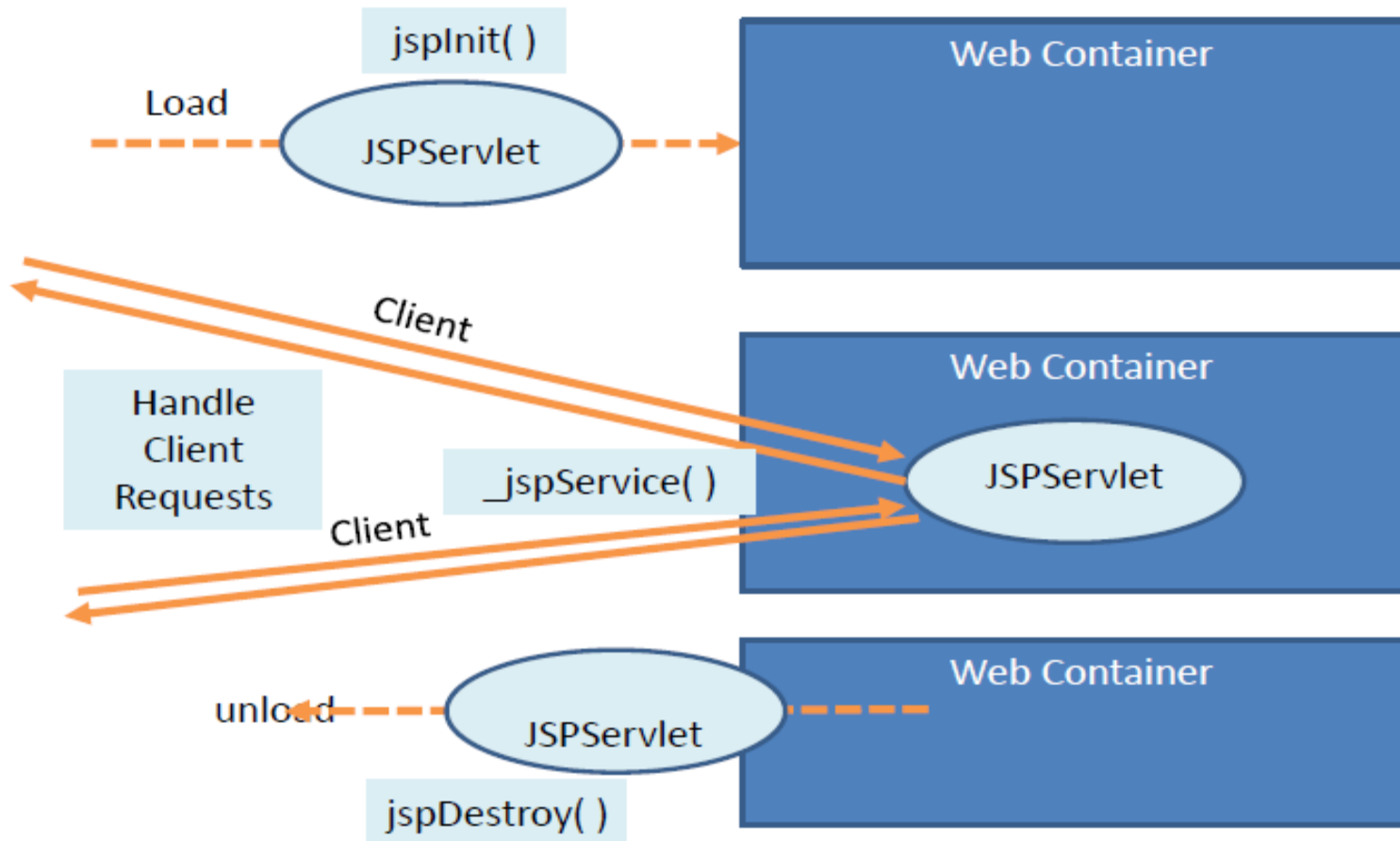
# Life Cycle of JSP Page

## JSP Translation Phase / Request Processing Phase



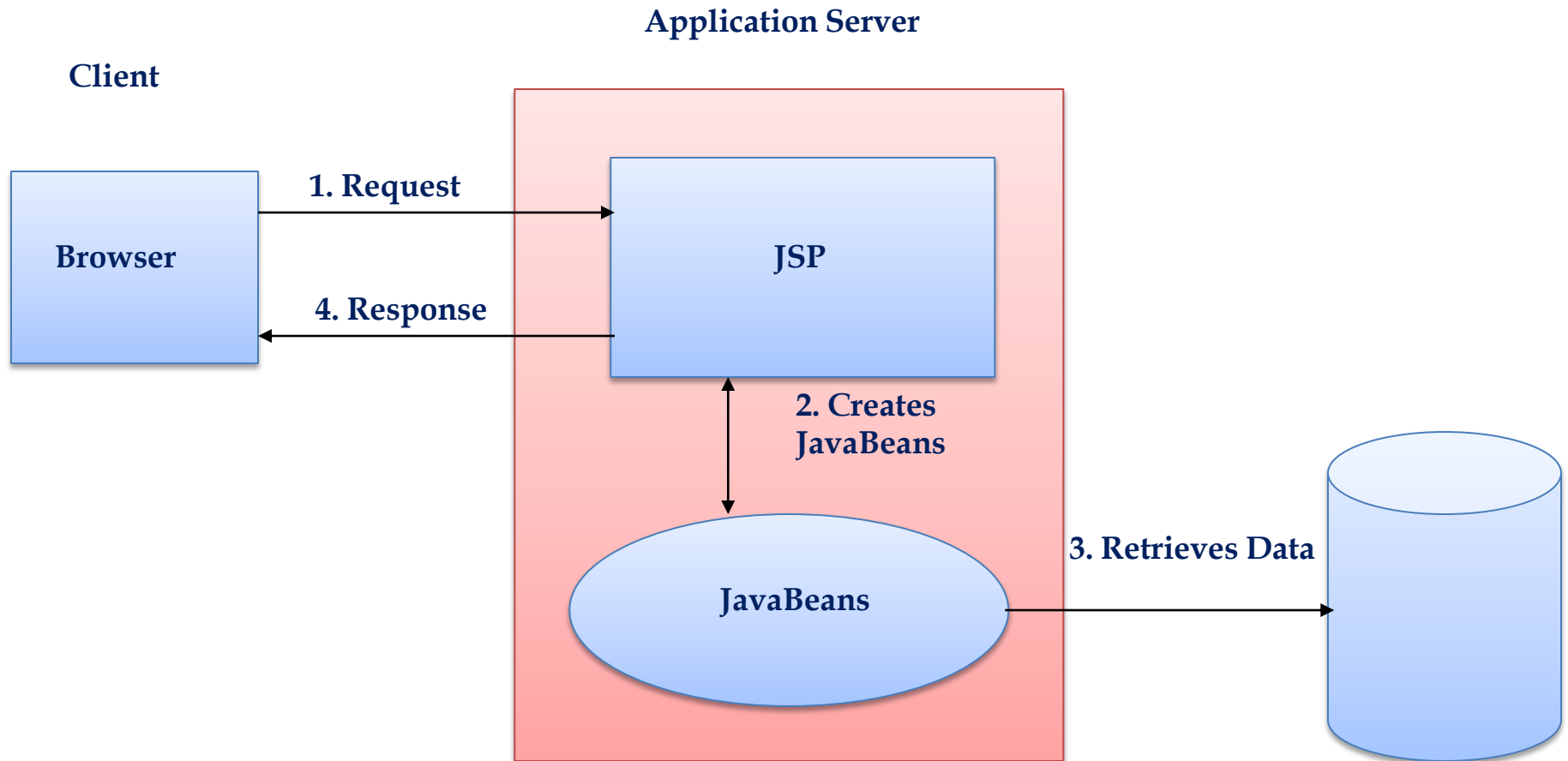
# JSP Life Cycle

## JSP life cycle methods

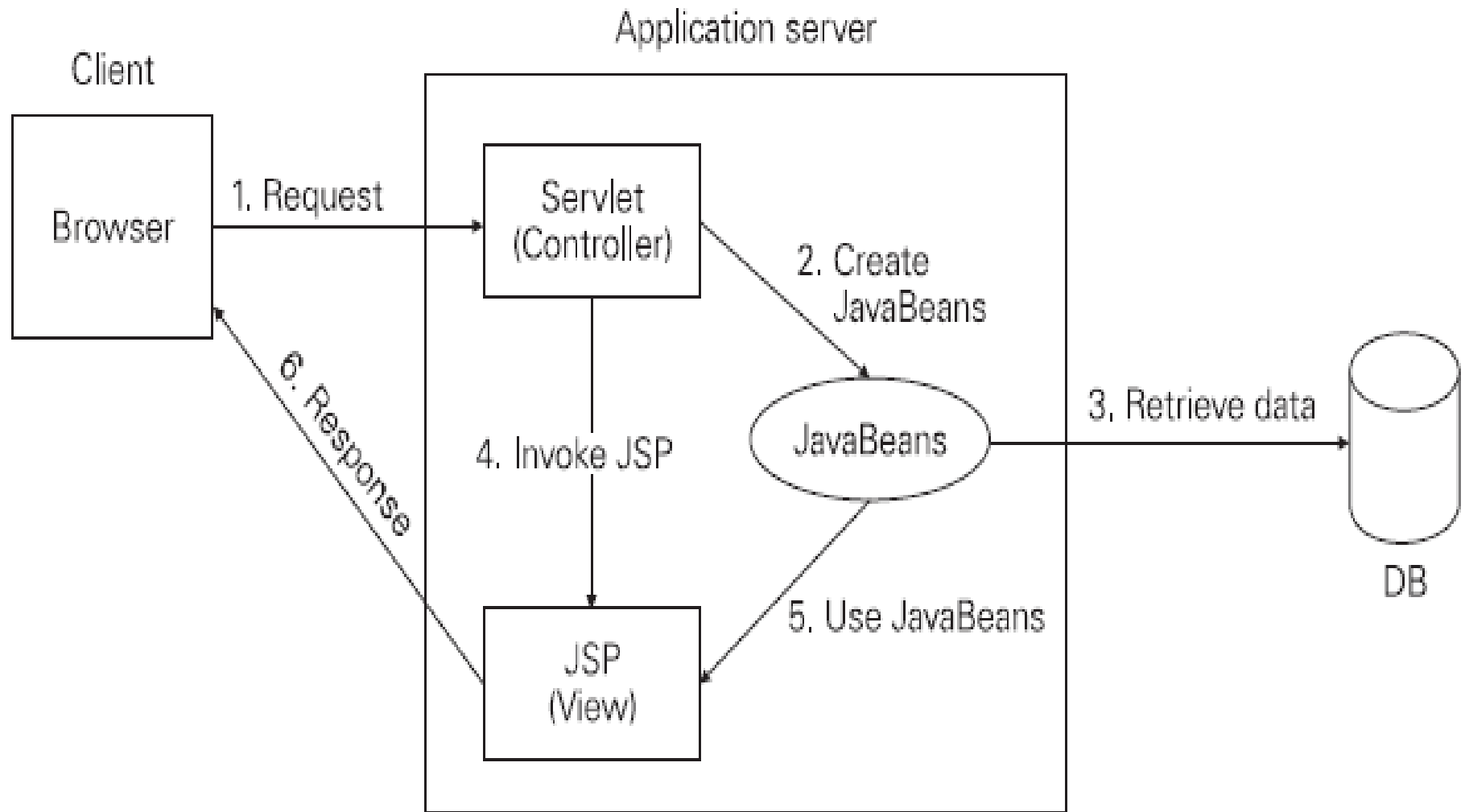




# Model 1 Architecture



# Model 2 Architecture



1. A JSP page contains ----- code and ----- tags.  
A. HTML and JSP
2. Init method is invoked by class loader True / False  
A. False ( Container )
3. In JSP pages, ----- and ----- provide code reusability and separation of concerns.  
A. Custom tags and JSP Expression Language

# Simple Jsp Creation

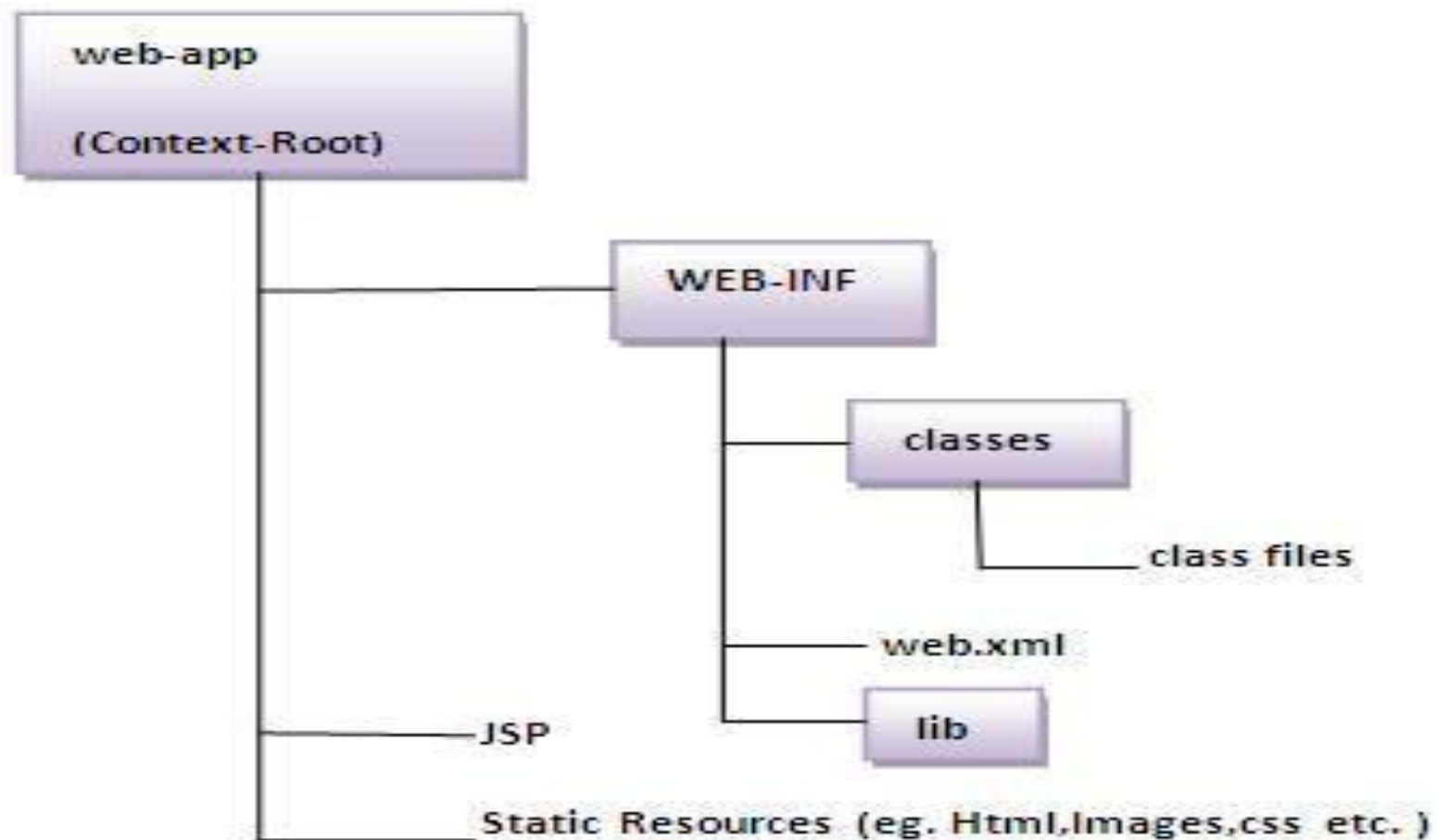
- ◆ Creating a simple JSP Page
  - We have to save the below file as index.jsp
  - Put it in a folder and paste the folder in the web-apps directory in apache tomcat to run the jsp page
- ◆ Sample code: index.jsp

```
<html>  
<body>  
JSP Page  
</body>  
</html>
```

## ◆ Directory structure of JSP

The directory structure of JSP page is same as servlet. We contains the jsp page outside the WEB-INF folder or in any directory.

# Simple Jsp Creation



# JSP Scripting Elements

## ◆ Scripting Elements in JSP

- Scriptlet tag
- Expression tag
- Declaration tag

## ◆ Scriptlet tag

- In Jsp java code can be written inside the jsp page using the scriptlet tag

## Syntax

`<% javasourcecode %>`

```
<html>
<body>
<%
  out.println(" Scriptlets in Jsp");
%>
</body>
</html>
```

# JSP Scripting Elements

## ◆ Expression Tag

- The code placed within expression tag is written to the output stream. It is mainly used to print the values of variable or method

### Syntax

`<%= statement %>`

```
<html>
<body>
<% ="Welcome to Jsp" %>
</body>
</html>
```

# JSP Scripting Elements

## ◆ Declaration Tag

- The Jsp Declaration tag is used to declare fields and methods.
- The code written inside the jsp declaration tag is placed outside the service() of the auto generated

### Syntax

`<% ! Statement %>`

```
<html>
<body>
<%! int data=100 %>
<% ="Value is :"+data %>
</body>
</html>
```



# Jsp Directives

- ◆ Directives are messages that tells the web container how to translate a JSP Page into corresponding servlet
- ◆ There are 3 types of directives
  - Page Directive
  - Include Directive
  - Taglib Directive

# Page Directive

- ◆ The page directive defines attributes that apply to an entire jsp page

Syntax `<%@ page attribute="value" %>`

## Attributes of JSP Page Directive

Import	Extends	contentType	autoFlush
Info	Buffer	Language	isELIgnored
isThreadSafe	autoFlush	Session	pageEncoding
errorPage	isErrorPage		

# Include Directive

- ◆ Include Directive is used to include the contents of any resource , file.html file or text file.
- ◆ The include directive includes the original content of the included resource at page translation time(the jsp page is translated only)
- ◆ Advantage : Code Reusability

Syntax : `<%@ include file="resourceName" %>`

# Taglib Directive

- ◆ The Jsp taglib directive is used to define a tag library that defines many tags.
- ◆ We use the TLD (Tag Library Descriptor) file to define the tags.

Syntax<%@ taglib uri="" preix="" %>

# Jsp Implicit Objects

- ◆ There are 9 implicit objects available for a JSP page
- ◆ The auto generated servlet contains many objects like out,request,response,config etc.

Object	Type
out	JspWriter
request	HttpServletRequest
response	HttpServletResponse
config	ServletConfig
application	ServletContext
session	HttpSession
pageContext	PageContext
page	Object
exception	Throwable

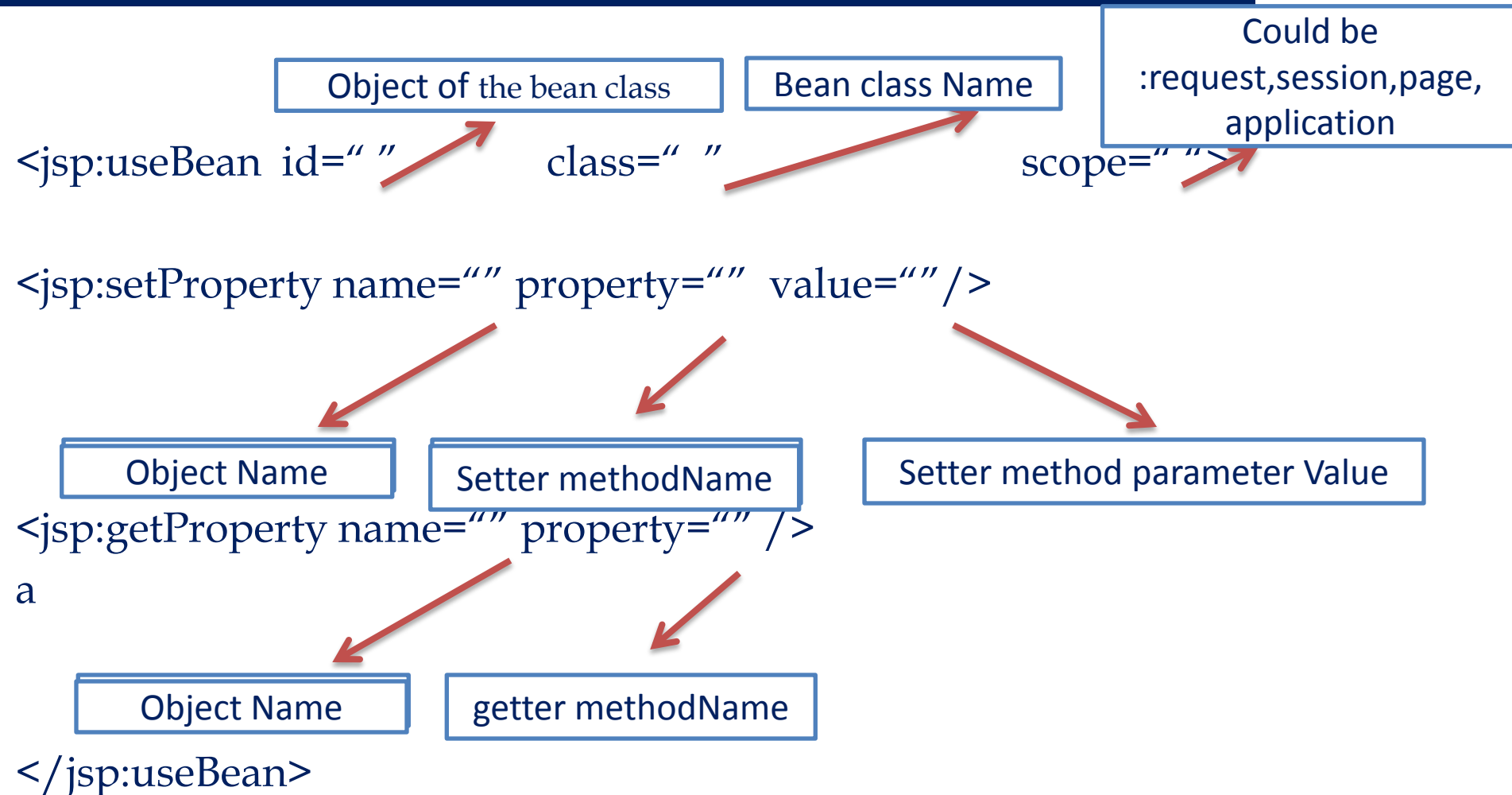
# JSP Action Tags

- ◆ Jsp Action Action Tags are used to perform specific task
- ◆ The action tags are basically used to control the flow between pages and to use Java Bean

Jsp Action Tags are as follows:

- `jsp:useBean`
- `jsp:setProperty`
- `jsp:getProperty`
- `jsp:include`
- `jsp:forward`
- `jsp:plugin`
- `jsp:fallback`

# jsp:UseBean Tag



# JSP Action Tags

- ◆ The Jsp forward action tag is used to forward the request to another resource it may be a jsp , html or another resource.

- ◆ Syntax for <jsp:forward> tag without parameter:

```
<jsp:forward page="relativeUrl" />
```

- ◆ Syntax for <jsp:forward> tag with parameter:

```
<jsp:forward page="relativeUrl">
```

```
<jsp:param name="parameterName" value="parameterValue" />
```

```
</jsp:forward>
```



# JSP Action Tags

- ◆ The Jsp include action tag is used to include the content of another resource
- ◆ It may be a jsp , html or servlet.
- ◆ The jsp include action tag includes the resource at request time
  - **Syntax for <jsp:include> tag without parameter:**  
`<jsp:include page="relativeUrl" />`
  - **Syntax for <jsp:include> tag with parameter:**  
`<jsp:include page="relativeUrl">`  
`<jsp:param name="parameterName" value="parameterValue" />`  
`</jsp:include>`

# Expression Language (EL)

- ◆ The expression language(EL) simplifies the accessibility of data stored in the Java Bean component and other objects like request , session,application etc
- ◆ There are many implicit objects and operators in EL ,
  - Syntax for Expression Language  
\${ name }

# Quiz

1. JSP action tags
  - a.jsp:useBean
  - b. jsp:setProperty
  - c. jsp:getProperty
  - d. All the Above

A. d
2. Identify implicit objects in jsp
  - a. Out , Session ,Request ,Response
  - b. Taglib
  - c. Forward
  - d. All of the above A

A. a