

## 1. What is JSP?

- JSP (JavaServer Pages) is a technology used to create dynamic web pages.
  - It is an extension of Servlets that allows embedding Java code within HTML.
  - JSP files have the extension .jsp.
- 

## 2. JSP Life Cycle

1. **Translation:** JSP is converted into a Servlet.
2. **Compilation:** The Servlet is compiled into a .class file.
3. **Initialization:** `jspInit()` is called once.
4. **Execution:** `jspService()` handles requests.
5. **Destruction:** `jspDestroy()` is called before unloading.

*What is a Scriptlet Tag?*

- The Scriptlet Tag (`<% %>`) is used to write Java code inside JSP.
- Code written inside a scriptlet is placed inside the `service()` method of the generated servlet.

### Example 1:

```
<%@ page language="java" contentType="text/html" %>

<html> <body> <% out.println("Hello from Scriptlet!"); %>

</body> </html>
```

### Example 2:

```
<%@ page language="java" contentType="text/html" %>

<html> <body>

<% int a = 10, b = 20; int sum = a + b; %>

<p>Sum = <%= sum %></p>

</body> </html>
```

Can write Java variables, loops, conditions, and method calls.

- ✓ Executes every time the JSP page is requested.
- ✓ Can modify HTML dynamically.

### *1. What is a Declarative Tag?*

- The Declarative Tag (<%! %>) is used to declare methods, variables, and constants in JSP.
- The declared variables and methods are initialized only once, when the JSP page is loaded.
- Unlike scriptlets, declarative code is outside the service() method of the servlet.

**Example:**

```
<%@ page language="java" contentType="text/html" %>
```

```
<html> <body>
```

```
<%! int count = 0; // Declared only once when JSP loads %>
```

```
<p>Count: <%= count %></p>
```

```
</body> </html>
```

- ✓ Used for declaring variables and methods.
- ✓ Variables retain their values across multiple requests.
- ✓ Runs only once when JSP is initialized.

### *1. What is an Expression Tag?*

- The Expression Tag (<%= %>) is used to print values (variables, expressions, or method results) directly into the JSP page.
- It is evaluated and sent as part of the response to the client.

**Example :**

```
<%@ page language="java" contentType="text/html" %>
```

```
<html> <body> <p>Message: <%= "Hello, JSP!" %></p>
```

```
</body> </html>
```

**Example:**

```
<html> <body>
```

```
<%! int multiply(int a, int b) { return a * b; } %>
```

```
<p>Multiplication Result: <%= multiply(5, 4) %></p>
```

```
</body> </html>
```

### ***What are JSP Implicit Objects?***

- **JSP provides 9 built-in objects, called implicit objects, which can be used without explicit declaration.**
- **These objects allow interaction with the request, response, session, application, etc.**

#### ***. List of JSP Implicit Objects***

<b>Object</b>	<b>Description</b>
<b>request</b>	<b>Represents HTTP request (Used to get form data).</b>
<b>response</b>	<b>Represents HTTP response (Used to send data to client).</b>
<b>session</b>	<b>Stores user data across multiple requests.</b>
<b>application</b>	<b>Stores global data accessible to all users.</b>
<b>out</b>	<b>Prints data to the client (like System.out.print).</b>
<b>config</b>	<b>Provides configuration info about the servlet.</b>
<b>pageContext</b>	<b>Provides access to request, session, application objects.</b>
<b>page</b>	<b>Refers to the current JSP page (like this in Java).</b>
<b>exception</b>	<b>Used to handle exceptions (only available in error pages).</b>

**JSP Implicit Objects simplify web development by providing pre-defined access to request, response, session, etc.**

**They eliminate the need for explicit object creation.**

**Use them wisely to manage data efficiently in JSP applications.**

### ***What is HttpSession?***

- **HttpSession is an interface in Java used to maintain user session data across multiple requests.**
- **It allows storing user-specific information (e.g., login details, shopping cart) between multiple pages.**