

# ใบงาน

รหัสวิชา 30901-2006

วิชา การพัฒนาเว็บด้วยเทคโนโลยีจาวา



ครูผู้สอน วรกิจ วิริยะเกษามงคล

แผนกวิชาเทคโนโลยีสารสนเทศ

วิทยาลัยเทคนิคชลบุรี

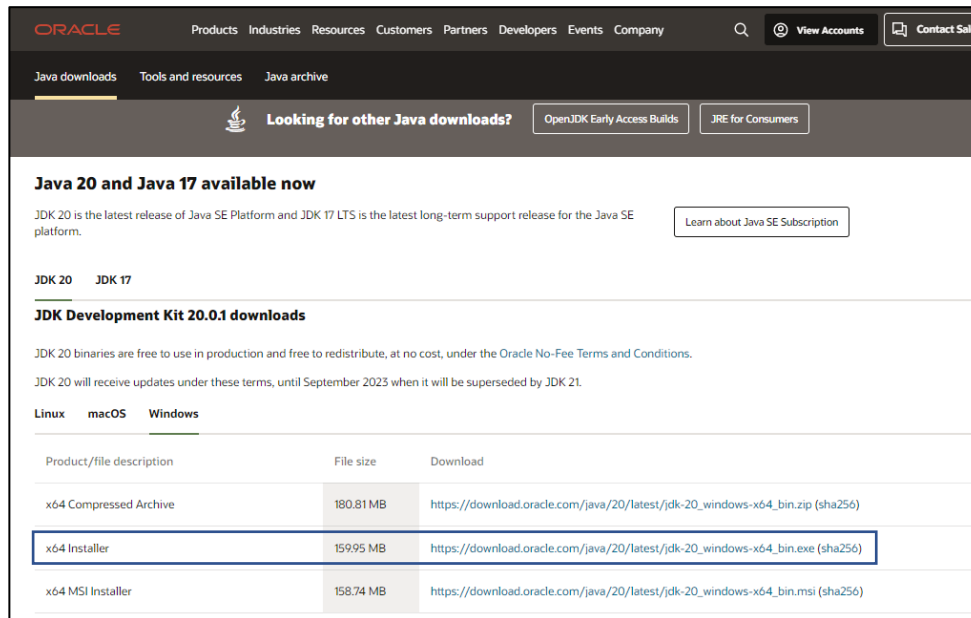
## ใบงาน

ชื่องาน ติดตั้งโปรแกรม JDK โปรแกรม Tomcat และ โปรแกรม Edit Plus

ชื่อวิชา การพัฒนาเว็บด้วยเทคโนโลยีจาวา รหัสวิชา 30901-2006

### ขั้นตอนการติดตั้งโปรแกรม JDK

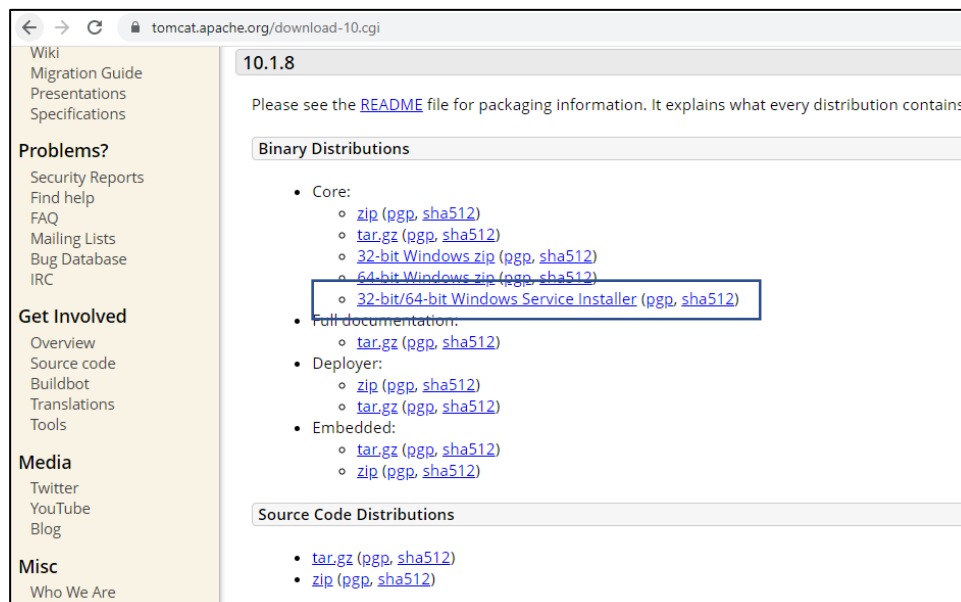
1. Download file program จากเว็บ <https://www.oracle.com/java/technologies/downloads/>



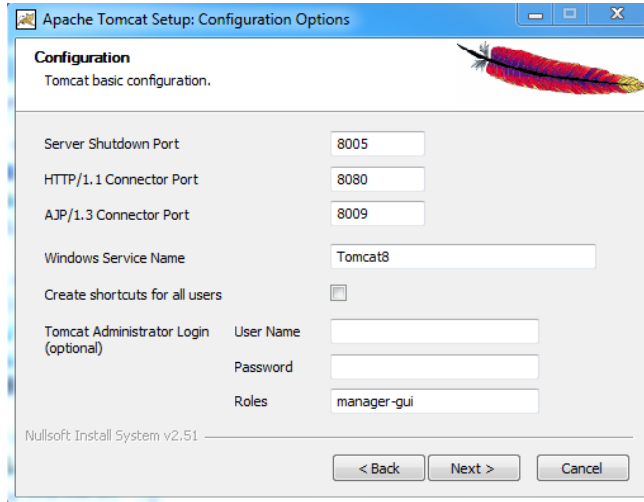
2. ทำการติดตั้งโดยไม่ต้องปรับแต่งอะไร

### ขั้นตอนการติดตั้งโปรแกรม Tomcat

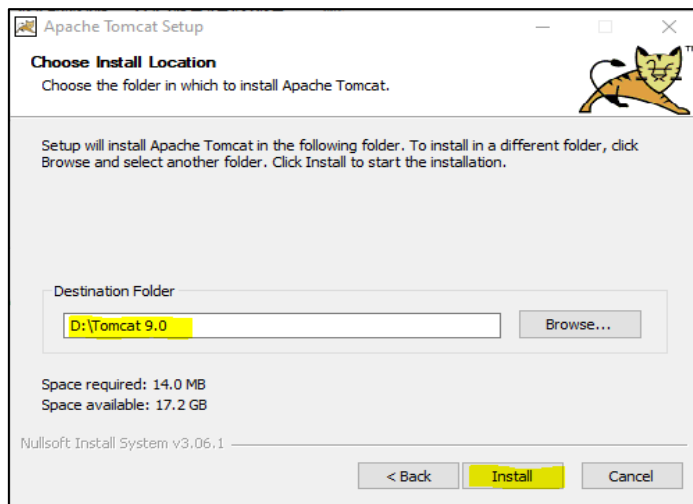
1. Download file program จากเว็บ <https://tomcat.apache.org/download-10.cgi>



2. ทำการติดตั้งโดยมีหน้าที่ต้องปรับแต่งคือ username และ password ไม่ต้องใส่

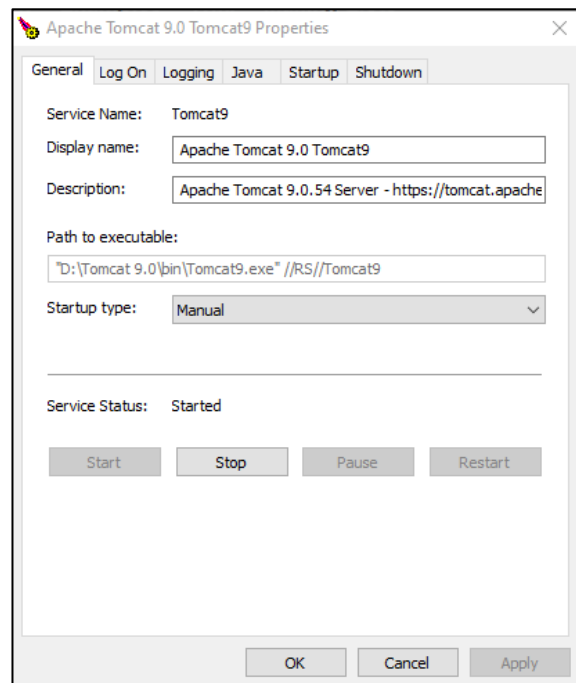
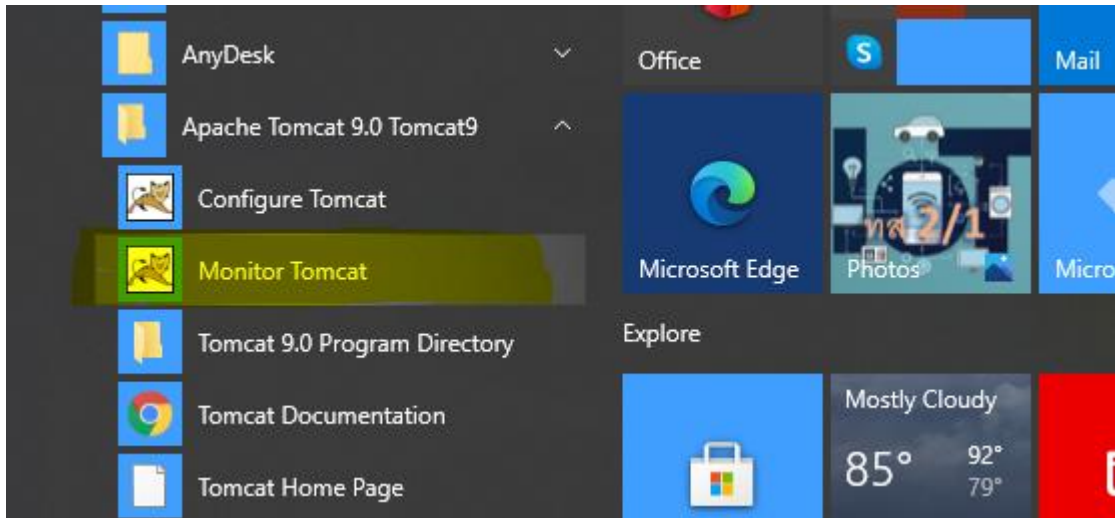


3. แก้ไขตำแหน่งการเก็บไฟล์ เนื่องจากโดย default จะบันทึกที่ c:/program File เมื่อทำการสร้างไฟล์ใหม่ แล้วทำการบันทึก ระบบ windows บางเครื่องมีการตั้งค่าความปลอดภัยของระบบไว้ จะทำการ save งานที่สร้างไม่ได้



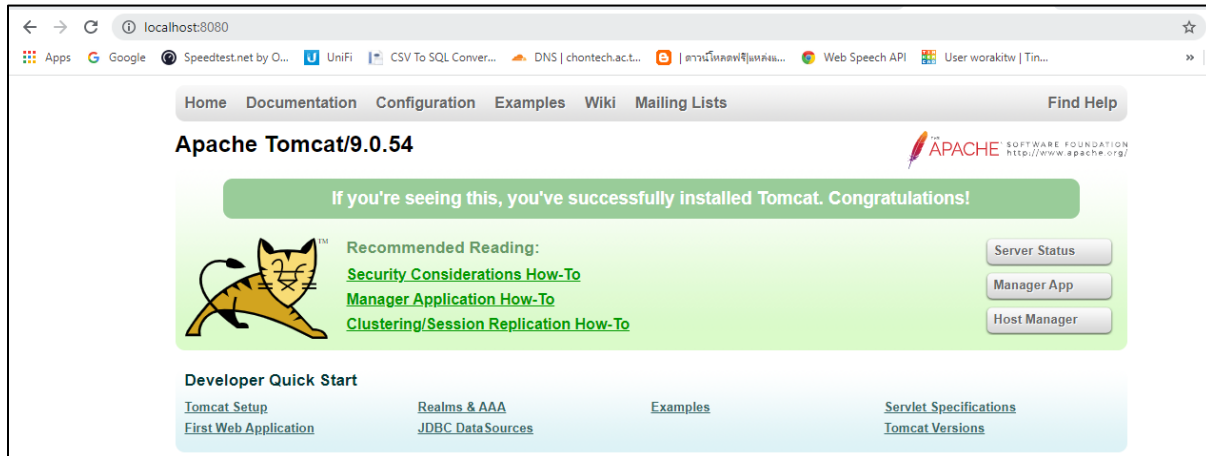
## วิธีการ RUN Tomcat

search => monitor tomcat



ทดสอบการทำงานเว็บจาวา ด้วย browser

<http://localhost:8080>



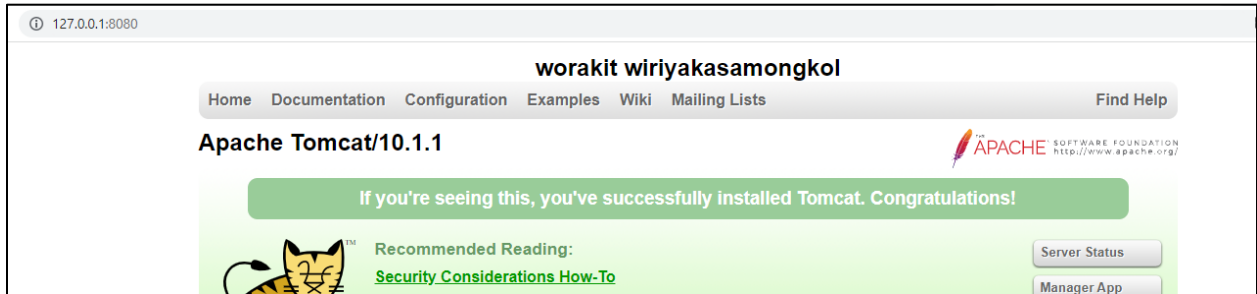
ทดสอบตำแหน่งการเก็บหน้าเว็บ

แก้ไขไฟล์ที่ D:/Tomcat\_10.1/webapp/ROOT/index.jsp

ใส่ชื่อภาษาอังกฤษใน tag body

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <title><%=request.getServletContext().getServerInfo() %></title>
  <link href="favicon.ico" rel="icon" type="image/x-icon" />
  <link href="tomcat.css" rel="stylesheet" type="text/css" />
</head>
<body>
  <h1>worakit wiriyakasamongkol</h1>
</body>
</html>
```

ผลลัพธ์



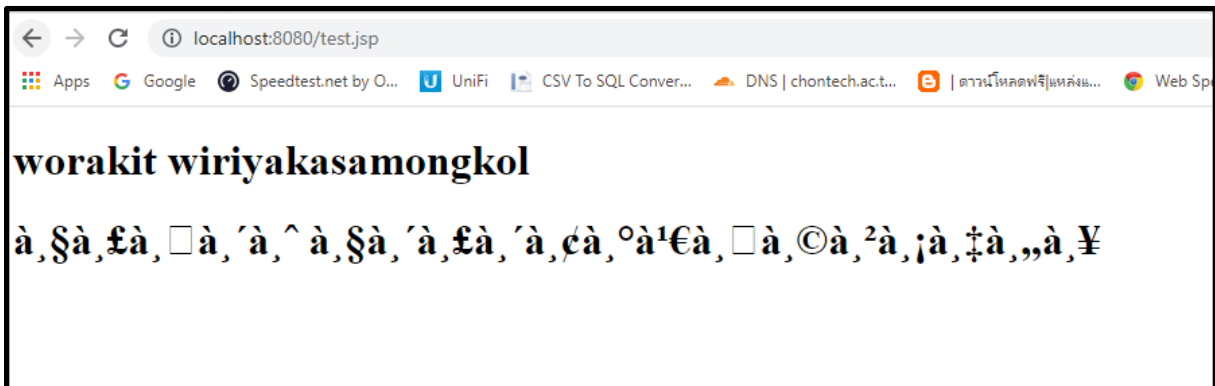
การแสดงผลภาษาไทย

แก้ไข code ที่ไฟล์ D:/Tomcat\_10.1/webapp/ROOT/index.jsp

บรรทัดที่ 2 ใส่ชื่อภาษาไทย

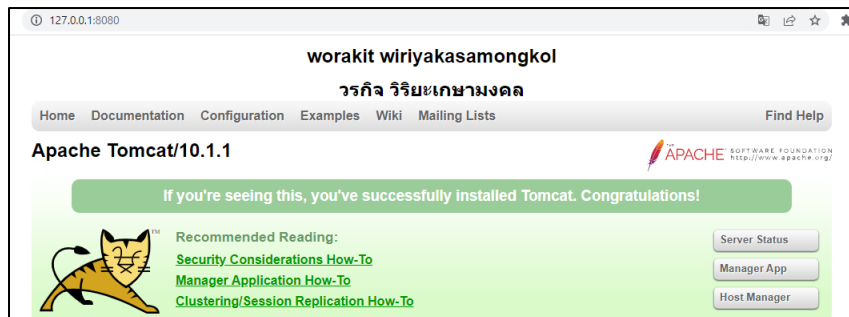
```
<body>
  <h1>worakit wiriyakasamongkol</h1>
  <h1>วรกิจ วิริยะเกษามงคล</h1>
</body>
```

ผลลัพธ์



แก้ไข code ให้สามารถแสดงภาษาไทยได้

ใส่ tag นี้บรรทัดบนสุด => `<%@page pageEncoding="UTF-8"%>`



## ใบงาน

ชื่องาน เขียนโปรแกรมใช้งาน servlet

ชื่อวิชา การพัฒนาเว็บด้วยเทคโนโลยีจาวา รหัสวิชา 30901-2006

---

### การตั้งค่าใช้งาน servlet

สร้างไฟล์ TestServlet.java บันทึกไฟล์ไว้ที่ C:\Tomcat 6.0\webapps\ROOT\WEB-INF\classes โดยการสร้าง ไดเรคทอรี ชื่อ classes ขึ้นมาใหม่

```
TestServlet.java
1  import javax.servlet.http.*;
2
3  public class TestServlet extends HttpServlet {
4  }
```

ทำการ compile ถ้ามีข้อความ error ดังต่อไปนี้ขึ้นมา แสดงว่ายังไม่สามารถเชื่อมต่อกับ Class ของ Java ได้

```
----- compile -----
TestServlet.java:1: package javax.servlet.http does not exist
import javax.servlet.http.*;
^
TestServlet.java:3: cannot find symbol
symbol: class HttpServlet
public class TestServlet extends HttpServlet {
                                ^
2 errors

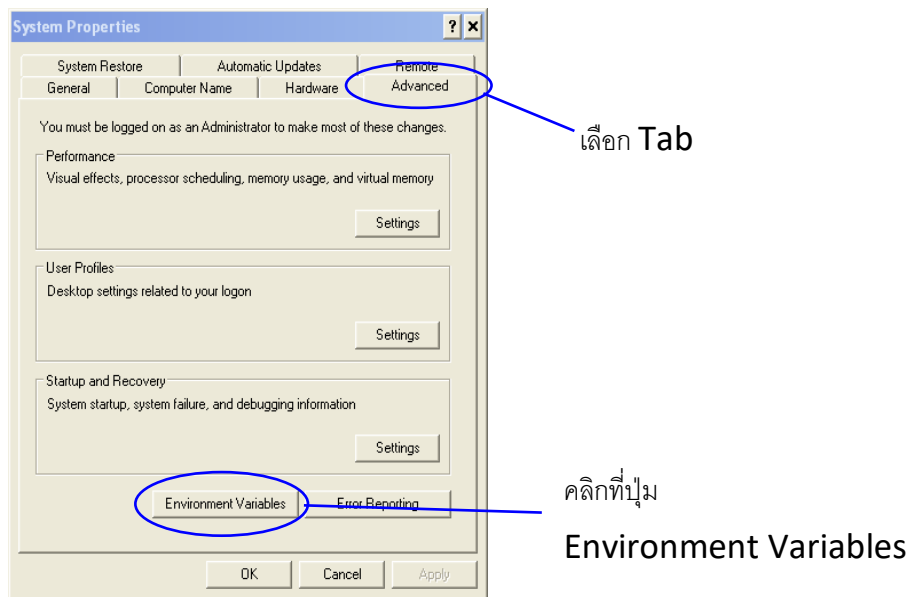
Output completed (0 sec consumed)
```

### การตั้งค่า Servlet ให้แสดงผลด้วย Browser

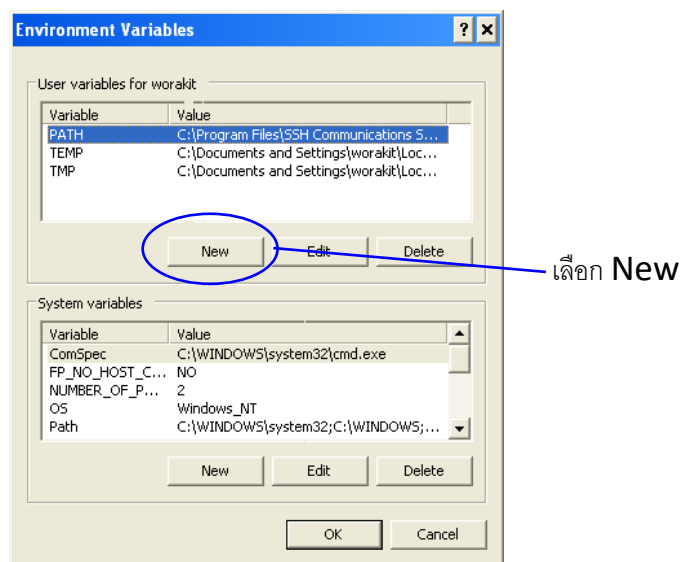
เพื่อให้ Servlet สามารถติดต่อกับ Class ได้ จำเป็นต้องมีการตั้งค่าของระบบ windows ให้เข้าถึง path directory ของ Class หรือเป็นการตั้งค่า Class Path ใน Environment Variables ดังนี้



1. เปิดหน้าต่าง System Properties ด้วยการคลิกขวาที่ Icon My Computer เลือก Properties หรือ เลือก Icon System ใน Control Panel
2. ในหน้าต่าง System Properties เลือก Tab Advanced คลิกที่ปุ่ม Environment Variables



3. ในหน้าต่าง Environment Variables จะแบ่งออกเป็นสองส่วน ส่วนของ User ที่ login ใช้งาน สิ่งที่กำหนดในส่วนนี้จะใช้เฉพาะ User คนนั้น และส่วนของ System ที่ไม่ว่า User คนไหน Login เข้ามาก็จะใช้ค่าที่กำหนดร่วมกัน สามารถตั้งค่า Class Path ที่ส่วนไหนก็ได้ เลือก New

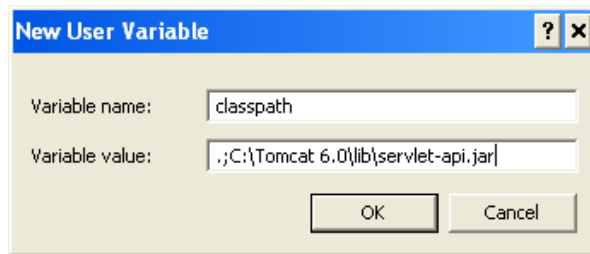


4. ในหน้าต่าง New User Variable ช่อง Variable name ใส่คำว่า classpath และช่อง Variable value ใส่ตำแหน่งของไฟล์ servlet.jar ในที่นี้คือ C:\Tomcat 6.0\lib\servlet-api.jar โดยใส่ในช่องว่างดังนี้



.;C:\Tomcat 6.0\lib\servlet-api.jar

การที่ต้องใส่เครื่องหมายจุดแล้วตามด้วย ; เพื่อเป็นการให้หาไฟล์ .class ในไดเรกทอรีที่เก็บไฟล์ก่อน ถ้าหาไม่พบจึงเรียกไฟล์ servlet-api.jar



5. เมื่อตั้งค่าเสร็จเรียบร้อยแล้ว ต้องทำการ Restart ระบบใหม่เพื่อให้มีการรับค่าที่ตั้งใหม่

### ขั้นตอนการปฏิบัติงาน

1. เขียนโปรแกรม Servlet ประกอบไปด้วยโครงสร้างพื้นฐานดังภาพ 3.12 ให้แสดงผลข้อความ “Hello Servlet”

```
import java.io.*;
import Java.servlet.*;
import javax.servlet.http.*;

public class processPara extends HttpServlet
{
    public void doGet(HttpServletRequest req, HttpServletResponse res)
        throws ServletException, IOException
    {
        .....
        .....
    }
}
```

2. เขียนคำสั่งภาษา java ลงใน method doGet() ทำการ save ด้วยชื่อ TestServlet .java
3. ทำการ compile ถ้ามี error ต้องแก้ไข error ที่เกิดขึ้น

- แก้ไขไฟล์ web.xml ให้สามารถเรียก class ที่ compile ได้

```
<servlet>
    <servlet-name>My Hello</servlet-name>
    <servlet-class>Hello</servlet-class>
</servlet>

<servlet-mapping>
```

- แสดงผลโดยการเรียกชื่อไฟล์จาก Browser
- ตรวจสอบความสำเร็จจากการแสดงผลจาก Browser

## ใบงาน

ชื่องาน เขียนโปรแกรมนับจำนวนครั้งการ refresh เบราวเซอร์

ชื่อวิชา การพัฒนาเว็บด้วยเทคโนโลยีจาวา รหัสวิชา 30901-2006

---

การเขียนโปรแกรม JSP จำเป็นต้องทราบถึง tag แบบต่างๆ ที่สามารถนำไปใช้งานได้ ซึ่ง tag ของ JSP แบ่งออกเป็น 4 ประเภท ดังตาราง

ชื่อของ tag	สัญลักษณ์ของ tag
Expression	<%= %>
Directive	<%@ %>
Scriptlet	<% %>
Declaration	<%! %>

### การใช้ Tag Expression

เป็น tag ที่ใช้สัญลักษณ์ <%= %> โดยที่เครื่องหมายเท่ากับ (=) ต้องอยู่ติดกับเครื่องหมายเปอร์เซ็นต์ (%) ถ้ามีการเว้นช่องว่างหรือมีการเคาะเว้นวรรค จะถือว่าเป็น tag ที่ไม่ถูกต้อง

ใช้สำหรับแสดงค่า output ออกทางจอภาพหรือทางโปรแกรมเบราวเซอร์ จะเป็นค่าของ ตัวแปรหรือเป็นการเรียกใช้ฟังก์ชันก็ได้ โดยที่ค่าของตัวแปรที่จะนำมาแสดงผลจะต้องมีการแปลง ให้มีรูปแบบเป็นตัวอักษรก่อนเสมอ

### ตัวอย่าง การใช้ tag Expression

บันทึกไฟล์ชื่อ Expression.jsp ไว้ตำแหน่ง C:\Tomcat 6.0\webapps\ROOT

Expression.jsp

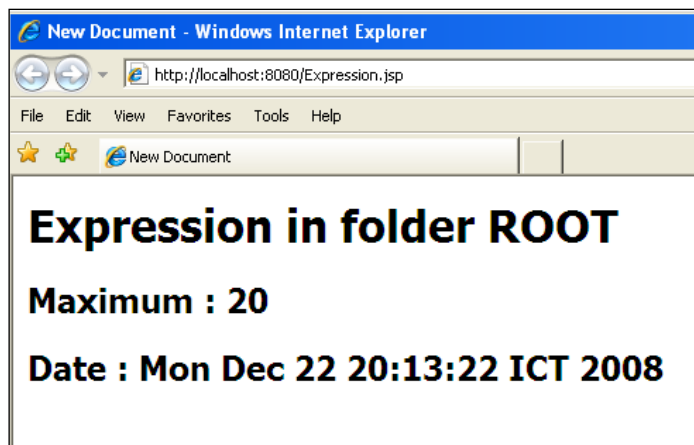
```
1 <HTML>
2 <HEAD>
3 <TITLE> New Document </TITLE>
4 </HEAD>
```

```

5 <BODY>
6 <H1>Expression in folder ROOT</H1>
7 <H2>Maximum : <%= java.lang.Math.max(20,10) %></H2>
8 <H2>Date : <%= new java.util.Date() %></H2>
9 </BODY>
10 </HTML>

```

ผลการทำงานของโปรแกรม Expression.jsp แสดงผลบนโปรแกรม Internet Explorer แสดงข้อความ แสดงค่าสูงสุดของเลขสองจำนวนจาก Method Max() และแสดงค่าของวันเวลาปัจจุบันด้วย Method Date()



### การใช้ Tag Scriptlet

เป็น tag ที่ใช้สัญลักษณ์ <% %> ใช้สำหรับแทรกโค้ดภาษา JAVA ลงใน JSP คำสั่งต่างๆ ของภาษา JAVA ต้องปิดท้ายด้วย ; (semicolon) ในแต่ละคำสั่ง

### ตัวอย่าง การใช้ tag Scriptlet

บันทึกไฟล์ชื่อ scriptlet.jsp ไว้ตำแหน่ง C:\Tomcat 6.0\webapps\ROOT

```

scriptlet.jsp
1 <HTML>
2 <HEAD>
3 <TITLE> test Scriptlet </TITLE>

```

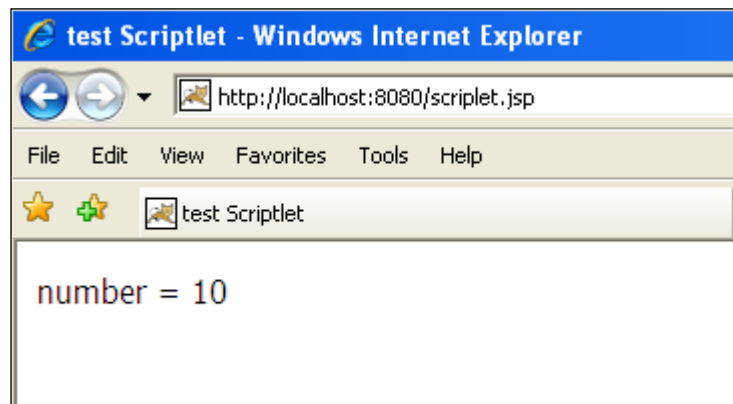
```
4 </HEAD>
5
6 <BODY>
7 <%
8   int num = 10;
9   out.print("number = "+num);
10 %>
11 </BODY>
12 </HTML>
```

บรรทัดที่ 7 และ 10 เป็นการเปิดและปิด tag scriptlet ฉะนั้น บรรทัดที่ 8-9 จะเป็นโค้ดภาษา JAVA

บรรทัดที่ 8 เป็นการประกาศตัวแปรแบบเลขจำนวนเต็มชื่อ num มีค่าเท่ากับ 10

บรรทัดที่ 9 เป็นการแสดงค่าของตัวแปรด้วยคำสั่ง out.print()

รันโปรแกรมจาก <http://localhost:8080/scriptlet.jsp> จะได้ผลลัพธ์ดังภาพ



## ตัวอย่าง การใช้ tag Scriptlet แทรกอยู่ระหว่าง tag HTML

บันทึกไฟล์ชื่อ scriptlet\_2.jsp ไว้ตำแหน่ง C:\Tomcat 6.0\webapps\ROOT

```
scriptlet2.jsp
1 <HTML>
2 <HEAD>
3 <TITLE> test Scriptlet </TITLE>
4 </HEAD>
5 <BODY>
6 <TABLE border=1>
7 <% for (int i=1;i<=10;i++){ %>
8 <TR>
9 <% for (int j=1;j<=10;j++){ %>
10 <TD><%= i*j %></TD>
11 <% } %>
12 </TR>
13 <% } %>
14 </TABLE>
15 </BODY>
16 </HTML>
```

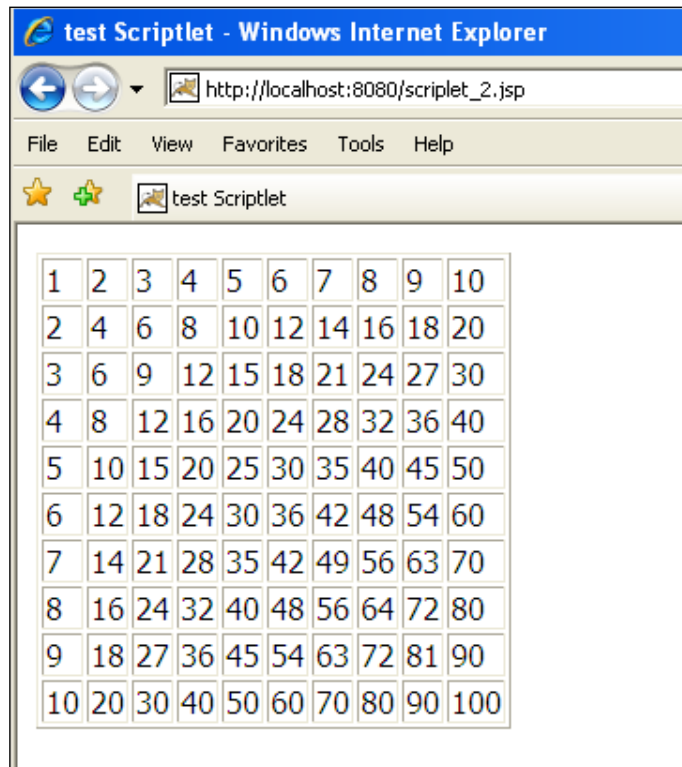
ตัวอย่างนี้มีการแทรกการวนลูปลงใน tag HTML ในการสร้างตารางขึ้นมา ประกอบการวนลูป 2 ชั้น ลูปนอกใช้ในการกำหนดจำนวนแถว ลูปในใช้กำหนดจำนวนคอลัมน์

บรรทัดที่ 6 และบรรทัดที่ 14 เป็นการเปิดและปิด tag สำหรับสร้างตาราง ในการสร้างตารางจะต้องประกอบด้วย tag ของแถวคือ <TR> และในแต่ละแถวจะมี tag <TD> เท่ากับจำนวนช่องในแต่ละแถว

บรรทัดที่ 7 เป็นการแทรก tag JSP แบบ for loop จำนวน 10 รอบ เข้าไประหว่าง tag HTML และมีการปิด for loop ในบรรทัดที่ 13 ส่วนบรรทัดที่อยู่ระหว่างบรรทัดที่ 7-13 นี้จะต้องทำงานทั้งหมด 10 รอบ

บรรทัดที่ 9 เป็นการแทรก tag JSP แบบ for loop จำนวน 10 รอบ เข้าไประหว่าง tag HTML และมีการปิด for loop ในบรรทัดที่ 11 ภายใน loop เป็นการแสดงค่า i คูณกับค่า j ซึ่งการแสดงผลคูณของ i กับ j อยู่ภายใต้ tag scriptlet เพื่อให้แสดงผลคูณออกมา ซึ่งผลคูณที่แสดงออกมาจะอยู่ภายใต้ tag HTML มีจำนวนเท่ากับจำนวนรอบที่ for loop กระทำ ในที่นี้จะกระทำ 10 รอบต่อ 1 แถว

รันโปรแกรมจาก [http://localhost:8080/scriptlet\\_2.jsp](http://localhost:8080/scriptlet_2.jsp) จะได้ผลลัพธ์ดังภาพ



1	2	3	4	5	6	7	8	9	10
2	4	6	8	10	12	14	16	18	20
3	6	9	12	15	18	21	24	27	30
4	8	12	16	20	24	28	32	36	40
5	10	15	20	25	30	35	40	45	50
6	12	18	24	30	36	42	48	54	60
7	14	21	28	35	42	49	56	63	70
8	16	24	32	40	48	56	64	72	80
9	18	27	36	45	54	63	72	81	90
10	20	30	40	50	60	70	80	90	100

## การใช้ Tag Directive

เป็น tag ที่ใช้สัญลักษณ์ `<%@ %>` โดยที่เครื่องหมายเท่ากับ ( @ ) ต้องอยู่ติดกับเครื่องหมายเปอร์เซ็นต์ ( % ) ถ้ามีการเว้นช่องว่างหรือมีการเคาะเว้นวรรค จะถือว่าเป็น tag ที่ไม่ถูกต้อง

ใช้สำหรับแทรกโค้ดของไฟล์อื่นเข้ามา ด้วยคำสั่ง `include` ต้องอยู่ภายใน tag นี้ หรือใช้ในการกำหนดค่าแอตทริบิวต์ (Attribute) ต่างๆ ของ JSP ด้วยคำสั่ง `page` ตามด้วยแอตทริบิวต์ และค่าของแอตทริบิวต์นั้นๆ เช่น

```
<%@ include file="header.html" %>
```



เป็นการเรียกใช้ไฟล์ header.html แทรกลงในไฟล์ JSP ในที่นี้เป็นการสร้างส่วนบนของหน้าจอ ซึ่งทุกหน้าจะมีส่วนบนเหมือนกัน ถ้าต้องเขียนโค้ดใหม่ทุกครั้ง ก็จะเป็นการเขียนโค้ดเดิมๆ ทุกหน้า เวลาแก้ไขก็ต้องเข้าไปแก้ไขทุกไฟล์ ถ้าใช้การ include เวลาแก้ไขสามารถแก้ไขไฟล์เดียว มีผลทุกหน้า

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

เป็นการกำหนดค่าแอตทริบิวต์ของ JSP เพื่อบอกให้ทราบว่าคลาสที่ถูกเรียกใช้นั้นสามารถหาได้จากที่ใด จากแพ็คเกจใด ในที่นี้ให้ไปหาจากแพ็คเกจ java.util

```
<%@ page autoFlush="true" %>
```

เป็นการกำหนดค่าแอตทริบิวต์ของ JSP เพื่อบอกให้ทราบว่าเมื่อบัฟเฟอร์เต็ม จะให้จัดการอย่างไรกับบัฟเฟอร์ ค่าที่สามารถใส่ได้คือ true กับ false ถ้าใส่ค่าเป็น true จะทำการเคลียร์บัฟเฟอร์ให้อัตโนมัติ

```
<%@ page buffer="64kb" %>
```

เป็นการกำหนดค่าแอตทริบิวต์ของ JSP เพื่อกำหนดขนาดของค่าของบัฟเฟอร์ใหม่ โดยปกติเมื่อไม่มีการกำหนดค่าใดๆ ขนาดของบัฟเฟอร์จะมีค่า 8 kb (หน่วยเป็นกิโลไบต์)

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

เป็นการกำหนดค่าแอตทริบิวต์ของ JSP เพื่อกำหนดค่าของการแสดงภาษาในหน้านั้นๆ ว่ามีการใช้การแปลงโค้ดอักขรแบบใด โดยปกติค่าเริ่มต้นจะเป็น text/html; charset=ISO8859-1 ซึ่งไม่สามารถแสดงตัวอักษรภาษาไทยได้ การกำหนดค่าให้แสดงภาษาไทยได้นั้นสามารถใช้ text/html; charset=TIS-620 ได้

```
<%@ page errorPage="error.jsp" %>
```

เป็นการกำหนดค่าแอตทริบิวต์ของ JSP เพื่อกำหนดว่าเมื่อเกิด error ขึ้นจะไปเรียก page ที่มีชื่อว่า error.php โดยการโยน exception ไปด้วย เมื่อไฟล์ error.php รับค่าแล้วสามารถแสดงหน้าหรือข้อความ error ออกมา

```
<%@ page isErrorPage="true" %>
```

เป็นการกำหนดค่าแอตทริบิวต์ของ JSP เพื่อกำหนดว่าเมื่อมีการโยน exception มาจะรับไว้หรือไม่ ค่าของแอตทริบิวต์นี้มีค่าเป็น true และ false โดยค่าเริ่มต้นจะเป็น false ถ้าไม่มีการกำหนดให้เป็น true เมื่อมีการโยน exception มากก็จะไม่แสดงหน้านี้ออกมา เวลาใช้งานจะใช้คู่กับ errorPage

```
<%@ page extend="myclass.MotherClass" %>
```

เป็นการกำหนดค่าแอตทริบิวต์ของ JSP เพื่อกำหนดว่ามีการสืบทอดคุณสมบัติของ class แม่มาจากที่ใด ซึ่งเมื่อมีการสืบทอดสามารถใช้ตัวแปรหรือ method ของ class แม่ได้

```
<%@ page pageEncoding="TIS-620" %>
```

เป็นการกำหนดค่าแอตทริบิวต์ของ JSP เพื่อกำหนดการใช้งานของ character encoding ที่หน้านั้นๆ ใช้สำหรับส่ง response กลับไป โดยถ้าต้องการแสดงภาษาไทยต้องกำหนด pageEncoding เป็น TIS-620 หรือ WINDOWS-874

```
<%@ page session="true" %>
```

เป็นการกำหนดค่าแอตทริบิวต์ของ JSP เพื่อกำหนดการใช้งานของหน้านั้นๆ ว่าต้องการดึงข้อมูลจาก session มาใช้หรือไม่ ค่าที่กำหนดได้มี 2 ค่า คือ เป็น true และ false โดยค่าเริ่มต้นจะเป็น true ซึ่งสามารถดึงค่าของ session มาใช้งานได้

### ตัวอย่าง การใช้งาน Tag Directive (include และ import)

สร้างไฟล์ header.html โดยนำรูป jspLogo มาใส่

```
header.html
```

```
1 <HTML>
2 <BODY>
3     <IMG SRC="pic/jspLogo.jpg">
4 </BODY>
5 </HTML>
```

สร้างไดเรกทอรี pic ขึ้นที่ C:\Tomcat 6.0\webapps\ROOT และนำไฟล์ jspLogo.jpg มาเก็บไว้ที่ C:\Tomcat 6.0\webapps\ROOT\pic ซึ่งเมื่อทดสอบการทำงานของ header.html โดยการเรียกโปรแกรมจาก <http://localhost:8080/header.html> จะได้ผลลัพธ์ดังภาพ



เราจะใช้ Tag Directive ในการแทรกไฟล์เข้ามาในอีกไฟล์หนึ่ง คือ include โดยการแทรก header.html เข้าไว้ในไฟล์ jsp แล้วบันทึกไฟล์ไว้ที่ C:\Tomcat 6.0\webapps\ROOT ตั้งชื่อไฟล์ว่า directive\_1.jsp

```
directive_1.jsp
```

```
1 <HTML>
2 <HEAD>
3 <TITLE> directive </TITLE>
4 </HEAD>
5
6 <BODY>
7     <%@ include file="header.html" %>
8     <%@ page import="java.util.*" %>
9     <H2>Date : <%= new Date() %></H2>
10 </BODY>
11 </HTML>
```

บรรทัดที่ 7 เป็นการใช้ include directive แทรกไฟล์ header.html เข้ามา

บรรทัดที่ 8 เป็นการใช้ import directive บอกตำแหน่งที่เก็บไฟล์สำหรับเรียกใช้งาน class ด้วยชื่อ class สั้นๆ ถ้าไม่มีการ import จะต้องเรียกโดยการระบุ part เต็มๆ โดยต้องระบุดังนี้

```
<%= new java.util.Date() %>
```

เมื่อมีการ import class ใช้งาน

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

เวลาเรียกใช้งาน class จึงสามารถเรียกสั้นๆ ดังบรรทัดที่ 9

รันโปรแกรมจาก [http://localhost:8080/directive\\_1.jsp](http://localhost:8080/directive_1.jsp) จะได้ผลลัพธ์ดังภาพ



## ใบงาน

ชื่องาน เขียนโปรแกรมส่งข้อมูล form textbox ไปแสดงผลใน JSP

ชื่อวิชา การพัฒนาเว็บด้วยเทคโนโลยีจาวา รหัสวิชา 30901-2006

---

รับข้อมูลด้วย textbox

เป็นการรับข้อมูลจากผู้ใช้งานในรูปแบบข้อความ เช่น ชื่อ นามสกุล รหัสผู้ใช้งาน ซึ่งเป็นข้อความที่มีความยาวไม่มากนัก ไม่เกิน 1 บรรทัด เราจะใช้ textbox ในการรับค่าของข้อมูล

รูปแบบ

```
<INPUT TYPE="text" NAME="ชื่อของ textbox"
      VALUE="ค่าของตัวแปร textbox"
      SIZE="ขนาดความยาวของช่อง textbox"
      MAXLENGTH="จำนวนตัวอักษรสูงสุดที่ textbox รับได้" >
```

ตัวอย่าง

```
<INPUT TYPE="text" NAME="name" VALUE="admin" SIZE="30" MAXLENGTH="10">
```

ผลลัพธ์



การส่งข้อมูลจาก form ไปประมวลผล

จากการที่ได้สร้าง form ขึ้นมาให้มีหน้าตาแบบต่างๆ เมื่อทำการคลิกเมาส์ที่ปุ่ม submit ค่าของตัวแปรต่างๆ จะต้องถูกส่งไปประมวลผลในไฟล์ JSP ฉะนั้นจะต้องมีไฟล์ 2 ไฟล์ คือไฟล์แรกเป็น form สำหรับรับข้อมูลจากผู้ใช้และมีการระบุข้อมูลใน form จะส่งไปยังไฟล์ใด ในการประมวลผล ไฟล์ที่สองเป็นไฟล์ JSP ที่ form ระบุมาว่าส่งมาที่ไฟล์ JSP ไต ไฟล์ JSP จะนำค่าต่างๆ ที่ถูกส่งมาทำการประมวลผลและแสดงผลทางเบราว์เซอร์

รูปแบบ tag ของ form ต้องมีรูปแบบและบอกว่าจะส่งข้อมูลที่รับมาไปยังที่ใด รูปแบบของ form มีลักษณะดังนี้

```
<FORM METHOD="POST" ACTION="ชื่อไฟล์ JSP ที่จะส่งไปประมวลผล">
```

..... รายละเอียดต่างๆ ของ form

```
</FORM>
```

ตัวอย่าง เช่น

```
<FORM METHOD="POST" ACTION="processData.jsp">
```

..... รายละเอียดต่างๆ ของ form

```
</FORM>
```

เมื่อทำการกดปุ่ม submit ค่าต่างๆ ที่อยู่ใน form จะถูกส่งไปประมวลผลที่ไฟล์ processData.jsp

### การรับข้อมูลจาก form ด้วยโปรแกรม JSP

เมื่อมีการส่งค่าต่างๆ เข้ามาในโปรแกรม JSP โปรแกรม JSP จะต้องรับค่าต่างๆ มาเก็บไว้ในรูปของตัวแปร โดยทั่วไปข้อมูลที่ถูกส่งเข้ามาจะมีชนิดเป็น String ในการรับค่าตัวแปรจะใช้คำสั่ง

```
request.getParameter("ชื่อของ Element ในแต่ละชนิด");
```

ตัวอย่าง เช่น form มีการส่งค่ามาดังนี้

```
<FORM METHOD=POST ACTION="processData.jsp">
```

```
ชื่อ/สกุล : <INPUT TYPE="text" NAME="user" size=50><br>
```

```
<INPUT TYPE="reset">
```

```
<INPUT TYPE="submit" value="ส่งข้อมูล">
```

```
</FORM>
```

ข้อมูล form มีการรับค่า 1 อย่าง คือชื่อ นามสกุล เมื่อผู้ใช้งานกรอกชื่อ นามสกุล แล้วกดปุ่ม submit ข้อมูลชื่อที่กรอกไว้จะถูกส่งมาที่ไฟล์ชื่อ processData.jsp ในรูปของตัวแปรที่ชื่อว่า user

ไฟล์ processData.jsp จะต้องทำการรับค่าที่ถูกส่งเข้ามา ในรูปของตัวแปรที่ชื่อว่า user โดยสามารถเขียนโค้ดดังนี้

```
request.getParameter("user");
```

## ใบงาน

ชื่องาน เขียนโปรแกรมส่งข้อมูล form checkbox ไปแสดงผลใน JSP

ชื่อวิชา การพัฒนาเว็บด้วยเทคโนโลยีจาวา รหัสวิชา 30901-2006

---

### รับข้อมูลด้วย checkbox

เป็นการรับข้อมูลจากผู้ใช้งานในรูปแบบตัวเลือกที่ผู้ใช้งานสามารถเลือกได้มากกว่า 1 รายการ โดยต้องการเลือกรายการใดก็ใช้เมาส์คลิกที่ช่องสี่เหลี่ยมหน้ารายการนั้น

#### รูปแบบ

```
<INPUT TYPE="checkbox" NAME="ชื่อของ checkbox"
      VALUE="ค่าที่กำหนดให้ตัวแปร checkbox"
      CHECKED >
```

#### ตัวอย่าง

```
<INPUT TYPE="checkbox" NAME="color" VALUE="red" CHECKED>red
<INPUT TYPE="checkbox" NAME="color" VALUE="green">green
<INPUT TYPE="checkbox" NAME="color" VALUE="blue">blue
```

#### ผลลัพธ์

red  green  blue

#### อธิบาย รูปแบบของ checkbox

การสร้าง checkbox มีชื่อว่า color และมี 3 รายการคือ red green blue โดยมีการเลือกไว้ 1 ค่าคือ red ผู้ใช้สามารถคลิกเมาส์เลือกรายการใหม่หรือคลิกเมาส์ยกเลิกสิ่งที่เลือกได้ ถ้ามีการเลือกรายการใด ค่าของตัวแปร color ก็จะมีค่าตามที่ได้เลือกไว้ ค่าของตัวแปรแบบ checkbox สามารถมีได้หลายค่า ตามแต่จะเลือกค่าใดไว้บ้าง โดยค่าของตัวแปรแบบ checkbox จะถูกเก็บไว้ในรูปแบบตัวแปร array

ถ้าต้องการให้มีการเลือกตั้งแต่แรกในการทำงาน ให้ใส่คำว่า CHECKED ไว้ใน tag นั้น



## การส่งข้อมูลจาก form ไปประมวลผล

จากการที่ได้สร้าง form ขึ้นมาให้มีหน้าตารับค่าแบบต่างๆ เมื่อทำการคลิกเมาส์ที่ปุ่ม submit ค่าของตัวแปรต่างๆ จะต้องถูกส่งไปประมวลผลในไฟล์ JSP ฉะนั้นจะต้องมีไฟล์ 2 ไฟล์ คือไฟล์แรกเป็น form สำหรับรับข้อมูลจากผู้ใช้และมีการระบุข้อมูลใน form จะส่งไปยังไฟล์ใด ในการประมวลผล ไฟล์ที่สองเป็นไฟล์ JSP ที่ form ระบุว่าส่งมาที่ไฟล์ JSP ไต ไฟล์ JSP จะนำค่าต่างๆ ที่ถูกส่งมาทำการประมวลผลและแสดงผลทางเบราว์เซอร์

รูปแบบ tag ของ form ต้องมีรูปแบบและบอกว่าจะส่งข้อมูลที่รับมาไปยังที่ใด รูปแบบของ form มีลักษณะดังนี้

```
<FORM METHOD="POST" ACTION="ชื่อไฟล์ JSP ที่จะส่งไปประมวลผล">
    ..... รายละเอียดต่างๆ ของ form
</FORM>
```

ตัวอย่าง เช่น

```
<FORM METHOD="POST" ACTION="processData.jsp">
    ..... รายละเอียดต่างๆ ของ form
</FORM>
```

เมื่อทำการกดปุ่ม submit ค่าต่างๆ ที่อยู่ใน form จะถูกส่งไปประมวลผลที่ไฟล์ processData.jsp

## JSP กับการรับข้อมูลที่มีมากกว่า 1 ค่า

การรับข้อมูลจาก textbox password radio textarea และ listbox สามารถรับค่าได้เหมือนตัวอย่างด้านบนเพราะมีค่าที่ส่งมา 1 ค่าต่อ 1 ตัวแปร แต่สำหรับ checkbox และ listbox ที่มีการกำหนด MULTIPLE มีการส่งค่ามากกว่า 1 ค่าต่อ 1 ตัวแปร ในการรับค่าจึงต้องใช้ตัวแปรแบบ array ในการรับค่า และคำสั่งในการรับค่าแบบหลายค่าคือ

```
request.getParameterValues("ชื่อของ Element");
```

## ใบงาน

ชื่องาน เขียนโปรแกรมส่งข้อมูล form checkbox ไปแสดงผลใน JSP

ชื่อวิชา การพัฒนาเว็บด้วยเทคโนโลยีจาวา รหัสวิชา 30901-2006

---

### รับข้อมูลด้วย checkbox

เป็นการรับข้อมูลจากผู้ใช้งานในรูปแบบตัวเลือกที่ผู้ใช้งานสามารถเลือกได้มากกว่า 1 รายการ โดยต้องการเลือกรายการใดก็ใช้เมาส์คลิกที่ช่องสี่เหลี่ยมหน้ารายการนั้น

รูปแบบ

```
<INPUT TYPE="checkbox" NAME="ชื่อของ checkbox"
        VALUE="ค่าที่กำหนดให้ตัวแปร checkbox"
        CHECKED >
```

ตัวอย่าง

```
<INPUT TYPE="checkbox" NAME="color" VALUE="red" CHECKED>red
<INPUT TYPE="checkbox" NAME="color" VALUE="green">green
<INPUT TYPE="checkbox" NAME="color" VALUE="blue">blue
```

ผลลัพธ์

red  green  blue

อธิบาย รูปแบบของ checkbox

การสร้าง checkbox มีชื่อว่า color และมี 3 รายการคือ red green blue โดยมีการเลือกไว้ 1 ค่าคือ red ผู้ใช้สามารถคลิกเมาส์เลือกรายการใหม่หรือคลิกเมาส์ยกเลิกสิ่งที่เลือกได้ ถ้ามีการเลือกรายการใด ค่าของตัวแปร color ก็จะมีค่าตามที่ได้เลือกไว้ ค่าของตัวแปรแบบ checkbox สามารถมีได้หลายค่า ตามแต่จะเลือกค่าใดไว้บ้าง โดยค่าของตัวแปรแบบ checkbox จะถูกเก็บไว้ในรูปแบบตัวแปร array

ถ้าต้องการให้มีการเลือกตั้งแต่แรกในการทำงาน ให้ใส่คำว่า CHECKED ไว้ใน tag นั้น

## การส่งข้อมูลจาก form ไปประมวลผล

จากการที่ได้สร้าง form ขึ้นมาให้มีหน้าตารับค่าแบบต่างๆ เมื่อทำการคลิกเมาส์ที่ปุ่ม submit ค่าของตัวแปรต่างๆ จะต้องถูกส่งไปประมวลผลในไฟล์ JSP ฉะนั้นจะต้องมีไฟล์ 2 ไฟล์ คือไฟล์แรกเป็น form สำหรับรับข้อมูลจากผู้ใช้และมีการระบุข้อมูลใน form จะส่งไปยังไฟล์ใด ในการประมวลผล ไฟล์ที่สองเป็นไฟล์ JSP ที่ form ระบุว่าส่งมาที่ไฟล์ JSP ไต ไฟล์ JSP จะนำค่าต่างๆ ที่ถูกส่งมาทำการประมวลผลและแสดงผลทางเบราว์เซอร์

รูปแบบ tag ของ form ต้องมีรูปแบบและบอกว่าจะส่งข้อมูลที่รับมาไปยังที่ใด รูปแบบของ form มีลักษณะดังนี้

```
<FORM METHOD="POST" ACTION="ชื่อไฟล์ JSP ที่จะส่งไปประมวลผล">
```

..... รายละเอียดต่างๆ ของ form

```
</FORM>
```

ตัวอย่าง เช่น

```
<FORM METHOD="POST" ACTION="processData.jsp">
```

..... รายละเอียดต่างๆ ของ form

```
</FORM>
```

เมื่อทำการกดปุ่ม submit ค่าต่างๆ ที่อยู่ใน form จะถูกส่งไปประมวลผลที่ไฟล์ processData.jsp

## JSP กับการรับข้อมูลที่มีมากกว่า 1 ค่า

การรับข้อมูลจาก textbox password radio textarea และ listbox สามารถรับค่าได้เหมือนตัวอย่างด้านบนเพราะมีค่าที่ส่งมา 1 ค่าต่อ 1 ตัวแปร แต่สำหรับ checkbox และ listbox ที่มีการกำหนด MULTIPLE มีการส่งค่ามากกว่า 1 ค่าต่อ 1 ตัวแปร ในการรับค่าจึงต้องใช้ตัวแปรแบบ array ในการรับค่า และคำสั่งในการรับค่าแบบหลายค่าคือ

```
request.getParameterValues("ชื่อของ Element");
```

## ใบงาน

### ชื่องาน เขียนโปรแกรมใช้งาน Cookie

ชื่อวิชา การพัฒนาเว็บด้วยเทคโนโลยีจาวา รหัสวิชา 30901-2006

---

#### การใช้งาน Cookie

หลังจากผู้ใช้งานเปิดเบราว์เซอร์ติดต่อไปยัง web server จะมีการตรวจสอบว่าเครื่องของผู้ใช้งานเครื่องนี้มีการเก็บค่า Cookie ไว้หรือไม่ ถ้ามีการเก็บค่าเอาไว้ ก็สามารถเข้าสู่เว็บไซต์ ได้โดยไม่ต้องทำการ login ก่อน ใช้ชื่อผู้ใช้งานตามข้อมูลที่เก็บใน Cookie แต่ถ้าไม่มีข้อมูลใน Cookie ก็จะเข้าสู่หน้า login ทำการใส่ username และรหัสผ่านครั้งแรกก่อน เมื่อการ login ถูกต้อง เครื่องที่ใช้งานนี้อยู่ก็จะถูกเขียนข้อมูลเล็กๆ ลงในเครื่องทำให้การเข้ามาในครั้งต่อไปไม่จำเป็นต้อง login

รูปแบบในการสร้างข้อมูล cookie ประกอบไปด้วย

1. การสร้างตัวแปร cookie และค่าของตัวแปร

```
Cookie ตัวแปร_Object = new Cookie ("ชื่อ_cookie",ค่า_cookie)
```

2. การกำหนดเวลาหมดอายุของ cookie

```
ตัวแปร_Object.setMaxAge(วินาที)
```

3. การบันทึกค่า cookie

```
response.addCookie(ตัวแปร_Object)
```

ในการดึงข้อมูล cookie ออกมาใช้คำสั่ง

```
ชื่อ_cookie.getValue()
```

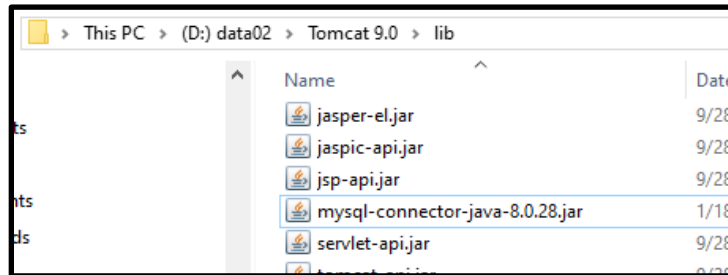
## ใบงาน

### ชื่องาน เขียนโปรแกรมติดต่อฐานข้อมูล Mysql

ชื่อวิชา การพัฒนาเว็บด้วยเทคโนโลยีจาวา รหัสวิชา 30901-2006

#### การเชื่อมต่อฐานข้อมูล

1. Download driver สำหรับ connect Mysql คือไลบรารี JDBC ที่ใช้สำหรับเชื่อมต่อฐานข้อมูล MySQL กับ Java โดยต้องดาวน์โหลดไฟล์ mysql-connector-java-8.0.28-bin.jar download ที่ <https://jar-download.com/artifacts/mysql/mysql-connector-java>
2. นำมาวางไว้ในโฟลเดอร์ lib ของโปรเจกต์ Java คือ ../Tomcat 10.1\lib\ หรือ ../Tomcat\_10.1\webapps\.....\WEB-INF\lib\
3. Restart service Tomcat



4. ตัวอย่าง code connect

```
<%@ page import="java.sql.*" %>
<%@ page pageEncoding="utf8" %>
<!doctype html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>Connect Database mysql</title>
</head>
<body>
<%
try{
    //สร้างตัวแปรเชื่อมต่อ ฐานข้อมูล mysql
    String connectionURL = "jdbc:mysql://localhost:3306/mysql";
    // ประกาศตัวแปร connection
```

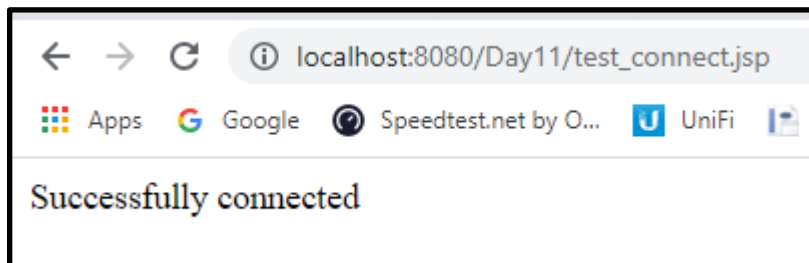
```
Connection connection = null;
// Load driver mysql
Class.forName("com.mysql.jdbc.Driver").newInstance();
connection = DriverManager.getConnection(connectionURL, "root", "1234");

if(!connection.isClosed()){
    out.println("Successfully connected");
    connection.close();
}

}

catch (Exception ex){
    out.println("connected error : "+ex);
}

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



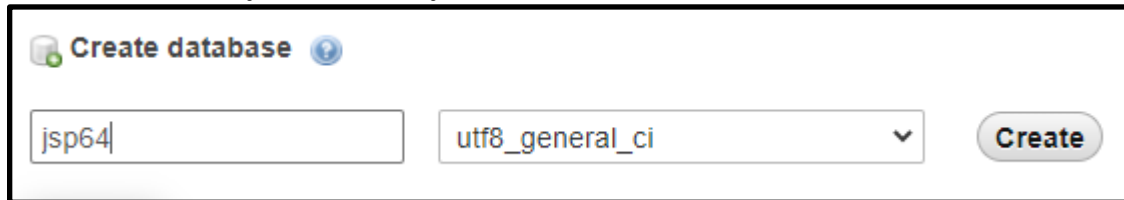
## ใบงาน

ชื่องาน เขียนโปรแกรมแสดงข้อมูลจากฐานข้อมูล Mysql

ชื่อวิชา การพัฒนาเว็บด้วยเทคโนโลยีจาวา รหัสวิชา 30901-2006

### การแสดงผลข้อมูลจากฐานข้อมูล

#### 1. สร้างฐานข้อมูลและตารางข้อมูล



Create database ?

▼ Create

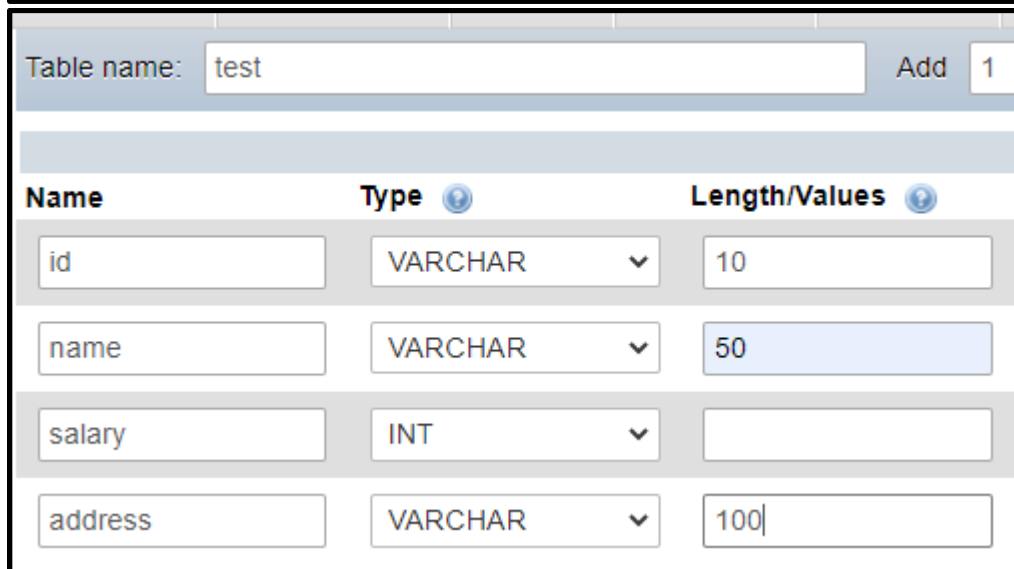


Table name:  Add 1

Name	Type	Length/Values
<input type="text" value="id"/>	<input type="text" value="VARCHAR"/> <span>▼</span>	<input type="text" value="10"/>
<input type="text" value="name"/>	<input type="text" value="VARCHAR"/> <span>▼</span>	<input type="text" value="50"/>
<input type="text" value="salary"/>	<input type="text" value="INT"/> <span>▼</span>	<input type="text"/>
<input type="text" value="address"/>	<input type="text" value="VARCHAR"/> <span>▼</span>	<input type="text" value="100"/>

#### 2. เพิ่มข้อมูล ลงในตารางที่สร้างไว้



	id	name	salary	address
<input type="checkbox"/> <span>Edit</span> <span>Copy</span> <span>Delete</span>	101	aaaaaa	20000	ชลบุรี
<input type="checkbox"/> <span>Edit</span> <span>Copy</span> <span>Delete</span>	102	bbbbbb	15000	กรุงเทพ

#### 3. เขียนคำสั่งแสดงข้อมูล

```
Statement stmt = connection.createStatement();  
  
if(!connection.isClosed()){  
    String sql="select * from test";  
    ResultSet rs= stmt.executeQuery(sql);  
    out.print("<h2>ข้อมูล test</h2>");  
    while (rs.next()){  
        out.print(rs.getString("id")+ " , ");  
    }  
}
```



```
        out.print(rs.getString("name")+ " , ");
        out.print(rs.getString("salary")+ " , ");
        out.print(rs.getString("address"));
        out.print("<br>");
    }
    //out.println("Successfully connected show_test");
    connection.close();
}
```

#### 4. ทดสอบการแสดงผลจาก Browser

### **ข้อมูล test**

101 , aaaaaa , 20000 , ชลบุรี  
102 , bbbbbb , 15000 , กรุงเทพ