ใบงาน

รหัสวิชา 30901-2006 วิชา การพัฒนาเว็บด้วยเทคโนโลยีจาวา

Rel MANAR PROPERTY AND A STATE OF A STATE OF

A M SI R

ส

9/

Ô

นิ

0



ใบงาน ชื่องาน ติดตั้งโปรแกรม JDK โปรแกรม Tomcat และ โปรแกรม Edit Plus

ชื่อวิชา การพัฒนาเว็บด้วยเทคโนโลยีจาวา **รหัสวิชา** 30901-2006

ขั้นตอนการติดตั้งโปรแกรม JDK

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

ORACLE Products Industries Re	esources Custome	ers Partners Deve	elopers Events	Company	۵	View Accounts	Contact Sales
Java downloads Tools and resources Java archiv	e						
Looking	for other Jav	a downloads?	OpenJDK Earl	ly Access Build	Is JRE for Cor	nsumers	
Java 20 and Java 17 available now							
JDK 20 is the latest release of Java SE Platform and JDK platform.	17 LTS is the latest	long-term support r	elease for the Ja	va SE	Learn about Java	a SE Subscription	
JDK 20 JDK 17							
JDK Development Kit 20.0.1 downloads	JDK Development Kit 20.0.1 downloads						
JDK 20 binaries are free to use in production and free to	IDK 20 binaries are free to use in production and free to redistribute at no cost, under the Oracle No-Fee Terms and Conditions						
JDK 20 will receive updates under these terms, until Sep	tember 2023 when	it will be superseder	d by JDK 21.				
Linux macOS Windows							
Product/file description	File size	Download					
x64 Compressed Archive	180.81 MB	https://download.	oracle.com/java,	/20/latest/jo	dk-20_windows-x	64_bin.zip (sha256)	
xó4 Installer	159.95 MB	https://download.	oracle.com/java,	/20/latest/jo	lk-20_windows-x	64_bin.exe (sha256)	
x64 MSI Installer	158.74 MB	https://download.	oracle.com/java	/20/latest/jo	lk-20_windows-x	64_bin.msi (sha256)	-

2. ทำการติดตั้งโดยไม่ต้องปรับแต่งอะไร

ขั้นตอนการติดตั้งโปรแกรม Tomcat

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

← → C 🔒 tomcat.apa	ache.org/download-10.cgi
Wiki Migration Guida	10.1.8
Presentations Specifications	Please see the <u>README</u> file for packaging information. It explains what every distribution contains.
Problems?	Binary Distributions
Security Reports Find help FAQ Mailing Lists Bug Database IRC Get Involved Overview Source code Buildbot Translations Tools Media Tuitter	 Core: zip (pgp, sha512) tar.g2 (pgp, sha512) 32-bit Windows zip (pgp, sha512) <u>64-bit Windows zip (pgp, sha512)</u> <u>32-bit/64-bit Windows Service Installer (pgp, sha512)</u> Full documentation: tar.g2 (pgp, sha512) <u>512 (pgp, sha512)</u> <u>512 (pgp, sha512)</u> <u>512 (pgp, sha512)</u> <u>512 (pgp, sha512)</u> <u>512 (pgp, sha512)</u>
YouTube	Source Code Distributions
Blog	(
Misc Who We Are	 tar.gz (pgp, sha512) zip (pgp, sha512)

2. ทำการติดตั้งโดยมีหน้าที่ต้องปรับแต่งคือ username และ password ไม่ต้องใส่

🛃 Apache Tomcat Setup: Co	nfiguration Opt	tions	
Configuration Tomcat basic configuration.			*
Server Shutdown Port		8005	
HTTP/1.1 Connector Port		8080	
AJP/1.3 Connector Port		8009	
Windows Service Name		Tomcat8	
Create shortcuts for all users	;		
Tomcat Administrator Login	User Name		
(opuonai)	Password		
	Roles	manager-gui	
Nullsoft Install System v2.51 —			
		< Back	lext > Cancel

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

🗮 Apache Tomcat Setup	_	- [×
Choose Install Location		0	- Fer
Choose the folder in which to install Apache Tomcat.			
Setup will install Apache Tomcat in the following folder. To install in a Browse and select another folder. Click Install to start the installatio	a differe n.	nt folder	, click
Destination Folder			
D:\Tomcat 9.0		Browse.	
Space required: 14.0 MB			
Space available: 17.2 GB			
Nullsoft Install System v3,06,1			
< Back	Install		Cancel

วิธีการ RUN Tomcat

search => monitor tomcat



Image: Apache Tomcat 9.0 Tomcat9 Properties				
General Log On	Logging Java	Startup	Shutdown	
Service Name:	Tomcat9			
Display name:	Apache Tom	cat 9.0 Tom	cat9	
Description:	Apache Tom	cat 9.0.54 S	erver - https:/	//tomcat.apache
Path to executat	de:			
"D:\Tomcat 9.0	\bin\Tomcat9.e>	ke" //RS//To	mcat9	
Startup type:	Manual			\sim
Service Status: Started				
Start Stop Pause Restart				
		ОК	Cancel	Apply

http://localhost:8080

← → C (i) localhost:8080		
🗰 Apps G Google 🌘 Speedtest.net	by O 🚺 UniFi [CSV To SQL Conver 🔺 DNS chontech.ac.t ตาวน์โหลดฟจี]แหล่งแ 🦁 Web Speech API 🔡 User worakitw Tin
Home D	ocumentation Configuration Examples Wiki Mailing Lists	Find Help
Apache	Fomcat/9.0.54	APACHE' SOFTWARE FOUNDATION
	If you're seeing this, you've successfully installed	d Tomcat. Congratulations!
X	Recommended Reading: Security Considerations How-To Manager Application How-To Clustering/Session Replication How-To	Server Status Manager App Host Manager
Develope	er Quick Start	
<u>Tomcat Sett</u> <u>First Web A</u>	upplication JDBC DataSources	Servlet Specifications Tomcat Versions

ทดสอบตำแหน่งการเก็บหน้าเว็บ

แก้ไขไฟล์ที่ D:/Tomcat_10.1/webapp/ROOT/index.jsp

ใส่ชื่อภาษาอังกฤษใน tag body

html			
<html lang="en"></html>			
<head></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"/>			
<body></body>			
<h1>worakit wiriyakasamongkol</h1>			

ผลลัพธ์

③ 127.0.0.1:8080		
	worakit wiriyakasamongkol	
	Home Documentation Configuration Examples Wiki Mailing Lists	Find Help
	Apache Tomcat/10.1.1	PACHE SOFTWARE FOUNDATION http://www.apache.org/
	If you're seeing this, you've successfully installed Tomcat. Congratulatio	ns!
	Recommended Reading: Security Considerations How-To	Server Status Manager App

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 โดยการ สร้าง ไดเร็คทอรี่ ชื่อ classes ขึ้นมาใหม่

Test	Servlet.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.*;

Λ

TestServlet.java:3: cannot find symbol

symbol: class HttpServlet

public class TestServlet extends HttpServlet {

Λ

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

System Properties ?>	<
System Restore Automatic Updates Remote General Computer Name Hardware Advanced	
You must be logged on as an Administrator to make most of these changes. Performance Visual effects, processor scheduling, memory usage, and virtual memory Settings	เลือก Tab
User Profiles Desktop settings related to your logon Settings	
Startup and Recovery System startup, system failure, and debugging information Settings	
Environment Variables Excer Reporting	คลิกที่ปุ่ม Environment Variables
	1

 ในหน้าต่าง Environment Variables จะแบ่งออกเป็นสองส่วน ส่วนของ User ที่ login ใช้งาน สิ่งที่ กำหนดในส่วนนี้จะใช้เฉพาะ User คนนั้น และส่วนของ System ที่ไม่ว่า User คนไหน Login เข้ามาก็จะใช้ค่าที่ กำหนดร่วมกัน สามารถตั้งค่า Class Path ที่ส่วนไหนก็ได้ เลือก New

Environment Varial	bles	? ×	
User variables for wo	rakit		
Variable	Value	-	
PATH	C:\Program Files\SSH Communications S		
TEMP	C:\Documents and Settings\worakit\Loc		
TMP	C:\Documents and Settings\worakit\Loc		
	New Edit Delete	1	เดือก Now
System variables			
Variable	Value		
ComSpec	C:\WINDOWS\system32\cmd.exe		
FP_NO_HOST_C	NO -	-	
NUMBER_OF_P	2		
OS	Windows_NI		
Path	C:(WINDOWS(system32)C:(WINDOWS)	-	
	New Edit Delete		
	OK Cancel		

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 โดยใส่ใน ช่องว่างดังนี้

วิทยาลัยเทคนิคชลบุรี	วรกิจ	วิริยะเกษามงคล	ผู้จัดทำ
----------------------	-------	----------------	----------

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

การที่ต้องใส่เครื่องหมายจุดแล้วตามด้วย ; เพื่อเป็นการให้หาไฟล์ .class ในไดเร็คทอรี่ที่เก็บไฟล์ ก่อน ถ้าหาไม่พบจึงเรียกไฟล์ servlet-api.jar

New User Variabl	e ? 🗙
Variable name: Variable value:	classpath .;C:\Tomcat 6.0\lib\servlet-api.jar
	OK Cancel

5. เมื่อตั้งค่าเสร็จเรียบร้อยแล้ว ต้องทำการ Restart ระบบใหม่เพื่อให้มีการรับค่าที่ตั้งใหม่

ขั้นตอนการปฏิบัติงาน

1. เขียนโปรแกรม Servlet ประกอบไปด้วยโครงสร้างพื้นฐานดังภาพ 3.12 ให้แสดงผลข้อความ "Hello Servlet"

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

4. แก้ไขไฟล์ web.xml ให้สามารถเรียก class ที่ compile ได้



- 5. แสดงผลโดยการเรียกชื่อไฟล์จาก Browser
- 6. ตรวจสอบความสำเร็จจากการแสดงผลจาก Browser

ใบงาน ชื่องาน เขียนโปรแกรมนับจำนวนครั้งการ refresh เบราว์เซอร์

การเขียนโปรแกรม JSP จำเป็นต้องทราบถึง tag แบบต่างๆ ที่สามารถนำไปใช้งานได้ ซึ่ง tag ของ JSP แบ่ง ออกเป็น 4 ประเภท ดังตาราง

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

การใช้ Tag Expression

เป็น tag ที่ใช้สัญลักษณ์ <%= %> โดยที่เครื่องหมายเท่ากับ(=)ต้องอยู่ติดกับเครื่องหมาย เปอร์เซ็นต์ (%) ถ้ามีการเว้นซ่องว่างหรือมีการเคาะเว้นวรรค จะถือว่าเป็น tag ที่ไม่ถูกต้อง

ใช้สำหรับแสดงค่า output ออกทางจอภาพหรือทางโปรแกรมเบราเซอร์ จะเป็นค่าของ ตัวแปรหรือ เป็นการเรียกใช้ฟังชั่นก็ได้ โดยที่ค่าของตัวแปรที่จะนำมาแสดงผลจะต้องมีการแปลง ให้มีรูปแบบเป็นตัวอักษร ก่อนเสมอ

ตัวอย่าง การใช้ tag Expression

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

Exp	pression.jsp
1	<html></html>
2	<head></head>
3	<title> New Document </title>
4	

5	<body></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	
10	

ผลการทำงานของโปรแกรม Expresstion.jsp แสดงผลบนโปรแกรม Internet Explorer แสดงข้อความ แสดง ค่าสูงสุดของเลขสองจำนวนจาก Method Max() และแสดงค่าของวันเวลาปัจจุบันด้วย Method Date()



การใช้ Tag Scriptlet

เป็น tag ที่ใช้สัญลักษณ์ <% %> ใช้สำหรับแทรกโค็ดภาษา JAVA ลงใน JSP คำสั่งต่างๆ ของ ภาษา JAVA ต้องปิดท้ายด้วย ; (semicolon) ในแต่ละคำสั่ง

ตัวอย่าง การใช้ tag Scriptlet

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

scri	plet.jsp	
1	<html></html>	
2	<head></head>	
3	<title> test Scriptlet </title>	

۵		
-		
5		
-		
6	<body></body>	
Ū.		
7	<%	
	(70	
8	int num = 10	
9	out print("number = $"+num$).	
·		
10	%>	
	, .	
11		
12		

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

scri	plet2.jsp	
1	<html></html>	
2	<head></head>	
3	<title> test Scriptlet </title>	
4		
5	<body></body>	
6	<table border="1"></table>	
7	<% for (int i=1;i<=10;i++){ %>	
8	<tr></tr>	
9	<% for (int j=1;j<=10;j++){ %>	
10	<td><%= i*j %></td>	<%= i*j %>
11	<% } %>	
12		
13	<% } %>	
14		

| 15 | |
| 16 | |
บันทึกไฟล์ชื่อ scriplet_2.jsp ไว้ตำแหน่ง C:\Tomcat 6.0\webapps\ROOT

ตัวอย่างนี้มีการแทรกการวนลูปลงใน tag HTML ในการสร้างตารางขึ้นมา ประกอบการวนลูป 2 ชั้น ลูป นอกใช้ในการกำหนดจำนวนแถว ลูปในใช้กำหนดจำนวนคอลัมน์

บรรทัดที่ 6 และบรรทัดที่ 14 เป็นการเปิดและปิด tag สำหรับสร้างตาราง ในการสร้างตารางจะต้อง ประกอบด้วย tag ของแถวคือ <TR> และในแต่ละแถวจะมี tag <TD> เท่ากับจำนวนช่องในแต่ละแถว บรรทัดที่ 7 เป็นการแทรก tag JSP แบบ for loop จำนวน 10 รอบ เข้าไประหว่าง tag HTML และมี การปิด for loop ในบรรทัดที่ 13 ส่วนบรรทัดที่อยู่ระหว่างบรรทัดที่ 7-13 นี้จะต้องทำงานทั้งหมด 10 รอบ

บรรทัดที่ 9 เป็นการแทรก tag JSP แบบ for loop จำนวน 10 รอบ เข้าไประหว่าง tag HTML และมี การปิด for loop ในบรรทัดที่ 11 ภายใน loop เป็นการแสดงค่า i คูณกับค่า j ซึ่งการแสดงผลคูณของ i กับ j อยู่ภายใต้ tag scriptlet เพื่อให้แสดงค่าผลคูณออกมา ซึ่งผลคูณที่แสดงออกมาจะอยู่ภายใต้ tag HTML มีจำนวน เท่ากับจำนวนรอบที่ for loop กระทำ ในที่นี้จะกระทำ 10 รอบต่อ 1 แถว

รันโปรแกรมจาก http://localhost:8080/scriptlet_2.jsp จะได้ผลลัพธ์ดังภาพ

🧷 te	est S	crip	tlet	- Wi	indo	ws	nte	net	Explo	rer	
0	٢	- [栏 h	ttp://	locall	nost:	8080,	/scrip	let_2.js	p	
File	Edit	Vie	w	Favo	rites	То	ools	Hel	p		
🚖 🤞	Ŷ	2	test S	5cript	let						
			_								
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(@)ต้องอยู่ติดกับเครื่องหมาย เปอร์เซ็นต์ (%) ถ้ามีการเว้นช่องว่างหรือมีการเคาะเว้นวรรค จะถือว่าเป็น tag ที่ไม่ถูกต้อง

ใช้สำหรับแทรกโค๊ดของไฟล์อื่นเข้ามา ด้วยคำสั่ง include ต้องอยู่ภายใน tag นี้ หรือใช้ในการกำหนดค่า แอตทริบิวต์ (Attribute) ต่างๆ ของ JSP ด้วยคำสั่ง page ตามด้วยแอตทริบิวต์ และค่าของแอตทริบิวต์นั้นๆ เช่น

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

เป็นการเรียกใช้ไฟล์ header.html แทรกลงในไฟล์ JSP ในที่นี้เป็นการสร้างส่วนบนของหน้าจอ ซึ่งทุกหน้าจะมีส่วนบนเหมือนๆกัน ถ้าต้องเขียนโค๊ดใหม่ทุกครั้ง ก็จะเป็นการเขียนโค๊ดเดิมๆ ทุกหน้า เวลาแก้ไขก็ ต้องเข้าไปแก้ไขทุกไฟล์ ถ้าใช้การ include เวลาแก้ไขสามารถแก้ไขไฟล์เดียว มีผลทุกหน้า

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

เป็นการกำหนดค่าแอตทริบิวต์ของ JSP เพื่อบอกให้ทราบว่าคลาสที่ถูกเรียกใช้นั้นสามารถหาได้จากที่ใด จากเพ็กเกจใด ในที่นี้ให้ไปหาจากเพ็กเกจ java.util

<%@ page autoFlush="true" %>

เป็นการกำหนดค่าแอตทริบิวต์ของ JSP เพื่อบอกให้ทราบว่าเมื่อบัฟเฟอร์เต็ม จะให้จัดการอย่างไรกับ บัฟเฟอร์ ค่าที่สามารถใส่ได้คือ true กับ false ถ้าใส่ค่าเป็น true จะทำการเคลียบัฟเฟอร์ให้อัตโนมัติ

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

เป็นการกำหนดค่าแอตทริบิวต์ของ JSP เพื่อกำหนดขนาดของค่าของบัฟเฟอร์ใหม่ โดยปกติเมื่อไม่มีการ กำหนดค่าใดๆ ขนาดของบัฟเฟอร์จะมีค่า 8 kb (หน่วยเป็นกิโลไบต์)

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

เป็นการกำหนดค่าแอตทริบิวต์ของ JSP เพื่อกำหนดค่าของการแสดงภาษาในหน้านั้นๆ ว่ามีการใช้การ แปลงโค๊ดอักษรแบบใด โดยปกติค่าเริ่มต้นจะเป็น text/html; charset=ISO8859-1 ซึ่งไม่สามารถแสดงตัวอักษร ภาษาไทยได้ 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 ค่า ของแอตทริบิวต์นี้มีค่าเป็น true และ false โดยค่าเริ่มต้นจะเป็น false ถ้าไม่มีการกำหนดให้เป็น true เมื่อมีการ โยน exception มาก็จะไม่แสดงหน้านี้ออกมา เวลาใช้งานจะใช้คู่กับ errorPage

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

เป็นการกำหนดค่าแอตทริบิวต์ของ JSP เพื่อกำหนดว่ามีการสืบทอดคุณสมบัติของ class แม่มาจากที่ใด ซึ่งเมื่อมีการสืบทอดสามารถที่จะใช้ตัวแปรหรือ method ของ class แม่ได้

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

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

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

เป็นการกำหนดค่าแอตทริบิวต์ของ JSP เพื่อกำหนดการใช้งานของหน้านั้นๆว่าต้องการดึงข้อมูลจาก session มาใช้หรือไม่ ค่าที่กำหนดได้มี 2 ค่า คือ เป็น true และ false โดยค่าเริ่มต้นจะเป็น true ซึ่งสามารถดึงค่า ของ session มาใช้งานได้

ตัวอย่าง การใช้งาน Tag Directive (include และ import)

สร้างไฟล์ header.html โดยนำรูป jspLogo มาใส่

he	eader.html	
1	<html></html>	
2	<body></body>	
3		
4		
5		

สร้างไดเร็กทอรี่ pic ขึ้นที่ C:\Tomcat 6.0\webapps\ROOT และนำไฟล์ jspLogo.jpg มาเก็บไว้ที่ C:\Tomcat 6.0\webapps\ROOT\pic ซึ่งเมื่อทดสอบการทำงานของ header.html โดยการเรียกรันโปรแกรม จาก 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

dire	ctive_1.jsp
1	<html></html>
2	<head></head>
3	<title> directive </title>
4	
5	
6	<body></body>
7	<%@ include file="header.html" %>
8	<%@ page import="java.util.*" %>
9	<h2>Date : <%= new Date() %></h2>
10	
11	

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 จะได้ผลลัพธ์ดังภาพ



ใบงาน ชื่องาน เขียนโปรแกรมส่งข้อมูล form textbox ไปแสดงผลใน JSP **ชื่อวิชา** การพัฒนาเว็บด้วยเทคโนโลยีจาวา **รหัสวิชา** 30901-2006

รับข้อมูลด้วย textbox

เป็นการรับข้อมูลจากผู้ใช้งานในรูปแบบข้อความ เช่น ชื่อ นามสกุล รหัสผู้ใช้งาน ซึ่งเป็นข้อความที่มี ความยาวไม่มากนัก ไม่เกิน 1 บรรทัด เราจะใช้ textbox ในการรับค่าของข้อมูล

ຽູປແບບ

<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

admin

การส่งข้อมูลจาก form ไปประมวลผล

จากการที่ได้สร้าง form ขึ้นมาให้มีหน้าตารับค่าแบบต่างๆ เมื่อทำการคลิกเมาส์ที่ปุ่ม submit ค่าของตัว แปรต่างๆ จะต้องถูกส่งไปประมวลผลในไฟล์ JSP ฉะนั้นจะต้องมีไฟล์ 2 ไฟล์ คือไฟล์แรกเป็น form สำหรับรับ ข้อมูลจากผู้ใช้และมีการระบุว่าข้อมูลใน form จะส่งไปยังไฟล์ใด ในการประมวลผล ไฟล์ที่สองเป็นไฟล์ JSP ที่ form ระบุมาว่าส่งมาที่ไฟล์ JSP ใด ไฟล์ JSP จะนำค่าต่างๆ ที่ถูกส่งมาทำการประมวลผลและแสดงผลทางเบรา เซอร์

รูปแบบ tag ของ form ต้องมีรูปแบบและบอกว่าจะส่งข้อมูลที่รับมาไปยังที่ใด รูปแบบของ form มี ลักษณะดังนี้

```
<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 โดยสามารถ เขียนโค๊ดดังนี้

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

ใบงาน ชื่องาน เขียนโปรแกรมส่งข้อมูล form checkbox ไปแสดงผลใน JSP **ชื่อวิชา** การพัฒนาเว็บด้วยเทคโนโลยีจาวา **รหัสวิชา** 30901-2006

รับข้อมูลด้วย checkbox

เป็นการรับข้อมูลจากผู้ใช้งานในรูปแบบตัวเลือกที่ผู้ใช้งานสามารถเลือกได้มากกว่า 1 รายการ โดย ต้องการเลือกรายการใดก็ใช้เมาส์คลิกที่ช่องสี่เหลี่ยมหน้ารายการนั้น

ູລູປແບບ

<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 ค่าของตัว แปร color ก็จะมีค่าตามที่ได้เลือกไว้ ค่าของตัวแปรแบบ checkbox สามารถมีได้หลายค่า ตามแต่จะเลือกค่าใดไว้ บ้าง โดยค่าของตัวแปรแบบ checkbox จะถูกเก็บไว้ในรูปแบบตัวแปร array

ถ้าต้องการให้มีการเลือกตั้งแต่แรกในการทำงาน ให้ใส่คำว่า CHECKED ไว้ใน tag นั้น

การส่งข้อมูลจาก form ไปประมวลผล

จากการที่ได้สร้าง form ขึ้นมาให้มีหน้าตารับค่าแบบต่างๆ เมื่อทำการคลิกเมาส์ที่ปุ่ม submit ค่าของตัว แปรต่างๆ จะต้องถูกส่งไปประมวลผลในไฟล์ JSP ฉะนั้นจะต้องมีไฟล์ 2 ไฟล์ คือไฟล์แรกเป็น form สำหรับรับ ข้อมูลจากผู้ใช้และมีการระบุว่าข้อมูลใน form จะส่งไปยังไฟล์ใด ในการประมวลผล ไฟล์ที่สองเป็นไฟล์ JSP ที่ form ระบุมาว่าส่งมาที่ไฟล์ JSP ใด ไฟล์ JSP จะนำค่าต่างๆ ที่ถูกส่งมาทำการประมวลผลและแสดงผลทางเบรา เซอร์

รูปแบบ tag ของ form ต้องมีรูปแบบและบอกว่าจะส่งข้อมูลที่รับมาไปยังที่ใด รูปแบบของ form มี ลักษณะดังนี้

```
<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 สามารถรับค่าได้เหมือน ตัวอย่างด้านบนเพราะมีค่าที่ส่งมา 1 ค่าต่อ 1 ตัวแปร แต่สำหรับ checkbox และ listbox ที่มีการกำหนด MULTIPLE มีการส่งค่ามากกว่า 1 ค่าต่อ 1 ตัวแปร ในการรับค่าจึงต้องใช้ตัวแปรแบบ array ในการรับค่า และ คำสั่งในการรับค่าแบบหลายค่าคือ

request.getParameterValues("ชื่อของ Element");

ใบงาน ชื่องาน เขียนโปรแกรมส่งข้อมูล form checkbox ไปแสดงผลใน JSP **ชื่อวิชา** การพัฒนาเว็บด้วยเทคโนโลยีจาวา **รหัสวิชา** 30901-2006

รับข้อมูลด้วย checkbox

เป็นการรับข้อมูลจากผู้ใช้งานในรูปแบบตัวเลือกที่ผู้ใช้งานสามารถเลือกได้มากกว่า 1 รายการ โดย ต้องการเลือกรายการใดก็ใช้เมาส์คลิกที่ช่องสี่เหลี่ยมหน้ารายการนั้น

ູລູປແບບ

<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 ค่าของตัว แปร color ก็จะมีค่าตามที่ได้เลือกไว้ ค่าของตัวแปรแบบ checkbox สามารถมีได้หลายค่า ตามแต่จะเลือกค่าใดไว้ บ้าง โดยค่าของตัวแปรแบบ checkbox จะถูกเก็บไว้ในรูปแบบตัวแปร array

ถ้าต้องการให้มีการเลือกตั้งแต่แรกในการทำงาน ให้ใส่คำว่า CHECKED ไว้ใน tag นั้น

การส่งข้อมูลจาก form ไปประมวลผล

จากการที่ได้สร้าง form ขึ้นมาให้มีหน้าตารับค่าแบบต่างๆ เมื่อทำการคลิกเมาส์ที่ปุ่ม submit ค่าของตัว แปรต่างๆ จะต้องถูกส่งไปประมวลผลในไฟล์ JSP ฉะนั้นจะต้องมีไฟล์ 2 ไฟล์ คือไฟล์แรกเป็น form สำหรับรับ ข้อมูลจากผู้ใช้และมีการระบุว่าข้อมูลใน form จะส่งไปยังไฟล์ใด ในการประมวลผล ไฟล์ที่สองเป็นไฟล์ JSP ที่ form ระบุมาว่าส่งมาที่ไฟล์ JSP ใด ไฟล์ JSP จะนำค่าต่างๆ ที่ถูกส่งมาทำการประมวลผลและแสดงผลทางเบรา เซอร์

รูปแบบ tag ของ form ต้องมีรูปแบบและบอกว่าจะส่งข้อมูลที่รับมาไปยังที่ใด รูปแบบของ form มี ลักษณะดังนี้

```
<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 สามารถรับค่าได้เหมือน ตัวอย่างด้านบนเพราะมีค่าที่ส่งมา 1 ค่าต่อ 1 ตัวแปร แต่สำหรับ checkbox และ listbox ที่มีการกำหนด MULTIPLE มีการส่งค่ามากกว่า 1 ค่าต่อ 1 ตัวแปร ในการรับค่าจึงต้องใช้ตัวแปรแบบ array ในการรับค่า และ คำสั่งในการรับค่าแบบหลายค่าคือ

request.getParameterValues("ชื่อของ Element");

ใบงาน ชื่องาน เขียนโปรแกรมใช้งาน Cokkie

ชื่อวิชา การพัฒนาเว็บด้วยเทคโนโลยีจาวา **รหัสวิชา** 30901-2006

การใช้งาน Cookie

หลังจากผู้ใช้งานเปิดเบราเซอร์ติดต่อไปยัง web server จะมีการตรวจสอบว่าเครื่องของผู้ใช้งานเครื่องนี้มี การเก็บค่า Cookie ไว้หรือไม่ ถ้ามีการเก็บค่าเอาไว้ ก็สามารถเข้าสู่เว็บไซด์ ได้โดยไม่ต้องทำการ login ก่อน ใช้ชื่อ ผู้ใช้งานตามข้อมูลที่เก็บใน Cookie แต่ถ้าไม่มีข้อมูลใน Cookie ก็จะเข้าสู่หน้า login ทำการใส่ username และ รหัสผ่านครั้งแรกก่อน เมื่อการ login ถูกต้อง เครื่องที่ใช้งานนี้อยู่ก็จะถูกเขียนข้อมูลเล็กๆ ลงในเครื่องทำให้การเข้า มาในครั้งต่อไปไม่จำเป็นต้อง 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()

ในงาน ้ชื่องาน เขียนโปรแกรมติดต่อฐานข้อมูล Mysql

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
วิทยาลัยเทคนิคชลบุรี
```

วรกิจ วิริยะเกษามงคล ผู้จัดทำ

```
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

การแสดงข้อมูลจากฐานข้อมูล

1. สร้างฐานข้อมูลและตารางข้อมูล

🕞 Create database 💿					
jsp64	utf8_g	eneral_ci		~	Create
Table name: test				Add	1
Name	Туре 😡		Lengt	n/Values 😡	_
id	VARCHAR	~	10		
name	VARCHAR	~	50		
salary	INT	~			
address	VARCHAR	~	100		
 เพิ่มข้อมูล ลงในตาราง 	ที่สร้างไว้				
←T→	▼ id	name	salary	address	
🗆 🥜 Edit 👫 Copy 🌾	Delete 101	aaaaaa	20000	ชลบุรี	
🔲 🕜 Edit 👫 Copy 🌾	Delete 102	bbbbb	15000	กรุงเทพ	
 เขียนคำสั่งแสดงข้อมูล 					J
Statement stmt = conne	ection.createSta	tement();			
if(!connection.isClo	sed()){				
String sql="select	* from test";				
ResultSet rs= stm	t.executeQuery	(sql);			
out.print(" <h2></h2>	>ข้อมูล test	>");			
while (rs.next()){					
out.print(rs.get	tString("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 , bbbbb , 15000 , กรุงเทพ