

Module 3-4

Views Part 2

Variable Scope

- Generally speaking, variables in JSP's can be slotted into 4 different groups: page, request, session, and application. We can set these via the scope attribute.

```
<c:set var="myPageVar" value="Larry" scope="page" />  
<c:set var="myRequestVar" value="Curly" scope="request" />  
<c:set var="mySessionVar" value="Moe" scope="session" />
```

- Now consider the following attempt to inject myRequestVar into the HTML, which will output the value "Curly":

Value of `requestScope.myRequestVar` is: `${requestScope.myRequestVar}`

Variable Scope

- The default scope (if no scope attribute is present) is pageScope.
- Behind the scenes, these “implicit” objects - pageScope, requestScope, and sessionScope are implementation of java maps.
- You can refer to these without a scope, but at your own peril! What would happen if two variables had the same name but different scope?

```
<c:set var="ambiguousReference" value="ambiguousReference is in session scope" scope="session" />  
<c:set var="ambiguousReference" value="ambiguousReference is in request scope" scope="request" />
```

Let's See some Examples of Scoping

Scriptlets

Scriptlets are embedded pieces of Java code in your JSP file.

- They must be enclosed in with the following set of tags: `<% ... %>`
- Values defined within a scriptlet can be made accessible to an EL expression using the `pageContext` object.

Scriptlets Examples

Scriptlets are typically interleaved into the JSP code. Consider this eye sore:

```
<%  
boolean isSomething = true;  
if (!isSomething) {  
%>
```

Is False

```
<%  
} else {  
%>
```

Is True

```
<%  
}  
%>
```

The parts highlighted in red are chunks of Java code.

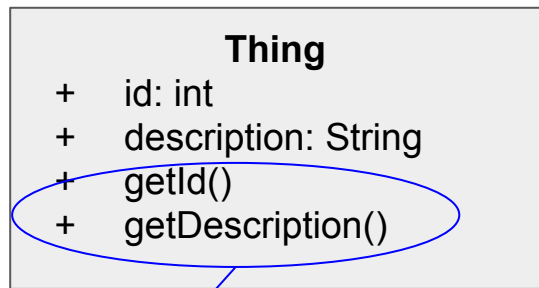
The output is:
Is True

Java Objects and JSP Mapping

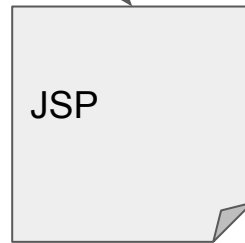
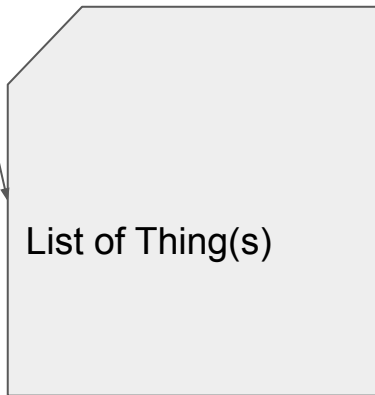
Ultimately we want a mapping between the objects we create on the server (in Java) and elements that are rendered in HTML through JSP's.

- We do this by following the JavaBeans naming convention on the server side.
- We deploy getters and setters to the data members for an object we want to expose to the end user through JSP's.
- On the JSP side, we can access these data members using Expression Language dot notation.

Java Objects and JSP Mapping



Coincidentally, these 2 methods observe the proper "javaBeans naming convention"



We will learn more about this part in the coming days.

We will be focused on this part today.

Revisiting the For Each Loop

- Here we will introduce an alternative syntax to the ForEach loop with JSTL. Suppose that data is the list object from the previous slide that has been transmitted to the JSP:

```
<c:forEach var="item" items="${data}">
<tr>
  <td>${item.id}</td>
  <td>${item.description}</td>
</tr>
</c:forEach>
```

Let's Do Some Coding!