



# บทที่ 4

## JSP

### (Java Server Pages)

ธีระยุทธ ทองเครือ

ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์

มหาวิทยาลัยขอนแก่น



# JSP

- ❖ JSP คือ เทคโนโลยีที่ใช้ในการสร้าง Web Application เช่นเดียวกับ Servlet
- ❖ JSP จะประมวลผลบน Server โดยสามารถรับ request และส่ง response ได้ เช่นเดียวกับ Servlet
- ❖ JSP จะคล้ายคลึงกับภาษา PHP คือ เขียน HTML, CSS หรือ JS และแทรกส่วนที่ทำงานในแท็กเฉพาะ

## PHP

```
<html>
<body>
  <?php
    echo "Hello World";
  ?>
</body>
</html>
```

## JSP

```
<html>
<body>
  <%
    out.println( "Hello World");
  %>
</body>
</html>
```



# การแทรกคำสั่งภาษา Java ใน JSP



❖ ใช้สัญลักษณ์ที่เรียกว่า Scriptlet

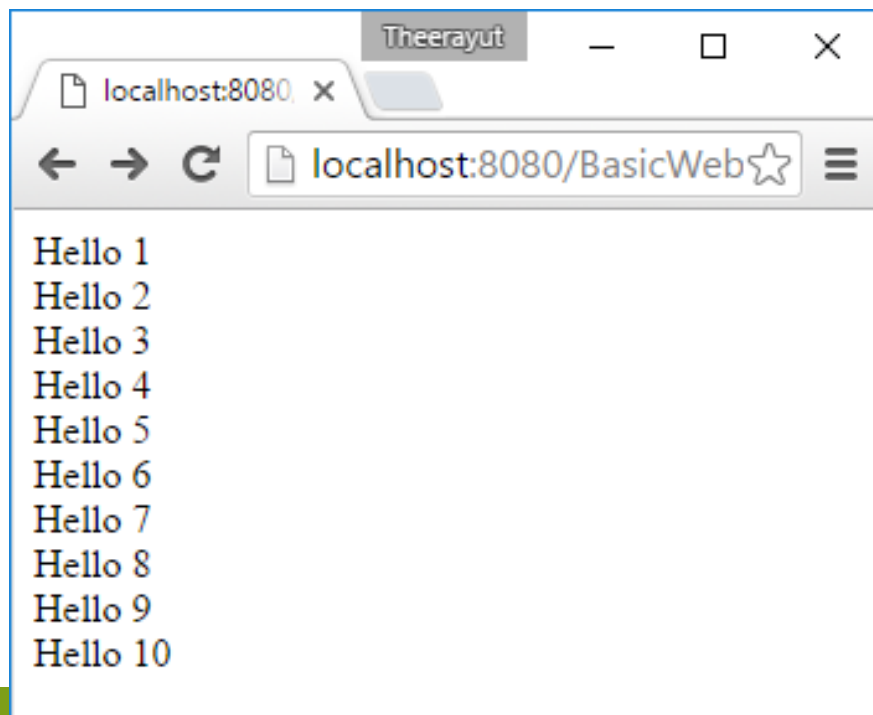
<%      โค้ดภาษา Java      %>

```
<html>
<body>
    <% int count = 0; %>
    The page count is now:
    <%
        out.println(count++);
    %>
</body>
</html>
```



# ตัวอย่าง

```
<html>
<body>
  <% for (int i=1; i<=10; i++) { %>
    Hello <%=i %> <br>
  <% } %>
</body>
</html>
```





# แท็ก Expression

❖ แท็ก Expression ใช้แทนคำสั่ง `out.println()` มีรูปแบบดังนี้

```
<%= expression %>
```

ไม่ใช่แท็ก Expression

ใช้แท็ก Expression

```
<html>
<body>
  <% int count = 0; %>
  The page count is now:
  <% out.println(count++); %>
</body>
</html>
```

```
<html>
<body>
  <% int count = 0; %>
  The page count is now:
  <%= count++ %>
</body>
</html>
```

!!! ข้อควรระวัง !!!  
แท็กนี้จะไม่ มี Semicolon





# Implicit object



- ❖ Object ที่ไฟล์ JSP สามารถเรียกใช้ได้เลย โดยไม่ต้องประกาศก่อน ได้แก่
  - out - ใช้สำหรับใส่ข้อมูล response ในส่วนที่จะส่งกลับไปแสดงผล
  - request – ใช้ดึงข้อมูลจาก HTTP request
  - response - ใช้เก็บข้อมูลผลลัพธ์ที่ส่งกลับ
  - session – ตัวแปร session

# ตัวอย่าง

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

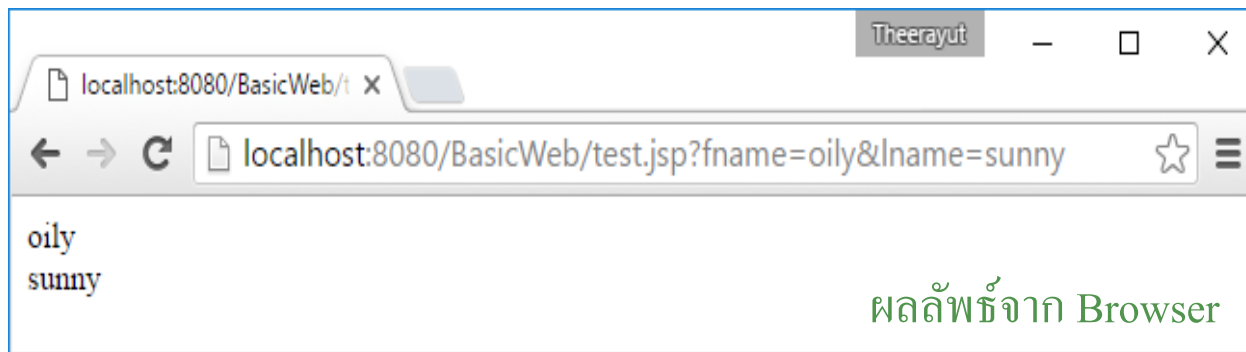
```
<%
```

```
String fname = request.getParameter("fname");  
out.println(fname + "<br>");
```

```
%>
```

```
<%= request.getParameter("lname") %>
```

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



```
<html><body>  
oily<br>  
sunny  
</body></html>
```

ผลลัพธ์ในรูปแบบโค้ด



# ภาษาไทยบน JSP

❖ ใช้คำสั่งดังนี้

```
<%@page contentType="text/html; charset=utf-8" pageEncoding="utf-8"%>
```

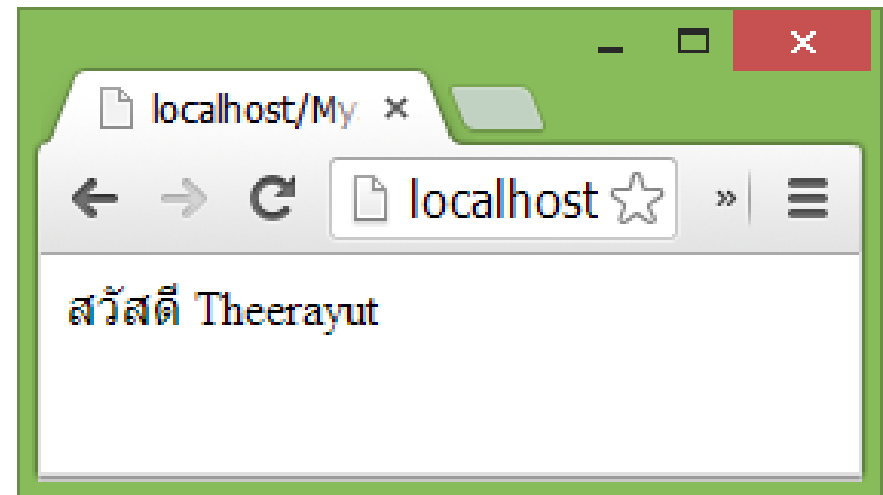
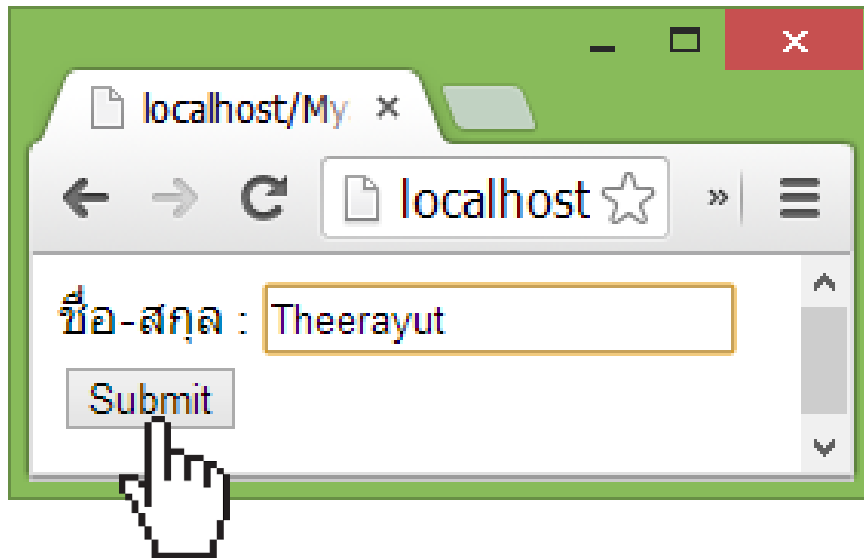
❖ ซึ่งเป็นคำสั่งเดียวกับใน Servlet

```
response.setCharacterEncoding("utf-8");  
response.setContentType("text/html; charset=utf-8");
```



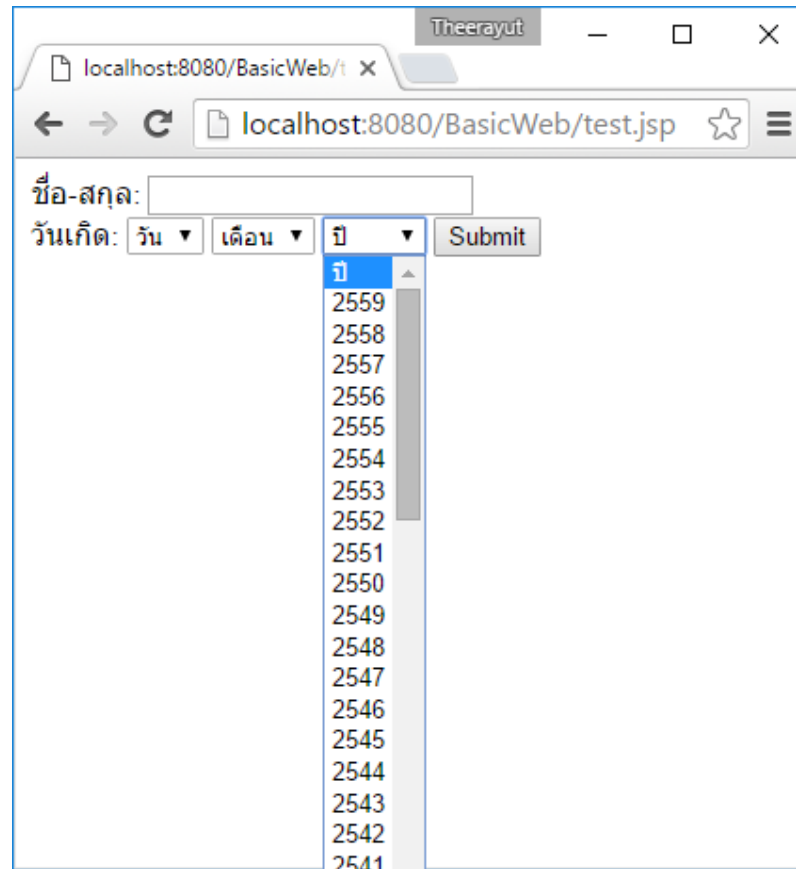
# กิจกรรม

- ❖ จงสร้างฟอร์ม HTML และส่งข้อมูลไปยังไฟล์ JSP อีกไฟล์หนึ่ง เพื่อแสดงผลดังตัวอย่าง



# กิจกรรม

- ❖ จงเปลี่ยนนามสกุลไฟล์ .html จากกิจกรรมก่อนหน้านี้เป็น .jsp และเพิ่มฟอร์มรับค่าวัน เดือนปีเกิด ที่เป็นแบบรายการให้ผู้ใช้เลือก โดยใช้วิธีการวนลูปด้วยคำสั่งภาษาจาวา ดังตัวอย่าง



The screenshot shows a web browser window with the address bar displaying 'localhost:8080/BasicWeb/test.jsp'. The page contains a form with the following elements:

- A text input field labeled 'ชื่อ-สกุล:'.
- A group of three dropdown menus for birth date, labeled 'วันเกิด:' with sub-labels 'วัน', 'เดือน', and 'ปี'.
- A 'Submit' button.

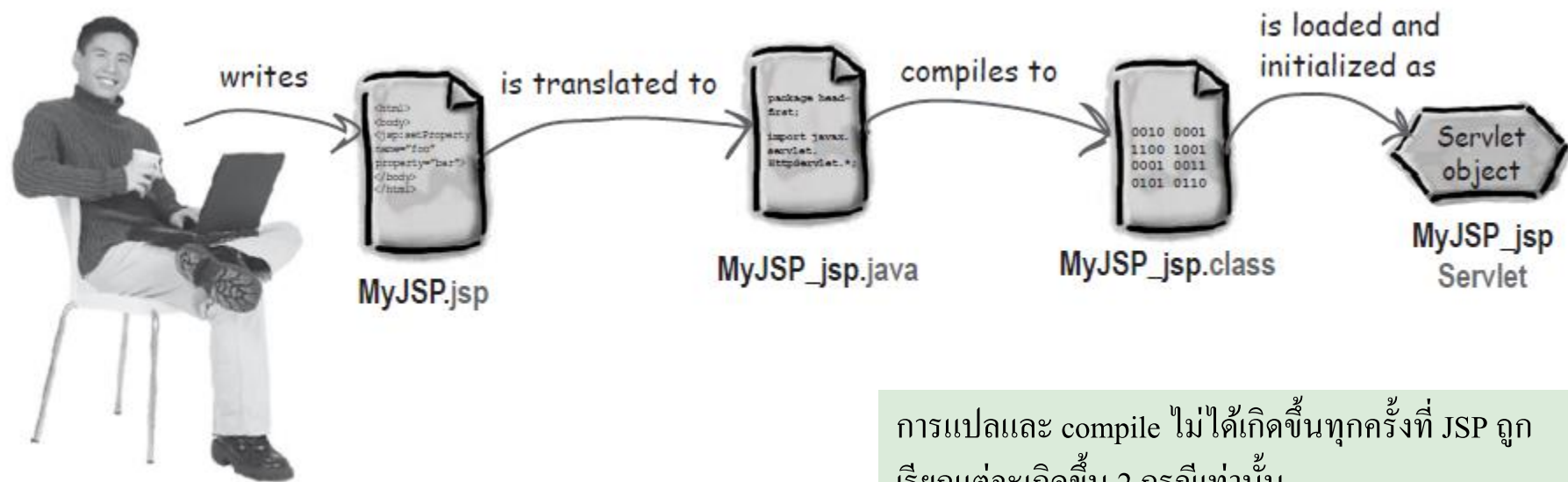
The 'ปี' (Year) dropdown menu is open, showing a list of years from 2541 to 2559. The year 2559 is currently selected.



# JSP จะถูกแปลเป็น Servlet



- ❖ JSP ที่สร้างขึ้น มีนามสกุลเป็น .jsp
- ❖ เมื่อไฟล์ JSP ถูกเรียกครั้งแรก Web Container จะแปลให้เป็น Servlet (.java) ก่อนเริ่มทำงาน หากมีแท็กต่างๆจะถูกแปลงเป็น content เก็บลงใน response



การแปลและ compile ไม่ได้เกิดขึ้นทุกครั้งที่ JSP ถูกเรียกแต่จะเกิดขึ้น 2 กรณีเท่านั้น

- ผู้ใช้เรียกไฟล์ JSP ครั้งแรก
- ไฟล์ JSP มีการแก้ไข



# JSP คือ Servlet

- ❖ โครงสร้างของ JSP จะมีความคล้ายคลึงกับ Servlet โดยมีเมธอด `service()` ที่ทำหน้าที่รับ request และส่ง response (JSP ใช้ชื่อเมธอด `_jspService`)

**This JSP:**

**Becomes this Servlet:**

```
public class basicCounter_jsp extends SomeSpecialHttpServlet {  
  
    public void _jspService(HttpServletRequest request,  
        HttpServletResponse response)throws java.io.IOException,  
        ServletException {  
  
        PrintWriter out = response.getWriter();  
        response.setContentType("text/html");  
        out.write("<html><body>");  
        int count=0;  
        out.write("The page count is now:");  
        out.print( ++count );  
        out.write("</body></html>");  
  
    }  
}
```

```
<html><body>  
<% int count=0; %>  
The page count is now:  
<%= ++count %>  
</body></html>
```

*The Container puts all the code into a generic service method. Think of it as a catch-all combo doGet/doPost.*

ไฟล์ JSP ที่ถูกแปลเป็น Servlet จะเก็บอยู่ใน



# การ import

- ❖ หาก JSP มีการเรียกใช้คลาสมาตรฐานของ Java หรือคลาสที่สร้างขึ้นเองจะต้อง import ในรูปแบบดังนี้

```
<%@page import="ชื่อ package และคลาส" %>
```

- ❖ ตัวอย่าง

```
<%@page import="java.util.*" %>
```

ใช้ \* เพื่อระบุว่าต้องการใช้  
ทุกคลาสใน package

```
<%@page import="java.util.*, mypack.*" %>
```

หากใช้คลาสจากหลาย  
package ให้คั่นด้วย comma

```
<%@page import="java.util.ArrayList, mypack.TestDB, mypack2.*" %>
```

สามารถระบุจำเพาะชื่อ  
คลาสที่ต้องการใช้หลังชื่อ  
package ได้

- ❖ เรียกแท็กชนิดนี้ว่า Directive





# การใช้ Comment



- ❖ Comment ใน JSP อยู่ในรูปแบบดังนี้

```
<%-- JSP Comment --%>
```

Comment แบบนี้จะไม่ถูกส่งไปกับ response (client ไม่เห็นใน code html)

- ❖ หากใช้ Comment ในรูปแบบภาษา html จะสามารถมองเห็นได้ที่ client

```
<!-- HTML Comment -->
```



# Include directive



- ❖ การแยกส่วนประกอบของเว็บเพจไว้คนละไฟล์จะช่วยให้ง่ายต่อการกลับมาแก้ไข
- ❖ สามารถใช้ Include directive เพื่อช่วยในการกำหนด Template ของเว็บไซต์ได้
- ❖ ไฟล์ html หรือ jsp ที่ต้องการนำมาประกอบเป็นส่วนหนึ่งของเว็บเพจจะใช้คำสั่งดังนี้

`<jsp:include page="path/ชื่อไฟล์ jsp หรือ html"/>`

## header.jsp

```
<div style="background-color: navy; color: white;">  
<h1>Header</h1>  
</div>
```

## footer.html

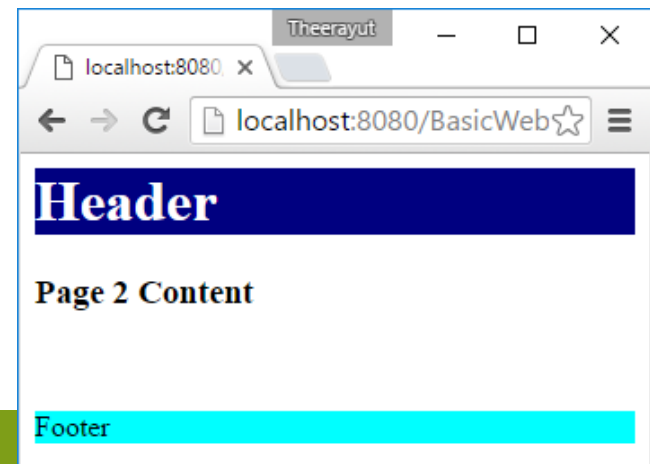
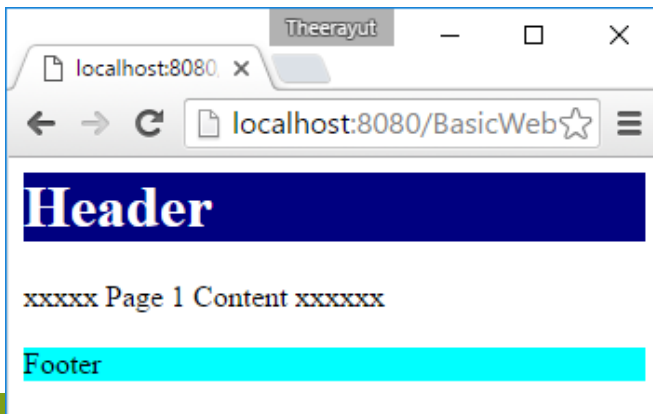
```
<br><br>  
<div style="background-color: aqua;">Footer</div>
```

## Page1.jsp

```
<html><body>  
<jsp:include page="header.jsp"/>  
  
xxxxx Page 1 Content xxxxxx  
  
<jsp:include page="footer.html"/>  
</body></html>
```

## Page2.jsp

```
<html><body>  
<jsp:include page="header.jsp"/>  
  
<h3>Page 2 Content</h3>  
  
<jsp:include page="footer.html"/>  
</body></html>
```





# Assignment#4

- ❖ นำโค้ดจาก Assignment#3 มาเพิ่มส่วนแก้ไขและลบสินค้าแต่ละชิ้น โดยใช้ JSP ในการทำงานแทนการใช้ Servlet กำหนดส่งวันพุธที่ 23 กุมภาพันธ์ 2560 ก่อนเวลา 23.00 น.

รหัส	ชื่อสินค้า	รายละเอียด	ราคา		
1	Centrum	วิตามินรวมจาก A ถึง Zinc	350	<a href="#">แก้ไข</a>	<a href="#">ลบ</a>
2	Caltrate	บำรุงกระดูก เสริมวิตามินดี	600	<a href="#">แก้ไข</a>	<a href="#">ลบ</a>
3	Ester-C	วิตามินซี 500 mg ไม่กัดกระเพาะ	1000	<a href="#">แก้ไข</a>	<a href="#">ลบ</a>
4	Glucosamine	บำรุงข้อต่อ ป้องกันข้อเสื่อม	1200	<a href="#">แก้ไข</a>	<a href="#">ลบ</a>
9	Vitamin B รวม	ป้องกันปลายประสาทอักเสบfdssffswdf	500	<a href="#">แก้ไข</a>	<a href="#">ลบ</a>

ชื่อสินค้า:

รายละเอียดสินค้า:

ราคา:

# เอกสารอ้างอิง

- ❖ Bryan Basham, Kathy Sierra, and Bert Bates. *Head First Servlets and JSP, 2nd Edition*, United States of America, O'Reilly Media, 2008.

