



Directives

=====

1. page directive
2. include directive
3. taglib directive

taglib directive

=====

=> We can use taglib directive to make custom tags available to our jsp page.

=> `<%@ taglib uri = 'Location of TLD file' prefix='[prefixName for a tag]' %>`

=> taglib directive contains 2 attributes prefix and uri.

=> uri represents the location of TLD file which inturn represents the TagHandler class.

eg::In SpringMVC to design a view using jsp we prefer working with form tag library given by pivotal team

index.jsp

=====

```
<%@ taglib uri = 'http://www.springframework.org/tags/form' prefix = 'form' %>
```

```
<form:form>
```

```
    <table>
```

```
        <tr>
```

```
            <td>First Name:</td>
```

```
            <td><form:input path="firstName" /></td>
```

```
        </tr>
```

```
        <tr>
```

```
            <td>Last Name:</td>
```

```
            <td><form:input path="lastName" /></td>
```

```
        </tr>
```

```
        <tr>
```

```
            <td colspan="2">
```

```
                <input type="submit" value="Save Changes" />
```

```
            </td>
```

```
        </tr>
```

```
    </table>
```

```
</form:form>
```

JSP Scripting Elements

=====

There are 2 types of Scripting elements

- a. Traditional Scripting elements.
- b. Modern Scripting elements.

Traditional Scripting elements

- a. Expression (`<%= expression %>`)
- b. Scriptlet (`<% any java code %>`)
- c. Declarative (`<%! any java declarations %>`)
- d. Comments (`<%-- JSP comments --%>`)

1. Expression tag

This tag is used to print java expression values to the JSP.

syntax: `<%= expression %>`

Inside the servlet the code would be replaced as

```
_ jspService(request,response)
{
    out.print(expression);
}
```

index.jsp

=====

<h1 style='color:blue;text-align:center;'>Working with Expression tag</h1>

<h1 style='color:green;text-align:center;'>

Username is :: <%= request.getParameter("username")%>

Password is :: <%= request.getParameter("password")%>

</h1>

request ==> http://localhost:9999/ThirdApp/index.jsp?

username=sachin&password=tendulkar

Conclusions

=====

1. Inside the expression tag, we are not supposed to keep ; symbol, if we keep it would result in 500 server error.

<%= new java.util.Date(); %> //invalid

2. Inside an expression tag we can use method calls also, but void return type method calls are not allowed.

<%= new java.util.ArrayList().size() %> //valid

<%= new java.util.ArrayList().clear() %> //invalid

3. Inside an expression tag space is not allowed between % and = , it would result in 500 server error.

<% =request.getParamter("username") %> //invalid

4. Inside an expression tag declaration statements are not allowed.

<%= String name ="dhoni" %> //invalid

which of the following are valid java expression?

1. <%= 27 %> //valid

2. <%= "27" %> //valid

3. <%= Math.random() %> //valid

4. <%= 10*20 %> //valid

5. <%= 10>20 %> //valid

6. <%= new Student() %> //valid

7. <%= String name= "Apeksha" %> //invalid

Scriptlet tag

=====

We use this tag to write java code inside jsp.

Syntax:

<%

Any java code

%>

Java code inside Scriptlet tag will be placed directly inside _jspService() method of generated servlet.

Every java statement present inside Scriptlet tag should be terminated with ; symbol.

eg:

index.jsp

=====

<h1 style='color:blue;text-align:center;'>Working with Scriptlet tag</h1>

<%

int x =10;

out.println(x);

%>

request -> http://localhost:9999/ThirdApp/

Note: It is not recommended to use Scriptlet tag inside jsp.

Declarative Tag(<%! %>)

=====

We can use Declarative tag to declare instance variables, static block, instance block, methods etc...

Syntax:

```
<%!  
    Any java declarations  
%>
```

Note: Any java declarations written inside declarative tag would be placed directly in the generated servlet but outside
 _ jspService() method.

index.jsp

=====

```
<h1 style='color:blue;text-align:center;'>Working with Declarative tag</h1>
```

```
<%!  
    int x =10;  
    static int y = 20;  
    int[] a = {10,20,30};  
    public String m1(){  
        System.out.println(a[0]);  
        System.out.println(x);  
        return "working with Declarative tag";  
    }  
%>
```

```
<%= m1()%>
```

request=> http://localhost:9999/ThirdApp/

index.jsp

=====

```
<%!  
    public String m1(){  
        int x = 10;  
        out.println(x);  
        return "working with Declarative tag";  
    }  
%>
```

```
<%= m1()%>
```

request=>http://localhost:9999/ThirdApp/

output: CE: out can't be resolved.

Note: All jsp implicit objects are local to _ jspService() method, but declaration tag code will be placed outside of _ jspService() method. so we can't use jsp implicit objects inside declaration tag. Where as scriptlet and expression tag code will be placed directly inside _ jspService(), so jsp implicit objects are accessible inside scriptlet and expression tag.

index.jsp

=====

```
<h1 style='color:blue;text-align:center;'>Working with Scripting Elements</h1>
```

```
<%!  
    java.util.Date d = null;
```

```

        String date =null;
%>
<%
        d = new java.util.Date();
        date = d.toString();
%>
<html>
    <head><title>Output</title></head>
    <body bgcolor='lightyellow'>
        <center>
            <b>
                <font size='6' color='red'>
                    Today date is <%= date%>
                </font>
            </b>
        </center>
    </body>
</html>

```

4. Comments inside JSP

Inside jsp can we write html code(html comments),java code(java comments),jsp code(jsp comments).

1. JSP comments

```
<%-- jsp comments --%>
```

2. HTML comments

```
<!-- HTML comments -->
```

3. Java comments

1. // single line comments

2. /*
 multiline comments
*/

3. /**
 Java documentation comments
*/

summary of jsp comments

```
=====
```

a. jsp comments

```

Is it visible in jsp      =====> yes
Is it visible in translated servlet =====> no
Is it visible in end user response source code =====> no

```

b. html comments

```

Is it visible in jsp =====> yes
Is it visible in translated servlet =====> yes
Is it visible in end user response source code =====> yes

```

c. java comments

```

Is it visible in jsp =====> yes
Is it visible in translated servlet =====> yes
Is it visible in end user response source code =====> no

```

Note: Among expression,scriptlet,declaration and jsp comments, we cannot use one inside another.

ie, nesting of these scripting elements is not possible, if we try to do it would result in 'CompileTimeError'.