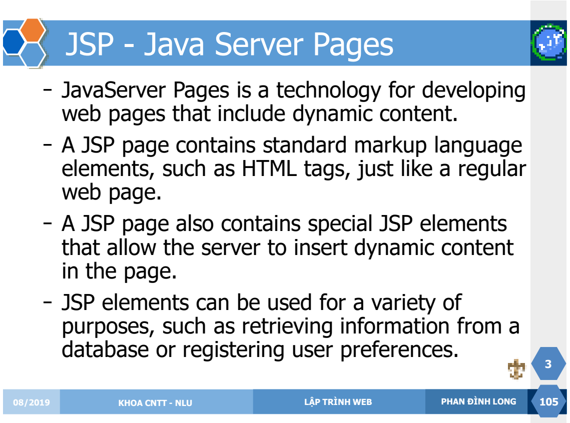
JSP/SERVLET

1

JSP

08/2019 KHOA CNTT - NLU LẬP TRÌNH WEB PHAN ĐÌNH LONG 2 105

2



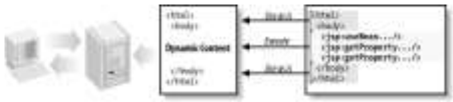
JSP - Java Server Pages

- JavaServer Pages is a technology for developing web pages that include dynamic content.
- A JSP page contains standard markup language elements, such as HTML tags, just like a regular web page.
- A JSP page also contains special JSP elements that allow the server to insert dynamic content in the page.
- JSP elements can be used for a variety of purposes, such as retrieving information from a database or registering user preferences.

08/2019 KHOA CNTT - NLU LẬP TRÌNH WEB PHAN ĐÌNH LONG 3 105

3

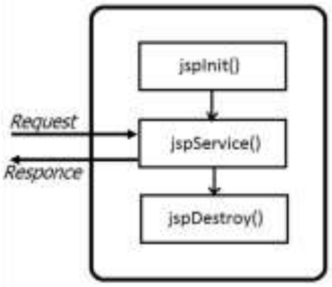
JSP - Java Server Pages



JSP - Java Server Pages

- **Why Use JSP?**
  - Embedding Dynamic Elements in HTML Pages
  - Using the Right Person for Each Task
  - Integration with Enterprise Java APIs

JSP - Java Server Pages

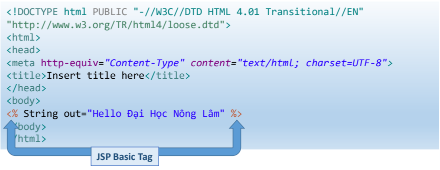


Core JSP syntax

- Directives
- Declarations
- Scriptlets
- Expressions

JSP Scriptlets

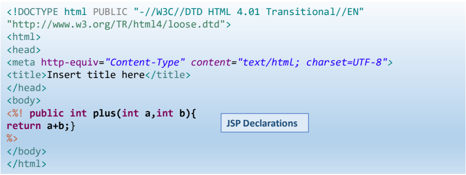
- Syntax: <% %>
- JSP Scriptlets using to writing code java to page.



```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
</head>
<body>
<% String out="Hello Đại Học Nông Lâm" %>
</body>
</html>
```

JSP Declarations

- Syntax :<%! ... %>
- JSP Declarations using to write method like other scripting language.

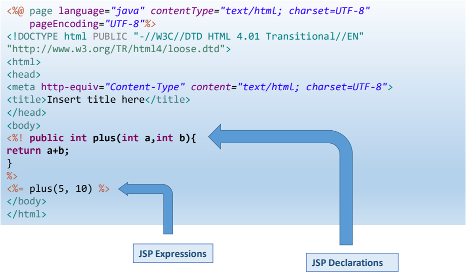


```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
</head>
<body>
<%! public int plus(int a,int b){
return a+b;}
%>
</body>
</html>
```

JSP Expressions

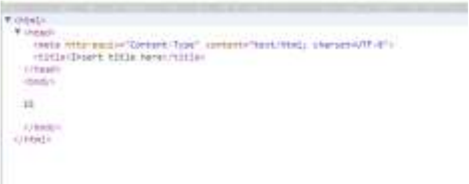
- Syntax: <%= … %>
- A **JSP expression** is used to insert the value of a scripting language expression, converted into a string, into the data stream returned to the client.

10



```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
</head>
<body>
<%! public int plus(int a,int b){
return a+b;
}
%>
<%= plus(5, 10) %>
</body>
</html>
```

11

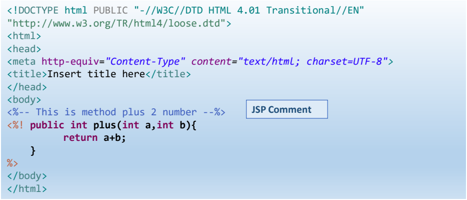


```
▼ <html>
  ▼ <head>
      <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
      <title>Insert title here</title>
    </head>
    <body>

    15

    </body>
</html>
```

12

JSP comment

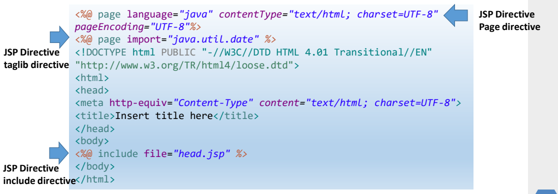
- Syntax :<%-- ... --%>
- Comment something in jsp page



```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
</head>
<body>
<%-- This is method plus 2 number --%>
<%! public int plus(int a,int b){
        return a+b;
    }
%>
</body>
</html>
```

JSP Directive

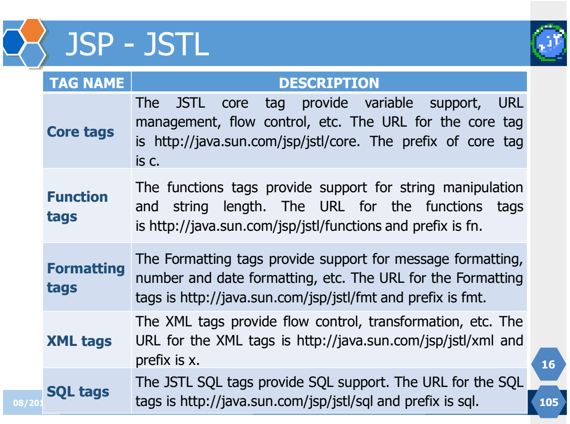
- <%@ %> JSP Directive: page directive, include directive, taglib directive



```
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<%@ page import="java.util.date" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
</head>
<body>
<%@ include file="head.jsp" %>
</body>
</html>
```

JSP - JSTL

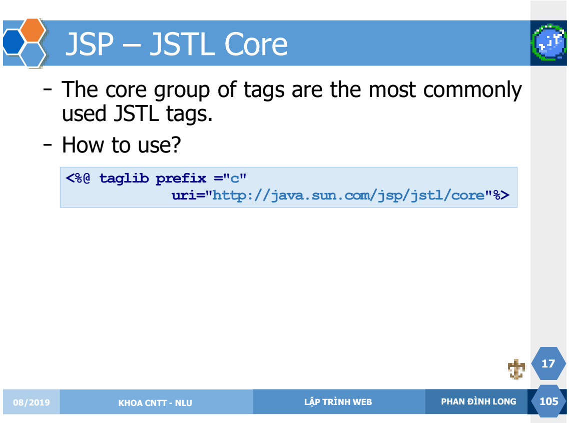
- The JSP Standard Tag Library (JSTL) represents a set of tags to simplify the JSP development.
- Advantage of JSTL
  - **Fast Development** JSTL provides many tags that simplify the JSP.
  - **Code Reusability** We can use the JSTL tags on various pages.
  - **No need to use scriptlet tag** It avoids the use of scriptlet tag.



JSP - JSTL

TAG NAME	DESCRIPTION
Core tags	The JSTL core tag provide variable support, URL management, flow control, etc. The URL for the core tag is http://java.sun.com/jsp/jstl/core. The prefix of core tag is c.
Function tags	The functions tags provide support for string manipulation and string length. The URL for the functions tags is http://java.sun.com/jsp/jstl/functions and prefix is fn.
Formatting tags	The Formatting tags provide support for message formatting, number and date formatting, etc. The URL for the Formatting tags is http://java.sun.com/jsp/jstl/fmt and prefix is fmt.
XML tags	The XML tags provide flow control, transformation, etc. The URL for the XML tags is http://java.sun.com/jsp/jstl/xml and prefix is x.
SQL tags	The JSTL SQL tags provide SQL support. The URL for the SQL tags is http://java.sun.com/jsp/jstl/sql and prefix is sql.

16

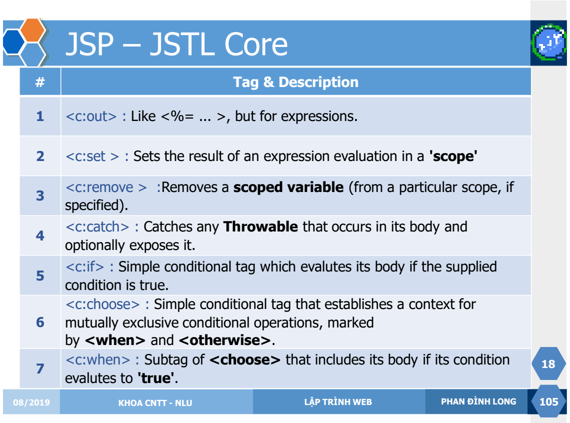


JSP – JSTL Core

- The core group of tags are the most commonly used JSTL tags.
- How to use?

```
<%@ taglib prefix ="c"
           uri="http://java.sun.com/jsp/jstl/core"%>
```

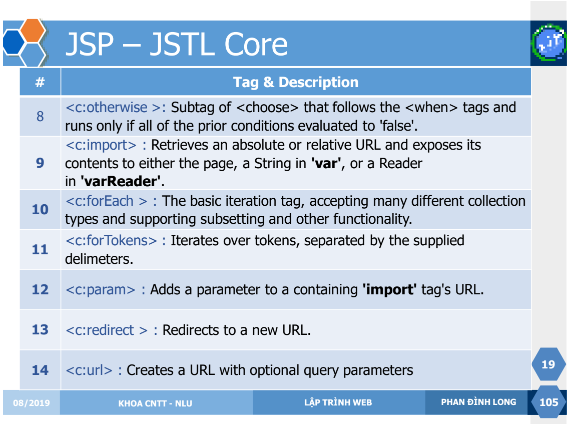
17



JSP – JSTL Core

#	Tag & Description
1	<c:out> : Like <%= ... >, but for expressions.
2	<c:set > : Sets the result of an expression evaluation in a **'scope'**
3	<c:remove > :Removes a **scoped variable** (from a particular scope, if specified).
4	<c:catch> : Catches any **Throwable** that occurs in its body and optionally exposes it.
5	<c:if> : Simple conditional tag which evalutes its body if the supplied condition is true.
6	<c:choose> : Simple conditional tag that establishes a context for mutually exclusive conditional operations, marked by **<when>** and **<otherwise>**.
7	<c:when> : Subtag of **<choose>** that includes its body if its condition evalutes to **'true'**.

18



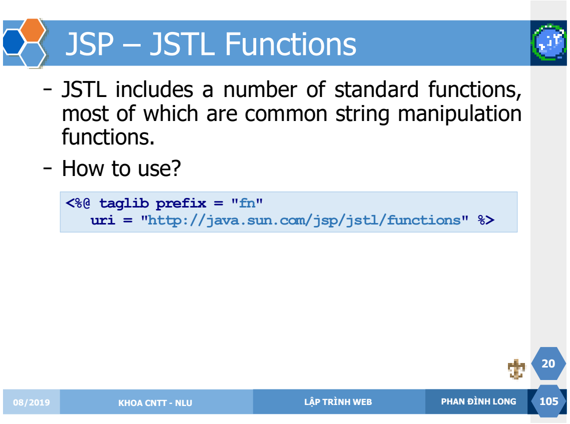
JSP – JSTL Core

#	Tag & Description
8	<c:otherwise >: Subtag of <choose> that follows the <when> tags and runs only if all of the prior conditions evaluated to 'false'.
9	<c:import> : Retrieves an absolute or relative URL and exposes its contents to either the page, a String in **'var'**, or a Reader in **'varReader'**.
10	<c:forEach > : The basic iteration tag, accepting many different collection types and supporting subsetting and other functionality.
11	<c:forTokens> : Iterates over tokens, separated by the supplied delimeters.
12	<c:param> : Adds a parameter to a containing **'import'** tag's URL.
13	<c:redirect > : Redirects to a new URL.
14	<c:url> : Creates a URL with optional query parameters

19

08/2019 KHOA CNTT - NLU LẬP TRÌNH WEB PHAN ĐÌNH LONG 105

19



JSP – JSTL Functions

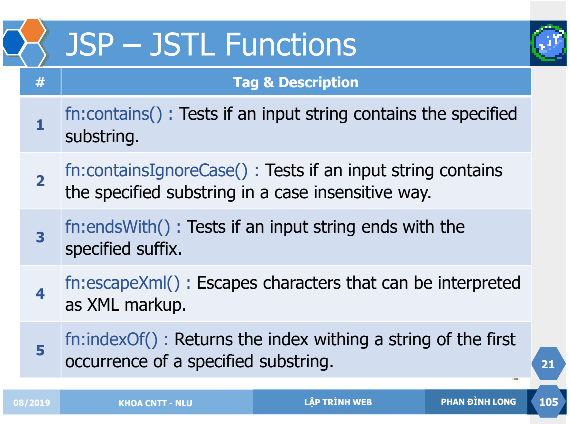
- JSTL includes a number of standard functions, most of which are common string manipulation functions.
- How to use?

```
<%@ taglib prefix = "fn"
   uri = "http://java.sun.com/jsp/jstl/functions" %>
```

20

08/2019 KHOA CNTT - NLU LẬP TRÌNH WEB PHAN ĐÌNH LONG 105

20



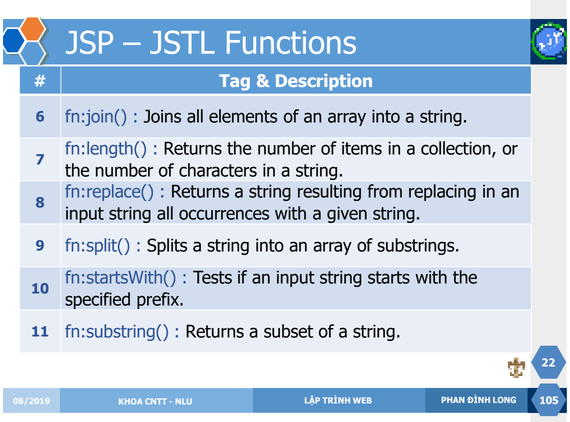
JSP – JSTL Functions

#	Tag & Description
1	fn:contains() : Tests if an input string contains the specified substring.
2	fn:containsIgnoreCase() : Tests if an input string contains the specified substring in a case insensitive way.
3	fn:endsWith() : Tests if an input string ends with the specified suffix.
4	fn:escapeXml() : Escapes characters that can be interpreted as XML markup.
5	fn:indexOf() : Returns the index withing a string of the first occurrence of a specified substring.

21

08/2019 KHOA CNTT - NLU LẬP TRÌNH WEB PHAN ĐÌNH LONG 105

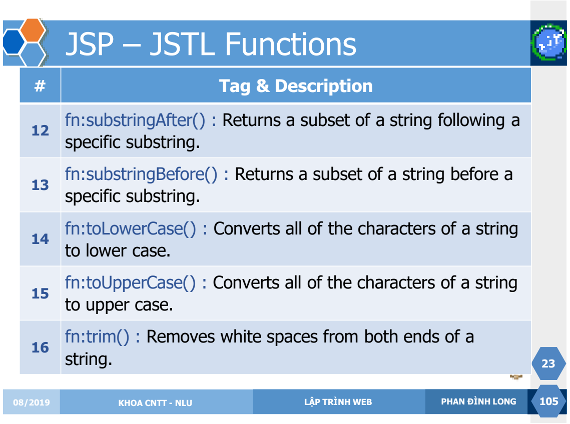
21



JSP – JSTL Functions

#	Tag & Description
6	fn:join() : Joins all elements of an array into a string.
7	fn:length() : Returns the number of items in a collection, or the number of characters in a string.
8	fn:replace() : Returns a string resulting from replacing in an input string all occurrences with a given string.
9	fn:split() : Splits a string into an array of substrings.
10	fn:startsWith() : Tests if an input string starts with the specified prefix.
11	fn:substring() : Returns a subset of a string.

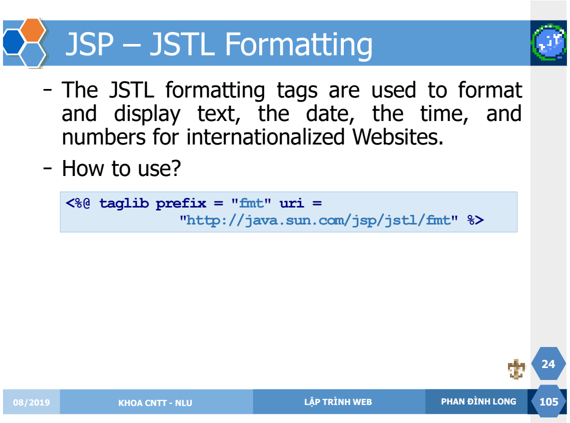
22



JSP – JSTL Functions

#	Tag & Description
12	fn:substringAfter() : Returns a subset of a string following a specific substring.
13	fn:substringBefore() : Returns a subset of a string before a specific substring.
14	fn:toLowerCase() : Converts all of the characters of a string to lower case.
15	fn:toUpperCase() : Converts all of the characters of a string to upper case.
16	fn:trim() : Removes white spaces from both ends of a string.

23

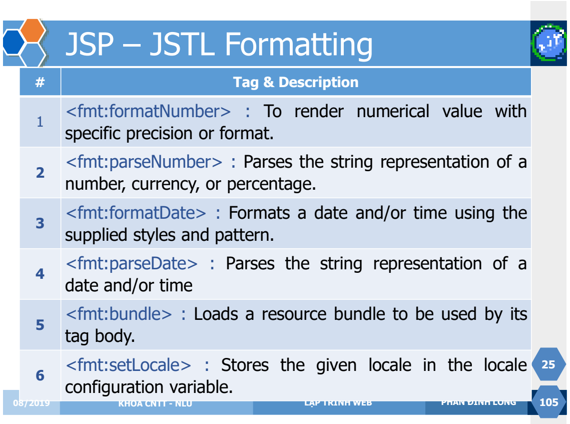


JSP – JSTL Formatting

- The JSTL formatting tags are used to format and display text, the date, the time, and numbers for internationalized Websites.
- How to use?

```
<%@ taglib prefix = "fmt" uri =
            "http://java.sun.com/jsp/jstl/fmt" %>
```

24

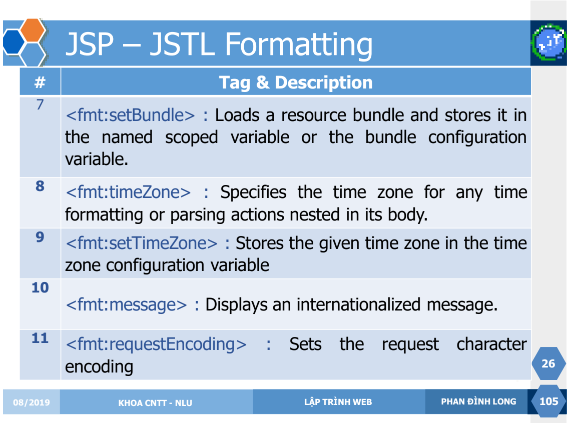


JSP – JSTL Formatting

#	Tag & Description
1	<fmt:formatNumber> : To render numerical value with specific precision or format.
2	<fmt:parseNumber> : Parses the string representation of a number, currency, or percentage.
3	<fmt:formatDate> : Formats a date and/or time using the supplied styles and pattern.
4	<fmt:parseDate> : Parses the string representation of a date and/or time
5	<fmt:bundle> : Loads a resource bundle to be used by its tag body.
6	<fmt:setLocale> : Stores the given locale in the locale configuration variable.

08/2019 KHOA CNTT - NLU LẬP TRÌNH WEB PHAN ĐÌNH LONG 25 105

25

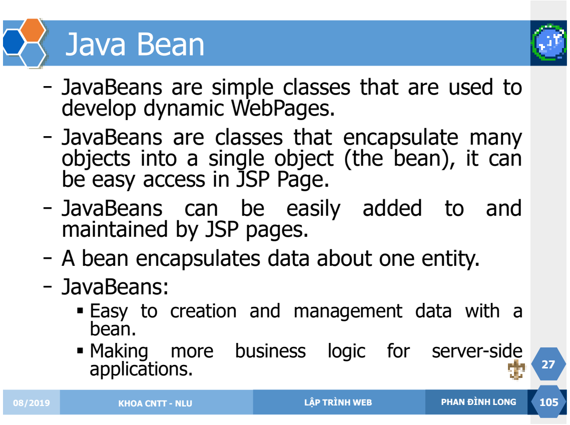


JSP – JSTL Formatting

#	Tag & Description
7	<fmt:setBundle> : Loads a resource bundle and stores it in the named scoped variable or the bundle configuration variable.
8	<fmt:timeZone> : Specifies the time zone for any time formatting or parsing actions nested in its body.
9	<fmt:setTimeZone> : Stores the given time zone in the time zone configuration variable
10	<fmt:message> : Displays an internationalized message.
11	<fmt:requestEncoding> : Sets the request character encoding

08/2019 KHOA CNTT - NLU LẬP TRÌNH WEB PHAN ĐÌNH LONG 26 105

26



Java Bean

- JavaBeans are simple classes that are used to develop dynamic WebPages.
- JavaBeans are classes that encapsulate many objects into a single object (the bean), it can be easy access in JSP Page.
- JavaBeans can be easily added to and maintained by JSP pages.
- A bean encapsulates data about one entity.
- JavaBeans:
  - Easy to creation and management data with a bean.
  - Making more business logic for server-side applications.

08/2019 KHOA CNTT - NLU LẬP TRÌNH WEB PHAN ĐÌNH LONG 27 105

27

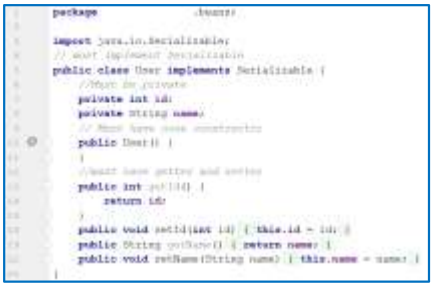
Java Bean

- A JavaBeans component is a Java class with the following features:
  - The class must implement **java.io.Serializable** or java.io.Externalizable interface to allow a tool to save and restore the bean's state.
  - A Java bean must be in a package (not default package).
  - A no-argument constructor.
  - Define getter and setter for each variable.
    - A readable property has a getter method.
    - A writable property has a setter method.
  - Class must define private instance variables.
-

28

Java Bean – how to use?

- Define bean.

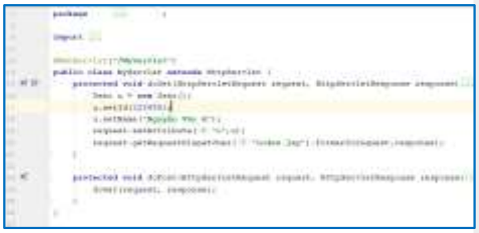


```
package [illegible].beans;

import java.io.Serializable;
// must implement Serializable
public class User implements Serializable {
    //Must be private
    private int id;
    private String name;
    // Must have none constructor
    public User() {
    }
    //must have getter and setter
    public int getId() {
        return id;
    }
    public void setId(int id) { this.id = id; }
    public String getName() { return name; }
    public void setName(String name) { this.name = name; }
}
```

29

Java Bean – how to use?

- Create and send bean to jsp.

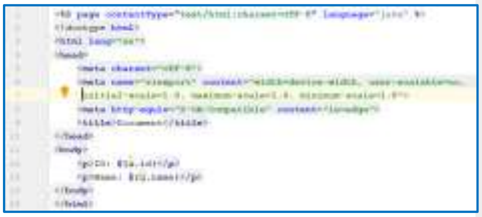


```
package [illegible];

import [illegible]

@WebServlet("/MyServlet")
public class MyServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response) [illegible]
        User u = new User();
        u.setId(123456);
        u.setName("Nguyễn Văn A");
        request.setAttribute("u", u);
        request.getRequestDispatcher("index.jsp").forward(request, response);
    }

    protected void doPost(HttpServletRequest request, HttpServletResponse response) [illegible]
        doGet(request, response);
    }
}
```

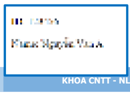
30

Java Bean – how to use?

- JSP page.



```
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<!doctype html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, user-scalable=no,
    initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Document</title>
</head>
<body>
    <p>ID: ${u.id}</p>
    <p>Name: ${u.name}</p>
</body>
</html>
```



```
ID: [illegible]
Name: Nguyễn Văn A
```

JSP Page output

31

Object and Scope in JSP

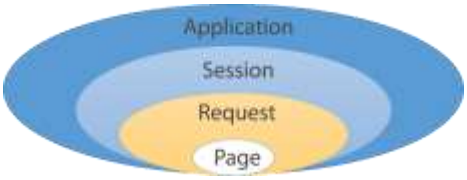
- Within both Scriptlets and Expressions there are certain implicit objects available for use.
  - Created by container
- Implicitly defined variables available for scripting:
  - request - HttpServletRequest object
  - response - HttpServletResponse object
  - pageContext - The PageContext for this JSP
  - session - HttpSession object (if any)
  - application - the ServletContext object
  - config - ServletConfig object for this JSP
  - out – JspWriter
  - page - Page's implementation class processing the current
  - exception - the Throwable object passed to this error page

32

Object and Scope in JSP

Most visible	**Application**	Objects accessible from pages that belong to the same application
	Session	Objects accessible from pages belonging to the same session as the one in which they were created
	Request	Objects accessible from pages processing the request where they were created
Least visible	**Page**	Object accessible only within pages where they were created

33



JSP scope

34

Application
Session
Request
Page

Objects with ***page scope*** are **accessible only within the page in which they're created**. The data is valid only during the processing of the current response; once the response is sent back to the browser, the data is no longer valid.

Objects of Page Scope

- response
- pagecontext
- out
- config
- page

35

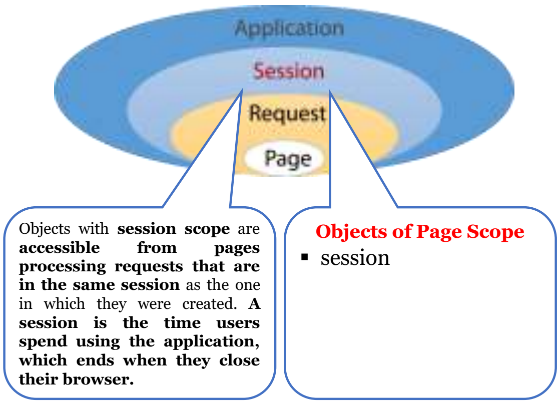
Application
Session
Request
Page

Objects with **request scope** are **accessible from pages processing the same request** in which they were created. Once the container has processed the request, the data is released.

Objects of request scope

- request

36

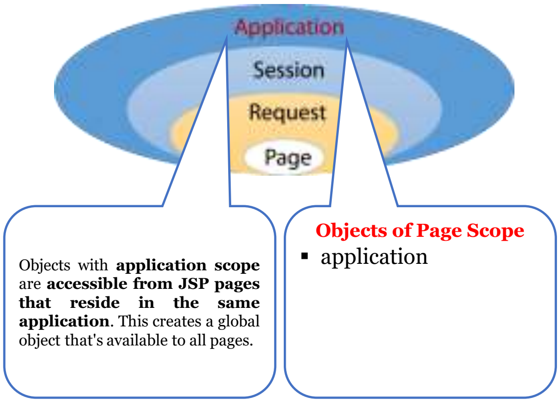


Objects with **session scope** are **accessible from pages processing requests that are in the same session** as the one in which they were created. **A session is the time users spend using the application, which ends when they close their browser.**

Objects of Page Scope

- session

37

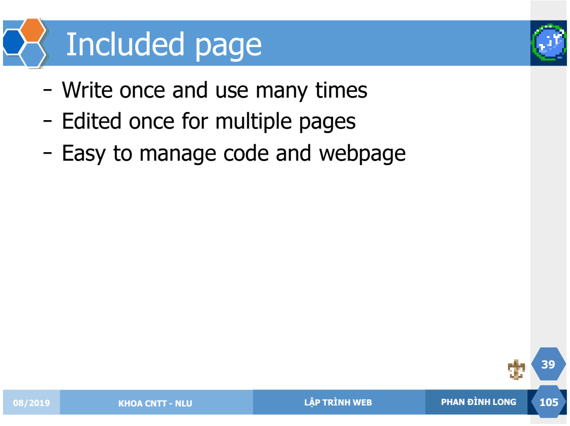


Objects with **application scope** are **accessible from JSP pages that reside in the same application**. This creates a global object that's available to all pages.

Objects of Page Scope

- application

38



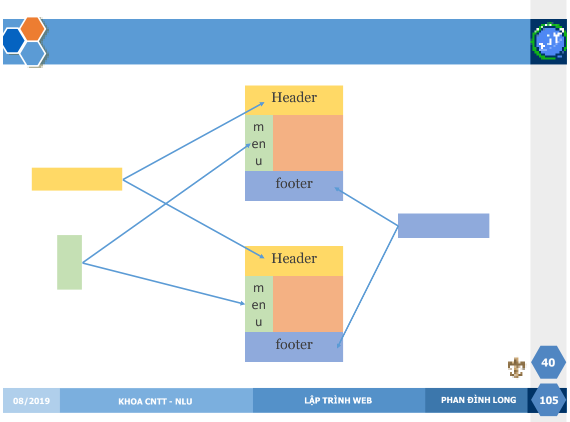
Included page

- Write once and use many times
- Edited once for multiple pages
- Easy to manage code and webpage

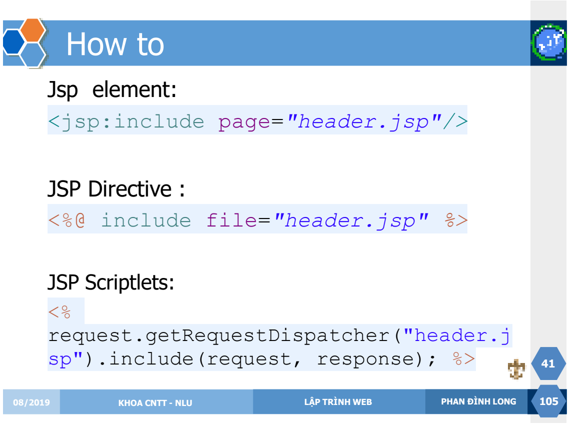
39

08/2019 KHOA CNTT - NLU LẬP TRÌNH WEB PHAN ĐÌNH LONG 105

39



40



How to

Jsp element:

```
<jsp:include page="header.jsp"/>
```

JSP Directive :

```
<%@ include file="header.jsp" %>
```

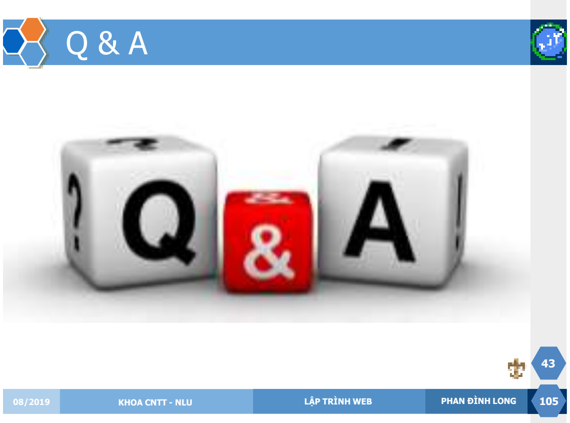
JSP Scriptlets:

```
<%
request.getRequestDispatcher("header.j
sp").include(request, response); %>
```

41

DEMO

42



Q & A

43

43