

**ĐẠI HỌC KHOA HỌC TỰ NHIÊN  
KHOA CÔNG NGHỆ THÔNG TIN**

# **JAVASERVER PAGE**

**Nguyễn Hoàng Anh**  
**[nhanh@fit.hcmus.edu.vn](mailto:nhanh@fit.hcmus.edu.vn)**

# Nội dung

- Giới thiệu JSP
- Các thành phần cơ bản trong JPS

# GIỚI THIỆU JSP

# Servlet

- Thuận lợi
  - Đọc dữ liệu từ Form
  - Đọc các HTTP Request Header
  - Gán HTTP Status Code và Response Header
  - Sử dụng Cookie và Session
  - Chia sẻ dữ liệu giữa các Servlet
  - Xử lý cơ sở dữ liệu, ...
- Bất lợi
  - Sử dụng câu lệnh println để phát sinh HTML

# Servlet

- Bạn có nghĩ bạn sẽ tạo được giao diện web đẹp với Servlet?
- Bạn có thể đọc được một trang web được tạo ra với Servlet?
- Bạn sẽ bảo trì các giao diện web này như thế nào?

→ Servlet gây khó khăn rất lớn cho lập trình viên khi phải tạo ra giao diện web.

# Servlet

- Servlet rất mạnh về xử lý và điều phối, nhưng Servlet lại rất yếu về tạo giao diện
- Khắc phục điểm yếu của Servlet như thế nào?

→ Sử dụng công nghệ JSP

# JSP

- Ý tưởng
    - Thiết kế các trang web sử dụng **HTML** chuẩn
    - **Vị trí nào cần tạo ra nội dung động chỉ cần chèn các thẻ Java vào bên trong HTML.**
    - Toàn bộ trang JSP được thông dịch sang Servlet (một lần) và Servlet được thực thi khi yêu cầu của client gửi đến
- Dễ dàng hơn rất nhiều cho lập trình viên

# JSP

```
1  <%@page contentType="text/html" pageEncoding="UTF-8"%>
2  <html>
3      <head>
4          <meta http-equiv="Content-Type"
5              content="text/html; charset=UTF-8">
6          <title>JSP Page</title>
7      </head>
8      <body>
9          <h1> Xin chào:
10             <%=request.getParameter("tenNguoiDung") %>
11          </h1>
12      </body>
13 </html>
```



# JSP

- Thuận lợi
  - Thuận tiện khi tạo ra trang web HTML
  - Dễ đọc và dễ bảo trì trang web HTML
- Có nhiều công cụ hỗ trợ thiết kế HTML
  - Dream Weaver
  - Adode GoLive,...
- Phân cách thiết kế web và xử lý mã nguồn java
  - Đội ngũ thiết kế HTML chuyên nghiệp hơn lập trình viên java

# JSP

- Thuận lợi hơn Servlet
    - Thuận tiện trong việc tạo ra trang web HTML
    - Sử dụng các công cụ thiết kế như DreamWeaver
    - Phân cách xử lý và giao diện
  - JSP ra đời để thay thế Servlet?
- Lập trình viên JSP rất cần kiến thức về lập trình web với Servlet

# Xây dựng web với java

- Servlet mạnh về xử lý nghiệp vụ và điều phối nhưng lại rất yếu về hiển thị
- JSP mạnh về xử lý hiển thị nhưng lại yếu về xử lý nghiệp vụ và điều phối
- Vậy chọn công nghệ nào? JSP hay Servlet? Tại sao?

# Xây dựng web với java

- Thường thì chúng ta sẽ **kết hợp** sức mạnh của **Servlet** và **JSP** vào mô hình **MVC** (Model-View-Controller)
  - Các **Servlet** đóng vai trò làm **Controller**
  - Các trang **JSP** đóng vai trò làm **View**

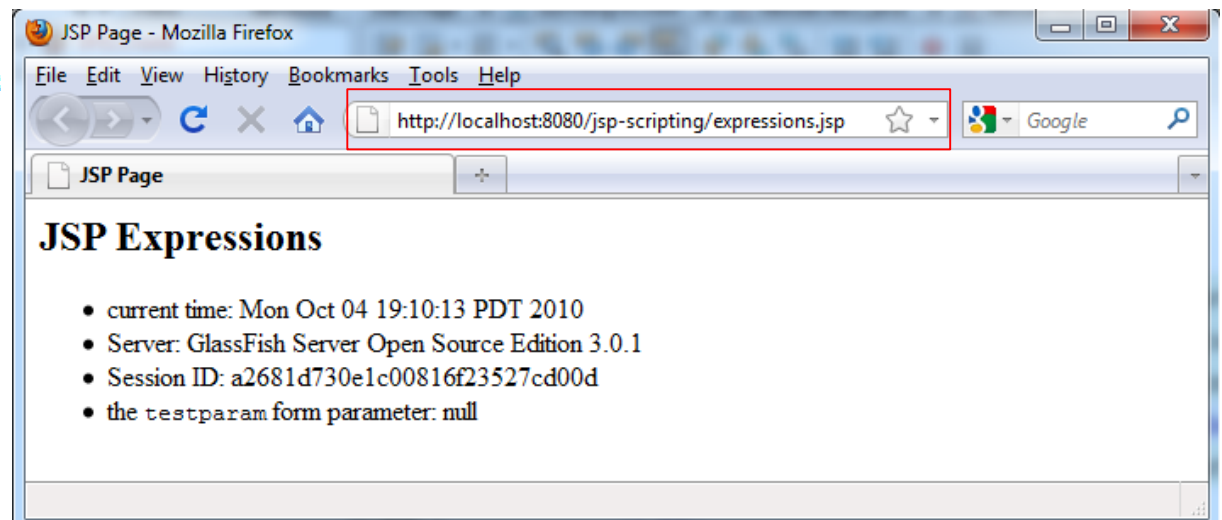
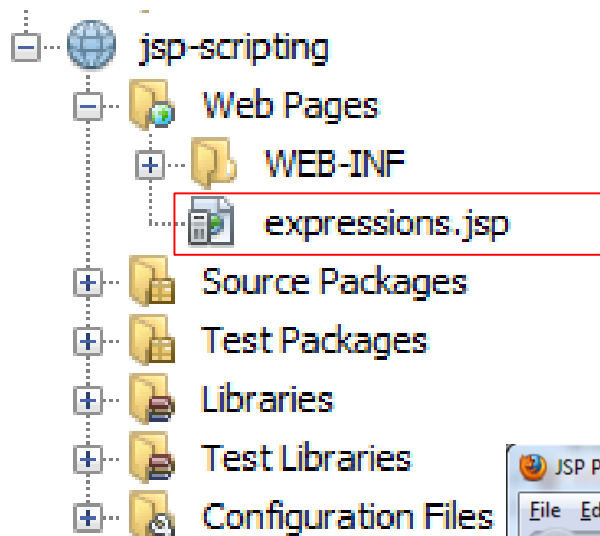
# Thiết lập môi trường khi sử dụng JSP

- Thiết lập CLASSPATH: **Không**
- Biên dịch: **Không**
- Sử dụng các Packaged để tránh đụng độ tên: **Không**
- Trang JSP phải để vào thư mục đặc biệt: **Không**
  - Để vào Web Pages trong netbeans, giống với HTML, CSS, JavaScript, GIF, JPEG, ...
- Sử dụng các URL đặc biệt cho trang JSP: **Không**
  - Giống như URL của trang HTML (ngoại trừ phần mở rộng của trang)

# JSP

```
1  <%@page contentType="text/html" pageEncoding="UTF-8"%>
2  <html>
3      <head>
4          <meta http-equiv="Content-Type"
5              content="text/html; charset=UTF-8">
6          <title>JSP Page</title>
7      </head>
8      <body>
9          <h2>JSP Expressions</h2>
10         <ul>
11             <li>current time: <%= new java.util.Date() %>
12             <li>Server: <%= application.getServerInfo() %>
13             <li>Session ID: <%= session.getId() %>
14             <li>the <code>testparam</code> form parameter:
15                 <%= request.getParameter("testparam") %>
16         </ul>
17     </body>
18 </html>
```

# JSP



# Cú pháp cơ bản

Các thẻ HTML	
Mô tả	Khi chuyển từ server đến client, nội dung HTML không thay đổi
Ví dụ	<code>&lt;h1&gt; Xin chào &lt;/h1&gt;</code>
Ghi chú trong HTML	
Mô tả	Thẻ ghi chú được gửi đến client nhưng không được hiển thị
Ví dụ	<code>&lt;!-- Nội dung ghi chú --&gt;</code>
Ghi chú trong JSP	
Mô tả	Thẻ ghi chú của lập trình viên không được gửi đến client
Ví dụ	<code>&lt;%-- Nội dung ghi chú --&gt;</code>



# Cú pháp cơ bản

JSP Expression	
Mô tả	Biểu thức được tính toán và gửi kết quả về khi client request
Ví dụ	<code>&lt;%= Giá trị %&gt;</code>
JSP Scriptlet	
Mô tả	Thực thi các câu lệnh khi client request
Ví dụ	<code>&lt;% Câu lệnh hoặc các câu lệnh%&gt;</code>
JSP Declaration	
Mô tả	Khai báo thuộc tính và định nghĩa phương thức trong trang
Ví dụ	<code>&lt;%! Khai báo thuộc tính %&gt;</code> <code>&lt;%! Định nghĩa phương thức %&gt;</code>

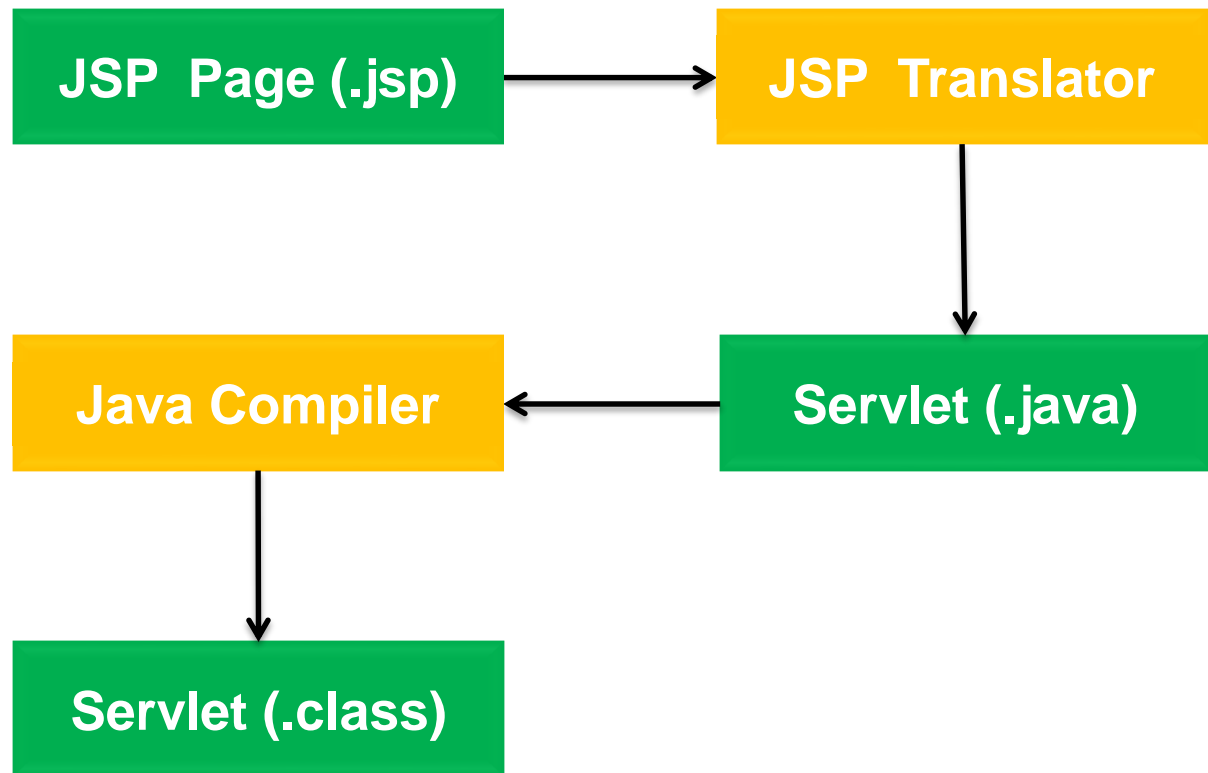
# Cú pháp cơ bản

JSP Directive	
Mô tả	<ul style="list-style-type: none"><li>- Thông tin về cấu trúc trang : <b>page</b></li><li>- Chèn trang khác vào : <b>include</b></li><li>- Các thẻ do người dùng định nghĩa : <b>taglib</b></li></ul>
Ví dụ	<b>&lt;%@ directive att="val" %&gt;</b>
JSP Action	
Mô tả	Thực thi Action khi client request
Ví dụ	<b>&lt;jsp:forward page="success.jsp" /&gt;</b>
JSP Expression Language Element	
Mô tả	Viết tắt của JSP Expression
Ví dụ	<b>\${biểu thức}</b>

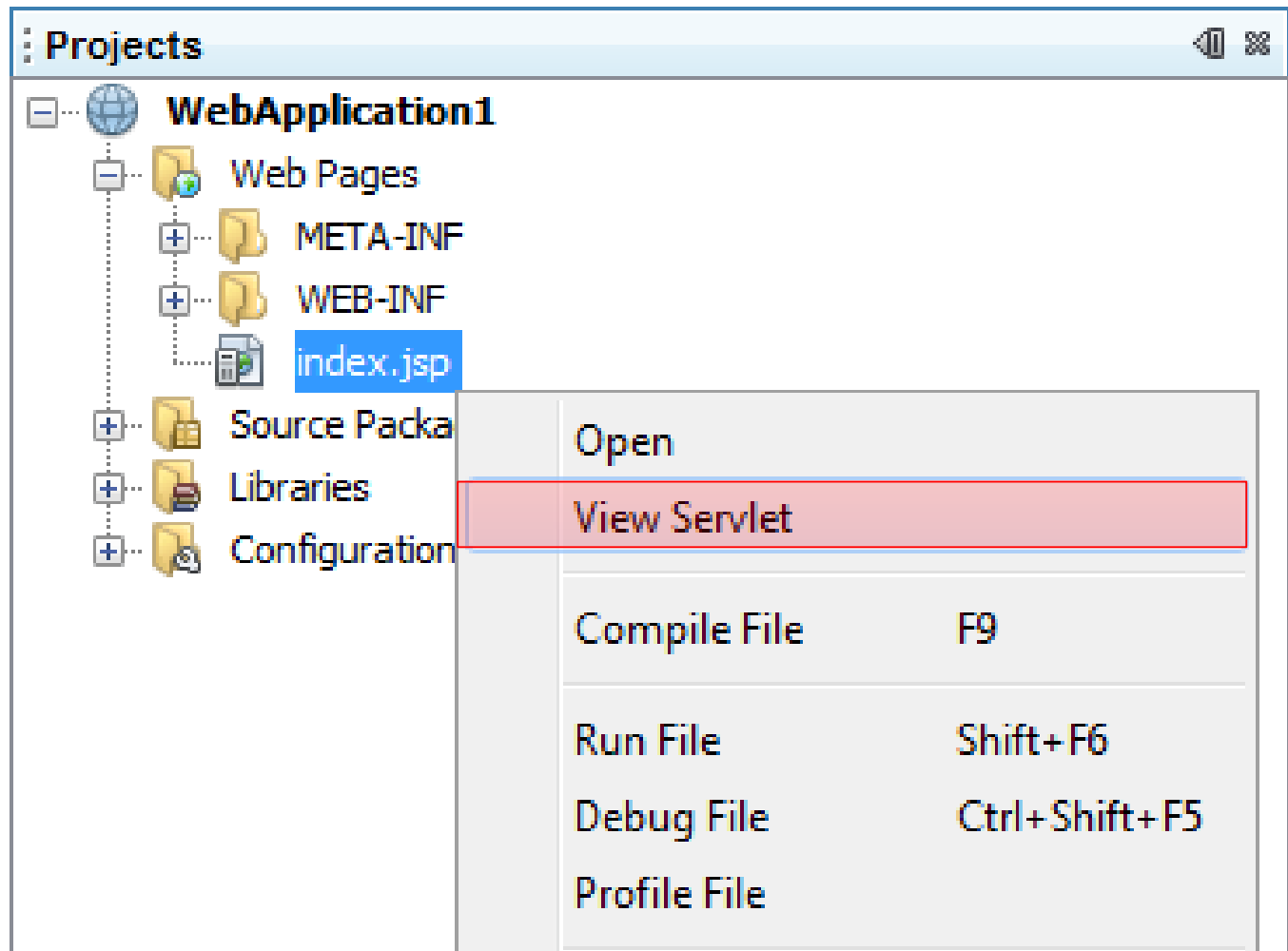
# Cú pháp cơ bản

Custom Tag	
Mô tả	Thẻ do người dùng định nghĩa
Ví dụ	<pre>&lt;prefix:name&gt;     . . . &lt;/prefix:name&gt;</pre>

# JSP là Servlet



# JSP là Servlet



# index.jsp

```
7  <%@page contentType="text/html" pageEncoding="UTF-8"%>
8  <!DOCTYPE html>
9  <html>
10     <head>
11         <meta http-equiv="Content-Type" content="text/html;
12             charset=UTF-8">
13         <title>JSP Page</title>
14     </head>
15     <body>
16         <h1>Hello World!</h1>
17         <%=new java.util.Date()%>
18     </body>
19 </html>
```

# index\_jsp.java - \_jspService

```
1 public final class index_jsp extends HttpJspBase, JspSourceDependent {  
2  
3     public void _jspService(final HttpServletRequest request,  
4                             final HttpServletResponse response)  
5         throws IOException, ServletException {  
6         final PageContext pageContext;  
7         HttpSession session = null;  
8         final ServletContext application;  
9         final ServletConfig config;  
10        JspWriter out = null;  
11        final java.lang.Object page = this;  
12        JspWriter _jspx_out = null;  
13        PageContext _jspx_page_context = null;  
14        ...  
15    }  
16 }  
17
```

## index\_jsp.java - \_jspService

```
1    ...
2    response.setContentType("text/html;charset=UTF-8");
3    pageContext = _jspxFactory.getPageContext(this, request, response,
4                                              null, true, 8192, true);
5    _jspx_page_context = pageContext;
6    application = pageContext.getServletContext();
7    config = pageContext.getServletConfig();
8    session = pageContext.getSession();
9    out = pageContext.getOut();
10   ...
11
```



# index\_jsp.java - \_jspService

```
54 out.write("\n");
55 out.write("\n");
56 out.write("\n");
57 out.write("<!DOCTYPE html>\n");
58 out.write("<html>\n");
59 out.write("    <head>\n");
60 out.write("        <meta http-equiv=\"Content-Type\" content=\"text/html;\n");
61 out.write("            charset=UTF-8\">\n");
62 out.write("        <title>JSP Page</title>\n");
63 out.write("    </head>\n");
64 out.write("    <body>\n");
65 out.write("        <h1>Hello World!</h1>\n");
66 out.write("    ");
67 out.print(new java.util.Date());
68 out.write("\n");
69 out.write("    </body>\n");
70 out.write("</html>\n");
```

# Các đối tượng được tạo sẵn trong trang JSP

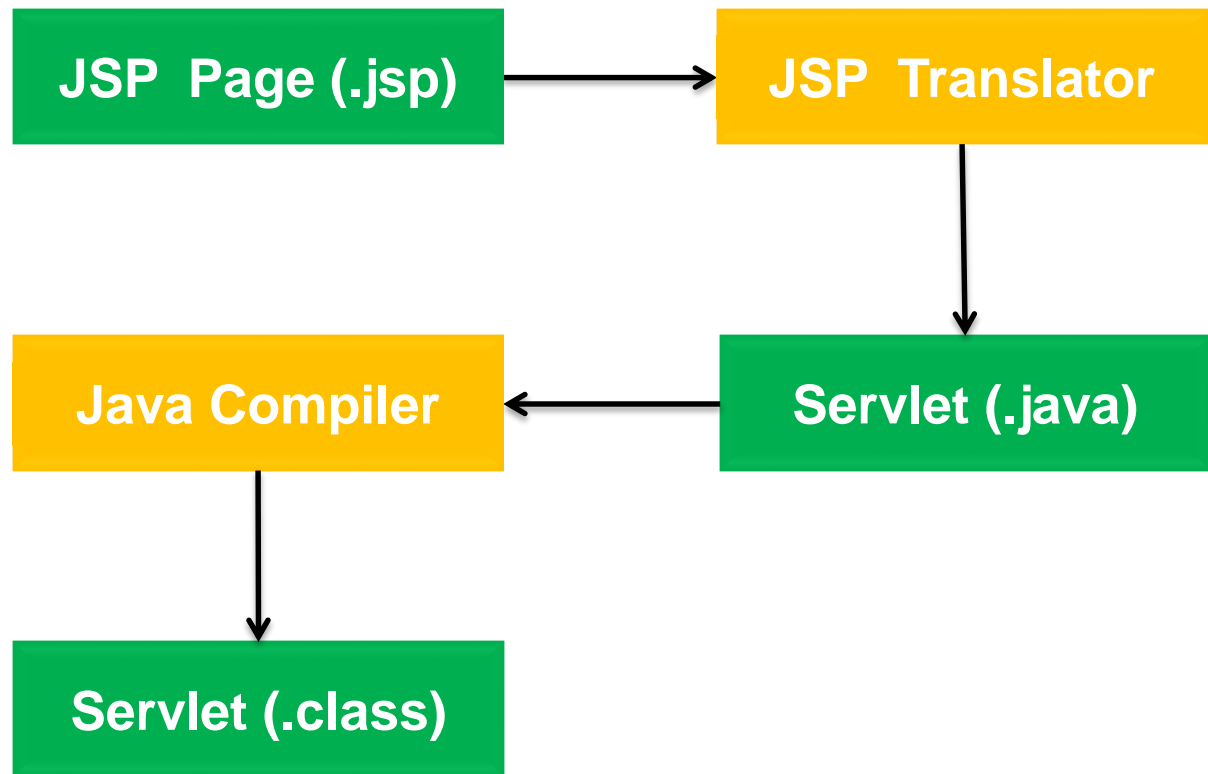
- **application** : ServletContext
- **config** : ServletConfig
- **exception** : Throwable
- **out** : JspWriter
- **page** : Object
- **PageContext** : PageContext
- **request** : HttpServletRequest
- **response** : HttpServletResponse
- **session** : HttpSession

# JSP SCRIPTING ELEMENTS

# JSP SCRIPTING ELEMENTS

- JSP Expression
- JSP Scriptlet
- JSP Declaration

# JSP là Servlet



# JSP Scripting Element

- Cho phép chèn các đoạn mã nguồn java vào bên trong trang JSP
- Khi trang JSP thông dịch các đoạn mã nguồn này sẽ được chèn vào bên trong phương thức `_jspService` của `Servlet`.
- Có 3 dạng
  - JSP Expression : `<%= Java Expression %>`
  - JSP Scriptlet : `<% Java Code %>`
  - JSP Declaration : `<%! Declarations %>`

# JSP Expression

- Định dạng:

**JSP : `<%= Java Expression %>`**

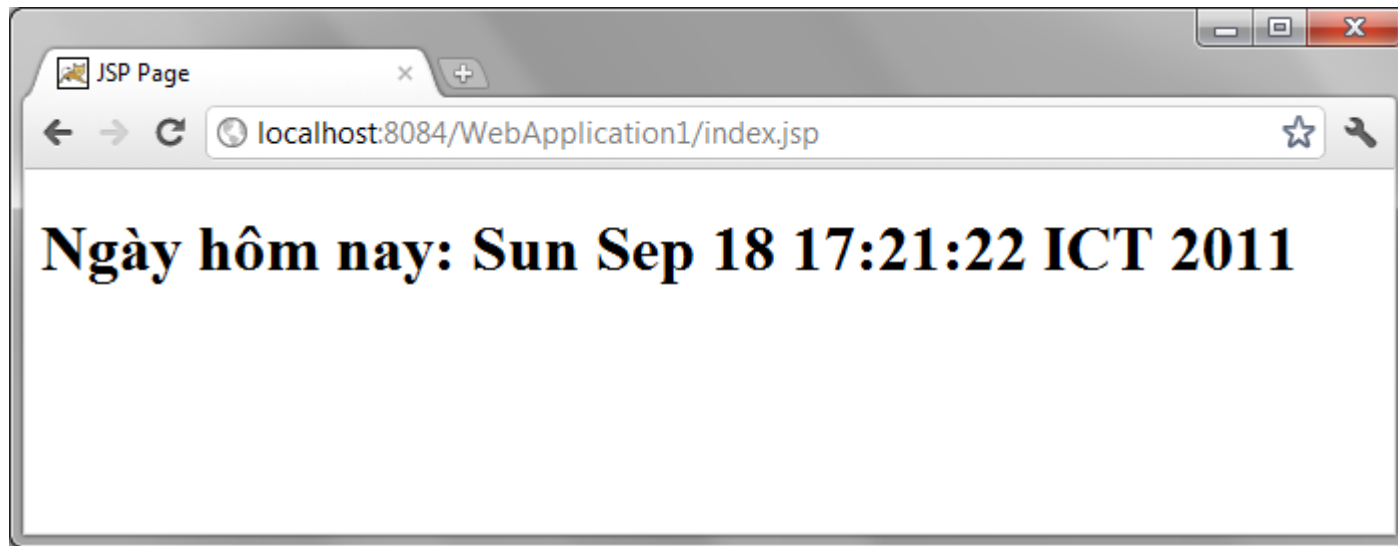
- **Lưu ý:-** Không được phép sử dụng **dấu ;** trong các Expression

# JSP Expression

- Kết quả
  - Expression sau tính toán ra **kết quả** sẽ được **chuyển thành một String**
  - **String** được chèn trực tiếp vào bên trong Output Stream của Servlet.
  - Kết quả tương tự như:  
**`out.println(Expression) ;`**
  - Trong **Expression** có thể sử dụng các biến:
    - Các biến được **định nghĩa tường minh**
    - Các đối tượng **được tạo sẵn ngầm định**



# Sử dụng JSP Expression



# Sử dụng JSP Expression

```
7  <%@page contentType="text/html" pageEncoding="UTF-8"%>
8  <!DOCTYPE html>
9  <html>
10     <head>
11         <meta http-equiv="Content-Type" content="text/html;
12             charset=UTF-8">
13         <title>JSP Page</title>
14     </head>
15     <body>
16         <h1>Ngày hôm nay: <%=new java.util.Date()%> </h1>
17     </body>
18 </html>
```

# JSP Expression

```
33 response.setContentType("text/html;charset=UTF-8");
34 PrintWriter out = response.getWriter();
35 out.println("<!DOCTYPE html>");
36 out.println("<html>");
37 out.println("    <head>");
38 out.println("        <meta http-equiv='Content-Type' content='text/html;";
39 out.println("            charset=UTF-8'>");
40 out.println("        <title>JSP Page</title>");
41 out.println("    </head>");
42 out.println("    <body>");
43 out.println("        <h1>Ngày hôm nay:  "+ new java.util.Date() +" </h1>");
44 out.println("    </body>");
45 out.println("</html>");
```

# Sử dụng JSP Expression

```
7  <%@page contentType="text/html" pageEncoding="UTF-8"%>
8  <!DOCTYPE html>
9  <html>
10     <head>
11         <meta http-equiv="Content-Type" content="text/html;
12             charset=UTF-8">
13         <title>JSP Page</title>
14     </head>
15     <body>
16         <h1>Ngày hôm nay:
17             <%=new java.util.Date()%>
18         </h1>
19     </body>
20 </html>
```

# Sử dụng JSP Expression

```
33 response.setContentType("text/html;charset=UTF-8");
34 PrintWriter out = response.getWriter();
35 out.println("<!DOCTYPE html>");
36 out.println("<html>");
37 out.println("    <head>");
38 out.println("        <meta http-equiv='Content-Type'");
39 out.println("            charset=UTF-8'>");
40 out.println("        <title>JSP Page</title>");
41 out.println("    </head>");
42 out.println("    <body>");
43 out.println("        <h1>Ngày hôm nay:    ");
44 out.println(new java.util.Date());
45 out.println("        </h1>");
46 out.println("    </body>");
47 out.println("</html>");
```

# JSP Scriptlet

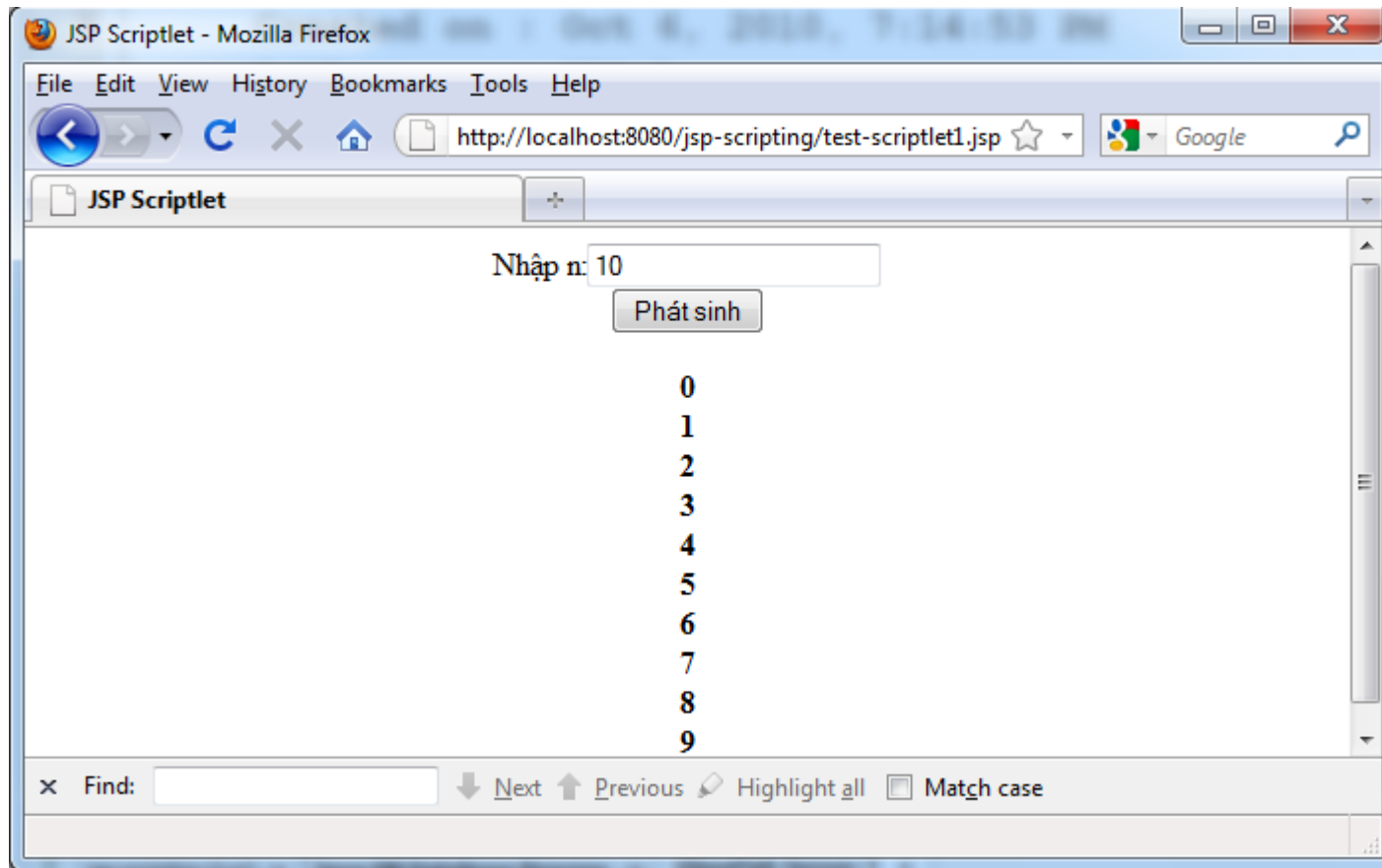
- Định dạng:

**JSP : `<% Java Code %>`**

# JSP Scriptlet

- Kết quả:
  - Sau khi trang JSP được thông dịch sang Servlet, **mã nguồn java** trong scriptlet **được chèn** tương ứng vào bên trong phương thức **\_jspService()**
- Trong **Scriptlet** có thể sử dụng các biến:
  - Các biến được **định nghĩa tường minh**
  - Các đối tượng **được tạo sẵn ngầm định**
- Trong scriptlet được phép khai báo biến, sử dụng các câu lệnh điều kiện, vòng lặp, gọi phương thức,...

# Sử dụng Scriptlet





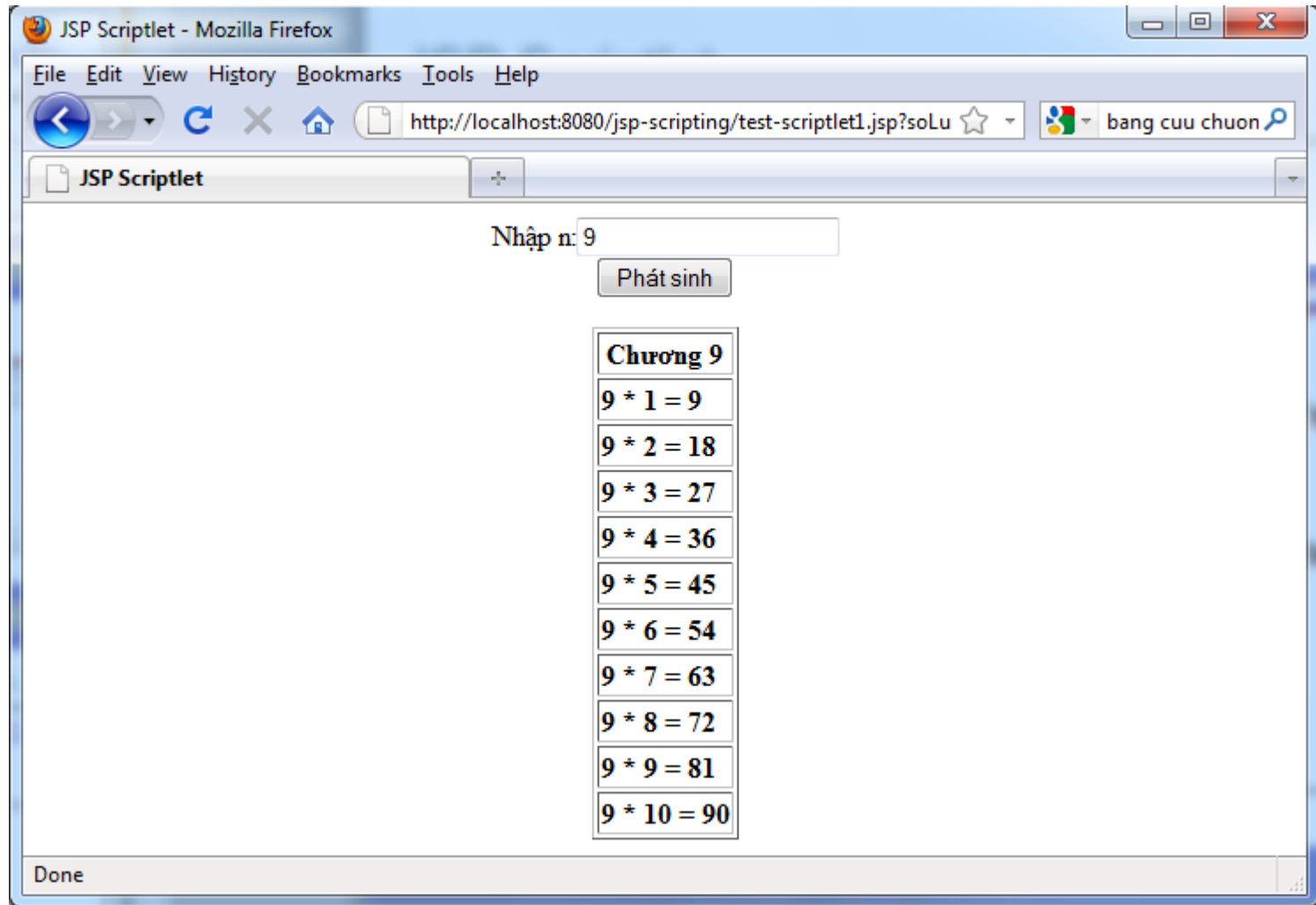
# Sử dụng Scriptlet

```
1 <body>
2   <center>
3     <form name="frm" method="get">
4       Nhập n:<input type="text" name="soLuong"/><br/>
5       <input type="submit" name="bt" value="Phát sinh"/>
6     </form>
7     <%
8       String s = request.getParameter("soLuong");
9       if (s != null) {
10        int n = Integer.parseInt(s);
11        for (int i = 0; i < n; i++) {
12          out.println("<b>" + i + "</b>");
13          out.println("<br/>");
14        }
15      }
16    %>
17   </center>
18 </body>
```

# Sử dụng Scriptlet

```
33 response.setContentType("text/html;charset=UTF-8");
34 PrintWriter out = response.getWriter();
35 out.println("<body>");
36 out.println("    <center>");
37 out.println("        <form name='frm' method='get'>");
38 out.println("            Nhập n:<input type='text' name='soLuong' /><br/>");
39 out.println("            <input type='submit' name='bt' value='Phát sinh' />");
40 out.println("        </form>");
41 String s = request.getParameter("soLuong");
42 if (s != null) {
43     int n = Integer.parseInt(s);
44     for (int i = 0; i < n; i++) {
45         out.println("<b> " + i + "</b>");
46         out.println("<br/>");
47     }
48 }
49 out.println("</center>");
50 out.println("</body>");
```

# Sử dụng Scriptlet + Expression + HTML



# Sử dụng Scriptlet + Expression + HTML

```
15 <body>
16 <form name="frm" method="get">
17     Nhập n:<input type="text" name="soLuong"/><br/>
18     <input type="submit" name="bt" value="Phát sinh"/>
19 </form>
20 <%
21     String s = request.getParameter("soLuong");
22     if (s != null) {
23         int n = Integer.parseInt(s);
24     }%>
25 <table border="1"><tr><th>Chương <%=n%></th></tr>
26 <%for (int i = 1; i <= 10; i++) {
27     %>
28     <tr><td><b>
29         <%= n + " * " + i + " = " + i * n%>
30     </b></td></tr>
31 <%}%>
32 </table>
33 <%}%>
34 </body>
```

# Sử dụng Scriptlet + Expression + HTML

```
33 response.setContentType("text/html;charset=UTF-8");
34 PrintWriter out = response.getWriter();
35 out.println("<body>");
36 out.println("        <form name='frm' method='get'>");
37 out.println("            Nhập n:<input type='text' name='soLuong'><br/>");
38 out.println("            <input type='submit' name='bt' value='Phát sinh'>");
39 out.println("        </form>");
40 String s = request.getParameter("soLuong");
41 if (s != null) {
42     int n = Integer.parseInt(s);
43     out.println("        <table border='1'><tr><th>Chương <%=n%></th></tr>");
44     for (int i = 1; i <= 10; i++) {
45         out.println("            <tr><td><b>");
46         out.println(n + " * " + i + " = " + i * n);
47         out.println("            </b></td></tr>");
48     }
49     out.println("        </table>");
50 }
51 out.println("    </body>");
```

# JSP Declaration

- Định dạng:

**JSP : <%! Khai báo các thuộc tính**

**Định nghĩa các phương thức %>**

# JSP Declaration

- Sau khi trang JSP được thông dịch thành Servlet thì các khai báo thuộc tính và định nghĩa phương thức được chèn vào bên trong Servlet.
- JSP Declaration được sử dụng với Scriptlet và Expression

# JSP Declaration

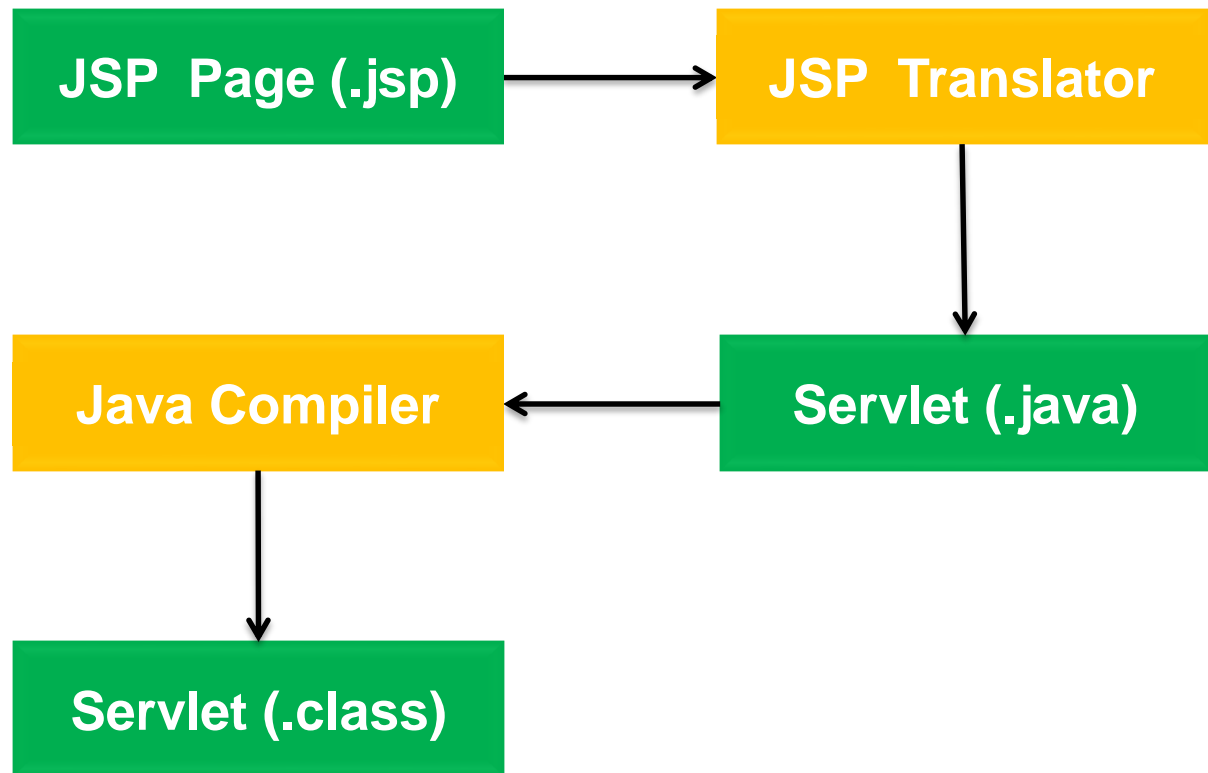
- JSP Declaration cho phép
  - Định nghĩa phương thức mới
  - Cài đặt lại các phương thức `jspInit()`, `jspDestroy`
  - Không được phép cài đặt lại phương thức `_jspService()`
- JSP Declaration không được phép sử dụng các đối tượng được định nghĩa ngầm định.



# Khai báo JSP Declaration

```
1  <%@page contentType="text/html" pageEncoding="UTF-8"%>
2  <%@page import="dao.*, pojo.*"%>
3  <%!
4      private ArrayList<DanhMuc> ds;
5      // Cài đặt lại phương thức jspInit
6      public void jspInit() {
7          this.ds = DanhMucDAO.layDanhSachDanhMuc();
8      }
9      //Cài đặt lại phương thức jspDestroy
10     public void jspDestroy() {
11         this.ds = null;
12     }
13     //Định nghĩa phương thức mới
14     public void xxx() {
15         . . .
16     }
17 %>
18 <html><head><title>DanhSachSach</title></head>
19     <body> ... </body>
</html>
```

# JSP là Servlet



# JSP Scripting Element

- Cho phép chèn các đoạn mã nguồn java vào bên trong trang JSP
- Khi trang JSP thông dịch các đoạn mã nguồn này sẽ được chèn vào bên trong phương thức `_jspService` của `Servlet`.
- Có 3 dạng
  - JSP Expression : `<%= Java Expression %>`
  - JSP Scriptlet : `<% Java Code %>`
  - JSP Declaration : `<%! Declarations %>`



# HỎI VÀ ĐÁP

# Reference

- [1] Marty Hall; Larry Brown; Yaakov Chaikin. (2007). Core Servlets and Javaserver Pages: Advanced Technologies, Vol. 2, 2<sup>nd</sup> Edition. Prentice Hall.
- [2] Bryan Basham; Kathy Sierra; Bert Bates. (2008). Head First Servlets and JSP, 2<sup>nd</sup> Edition. O'Reilly Media.