

1. What is the primary purpose of using `SimpleTagSupport` in JSP custom tags?
- A. To simplify database access within JSP pages
 - B. To provide a helper class for creating tag handlers with the Simple Tag model
 - C. To support complex iteration and body evaluation in Classic Tag handlers
 - D. To enable direct Java code execution within JSP without scripting elements

Answer: B

2. Which method must be overridden when extending `SimpleTagSupport` to create a custom tag?
- A. `doStartTag()`
 - B. `doTag()`
 - C. `doEndTag()`
 - D. `doInitBody()`

Answer: B

3. How do you access the body content of a custom tag when using `SimpleTagSupport`?
- A. Using `getBodyContent().getString()`
 - B. Using `getJspBody().invoke(null)`
 - C. Using `pageContext.getAttribute("body")`
 - D. Body content is automatically processed and cannot be accessed directly

Answer: B

4. Which object is used within a `SimpleTagSupport` subclass to write output directly to the JSP response?
- A. `HttpServletResponse`
 - B. `Writer` obtained via `getJspContext().getOut()`
 - C. `PrintWriter` from `response.getWriter()`
 - D. `ServletOutputStream`

Answer: B

5. In `SimpleTagSupport`, how can you pass data from the tag handler to the JSP page?
- A. Using `pageContext.setAttribute()`
 - B. Using `getJspContext().setAttribute()`
 - C. Storing data in instance variables of the tag handler
 - D. Using `response.addHeader()`

Answer: B