

Web Development with Jakarta Server Pages and Servlets

Session: 13

Jakarta Server Pages Standard Tag Library and Custom Tags



Objectives

- ❖ Explain the concept and necessity for JSTL
- ❖ List the advantages of using JSTL
- ❖ Describe different tag libraries available in JSTL
- ❖ Explain `jakarta.servlet.jsp.tagext`
- ❖ Explain how to configure the JSTL library in NetBeans Eclipse
- ❖ Identify the purpose of general-purpose tags
- ❖ Outline decision-making in tags
- ❖ Describe iteration tags in the core tag library
- ❖ Identify different tags available in the SQL tag library
- ❖ Explain custom tags in JSP
- ❖ Explain the use of Tag Extension API
- ❖ Explain Simple Tags API

Introduction

Advantages of JSTL

Helps in development of the JSP application.

Saves development time.

Helps HTML generation tools to parse the JSTL code.

Supports both formatting and multilingual support.

Enables the programmer to develop custom tags.

taglib Map in web.xml

Implicit Map Entries from TLDs

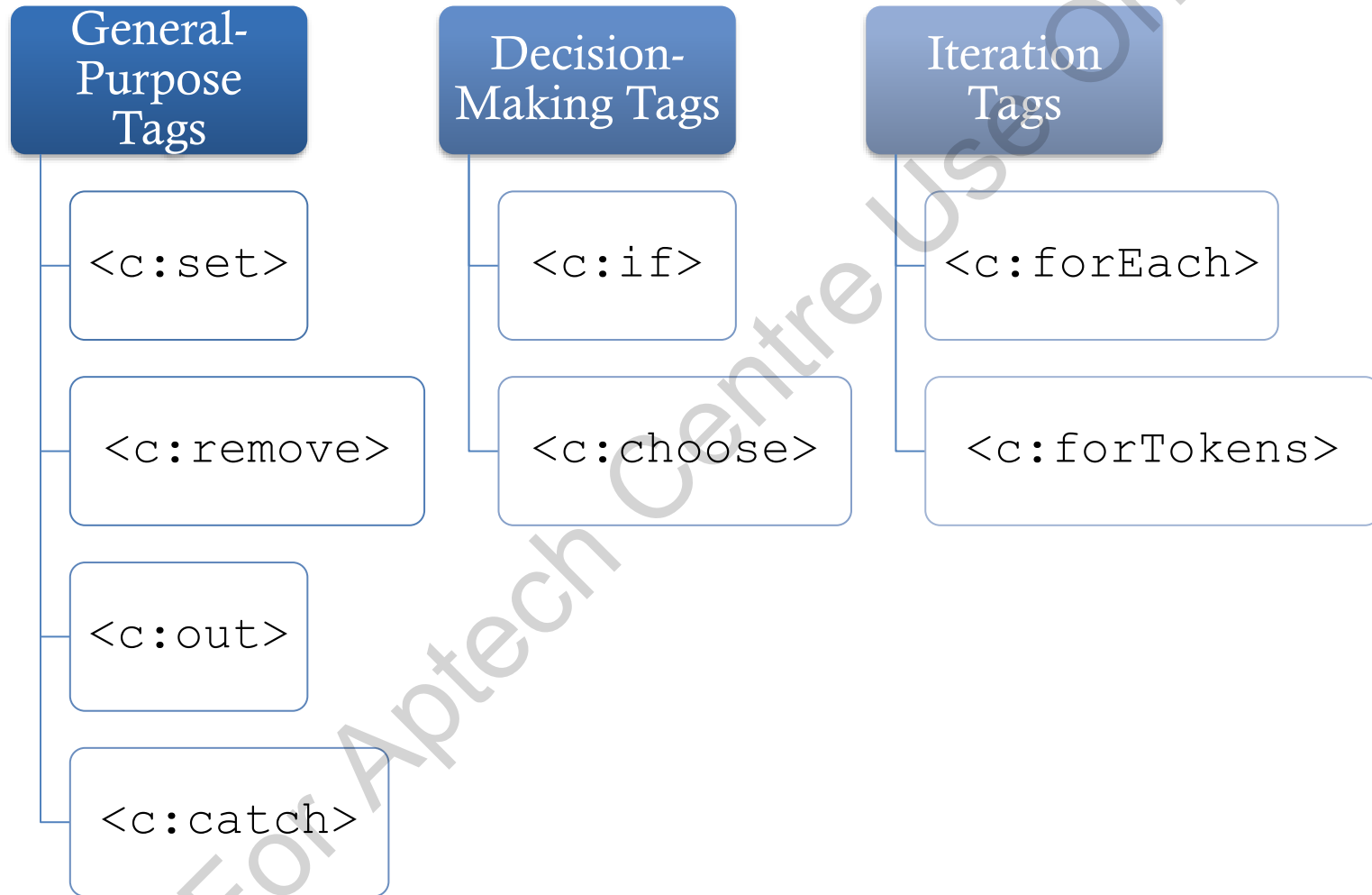
TLDs in JAR files in WEB-INF/lib

TLDs under WEB-INF

Implicit Map Entries from the Container

Map construction sequence of taglib

Core Tag Library



SQL Tag Library

Passing
database queries

- Provides users with the SQL tag library database.

Accessing query
results

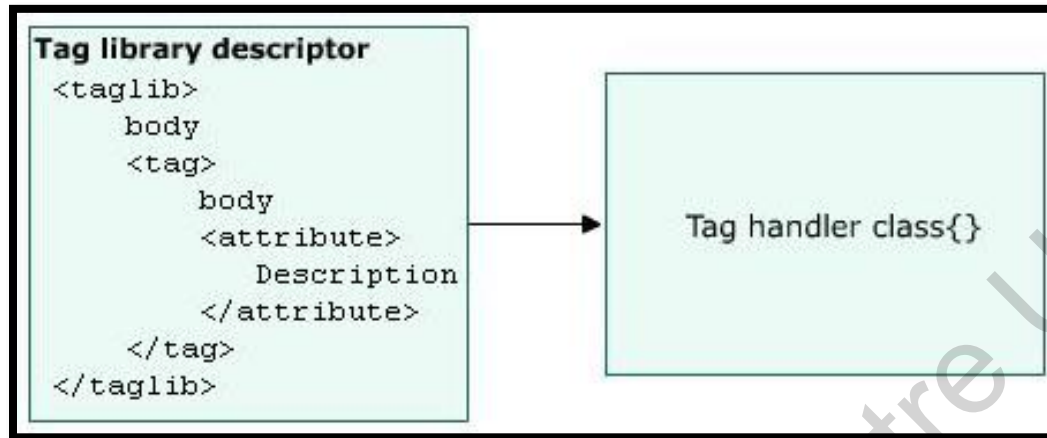
- Allows users to access results for queries.

Database
Modifications

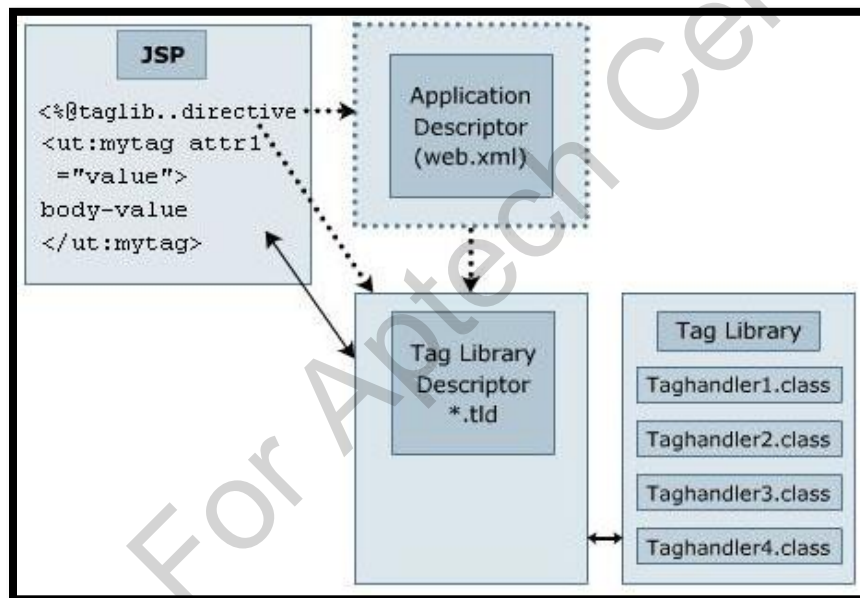
- Helps in modifying database.

Functionalities of JSTL SQL Tag

JSP Custom Tags



Custom Tags
Terminology



Working of Custom
Tag Libraries

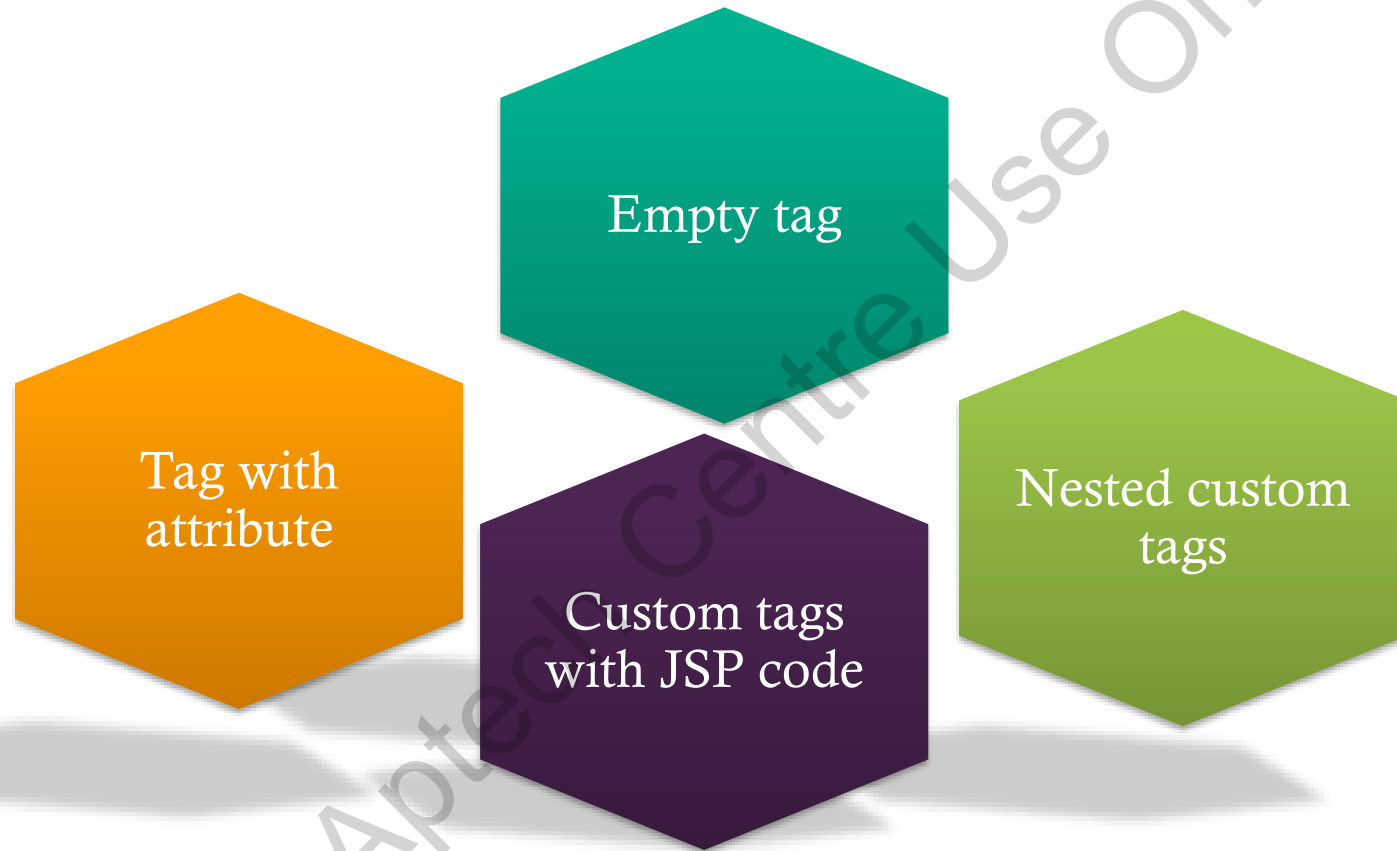
Custom Tag Library

Analyze the JSP file and compile them to servlet code.

Allow a Java object to be pooled and reused any time from memory.

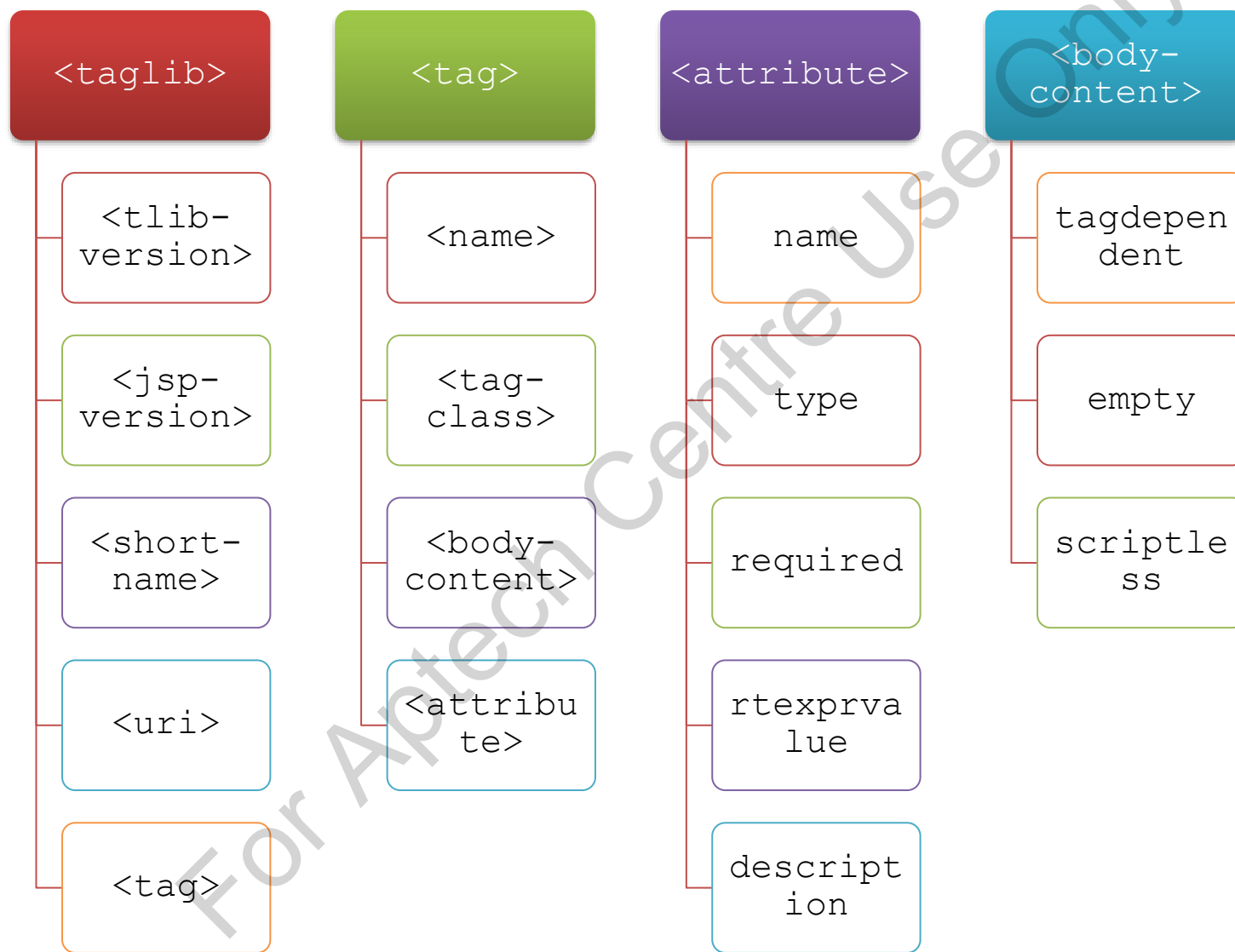
Functionality of Jaspers 2

Using Custom Tags in JSP Pages



Types of Custom Tags

Classic Custom Tags



Tag Extension API

Tag

IterationTag

BodyTag

Interfaces

TagSupport

BodyTagSupport

Classes

JspTagException

JspException

Exceptions

Implementing Tag Interface

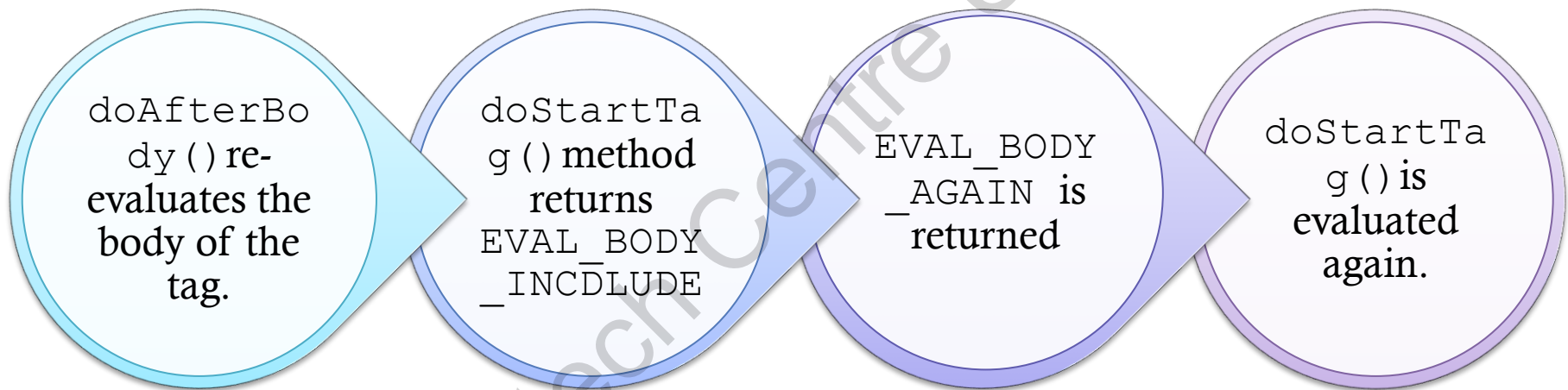
`doStartTag()`

`setPageContext()`

`doEndTag()`

`release()`

Implementing IterationTag Interface



BodyTag Interface

**setBodyContent
()**

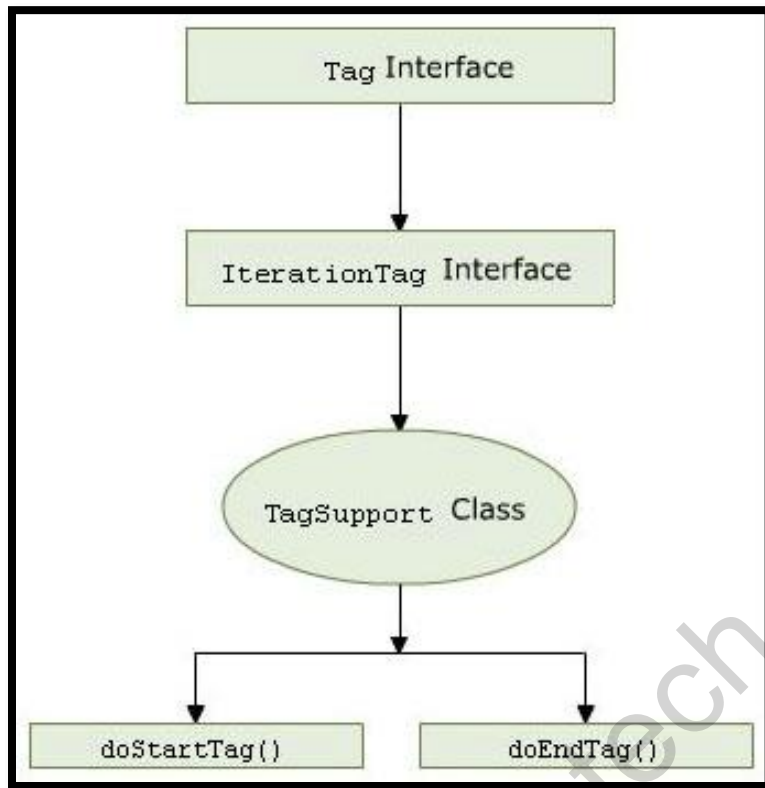
- `public void setBodyContent (BodyContent b) {}`

doInitBody ()

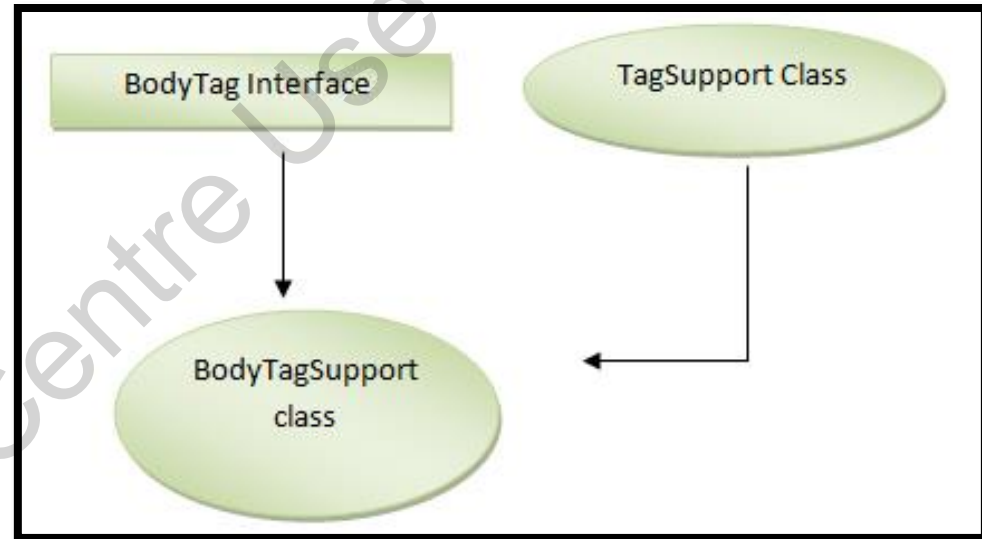
- `public void doInitBody () throws JspException { }`

Methods of BodyTag Interface

Extending TagSupport and BodyTagSupport

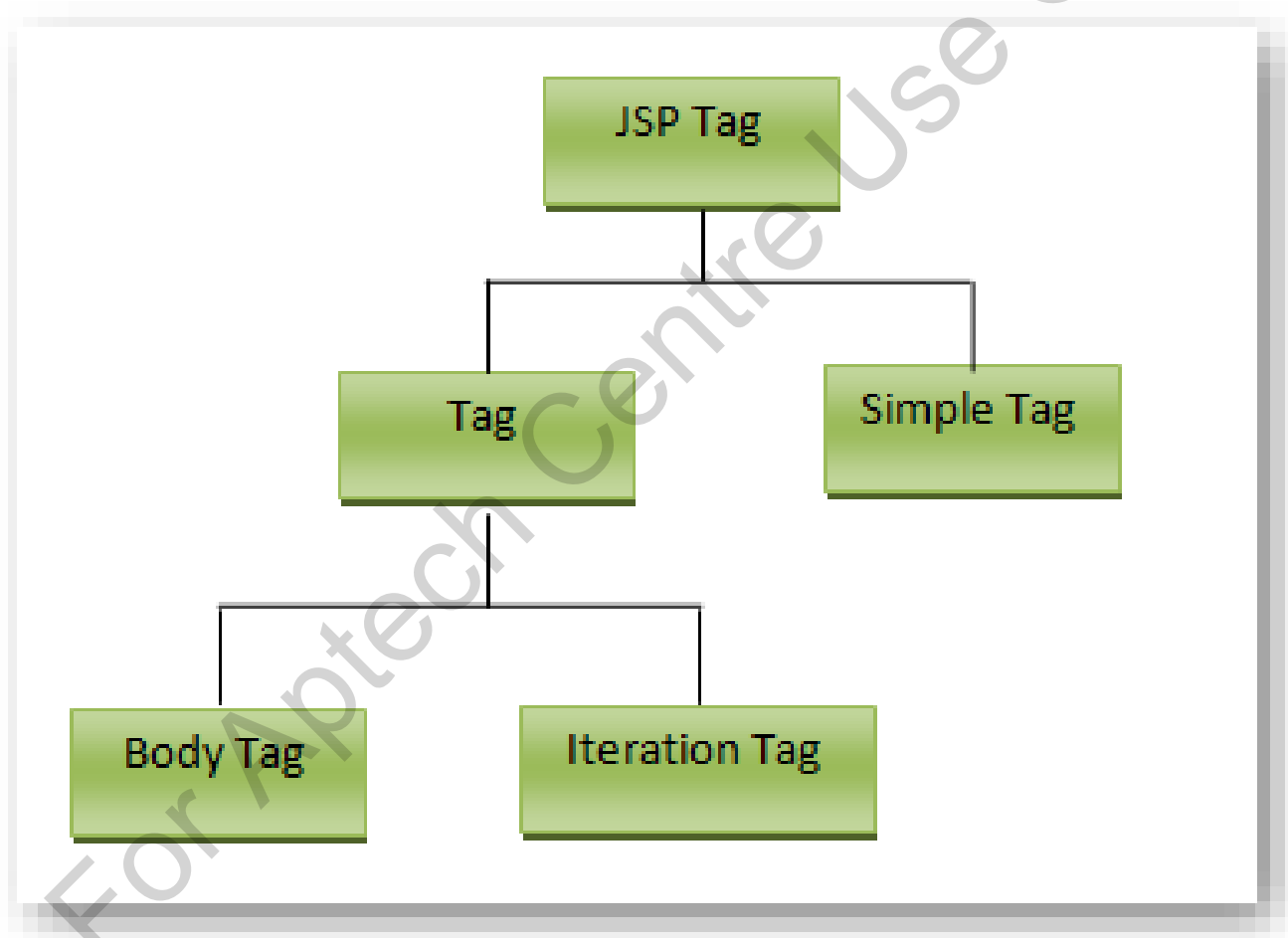


Hierarchy of TagSupport Class



Hierarchy of BodyTagSupport Class

JSP Tag Interfaces Hierarchy



Summary

- ❖ The JSTL tag library helps Web designers to integrate the Java technology code into JSP, by writing a simple code.
- ❖ The tag extension API in the JSP adds the tag functionalities to the language. It includes information on the tag library and its tags.
- ❖ `taglib` directive on a JSP page informs the Servlet that the page uses a tag library.
- ❖ The Tag extension API in the JSP adds the tag functionalities to the language.
- ❖ The Tag interface defines the lifecycle and the methods to be invoked at the start and end tag.
- ❖ The URI in the `taglib` directive points to a context relative path; this path hosts components that starts with `/` and is called the TLD resource path.
- ❖ JSTL provides decision-making tags to support conditions in a JSP page.
- ❖ The BodyTag interface extends IterationTag and adds new functionality that lets the tag handler evaluate its body content in a temporary buffer.
- ❖ JSTL is based on XML, which is very similar to HTML.
- ❖ The use of tag libraries, such as JSTL, helps people involved in the application development to focus on their area of expertise.