2. JSP Fundamentals

Contents

[1. Introduction 1](#_Toc23551)

[2. Demo: Creating First JSP Page - Hello World 1](#_Toc9869)

[3. Expression Tag 1](#_Toc1726)

[4. Declaration Tag 1](#_Toc25806)

[5. Scriptlet Tag 2](#_Toc21541)

[6. Summary 2](#_Toc6457)

# 1. Introduction

=>slides: Pg. 1

In this module we shall understand the JSP fundamentals which is required for getting started with the JSP development.

=>slides: Pg. 2

We shall start with creating our first JSP page using Esectionse and then we shall understand the expression declaration, and scriptlet tags with simple demos, and we know that whenever we create a JSP page, it generates a servlet dynamically, so we shall also understand where the code writt

# Demo: Creating First JSP Page - Hello World

=>slides: Pg. 3

I already opened Esectionse and now in order to create a new JSP application, click on the File menu, New, and we can select Dynamic Web Project.



Within the New Dynamic Web Project window,



provide a meaningful name for the project. Let me type in JSPFundamentals, and then we need to set the target runtime. So let me click on New Runtime and let me select Apache Tomcat version 8. 0. And click on Next. Now we need to select the Tomcat installation directory. So let me click on the Browse button and select the Apache Tomcat directory and click on Finish button. And once the installation directory is selected, then click on Next button, and finally at configure web module settings,



check the generate web. xml checkbox, and then click on Finish button. It is advisable to maintain all the HTML and the JSP pages within the WebContent folder.



So let me right click on the WebContent folder and then click on New and then select the JSP File. Provide a meaningful name for the file. For example, let me type in HelloWorld. jsp, then click on Next button.

We can observe a template selection page, which allows the user to select the template file for creating the JSP page.



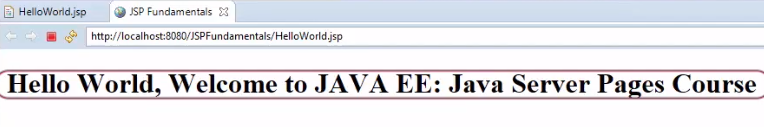
Let me use the default template for the preference page and click on the Finish button. We can observe a set of HTML instructions which creates a basic template for the page and also we can observe the first line, which is having a statement page language equal to Java, et cetera. This line is set to be a page directive, which I will explain in detail while explaining about the directives in the next module.



Now let me update the title for this page as JSP Fundamentals and within the body tag,



let me add a header tag h1 and let me type Hello World, welcome to JAVA EE: Java Server Pages Course. Now let me execute the page.

We can observe the output of our first JSP page is displayed. Quite a simple page to get started with. In the next section we shall start understanding the scripting elements of JSP.

# Expression Tag

=>slides: Pg. 4

Once we have understood how we can write a JSP page, in order to get started with the development, we need to understand some fundamentals in the scripting elements.

=>slides: Pg. 5

So let us first start with the expression tag. JSP expressions are used to compute an expression and the result of that is included in HTML page that is written to the browser. The expression scripting element can contain any valid Java expression. The scripting element will evaluate the expression and converts that value into string format and then it will be sent to the browser.

=>slides: Pg. 6

In order to define a JSP expression, syntax will be angular brackets, percentile, and an equal to, followed by the Java expression to be evaluated. For example, let us say I wanted to display the current date time, so within the JSP page I can type in paragraph, current date and time,

=>slides: Pg. 7

and then I can write an expression as new java. util. Date. Once this expression tag is processed by the server, then the generated HTML is simply the toString version of the given data object.

=>slides: Pg. 8

Any expression specified within the expression scripting element will be placed inside the jspService method, generated by the JSP container during the translation phase.

=>slides: Pg. 9

Now let us understand practically with some more examples of using JSP expressions. Let me open Esectionse and let us write the code. And let me create a new JSP file. So let me right click on the WebContent folder and I say New, JSP File.



And for the name of this file I will simply call it TestExpression. jsp. And once we provided the name, we can go ahead and click on the Finish button. We have a very basic HTML file here.



Now let me update the title for this page.

So let's go ahead and type out the first bit of converting a string to an upper case. So what I would like to do is add JSP expression.



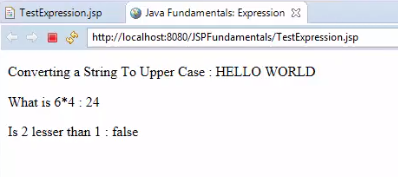
So let me type in angular brackets, percentile, equal to, here I say new string of Hello World and a. toUpperCase. Let me execute the page. We can observe the HELLO WORLD in uppercase. Now I wanted to do an example using a mathematical expression.



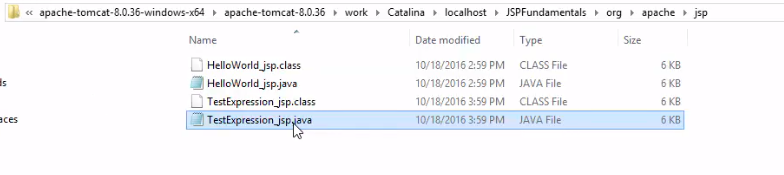
I just wanted to find what does 6 multiplied by 4 equals. Now again, I will make use of my JSP expression angular brackets, percentile with an equal symbol 6 times 4. Let me come back to the browser and all I have to do is simply reload this page and the result will show up. So let me hit the reload or the refresh button here and there it is, 6 multiplied by 4 equals 24. Again, a nice thing here with JSP is if you make any changes to the file, we don't have to rerun it or redeploy it, we can simply just reload that page and we are ready to go. Okay let us do one more thing, let us play around with the Boolean expression.



So let me just move down here, I will just have a very basic Boolean expression, example is 2 less than 1? And I will let the Java compute this for me again. I'm just showing an example here of a Boolean expression. So let me type in angular brackets, percentile, equal to is 2 less than 1? This will evaluate if it is true or false. Of course we know that this will return false, but we will see how the JSP program will help us out with this. Now let me save this file and once again let me reload the page.



There it is, is 2 less than 1: false. We saw a really simply example of using JSP expressions for Boolean, also for math and also for string objects. We know that the code returned within the scripting elements will be placed within the generated servlet page.



So to observe that, let me open the generated servlet file for the JSP page and let me scroll down to the jspService method. We can observe whatever the code we have written within the expression scripting element is present within the jspService method. Later in our course, we will be using these expressions quite often. In the next section we shall understand declaration scripting element.

# Declaration Tag

=>slides: Pg. 10

Once we have understood the expression tag, now let us understand the next scripting element declaration tag. Declaration tag is used to declare the variables and the methods within the JSP page and the methods defined within the declaration scripting elements can be called from the same JSP page. It is very useful when we need to execute some code over and over again. So we can simply encapsulate the code within the JSP method declarations.

=>slides: Pg. 11

The syntax in order to define the declaration tag will be angular brackets, percentile and exclamation followed by the variables and the methods to be defined. The declaration scripting elements can contain both instance and the static members. For example, to define an integer we use angular brackets, percentile, exclamation, int id equal to 10. And to define a static string variable, we can use angular brackets, percentile, exclamation, static String department equal to the value, for example, SALES. We know that declaration tags also supports to define the methods. In order to define a method, we can use angular brackets, percentile, exclamation, decimal getBonus of decimal salary, return salary multiplies 0. 10. Whenever we declare any variable of methods within declaration tag, then that code will be implicitly placed inside the class that is generated by the JSP container as declarations and the code can be accessed from the JSP lifecycle methods.

=>slides: Pg. 12

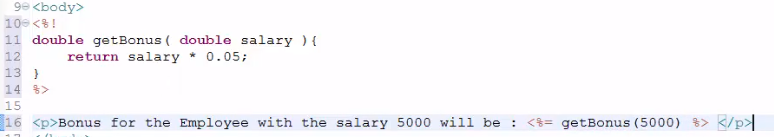
Now let us understand how to work with the declaration tags practically. Let me open the Esectionse and let us write the code and let me create a new JSP file. So let me right click on the WebContent folder and I will say New, JSP File.



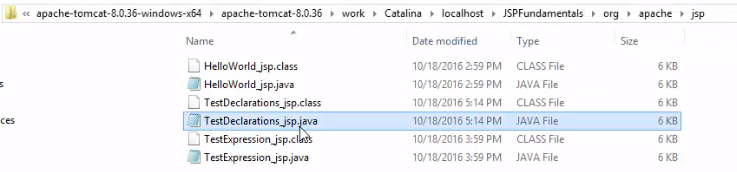
And then name of the file I would like to provide is TestDeclarations. jsp. Once we provide it the file name, we can go ahead and click the Finish button.



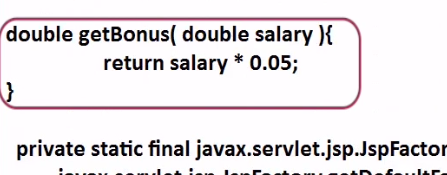
Now let me update the title for this page within the body.



Now what I would like to do is actually declare a method so I can use the JSP syntax, the angular brackets, percentile with the exclamation point. So at this point I can actually start writing any normal Java code. So I'm going to write a simple Java method here. I would like to provide the return type of the method as double and the method name as getBonus and the input data salary as double, and I will actually write the real implementation code and, again, this is very trivial example. I will just return salary multiplies 0. 05. Now we have the method declaration. Let us use the method, so let me type in paragraph Bonus for the Employee with the salary will be, let me write an expression, angular brackets, percentile, equal to getBonus of 5000. Now let me save the file and execute the file. We can observe the bonus details.



Now let me flip to the folder where the Java Server Page has been generated for the JSP file. And let me open the file.



We can observe that declarations of JSP file are maintained within the global declarations section of the generated servlet code. In the next section we shall understand another important scripting element, scriplet tag.

# Scriptlet Tag

=>slides: Pg. 13

We already understood two scripting elements of JSP, one was expression and the other one was declaration. Now let us understand one more important scripting element of JSP, that is JSP scriptlets. =>slides: Pg. 14

So what exactly is a JSP scriptlet? Well a scriptlet is a scripting element which allows us to add any variable declarations and any computed logic within the JSP page.

=>slides: Pg. 15

The syntax in order to define a JSP scriptlet is an angular bracket and a percentile followed by any number of valid Java code. The Java code specified in the scriptlet will be placed inside jspService method by the JSP container during the translation phase. A JSP file can contain any number of scriptlet elements. The code will be executed top to down in a sequential order when the page is processed. Before we continue with the demo, let us understand a small difference between the declaration and the scriptlet.

=>slides: Pg. 16

The code returned at the declaration element will be placed within the declarations of the generated servelet source file at the translation phase. The code returned within the scriptlet will be placed inside the jspService method. Declaration can contain both variables and the methods, whereas the scriptlet can contain only variables and any computed logic. It cannot have any method definitions within it.

=>slides: Pg. 17

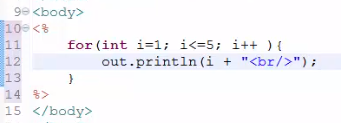
Now let us understand practically how to work with scriptlet. Let me open the Esectionse again and let us write the code. And what I would like to do is continue to use the previous project, JSP Fundamentals. And let me create a new JSP file. So let me right click on the web content folder and click on New, JSP File, and let me provide a meaningful name for the file.



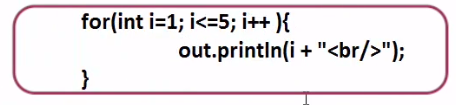
For example, TestScriptlets. jsp, and let me click on the Finish button.



Now let me update the title for this page. Now within the body tag, let us write our code.



So let us make use of scriptlet. To do, let me place an angular brackets, percentile, and then we can start writing our Java code here. So I will simply set up a for loop for i=1 and i is less than or equal to 5; i++, then I wanted to print out some information some information included in the page. So I will make use of out. println method. So let me type in out. println of i + break tag where out is a JSP built-in object which references the JSP writer class. I will explain about the JSP built-in objects later in our course. Now let me save the file and execute the file. We can observe the loop has been executed and we can see the result. Now let me open the generated file for the JSP page and let me scroll down to the jspService method.



We can observe whatever the code we have written within the scriptlet is available within the jspService method.

# Summary

=>slides: Pg. 18

In this module we have understood the JSP fundamentals to get started with the JSP development, and also we have understood practically some of the concepts what we have discussed within the module 1. Now let us understand some best practices while using JSP scripting elements. We need to minimize the number of scripting elements used in a JSP page and also we need to avoid writing thousands of lines of code in the JSP. Also we need to refactor this into separate Java classes or make use of MVC, and I will cover all these concepts in the later modules of our course. If you observe in all the JSP pages we created, the very first line you can notice, there will be an angular bracket, percentile, and an at symbol, followed by a page, which is one of the directives provided by the JSP. Now in the next module we shall understand working with the directives, which is one of the most important features of JSP programming.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*