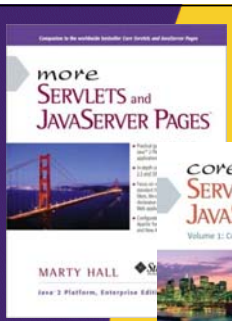
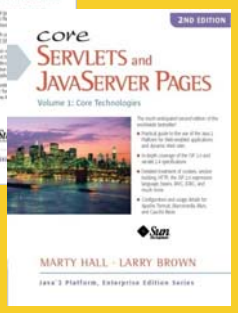
An Overview of Servlet & JSP Technology

Originals of slides and source code for examples: http://courses.coreservlets.com/Course-Materials/csajsp2.html
Also see the JSF tutorial – http://www.coreservlets.com/JSF-Tutorial/jsf2/
customized onsite JSP and servlet training courses – http://courses.coreservlets.com/servlet+jsp-training.html
and customized JSF2 and PrimeFaces training courses – http://courses.coreservlets.com/jsf-training.html





For live training on JSP, JSF 2, and/or PrimeFaces, email hall@coreservlets.com.

Marty is also available for consulting and development support.

Taught by the author of *Core Servlets and JSP*, this tutorial, and JSF 2.2 version of *Core JSF*. Available at public venues, or customized versions can be held on-site at your organization.

- Courses developed and taught by Marty Hall
  - JSF 2.2, PrimeFaces, servlets/JSP, Ajax, jQuery, Android development, Java 7 or 8 programming, custom mix of topics
  - Courses available in any state or country. Maryland/DC area companies can also choose afternoon/evening courses.
- Courses developed and taught by coreservlets.com experts (edited by Marty)
  - Hadoop, Spring, Hibernate/JPA, GWT, RESTful Web Services

Contact hall@coreservlets.com for details

Agenda

- **JSP vs. JSF**
- **What servlets and JSP are all about**
  - Understanding the role of servlets
  - Building Web pages dynamically
  - Evaluating servlets vs. other technologies
  - Understanding the role of JSP
- **Testing Tomcat with Eclipse**
  - Installing Tomcat
  - Installing and starting Eclipse
  - Telling Eclipse about Tomcat
  - Deploying and running Web apps from Eclipse
  - Making new Web apps in Eclipse

Prologue: JSP vs. JSF

JSP vs. JSF 2

- **Servlets and JSP (JavaServer Pages)**
  - Original, widely-deployed standard
  - Used by google.com, ebay.com, walmart.com, and thousands of other popular sites
  - Low level by today's standards
  - Covered in this tutorial
- **JSF (JavaServer Faces) Version 2**
  - An official part of Java EE as of Java EE 6
    - But runs in any recent Java-enabled server, including Tomcat
  - Higher-level features: integrated Ajax support, field validation, page templating, rich third-party component libraries such as PrimeFaces, etc. Designed around the MVC architecture.
  - Recommended for almost all new projects
  - Covered at http://www.coreservlets.com/JSF-Tutorial/jsf2/

JSP vs. JSF: When to Use Which

- **Servlets and JSP**
  - For maintaining and extending existing legacy projects
- **Servlets only**
  - For apps with front ends that do not use a server-side framework
    - E.g., HTML with jQuery and jQuery UI
  - Servlets primarily handle the Ajax requests from jQuery and do not build full pages
- **JSF 2**
  - For almost all new projects that involve dynamic pages
  - Usually combined with a rich component toolkit such as PrimeFaces
    - See http://www.coreservlets.com/JSF-Tutorial/primefaces/

Technologies Used Internally with JSF

- **Servlets**
  - Servlets are still used behind the scenes, and javax.faces.webapp.FacesServlet controls everything
  - Many servlet APIs important in JSF. See next page
- **JSP**
  - Used in JSF 1
    - JSF 1 was cumbersome and relatively weak
  - Replaced by facelets in JSF 2
    - JSF version 2 is dramatically simpler and more powerful than JSF 1, and does not use JSP at all except for backward compatability with legacy JSF 1 projects
    - Facelets are more-or-less HTML pages with tags to insert results, but with no explicit Java-based scripting directly in the page

Servlet Technologies Still Applicable with JSF 2

- **Knowing servlets still useful with JSF 2**
  - Direct servlet APIs not used all that frequently, but are available and still used in some important areas
- **Servlet APIs most commonly used with JSF**
  - Cookies (especially long-lived ones)
  - Setting response headers and response status codes
  - Changing output based on User-Agent
  - Explicit session manipulation
    - E.g., changing inactive interval or invalidating session
  - Security (both programmatic and declarative)
  - More info
    - See “Managed Beans 3” in JSF tutorial at http://www.coreservlets.com/JSF-Tutorial/jsf2/

Bottom Line: If You Are …

- **Maintaining or extending a legacy project**
  - Go through this entire servlet and JSP tutorial in order
- **New to Java-based Web apps and are starting a new project**
  - Skip this entire tutorial and instead use the JSF 2 tutorial at http://www.coreservlets.com/JSF-Tutorial/jsf2/
- **Already experienced with JSF 2 and want to learn the underlying technologies**
  - Use this tutorial, but skip the JSP sections.
  - Concentrate on cookies, session tracking, request headers, response headers, and security

What Servlets and JSP are All About

Why Web Apps?

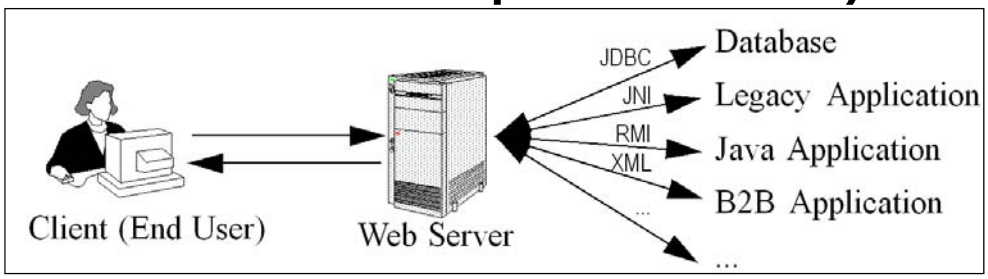
- **Downsides to browser-based apps**
  - GUI is poor
    - HTML is OK for static documents, but lousy for programs
  - Communication is inefficient
    - HTTP is poor protocol for the way we now use Web apps

Why Web Apps? (Continued)

- **So why does everyone want Web apps?**
  - Universal access
    - Everyone already has a browser installed
    - Any computer on the network can access content
  - Automatic "updates"
    - Content comes from server, so is never out of date


A Servlet's Job

- **Read explicit data sent by client (form data)**
- **Read implicit data sent by client (request headers)**
- **Generate the results**
- **Send the explicit data back to client (HTML)**
- **Send the implicit data to client (status codes and response headers)**

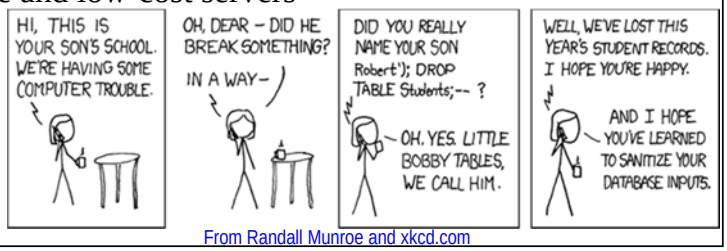


Why Build Web Pages Dynamically?

- **The Web page is based on data submitted by the user**
  - E.g., results page from search engines and order-confirmation pages at on-line stores
- **The Web page is derived from data that changes frequently**
  - E.g., a weather report or news headlines page
- **The Web page uses information from databases or other server-side sources**
  - E.g., an e-commerce site could use a servlet to build a Web page that lists the current price and availability of each item that is for sale.

The Advantages of Servlets Over "Traditional" CGI

- **Efficient**
  - Threads instead of OS processes, one servlet copy
- **Convenient**
  - Lots of high-level utilities
- **Powerful**
  - Sharing data, pooling, persistence
- **Portable**
  - Run on virtually all operating systems and servers
- **Inexpensive**
  - There are plenty of free and low-cost servers
- **Secure**
  - No shell escapes, no buffer overflows
- **Mainstream**
  - See next page



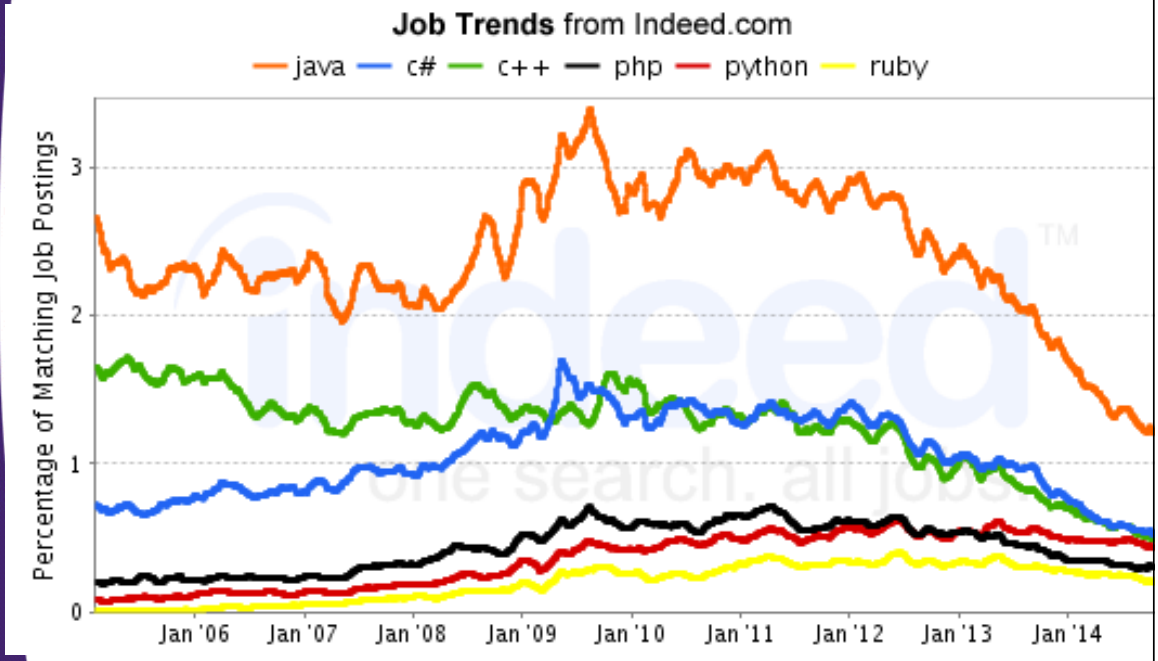
From Randall Munroe and xkcd.com

Mainstream

- **Popular:**
  - The single most common use of Java technology
  - The leading technology for medium/large Web applications
    - Google reports over 650 million Web pages using JSP
- **Supported by:**
  - Apache, Oracle, IBM, Sybase, BEA, Jetty, Caucho, Sun, New Atlanta, ATG, Fujitsu, Lutris, Silverstream, the World Wide Web Consortium (W3C), and many others
  - Plugins for IIS and Zeus
- **Runs on:**
  - Windows, Unix/Linux, MacOS, VMS, and IBM mainframe OSs
- **Used for:**
  - Airline companies, hotels, e-commerce sites, search engines, banks, financial sites, etc., etc., etc.

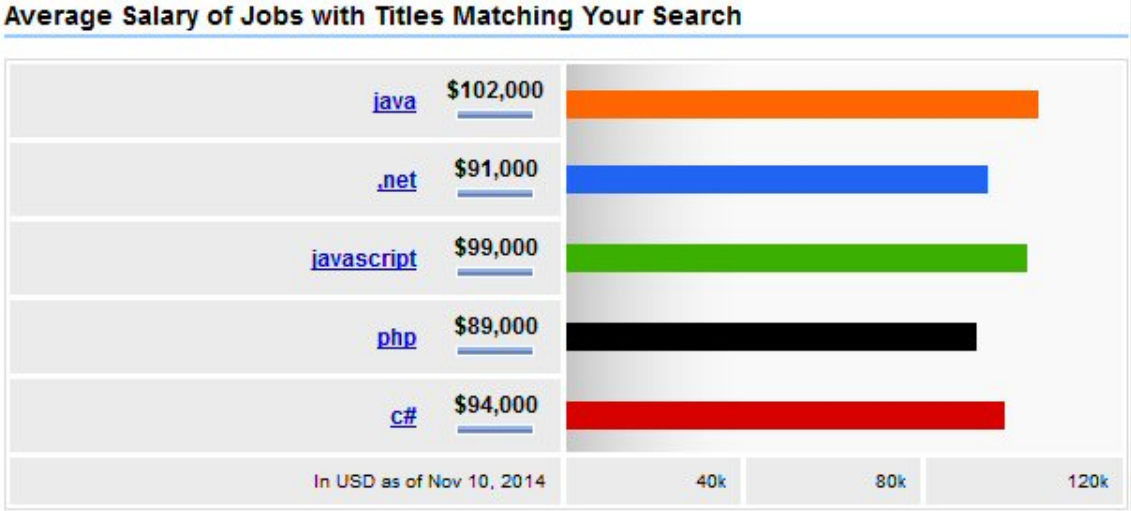
Server-side Java is driving the Web

Web App Language Popularity: Keywords in Job Postings



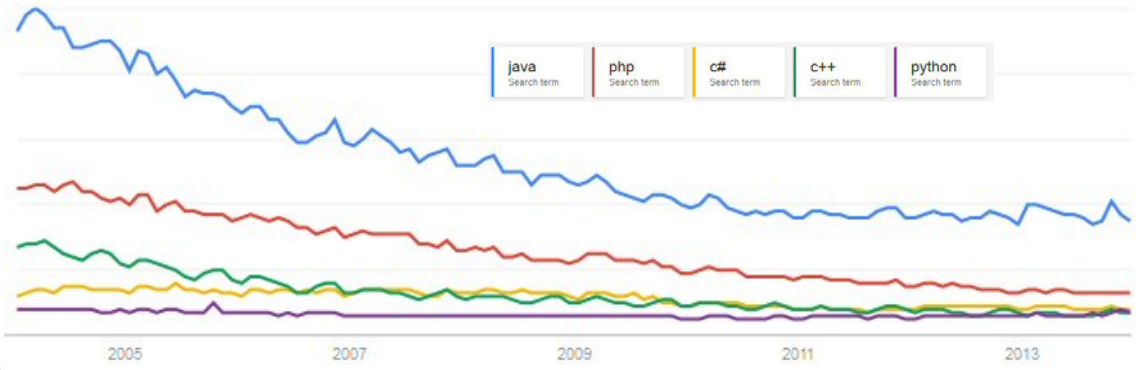
This reflects job postings that contain the keyword in the title or keywords. Since value is in percent, the specific ups and downs are not so relevant (perhaps there was an increase in teacher or construction jobs), but the relative values are instructive.

Web App Language Popularity: USA Salaries



Data for US jobs from indeed.com as of November 2014, averaged over all states.

Web App Language Popularity: Google Searches



This reflects searches at Google

Extending the Power of Servlets: JavaServer Pages (JSP)

- **Idea:**
  - Use regular HTML for most of page
  - Mark dynamic content with special tags
  - Details in second half of course

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD><TITLE>Welcome to Our Store</TITLE></HEAD>
<BODY>
<H1>Welcome to Our Store</H1>
<SMALL>Welcome,
<!-- User name is "New User" for first-time visitors -->
<%= coreservlets.Utils.getUserNameFromCookie(request) %>
To access your account settings, click
<A HREF="Account-Settings.html">here.</A></SMALL>
<P>
Regular HTML for  rest of on-line store's Web page
</BODY></HTML>
```

Accessing the Online Documentation

- **Servlets and JSP**
  - http://docs.coreservlets.com/servlet-3.0-api/
    - Servlets 3.0 and JSP 2.2 (Tomcat 7)
  - http://docs.oracle.com/cd/E17802_01/products/products/servlet/2.5/docs/servlet-2_5-mr2/
    - Servlets 2.5 (Tomcat 6)
  - http://docs.oracle.com/cd/E17802_01/products/products/jsp/2.1/docs/jsp-2_1-pfd2/
    - JSP 2.1 (Tomcat 6)
- **Java 7 and 8**
  - http://docs.oracle.com/javase/7/docs/api/
  - http://docs.oracle.com/javase/8/docs/api/
- **Java 8**
  - Java 8 can make for dramatically faster and more flexible/reusable code. It is highly recommended for use in Web applications.
    - See http://www.coreservlets.com/java-8-tutorial/

Setting Up Tomcat on Your PC

- **Tomcat 7 with Eclipse**
  - http://www.coreservlets.com/Apache-Tomcat-Tutorial/tomcat-7-with-eclipse.html
    - Or, just follow link at top left of www.coreservlets.com
    - More details in next section of this tutorial
- **Tomcat 6 with Eclipse**
  - http://www.coreservlets.com/Apache-Tomcat-Tutorial/
- **For manual execution**
  - http://www.coreservlets.com/Apache-Tomcat-Tutorial/
    - More details in last section.
    - Eclipse or another IDE *strongly* recommended
- **Bottom line**
  - Unzip Tomcat, then point Eclipse at the install folder

Installing Java and Tomcat

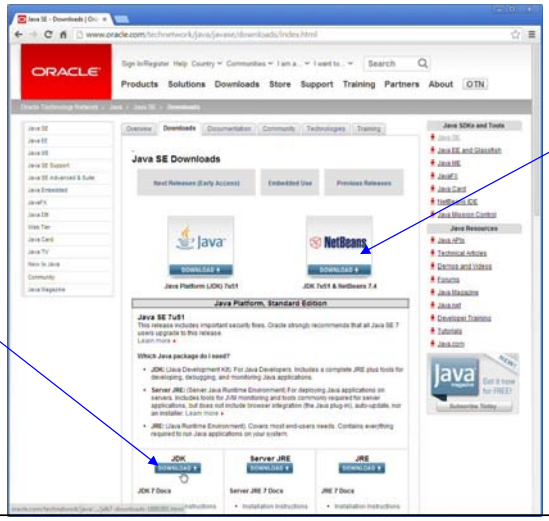
For even more detailed step-by-step instructions, see tutorials on using Eclipse with Tomcat at http://www.coreservlets.com/Apache-Tomcat-Tutorial/

Installing Java SE

- **Minimum Java version**
  - Technically, Java 6 legal, but Java 7 or 8 much better
    - Java 8 is *great* alternative if using Tomcat or Jetty instead of a full Java EE 7 sever
- **Downloading and installation**
  - Follow directions at Oracle site
    http://www.oracle.com/technetwork/java/javase/downloads/
  - Choose "JDK", not "JRE" or "Server JRE"
  - Install and accept all defaults
- **Bookmark the Java API ("JavaDocs")**
  - http://docs.oracle.com/javase/7/docs/api/ or
  - http://docs.oracle.com/javase/8/docs/api/

Installing Java SE

- **Install Java SE 7 or 8**
  - http://www.oracle.com/technetwork/java/javase/downloads/
    - Java 8 final available 3/2014, and is *huge* improvement. See Java 8 tutorial at http://www.coreservlets.com/java-8-tutorial/



Use this version. The "JDK – Java Development Kit" includes compiler for .java files, whereas the "JRE – Java Runtime Environment" is only for executing prebuilt .class files.

This tutorial uses Eclipse, but if you prefer the NetBeans IDE, it is very easy to adapt the instructions to that development environment. So, if you prefer NetBeans or your organization has standardized on it, use this download instead of (not in addition to) the one on the left.

Download and Unzip Tomcat

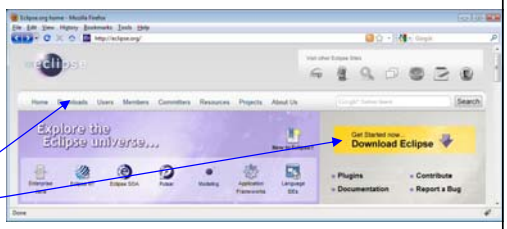
- **Start at http://tomcat.apache.org**
  - Choose download link on left, then ZIP version
    - Tomcat 7 or 8 (recommended)
    - Tomcat 6 (if you need compatibility with older servers, but will not support JSF 2.2 file upload component)
- **Or, go to http://www.coreservlets.com/**
  - Choose Tomcat tutorial from top left
  - This is preconfigured version
    - Set for development, not deployment mode
      - Port changed to 80, servlet reloading enabled, directory listings turned on, etc.
    - Otherwise unchanged
- **Either way, just unzip the file**
  - E.g., resulting in C:\apache-tomcat-7.0.34

Installing Eclipse

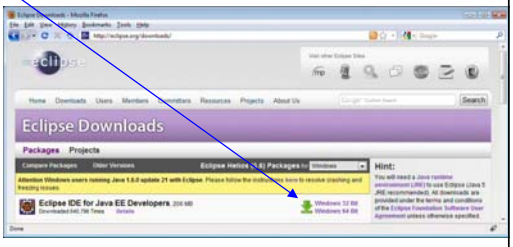
For even more detailed step-by-step instructions, see tutorials on using Eclipse with Tomcat at http://www.coreservlets.com/Apache-Tomcat-Tutorial/

Installing Eclipse

- **Overview**
  - Eclipse is a free open source IDE. Support for Java, Android, HTML, CSS, JavaScript, C++, PHP, JSF, servlets, and more.
    - http://eclipse.org/downloads/
    - Choose "Eclipse IDE for Java EE Developers"



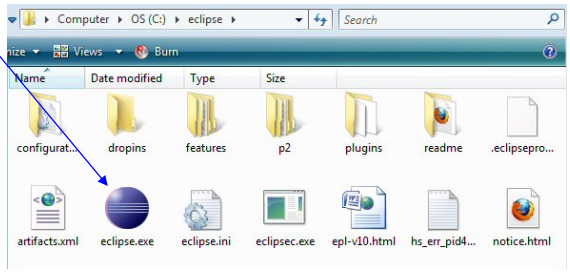
- **Features**
  - Checks your syntax as you type
  - Automatically compiles every time you save file
  - Many tools: refactoring, debugging, server integration, templates for common tasks, etc.
    - Low learning curve: beginners can use Eclipse without knowing these tools



Reminder: step-by-step guide at http://www.coreservlets.com/ (click "Apache Tomcat 7" in top left).

Running Eclipse

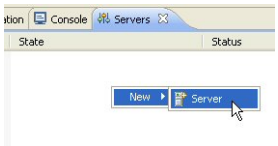
- **Unzip the downloaded file (no installer!)**
  - Call the folder you unzip into "installDir"
- **Double click eclipse.exe**
  - From *installDir*/bin
- **Click on "Workbench" icon**
  - Next time you bring up Eclipse, it will come up in workbench automatically



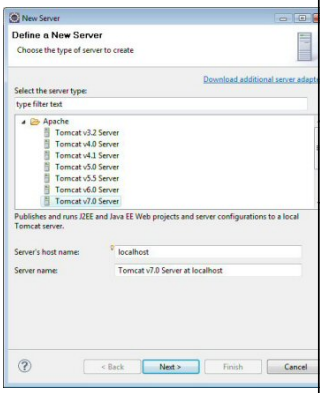
- **Shortcut**
  - Many developers put Eclipse link on their desktop
    - R-click eclipse.exe, Copy, then go to desktop, R-click, and Paste Shortcut (not just Paste!)

Configuring Eclipse

- **Tell Eclipse about Java version**
  - Window → Preferences → Java → Installed JREs → Press "Add", choose "Standard VM", navigate to JDK folder (not "bin" subdirectory)
    - E.g., C:\Program Files\Java\jdk1.7.0_51



- **Tell Eclipse about Tomcat**
  - Click on Servers tab at bottom. R-click in window.
  - New, Server, Apache, Tomcat v7.0, Next, navigate to folder, Finish.



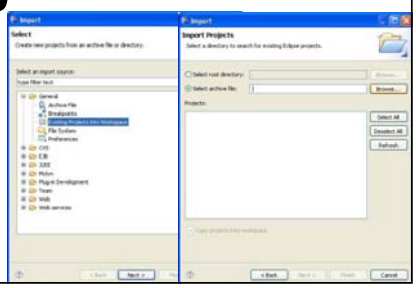
Most recent version (Luna as of 1/2015) is best choice. Old versions have JSF 2 support, but not explicit support for 2.2.

If you lose the "Servers" tab at the bottom of Eclipse, use Window menu, Show View, and hunt for "Servers".

Deploying Apps from Eclipse

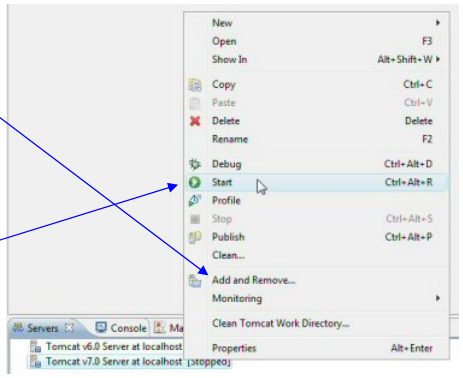
Download and Import Sample Project

- **Get test-app.zip from coreservlets.com**
  - Start at servlet and JSP tutorials
    - http://courses.coreservlets.com/Course-Materials/csajsp2.html
      - Go to first section (Overview and Setup)
  - Or, start at Apache Tomcat tutorial
    - http://www.coreservlets.com/Apache-Tomcat-Tutorial/
      - Choose Tomcat 7 (recommended) or Tomcat 6 version
- **Then, download test-app.zip**
  - Then, import into Eclipse.
    - File, Import, General, Existing Projects, Select archive file. Then click Browse and navigate to test-app.zip.



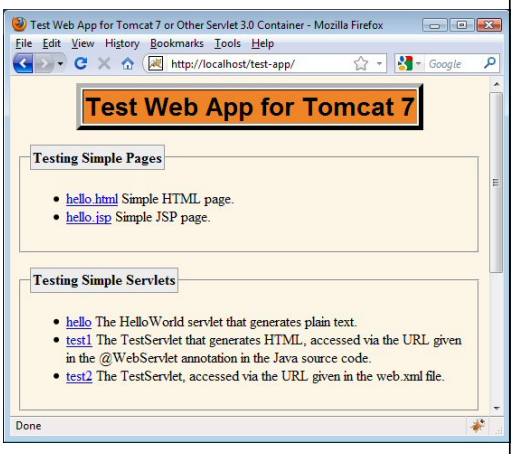
Deploying App in Eclipse

- **Deploy project**
  - Select “Servers” tab at bottom
  - R-click on Tomcat
  - Choose “Add and Remove”
  - Choose project
  - Press “Add”
  - Click “Finish”
- **Start Server**
  - R-click Tomcat at bottom
  - Start (use “Restart” if Tomcat already running)
- **Test URL**
  - http://localhost/test-app/ in any Web browser



Testing Deployed App in Eclipse

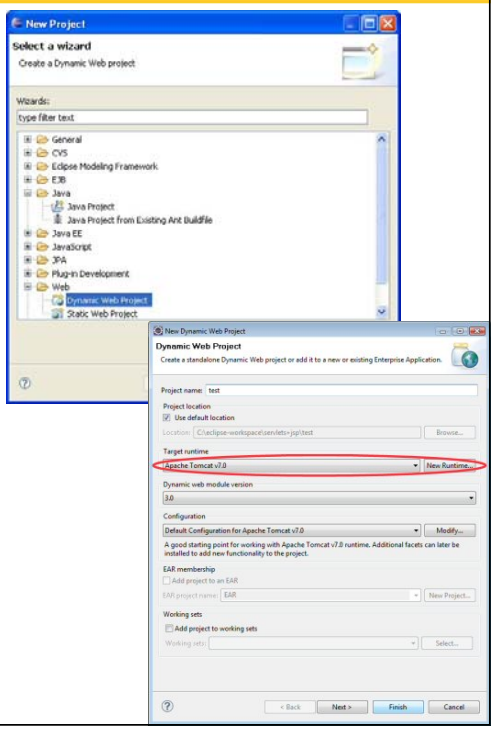
- **Start a browser**
  - Eclipse also has builtin browser, but I prefer to use Firefox, IE, or Chrome separately
- **Test base URL**
  - http://localhost/test-app/
- **Test Web content**
  - http://localhost/test-app/hello.html
  - http://localhost/test-app/hello.jsp
- **Test servlets**
  - http://localhost/test-app/hello
  - http://localhost/test-app/test1
  - http://localhost/test-app/test2



Making New Apps from Eclipse

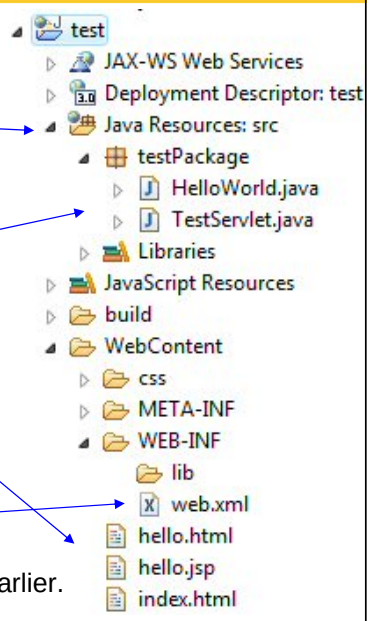
Making Web Apps in Eclipse

- **Make empty project**
  - File → New → Project → Web → Dynamic Web Project
  - For "Target runtime", choose "Apache Tomcat v7.0"
  - Give it a name (e.g., "test")
  - Accept all other defaults
- **Shortcut**
  - If you have made Dynamic Web Project recently in workspace, you can just do File → New → Dynamic Web Project



Adding Code to Eclipse Projects

- **Locations**
  - Java Resources: src
    - R-click and New → Package
    - Never use default package
  - src/testPackage
    - Java code in testPackage package
  - WebContent
    - Web files (HTML, JavaScript, CSS, JSP, images, etc.)
  - WebContent/some-subdirectory
    - Web content in subdirectory
    - R-click on WebContent, New → Folder
  - WebContent/WEB-INF
    - web.xml
      - Optional with servlets 3.0. Required in 2.5 & earlier.
      - Will be discussed later
- **Note**
  - Can cut/paste or drag/drop files into appropriate locations



Testing New App

- **Follow same procedure as "deploying app" from previous section**
  - Deploy project
    - Select "Servers" tab at bottom
    - R-click on Tomcat
    - Choose "Add and Remove"
    - Choose project
    - Press "Add"
    - Click "Finish"
  - Start Server
    - R-click Tomcat at bottom
    - Restart (use "Start" if Tomcat not already running)
  - Test URL
    - http://localhost/*appName*/ in any Web browser

Summary

- **General**
  - Servlets are efficient, portable, powerful, and widely accepted in industry
  - Regardless of deployment server, run a free server on your desktop for development
  - Using Eclipse (or another IDE like NetBeans or IntelliJ IDEA) greatly simplifies development and deployment
  - Strongly consider JSF 2 as an alternative for new projects
    - http://www.coreservlets.com/JSF-Tutorial/jsf2/
- **Getting started**
  - Start with test-app and TestServlet from coreservlets.com
  - Click on "Intermediate Servlets and JSP" tutorial in top-left corner and you can get pre-made Eclipse projects

Questions?

More info:
http://courses.coreservlets.com/Course-Materials/csajsp2.html – Servlet and JSP Tutorial
http://courses.coreservlets.com/servlet+jsp-training.html – Servlet and JSP training courses
http://www.coreservlets.com/Apache-Tomcat-Tutorial/tomcat-7-with-eclipse.html – Tutorial on integrating Apache Tomcat with Eclipse
http://www.coreservlets.com/JSF-Tutorial/jsf2/ – JSF 2.2 tutorial
http://www.coreservlets.com/JSF-Tutorial/primefaces/ – PrimeFaces tutorial
http://courses.coreservlets.com/jsf-training.html – Customized JSF and PrimeFaces training courses
http://coreservlets.com/ – JSF 2, PrimeFaces, Java 7 or 8, Ajax, jQuery, Hadoop, RESTful Web Services, Android, HTML5, Spring, Hibernate, Servlets, JSP, GWT, and other Java EE training