JSP Custom Tags Quiz

1. How do custom tags relate to JSTL?

   Ans:- Custom tags are user-defined tags. They eliminates the possibility of scriptlet tag and separates the business logic from the JSP page.

   The JSTL core tag provide variable support, URL management, flow control, etc.

   The functions tags provide support for string manipulation and string length

   The Formatting tags provide support for message formatting, number and date formatting, etc.

   The XML tags provide flow control, transformation, etc.

   The JSTL SQL tags provide SQL support.

2. What is the role of the URI in the TLD and the taglib directive?

Ans:- We can use the custom URI, to tell the web container about the tld file. In such case, we need to define the taglib element in the web.xml. The web container gets the information about the tld file from the web.xml file for the specified URI.

3. What is a tag handler class?

   Ans:- Create the Tag handler class and perform action at the start or at the end of the tag. Create the TagLibrary Descriptor (TLD) file and define tags Create the JSP file that uses the Custom tag defined in the TLD file

4. What is the role of attribute setters in a tag handler class?

   Ans:- Attribute lets you configure `getter` and `setter` functions for each attribute. These functions are invoked when the user calls Attribute's `get` and `set` methods, and provide a way to modify the value returned or the value stored respectively. You can also define a `validator` function for each attribute, which is used to validate the final value before it gets stored. All these functions receive the value and name of the attribute being set or retrieved, as shown in the example code below. The name is not used in this example, but is provided to support use cases where you may wish to share the same function between different attributes

6 . What is the role of the doTag() method in a tag handler class?

Ans:- the dotag() method is used to implement/override to code functionality of tag. And this is invoked when the end element of the tag encountered.

7. What does the operation getJspContext().getOut(). write(“Hi Bob”) do when called in a doTag() method?

Ans:- public class HelloAttributeTag extends SimpleTagSupport {
  private String name;
//the container calls back this setter to process attribute “name”
  public void setName(String name) {
this.name= name;
}
@Override
      public void doTag() throws JspException {
JspWriter out= getJspContext.getOut();
 try   {
     if(name!=null) {
      out.println(“<h1>Hello, “+name+”!</h1>”);
}
else {
        out.println(“<h1>Hello, everybody !</h1>”);
}
} catch(java.io.IOException ex) {
throw new JspException(“Error in HelloAttributeTag”, exe);
}

}

8. What does the operation getJspBody().invoke(null) do when called in a doTag() method?

Ans:- *When getJspBody().invoke(null) is called, it is the output resulting from the execution of the tag body's JSP content that gets passed to the client, not the JSP code itself.* Therefore, the doTag method has no way of accessing the tag body output. All it can do is pass it along.