

# Web课程



- 课程目标
- 前置知识
- 课程重难点
- 课程安排
- 学习建议

# 课程目标

- 本门课程主要讲解Web服务器Tomcat、Servlet、Jsp、Cookie、Session、EL、JSTL等技术内容，带领大家从入门到精通系统讲解Java Web开发中的相关知识
- 通过登录、跳转、创建Cookie、Session会话、监听等相关案例让大家达到熟练掌握WEB项目开发中常用技术的应用

# 前置知识

- 需要JavaSE基础知识

安装JDK1.8,配置环境变量、熟练掌握JavaSE基本语法

- 数据库知识

MySql、SQLServer、Oracle、GaussBD

- WEB前端知识

熟练掌握网页技术

# 课程重难点

- 重点

Servlet入门、JSP入门、EL与JSTL章节

- 难点

会话跟踪、监听器与过滤器、Servlet新功能章节

# 课程安排

- 第1章-Web快速入门
- 第2章-Servlet入门
- 第3章-JSP入门
- 第4章-会话跟踪
- 第5章-上下文

# 课程安排

- 第6章-监听器与过滤器
- 第7章-JSP其他主题
- 第8章-MVC模式
- 第9章-EL与JSTL
- 第10章-Servlet新功能

# 学习建议

- 一、听**明**白其原理
- 二、多**敲**代码练习
- 三、勤**查**帮助文档
- 四、**培养**“学习”能力



# Web开发快速入门

本章内容：共2小节，12个知识点

- 第1节：基本概念及Web应用入门
- 第2节：介绍第一个Web应用

# 本章目标

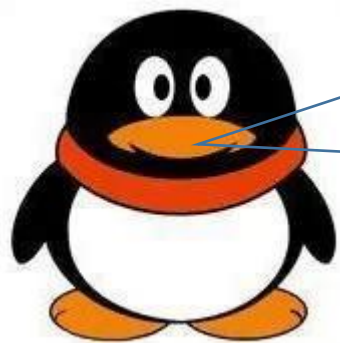
- 了解Web应用的一些基本概念;
- 理解Servlet的基本概念;
- 能够编写第一个简单的Web应用, 并部署运行;

# 第1节 【基本概念及Web应用入门】

- 知识点1: C/S、B/S与RIA
- 知识点2: Web站点、Web应用、HTTP服务器、Web应用服务器
- 知识点3: 开发Web应用的核心技术
- 知识点4: Tomcat介绍及安装
- 知识点5: 常见其他Servlet容器简介
- 知识点6: Servlet的概念及功能

# 知识点1 【C/S、B/S与RIA】 -1

- 说起网络应用，可以分为C/S和B/S两种；
- C/S结构是Client/Server（客户机/服务器）的简称，桌面应用程序采用的多是这种结构；
- B/S结构是Browser/Server（浏览器/服务器）的简称，特点是客户端无需安装特定的软件，只需要安装一个浏览器就可以与系统进行交互。



QQ就是C/S结构的网络应用，你要用QQ聊天，就得在电脑上安装QQ软件。

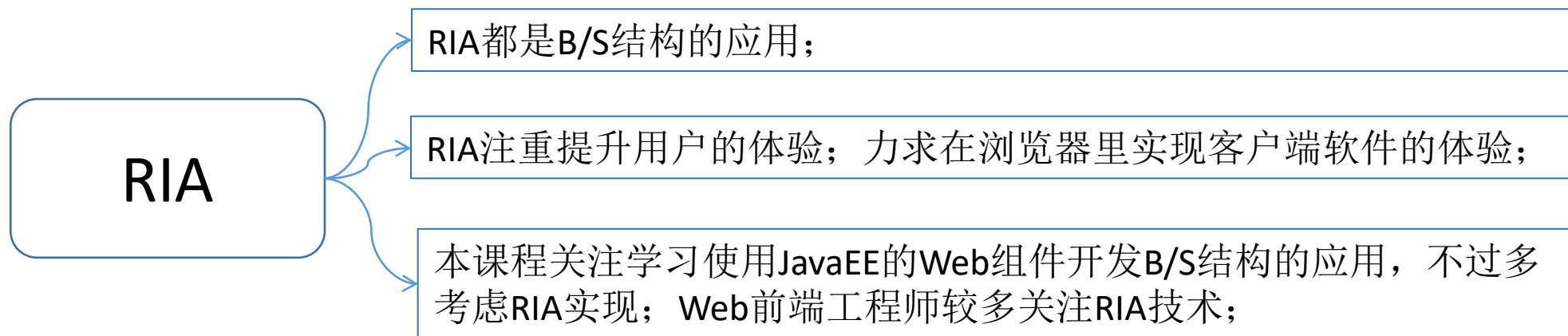


百度就是B/S结构的网络应用，你要使用百度，不需要安装什么，只要有浏览器，能上网就行了。



## 知识点1 【C/S、B/S与RIA】 -2

- C/S需要安装客户端软件，但是操作往往更为便捷；B/S不需要安装客户端软件，但是用户体验往往不如C/S好，安全性也较C/S要低；
- RIA是“Rich Internet Applications”的缩写，意为“富客户端网络应用”；
- RIA的主要目标：基于B/S结构，只要有浏览器就可以使用，在浏览器里实现与客户端软件类似的体验；例如：可以局部刷新、可以拖拽等；



## 知识点2 【Web站点、Web应用、HTTP服务器、Web应用服务器】 -1

- Web站点即Web Site，也被称为Web网站；比如某公司用来作为宣传使用的官方网站，就是Web站点；
- Web应用被称为Web Application，简单来说，就是通过浏览器访问的应用程序，从而为用户提供相关的服务，例如查询、购物、生成报表等；



中软国际公司官网，是一个Web站点，用户能够浏览公司信息，但是该站点不会提供服务。



京东，是一个Web应用，可以为用户提供查询、购物、退货等服务。

我们要学习的是开发Web应用。

## 知识点2 【Web站点、Web应用、HTTP服务器、Web应用服务器】 -2

- HTTP服务器：也称为Web服务器，主要功能是提供网上信息浏览服务，例如Apache、Nginx、IIS是比较常用的HTTP服务器；使用浏览器访问Web站点或者Web应用，则必须通过HTTP服务器；
- Web应用服务器：Web应用服务器能够运行服务器上的应用程序，并将结果返回给客户端浏览器；例如，Tomcat就是一种Web应用服务器；通常情况下，Web应用服务器兼具HTTP服务器的部分功能；

Tips

Web站点只需要浏览信息，所以只需要HTTP服务器即可；

Web应用需要HTTP服务器及Web应用服务器，因为不仅需要浏览信息，还需要运行应用程序；

很多Web应用服务器都兼具HTTP服务器的部分功能，因此本课程中，我们只使用Web应用服务器即可；



## 知识点3 【开发Web应用的核心技术】

- 有很多技术可以开发Web应用;



通用网关接口(Common Gateway Interface), 是一个WEB应用服务器提供信息服务的标准接口, 目前使用较少;



.Net是微软公司的企业应用框架, 其中Asp.net, ADO.net等, 都是Web应用开发的核心技术; 在中小型企业应用使用广泛;



Php是一种相对简单易学的Web开发技术, 目前广泛应用在互联网小型应用;



JavaEE是Oracle公司的企业应用框架, 在大型企业应用中应用广泛。其中Servlet、JSP是Web开发组件;

*本课程学习使用JavaEE中的Servlet、JSP组件来开发Web应用, 其中JSP的本质就是Servlet, 所以我们将先学习Servlet。*

## 知识点4 【Tomcat介绍及安装】 -1

- 我们的学习目标是使用Servlet和JSP开发Web应用，Servlet和JSP是JavaEE中的Web组件，Web组件都需要运行在Web应用服务器上；
- 本课程选用Apache软件基金会的Tomcat做为应用服务器，也称为Servlet容器(Container)；
- Tomcat开源、免费，在中小型系统和并发访问用户不是很多的场合下被普遍使用，是开发和调试Servlet/JSP程序的首选；

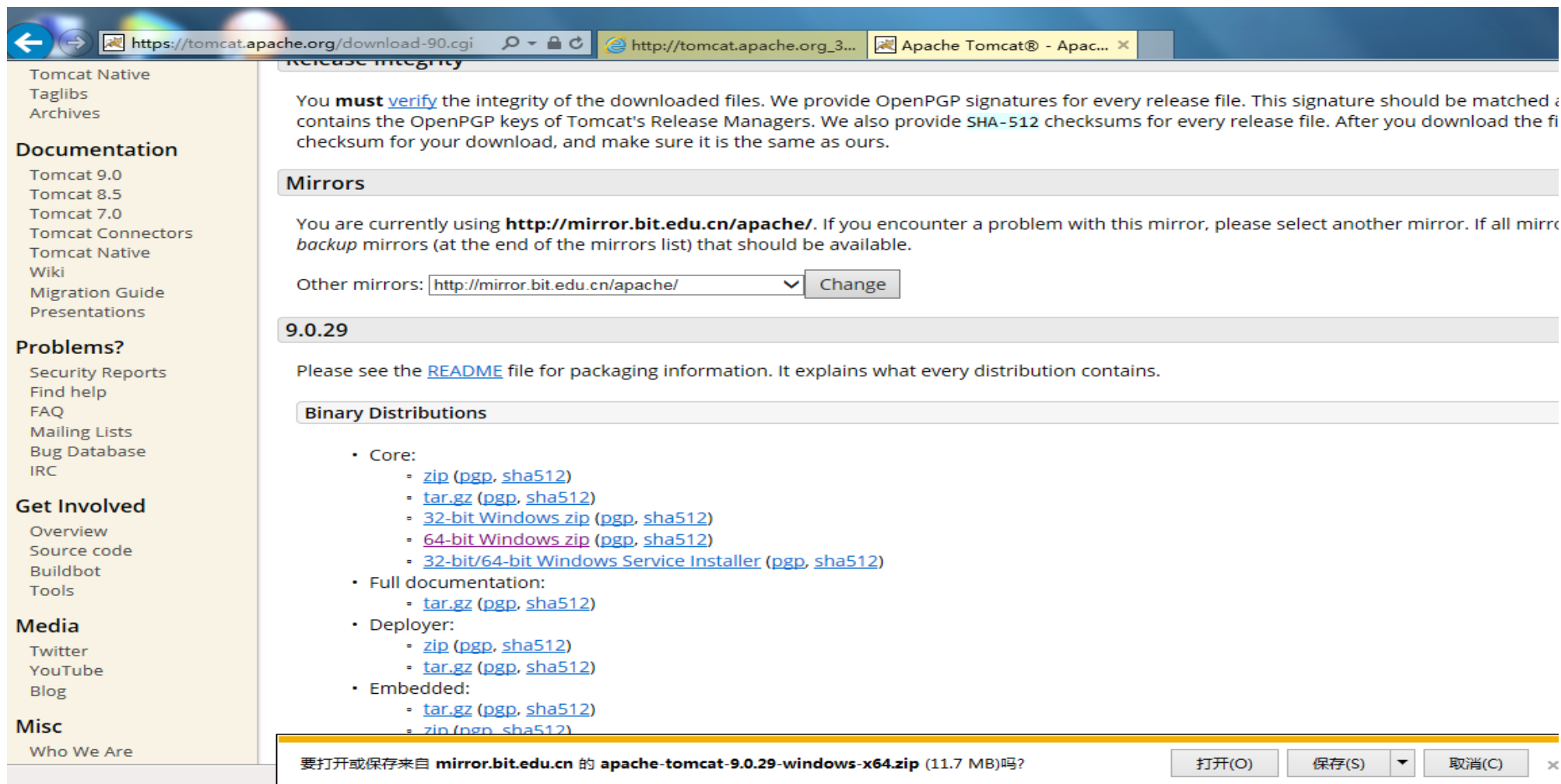


Apache Tomcat

下载地址：<http://tomcat.apache.org/>； 本课程采用Tomcat9.0版本；

安装：下载到的是一个压缩包，解压缩即可；【解压路径不要包含中文或空格等】

## 知识点4 【Tomcat介绍及安装】 -2



The screenshot shows the Apache Tomcat download page. The left sidebar contains navigation links for Tomcat Native, Taglibs, Archives, Documentation (Tomcat 9.0, 8.5, 7.0, Connectors, Native, Wiki, Migration Guide, Presentations), Problems? (Security Reports, Find help, FAQ, Mailing Lists, Bug Database, IRC), Get Involved (Overview, Source code, Buildbot, Tools), Media (Twitter, YouTube, Blog), and Misc (Who We Are).

The main content area is titled "Release integrity" and contains the following text:

You **must** [verify](#) the integrity of the downloaded files. We provide OpenPGP signatures for every release file. This signature should be matched against the OpenPGP keys of Tomcat's Release Managers. We also provide [SHA-512](#) checksums for every release file. After you download the file, calculate the checksum for your download, and make sure it is the same as ours.

The "Mirrors" section states: "You are currently using <http://mirror.bit.edu.cn/apache/>. If you encounter a problem with this mirror, please select another mirror. If all mirrors fail, you can try the [backup mirrors](#) (at the end of the mirrors list) that should be available." Below this is a dropdown menu showing the current mirror and a "Change" button.

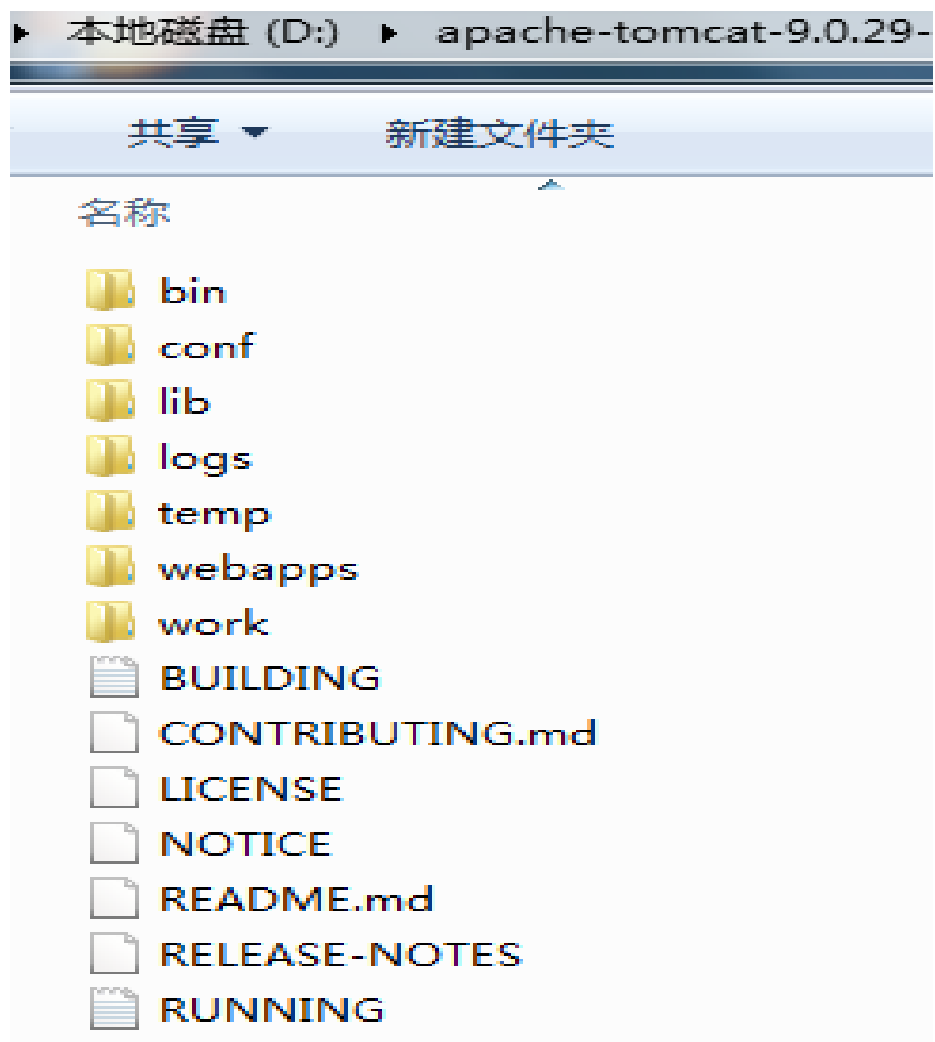
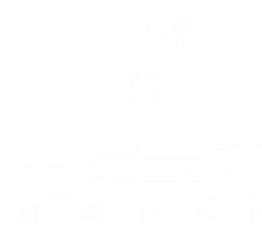
The "9.0.29" section contains the text: "Please see the [README](#) file for packaging information. It explains what every distribution contains."

The "Binary Distributions" section lists the following options:

- Core:
  - [zip](#) ([pgp](#), [sha512](#))
  - [tar.gz](#) ([pgp](#), [sha512](#))
  - [32-bit Windows zip](#) ([pgp](#), [sha512](#))
  - [64-bit Windows zip](#) ([pgp](#), [sha512](#))
  - [32-bit/64-bit Windows Service Installer](#) ([pgp](#), [sha512](#))
- Full documentation:
  - [tar.gz](#) ([pgp](#), [sha512](#))
- Deployer:
  - [zip](#) ([pgp](#), [sha512](#))
  - [tar.gz](#) ([pgp](#), [sha512](#))
- Embedded:
  - [tar.gz](#) ([pgp](#), [sha512](#))
  - [zip](#) ([pgp](#), [sha512](#))

A download dialog box is open at the bottom, asking: "要打开或保存来自 [mirror.bit.edu.cn](http://mirror.bit.edu.cn) 的 **apache-tomcat-9.0.29-windows-x64.zip** (11.7 MB)吗?" (Do you want to open or save the file **apache-tomcat-9.0.29-windows-x64.zip** (11.7 MB) from [mirror.bit.edu.cn](http://mirror.bit.edu.cn)?). The dialog has buttons for "打开(O)" (Open), "保存(S)" (Save), and "取消(C)" (Cancel).

## 知识点4 【Tomcat目录结构】 -1



## 知识点4 【Tomcat目录结构】 -2

- bin目录 用于存放 Tomcat的启动、停止等批处理脚本和Shell脚本
- conf目录 用于存放 Tomcat的相关配置文件
  - lib目录 Tomcat服务器依赖库目录，包含 Tomcat服务器运行环境  
依赖jar包
- logs Tomcat默认的日志存放路径
- temp目录 存放tomcat在运行过程中产生的临时文件
- webapps目录 Tomcat默认的Web应用部署目录
- work 存放Web应用JSP代码生成和编译后产生的class文件目录

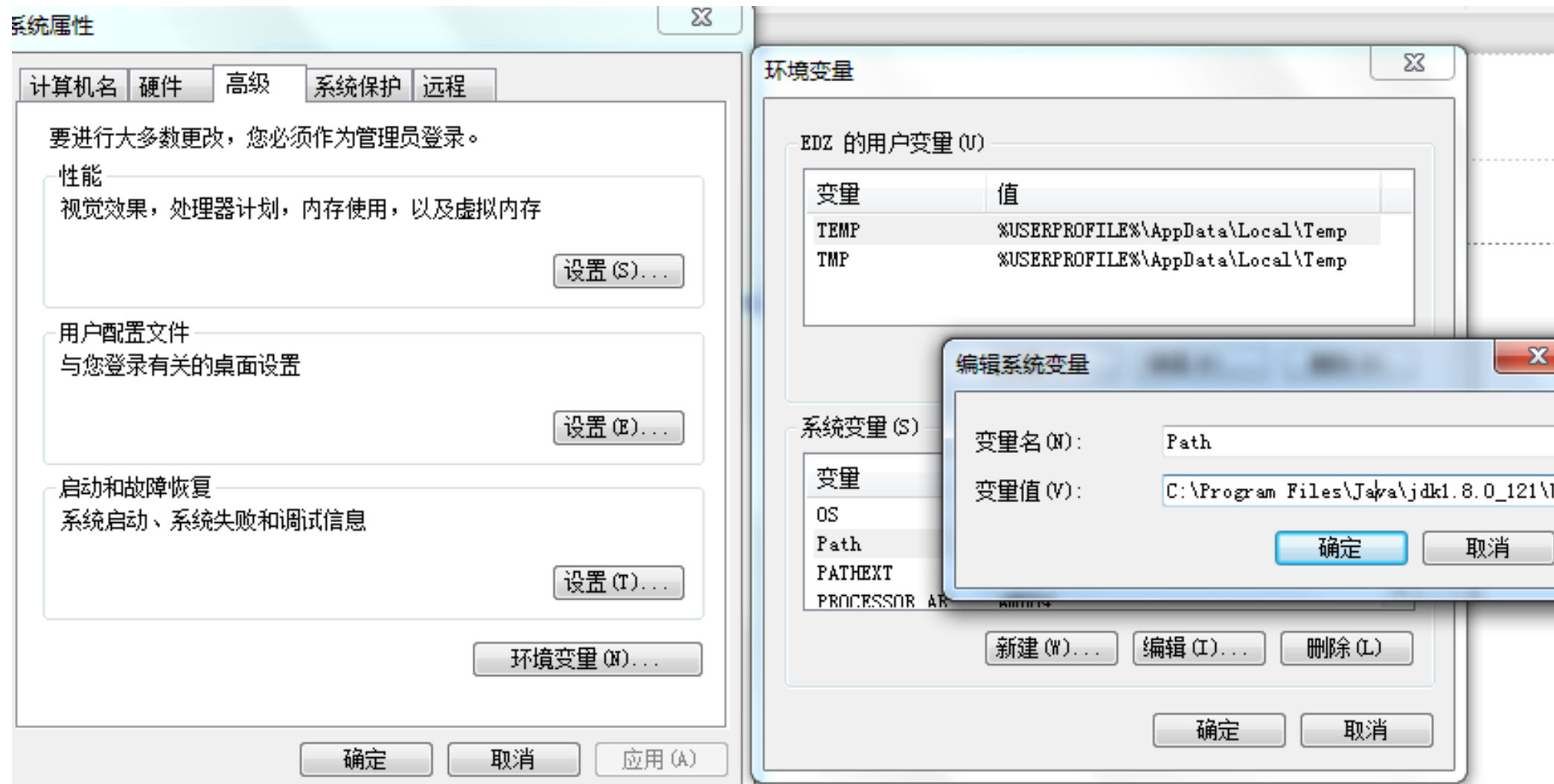
# Tomcat启动前需安装JDK



# JDK安装



# JDK环境变量配置





开始----->cmd



C:\Windows\system32\cmd.exe

```
C:\Users\EDZ>java -version
java version "1.8.0_121"
Java(TM) SE Runtime Environment (build 1.8.0_121-b13)
Java HotSpot(TM) 64-Bit Server VM (build 25.121-b13, mixed mode)

C:\Users\EDZ>_
```

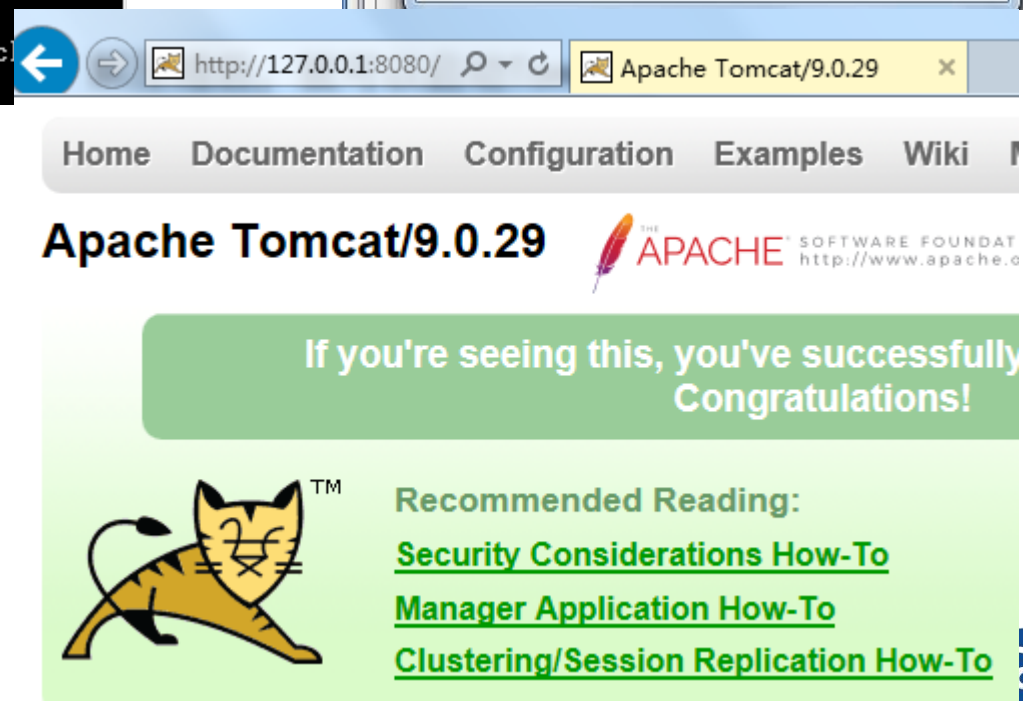
## 知识点4 【Tomcat介绍及安装】 -2

- 解压缩后，就可以启动Tomcat;
- 设置java\_home环境变量，将路径指定到JDK的安装目录下;
- 在cmd窗口，转到Tomcat安装目录的bin目录，运行startup

```
D:\apache-tomcat-9.0.29\bin>startup
Using CATALINA_BASE:   "D:\apache-tomcat-9.0.29"
Using CATALINA_HOME:   "D:\apache-tomcat-9.0.29"
Using CATALINA_TMPDIR: "D:\apache-tomcat-9.0.29\temp"
Using JRE_HOME:        "C:\Program Files\Java\jdk1.8.0_121"
Using CLASSPATH:       "D:\apache-tomcat-9.0.29\bin\bootstrap.jar;D:\apache-tomcat-9.0.29\bin\tomcat-juli.jar"
```



- 启动成功后，可以通过浏览器访问Tomcat
  - 输入 <http://127.0.0.1:8080> 其中127.0.0.1指的
  - 是访问本机，8080是Tomcat默认端口;
  - 访问到如下页面，说明Tomcat安装启动成功;



## 知识点5 【常见其他Servlet容器简介】

- 本课程选择使用Tomcat作为Servlet运行的环境，也就是Servlet容器；
- 除了Tomcat外，还有其他容器可以使用；



JBoss支持JavaEE标准，能够作为JavaEE应用服务器使用，Servlet容器只是其中一部分；



Resin是CAUCHO公司的产品，是一个非常流行的Servlet容器，性能也比较优良；



GlassFish支持JavaEE标准，是开源的、免费的，能够作为JavaEE应用服务器使用，Servlet容器只是其中一部分；

## 知识点6 【Servlet的概念及功能】

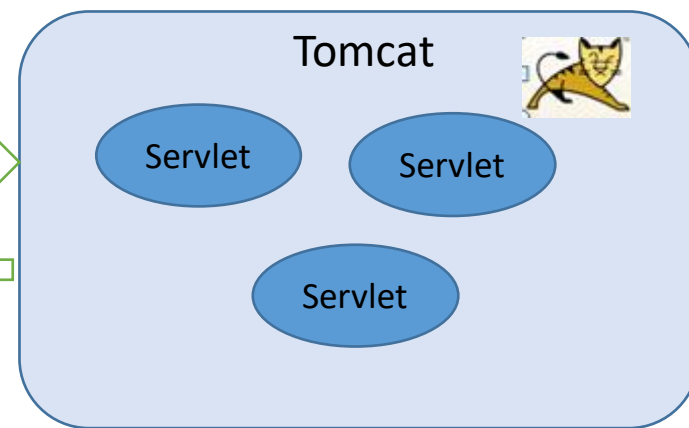
- Servlet是JavaEE规范中的Web开发组件;
- Servlet运行在服务器端, 需要Servlet容器的支持, 例如Tomcat;
- 可以通过浏览器访问Servlet, Servlet可以生成动态页面返回给浏览器;
- Servlet也是一段代码, 是一个Java类, 这个Java类需要遵守一定的编写规范, 例如, 必须继承于`javax.servlet.http.HttpServlet`类;

Servlet是本  
课程核心内  
容, JSP的  
本质也是  
Servlet~~



请求访问一个Servlet;

Tomcat运行Servlet,  
返回给客户端生成的  
动态页面;



## 本节总结提问【基本概念及Web应用入门】

- C/S和B/S有什么区别？
- Web站点和Web应用有什么区别？
- Servlet的概念和作用？
- 本课程主要的学习内容是什么？

## 本节总结 【基本概念及Web应用入门】

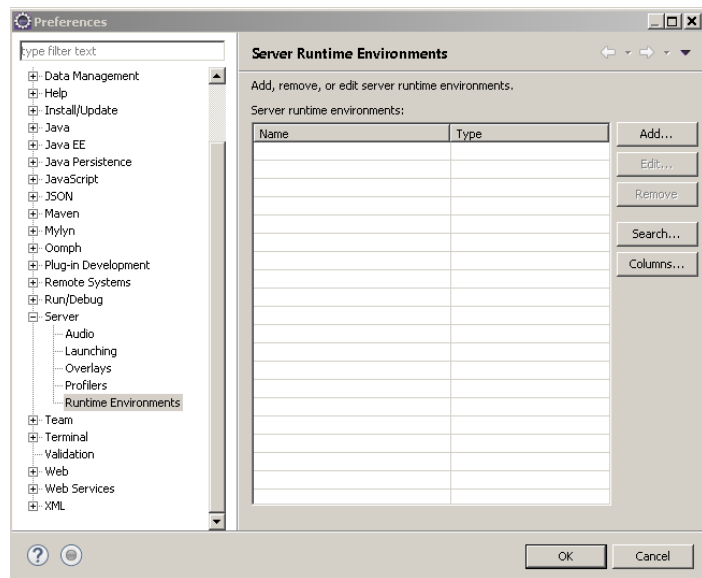
- C/S被称为胖客户端，用户使用必须安装软件；B/S被称为瘦客户端，用户使用不需要安装软件，只要有浏览器即可；
- Web站点只用来浏览信息，Web应用需要运行服务器端的程序，给客户提供服务；
- Servlet是JavaEE系列技术中的Web组件，是运行在Web服务器上的程序，用Java代码编写的类，必须遵守一定的规范；
- 本课程主要学习使用JavaEE规范中的Servlet和JSP开发Web应用，JSP的本质是Servlet；

## 第2节 【第一个Web应用】

- 知识点1: 使用IDE创建第一个Web应用
- 知识点2: 编写第一个Servlet
- 知识点3: web.xml中配置Servlet
- 知识点4: 部署到Tomcat
- 知识点5: 运行Web应用
- 知识点6: Servlet的生命周期

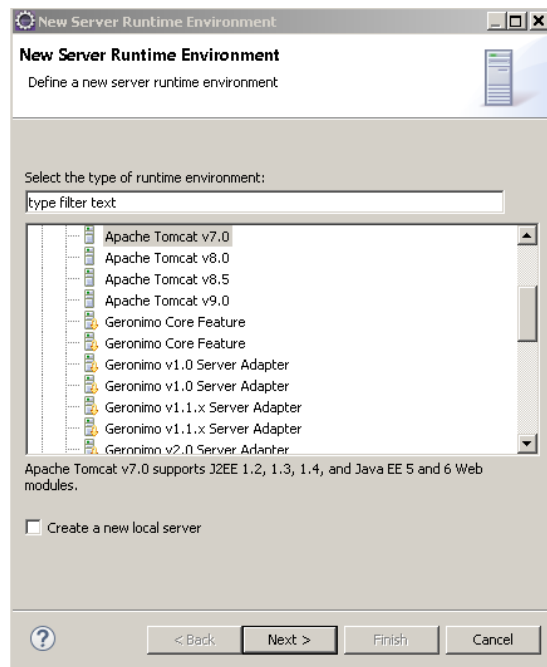
# 知识点1 【使用IDE创建第一个Web应用】 -1

- 本课程采用流行的JavaEE集成开发工具Eclipse;
- Web服务器采用Tomcat, 为了能够方便使用, 先在Eclipse中配置Tomcat;



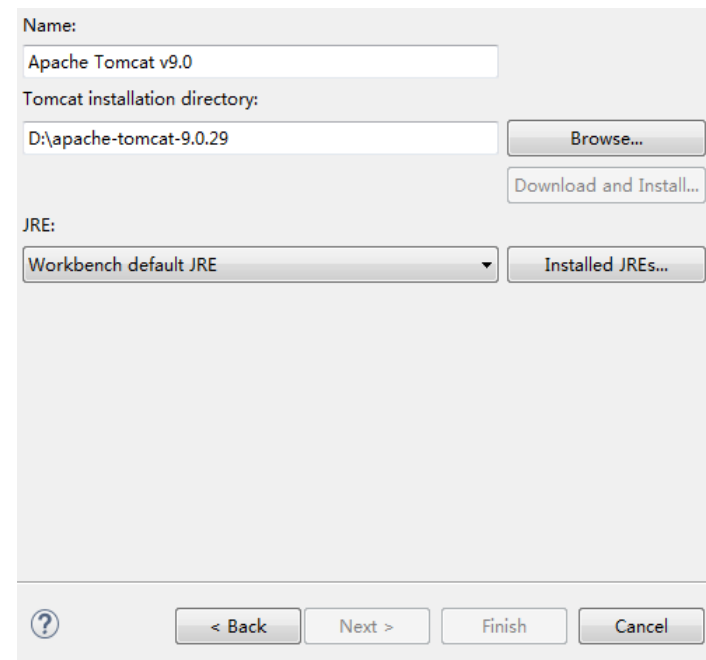
1

菜单: Window-preferences-server-runtime environments-add



2

选择Tomcat9.0版本, 点击next按钮;

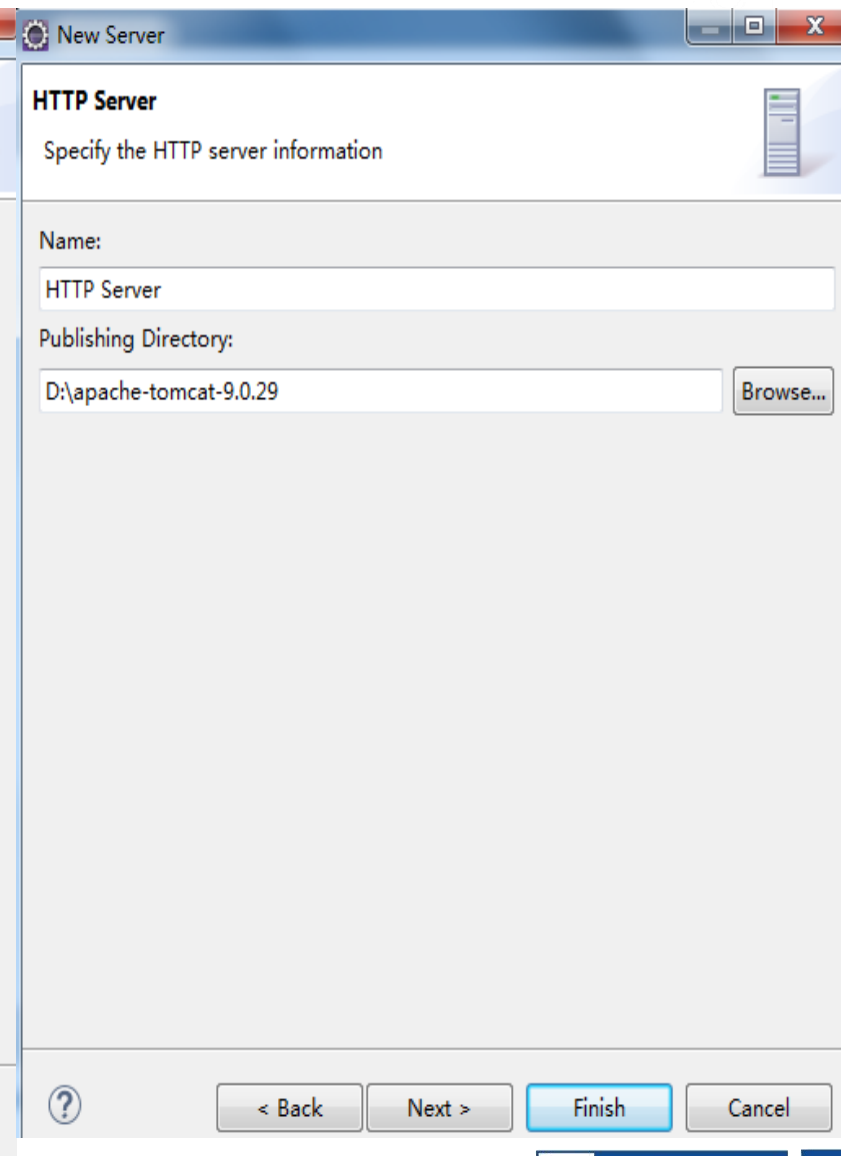
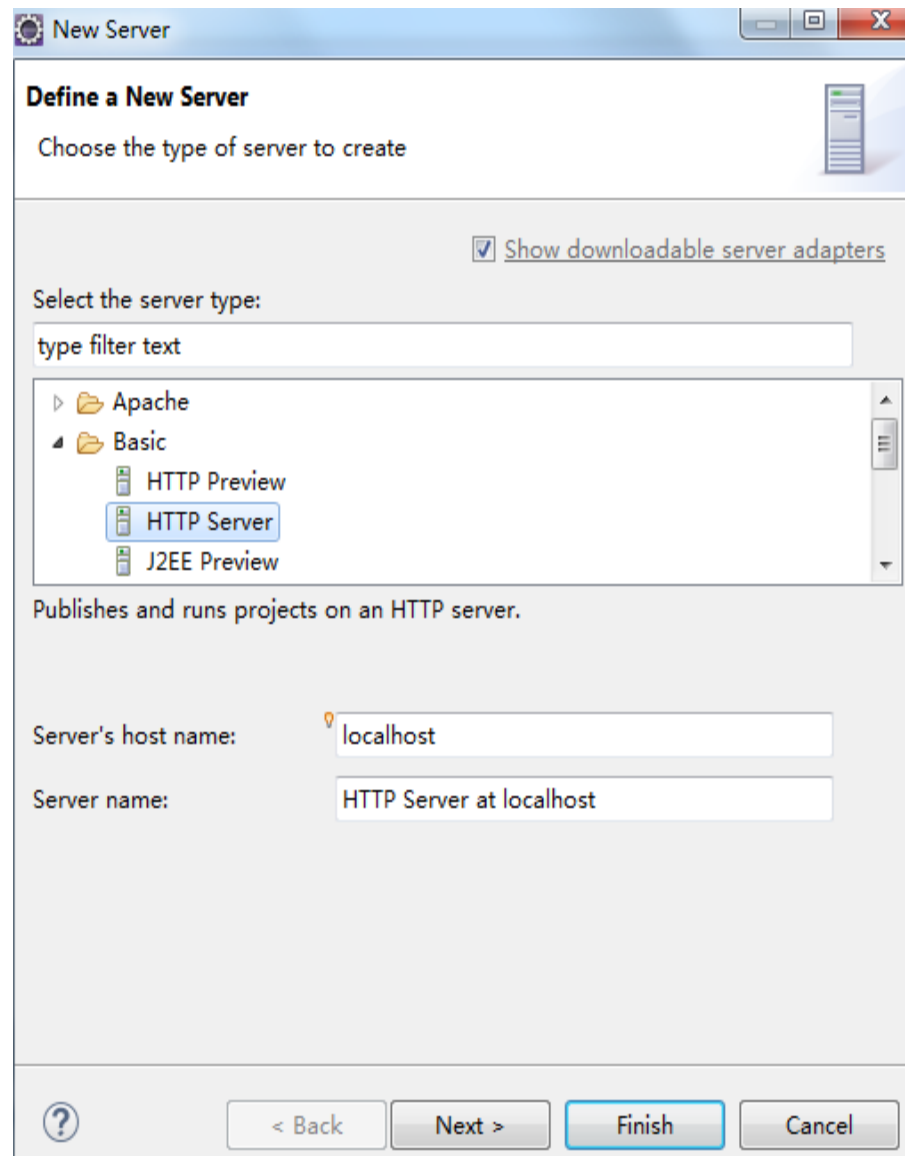
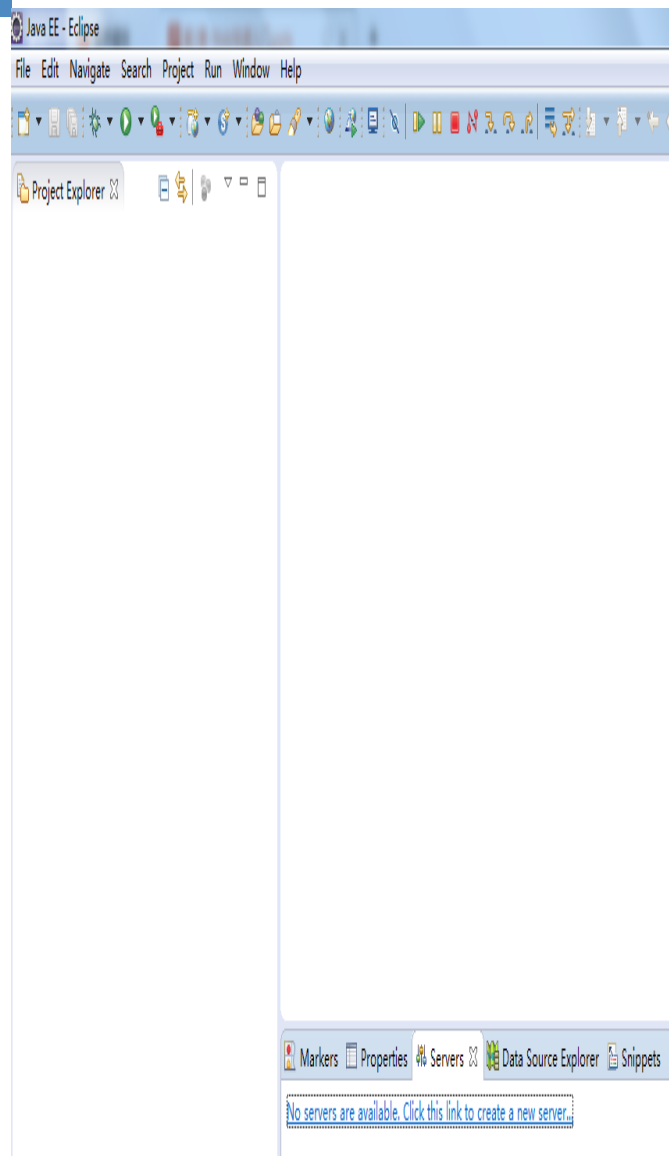


3

选择Tomcat安装根目录即可。

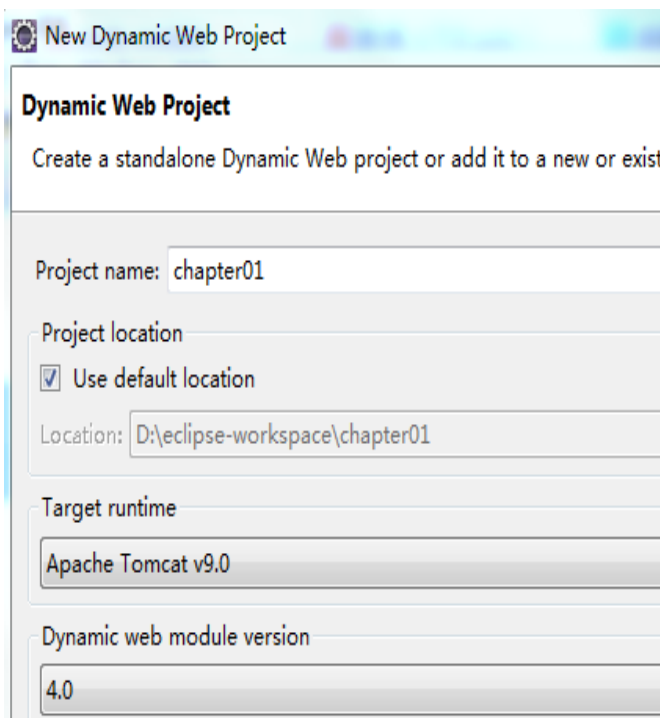


# 配置服务器



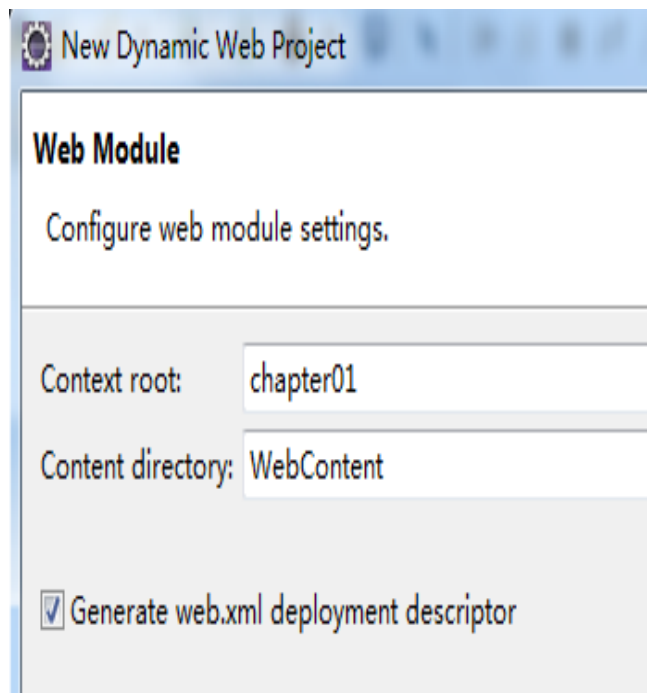
# 知识点1 【使用IDE创建第一个Web应用】 -2

- 创建一个名字为chapter01的Web应用;



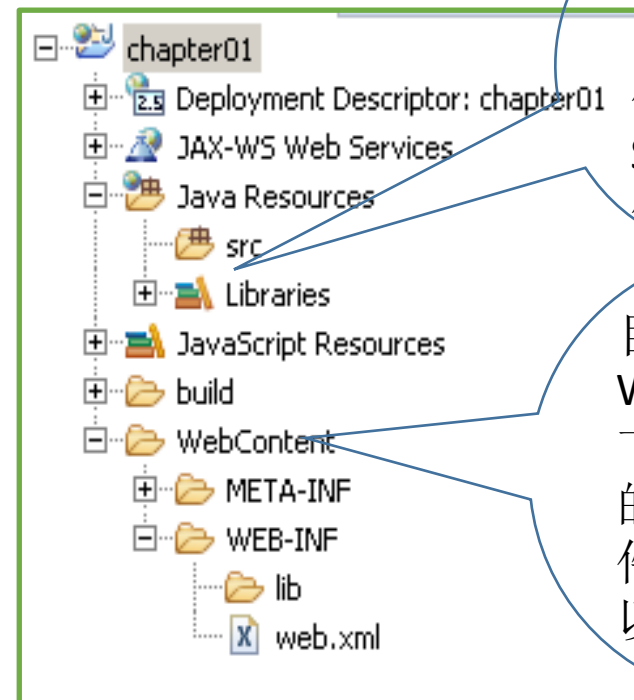
1

菜单: File-new-Dynamic Web Project,填写工程名称chapter01, 选用服务器为配置好的Tomcat



2

勾选自动生成web.xml文件;



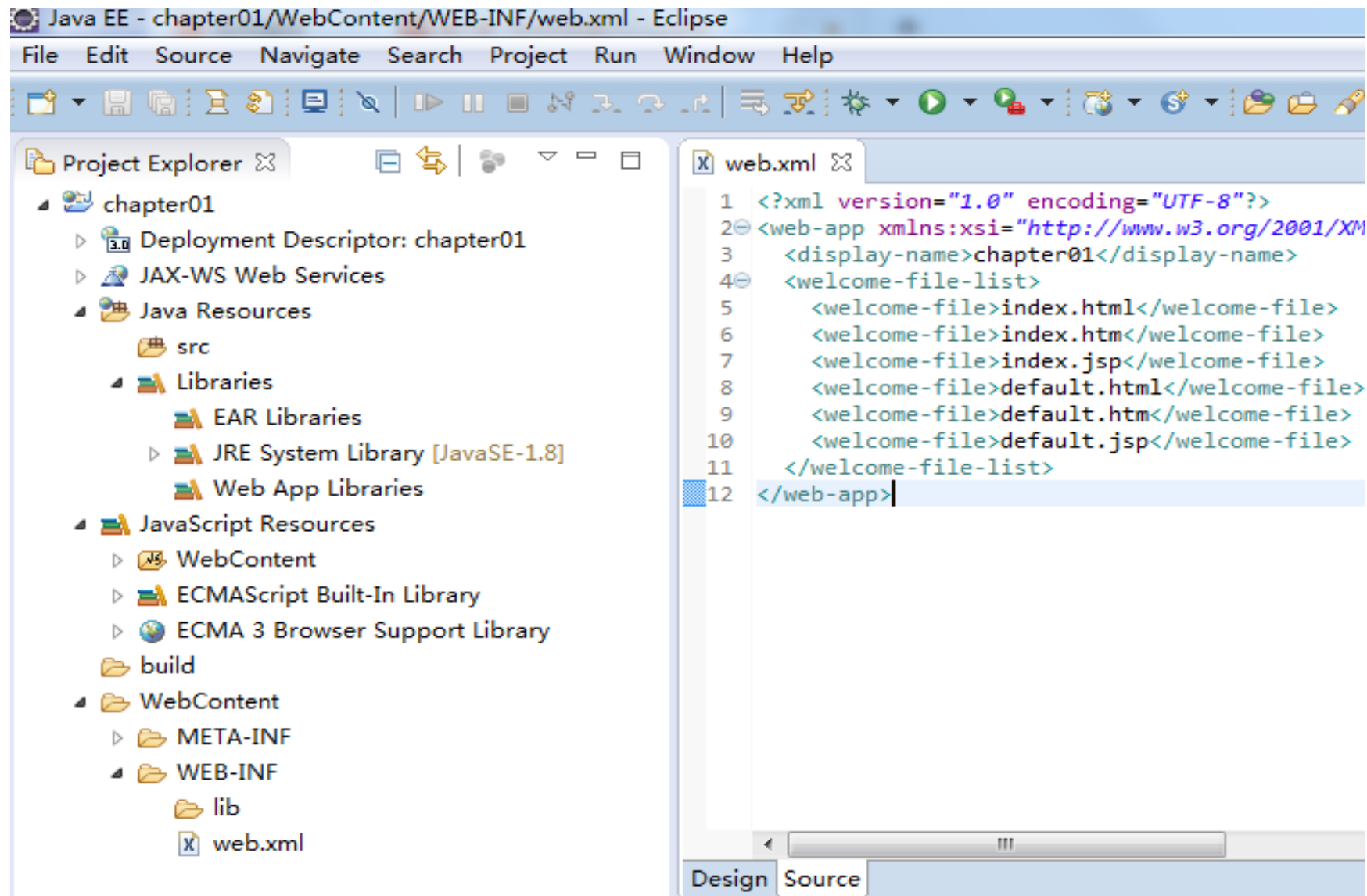
src目录下  
存放所有  
的.java文  
件, 包括  
Servlet文  
件

目录  
WebContent  
下存放所有  
的.jsp文  
件,HTML文件  
以及图片等。

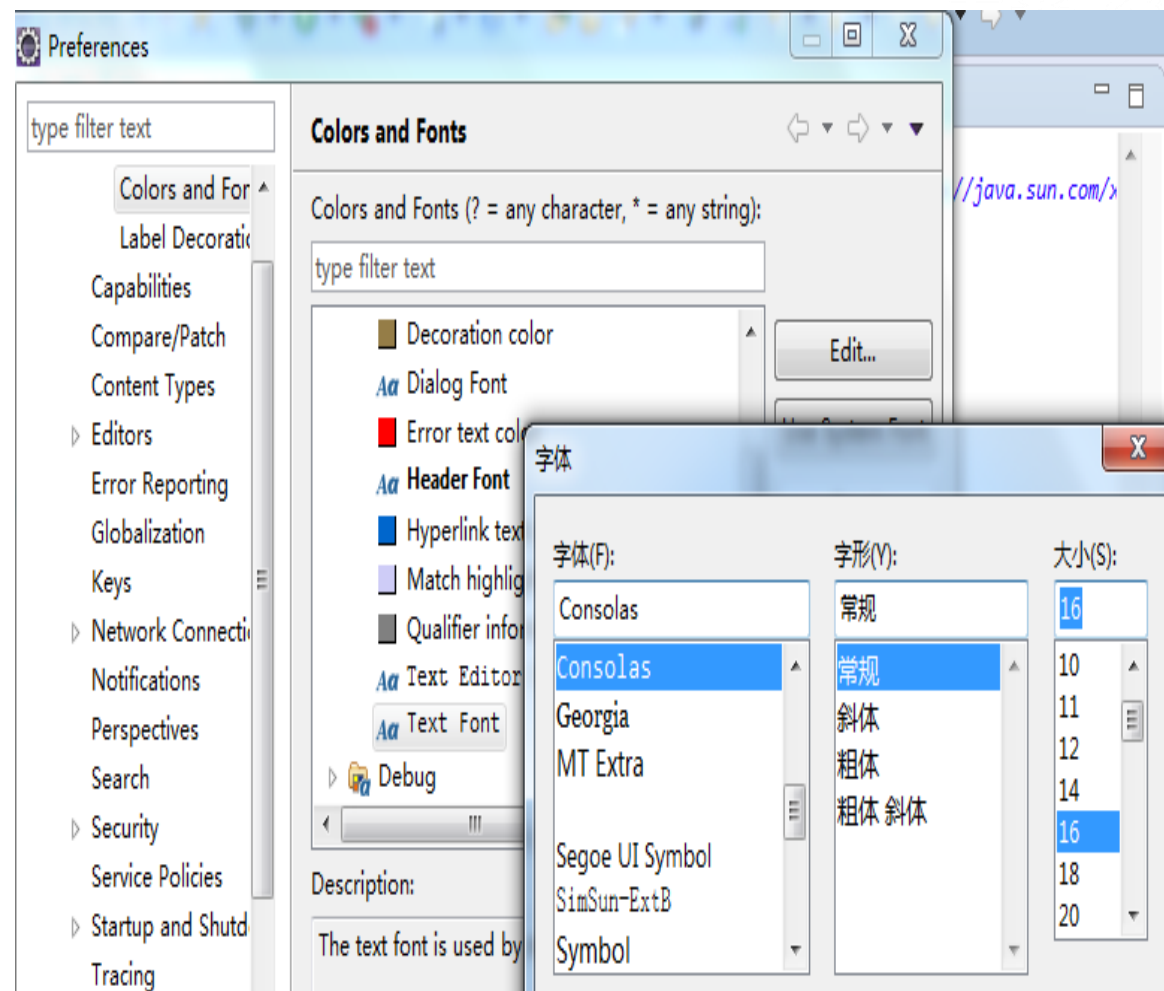
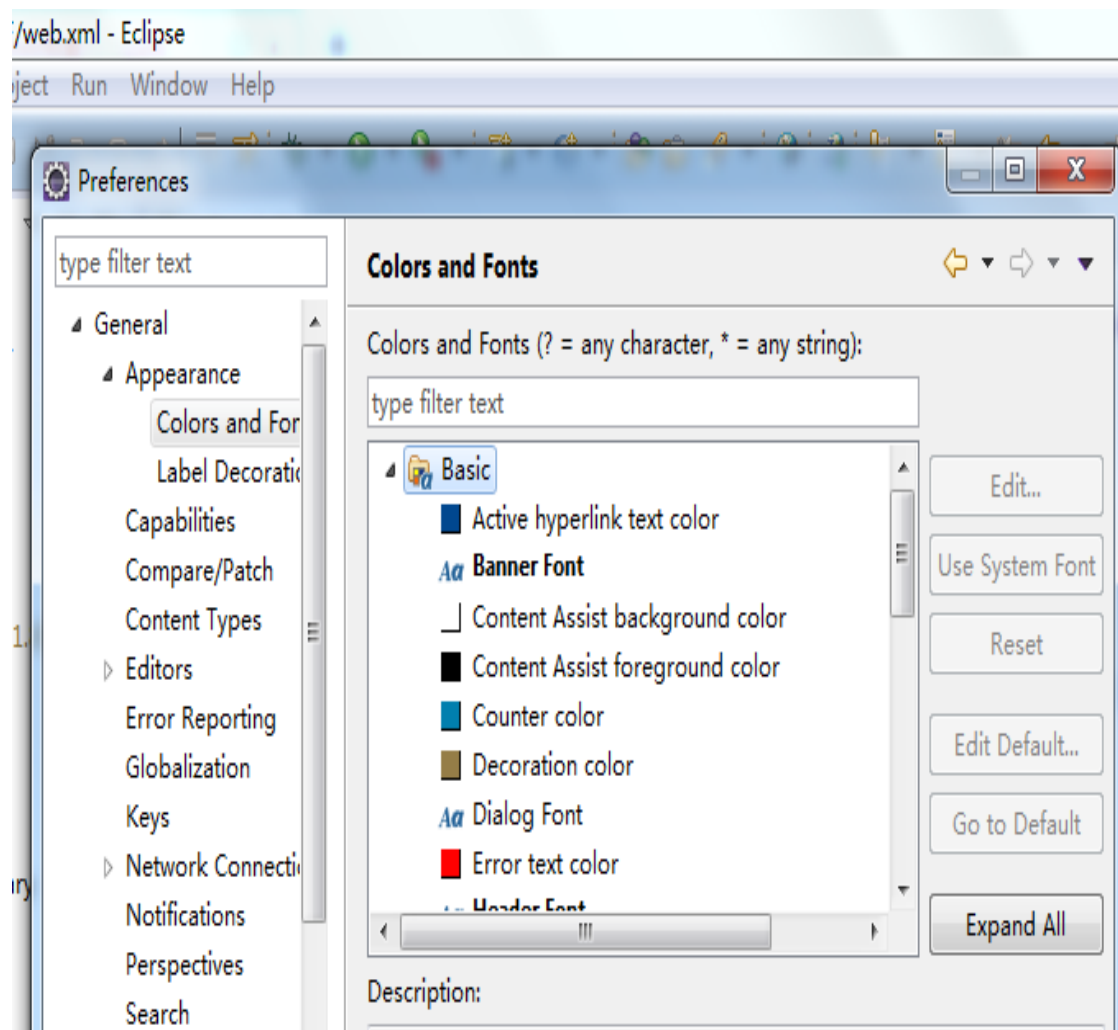
3

生成Web应用工程文件结构;

# 知识点1 【使用IDE创建第一个Web应用】 -3

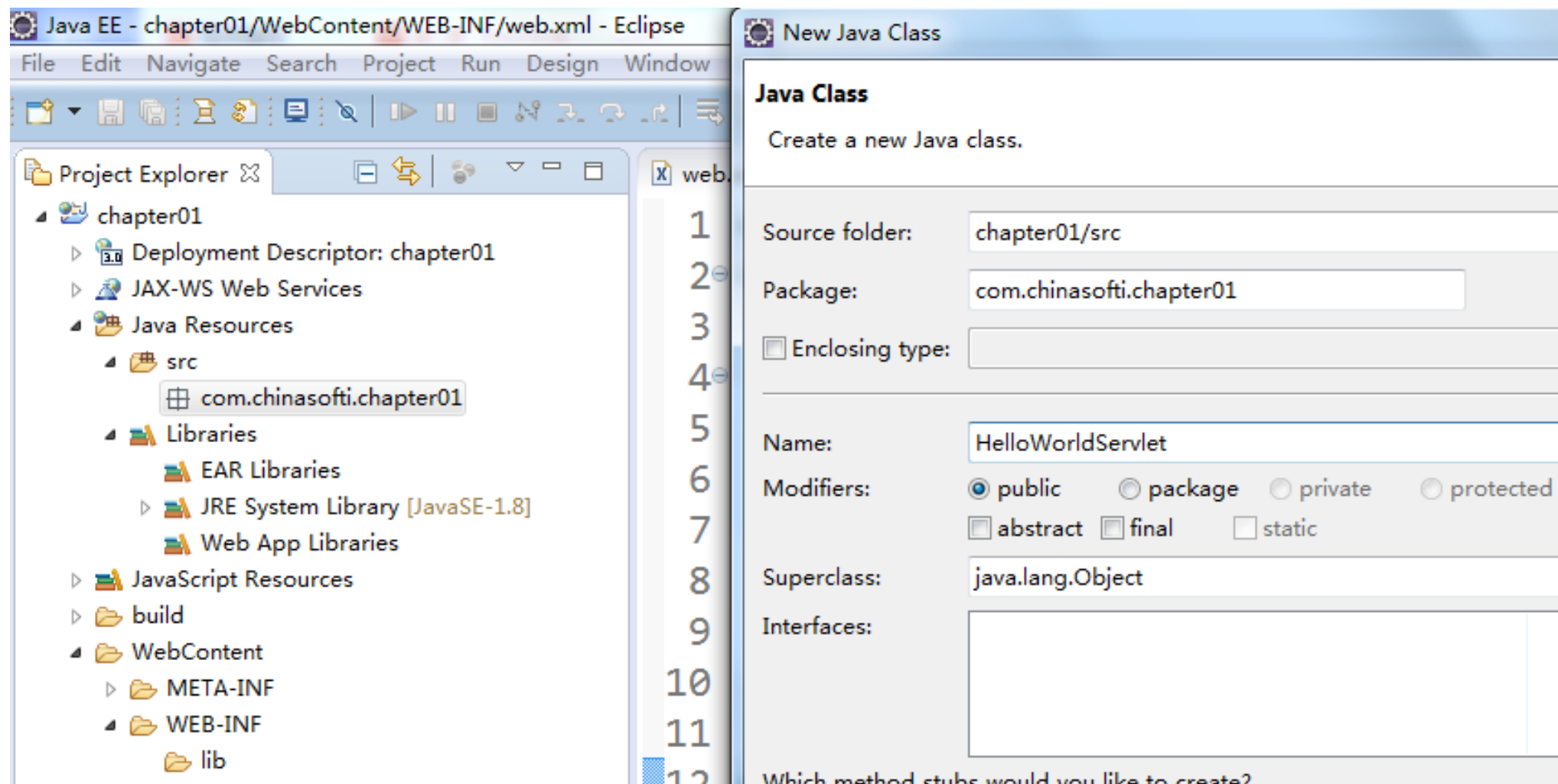


# 知识点1 【使用IDE创建第一个Web应用】 -4



菜单：Window-preferences-General-Appearance----Colors and Fonts

## 知识点2 【编写Servlet类】 -1



```
public class HelloWorldServlet extends HttpServlet {  
    public void doGet(HttpServletRequest request, HttpServletResponse response) throws  
IOException {  
        // 设置响应的内容类型  
        response.setContentType("text/html;charset=utf-8");  
        // 获得输出流  
        PrintWriter out = response.getWriter();  
        out.println("<html>");  
        out.println("<head><title>HelloWorld</title></head>");  
        out.println("<body bgcolor='red'>");  
        out.println("<center>HelloWorld 世界你好！ ！ ！ </center>");  
        out.println("</body></html>");  
    }  
    public void doPost(HttpServletRequest request, HttpServletResponse response)  
throws IOException {  
        this.doGet(request, response);  
    }  
}
```

## 知识点3：编写web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://java.sun.com/xml/ns/javaee"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-
app_3_0.xsd" id="WebApp_ID" version="3.0">
  <servlet>
    <servlet-name>HelloWorld</servlet-name>
    <servlet-class>com.chinasofti.chapter01.HelloWorldServlet</servlet-class>
  </servlet>

  <servlet-mapping>
    <servlet-name>HelloWorld</servlet-name>
    <url-pattern>/HelloWorld</url-pattern>
  </servlet-mapping>
</web-app>
```

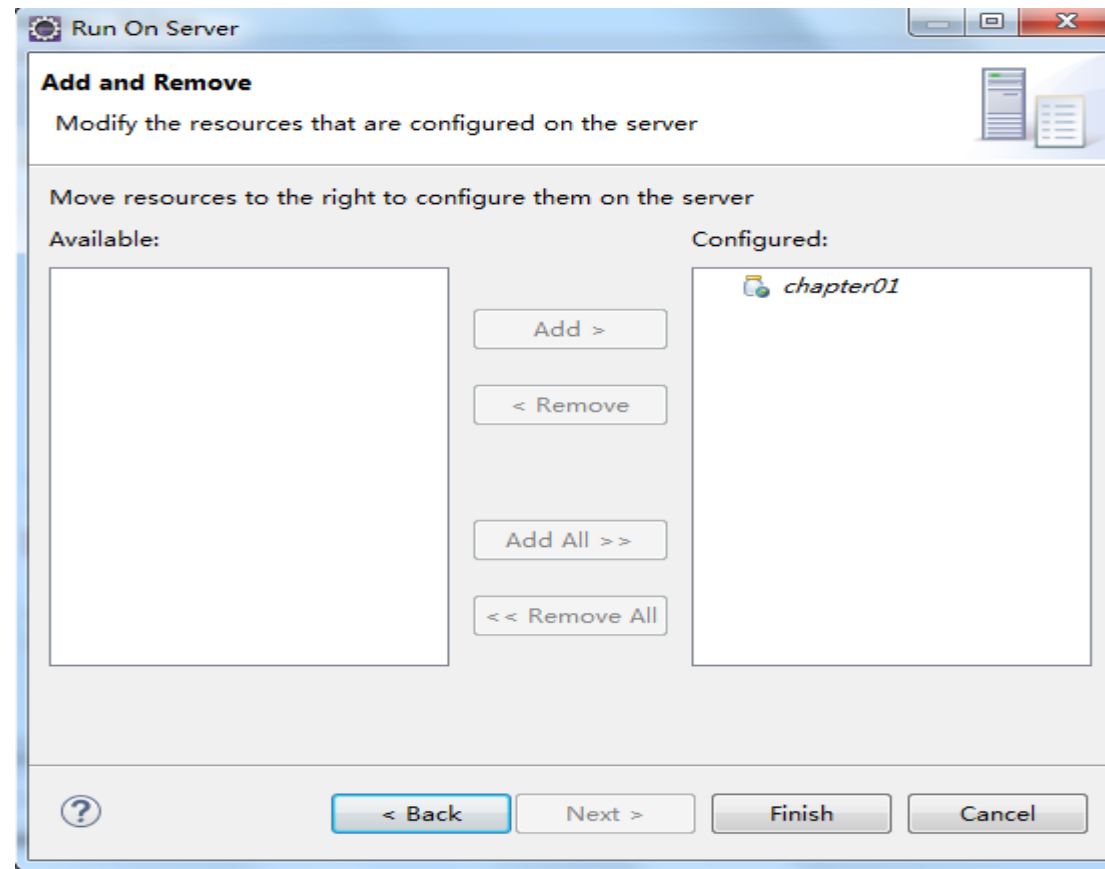
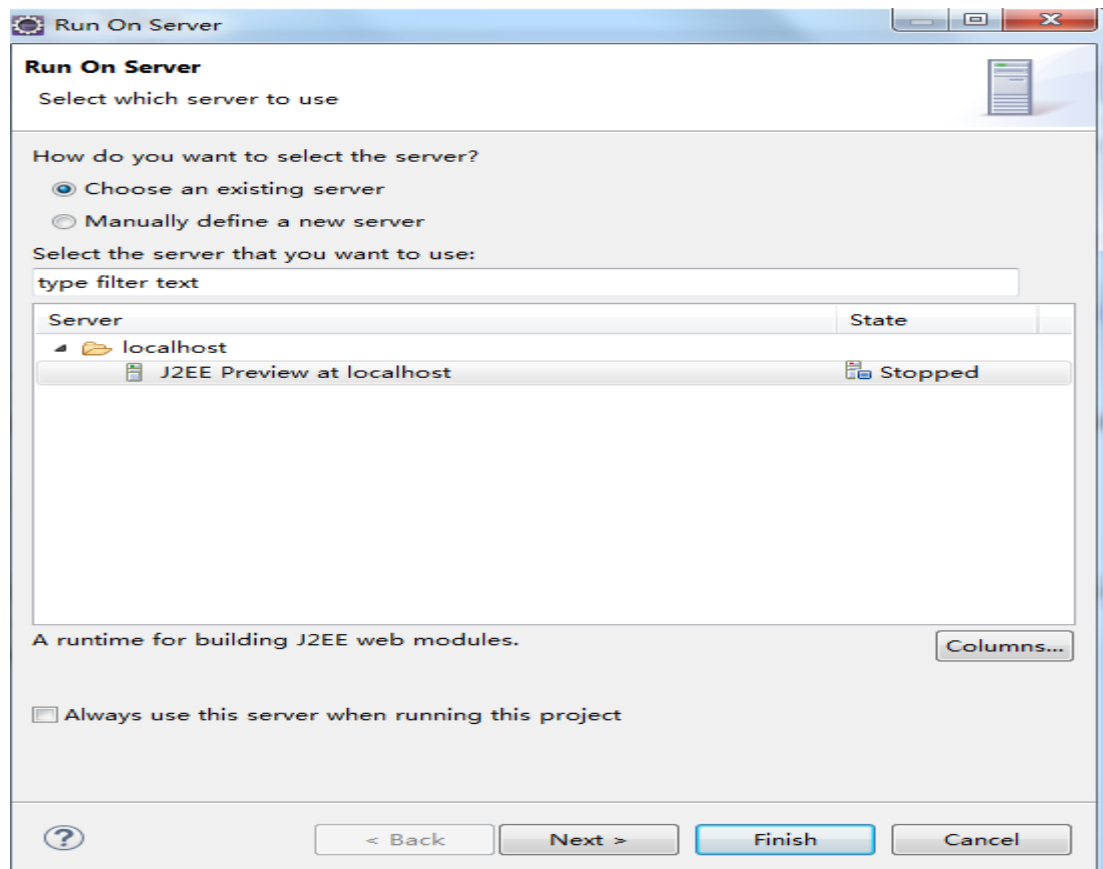
`<servlet>`元素包含`<servlet-name>`及`<servlet-class>`,其中`<servlet-name>`可以使用任意标识符, `<servlet-class>`是Servlet类的完整类名;

`<servlet-mapping>`元素包含`<servlet-name>`及`<url-pattern>`,其中`<servlet-name>`与已经定义的name对应, `<url-pattern>`是逻辑url, 非常非常重要, 访问Servlet就使用这个url-pattern, 必须以"/"开头。



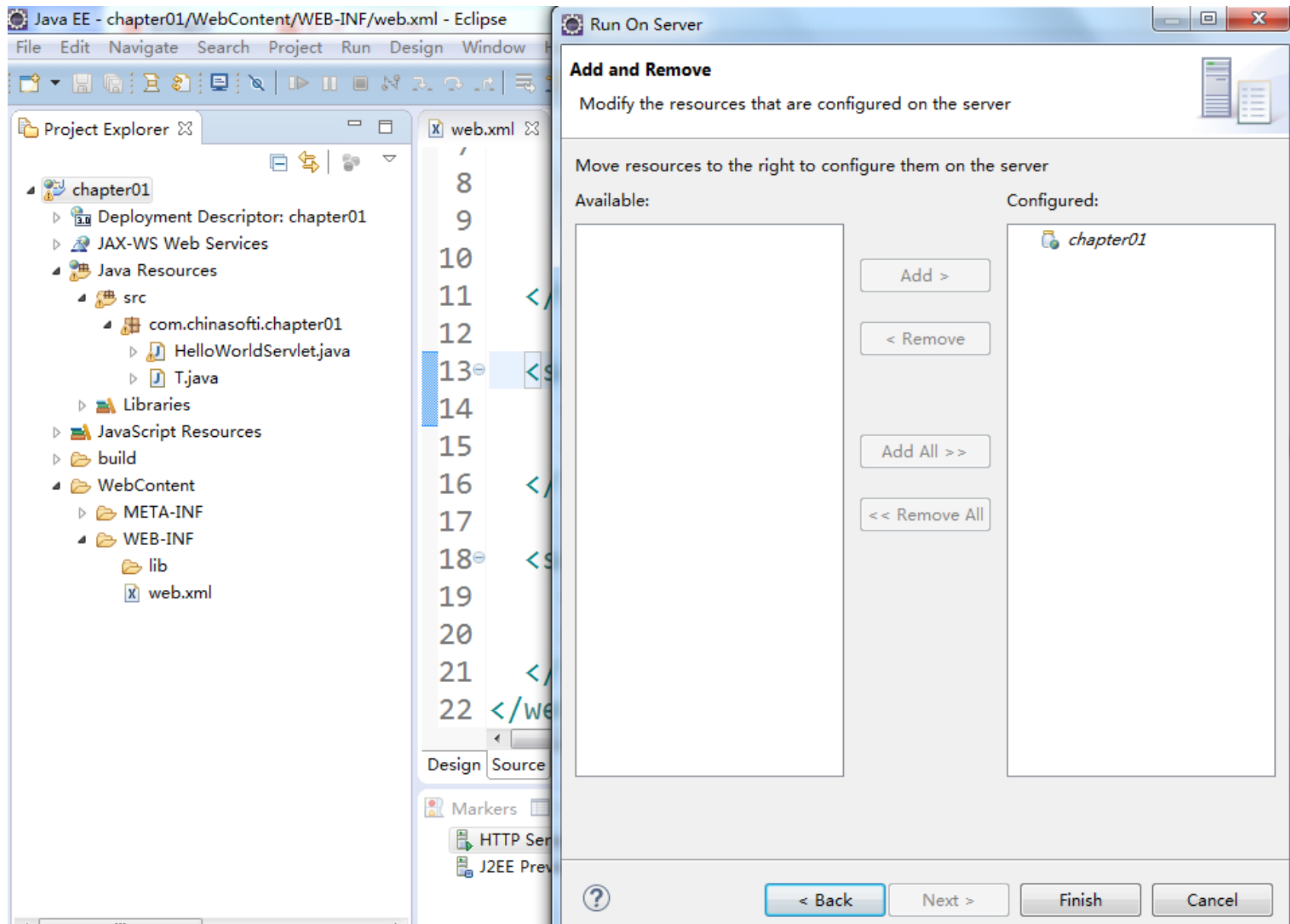
## 知识点4：部署到Tomcat-1

- 右键选中工程----->Run As ----->Run on Server-----Finish

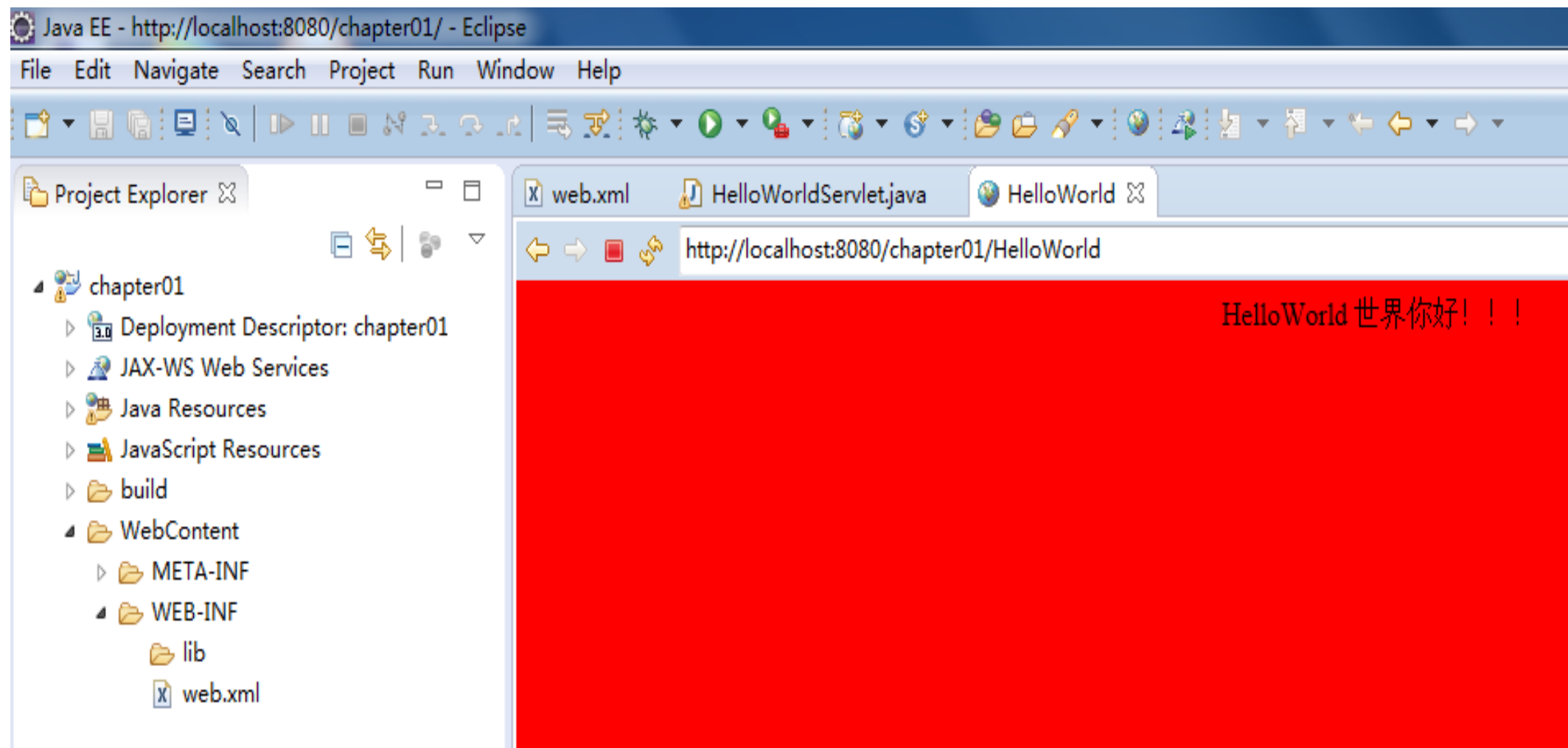




## 知识点4：启动tomcat服务器



## 知识点5：运行Web应用 通过url访问资源



```
public class LifeServlet extends HttpServlet {  
    private ServletConfig config;  
    //初始化Servlet  
    public void init(ServletConfig config) throws ServletException {  
        System.out.println("=====进入初始化Servlet方法中=====");  
        super.init(config);  
        this.config = config;  
        System.out.println("=====初始化Servlet=====");  
        System.out.println("==== 初始化Servlet结束=====");  
    }  
}
```

**//处理Get请求**

```
public void doGet(HttpServletRequest request, HttpServletResponse response) throws  
IOException, ServletException{  
    System.out.println("====进入doGet（）方法中====");  
    this.doPost(request, response);  
    System.out.println("=====调用doGet（）方法=====");  
    System.out.println("===调用doGet（）方法结束=====");  
}
```

//处理post请求

```
    public void doPost(HttpServletRequest request,HttpServletResponse response) throws  
    IOException,ServletException{
```

```
        System.out.println("====进入doPost（）方法中====");
```

```
        System.out.println("=====调用doPost（）方法=====");
```

```
        System.out.println("=====调用doPost（）方法结束=====");
```

```
    }
```

//调用destroy方法

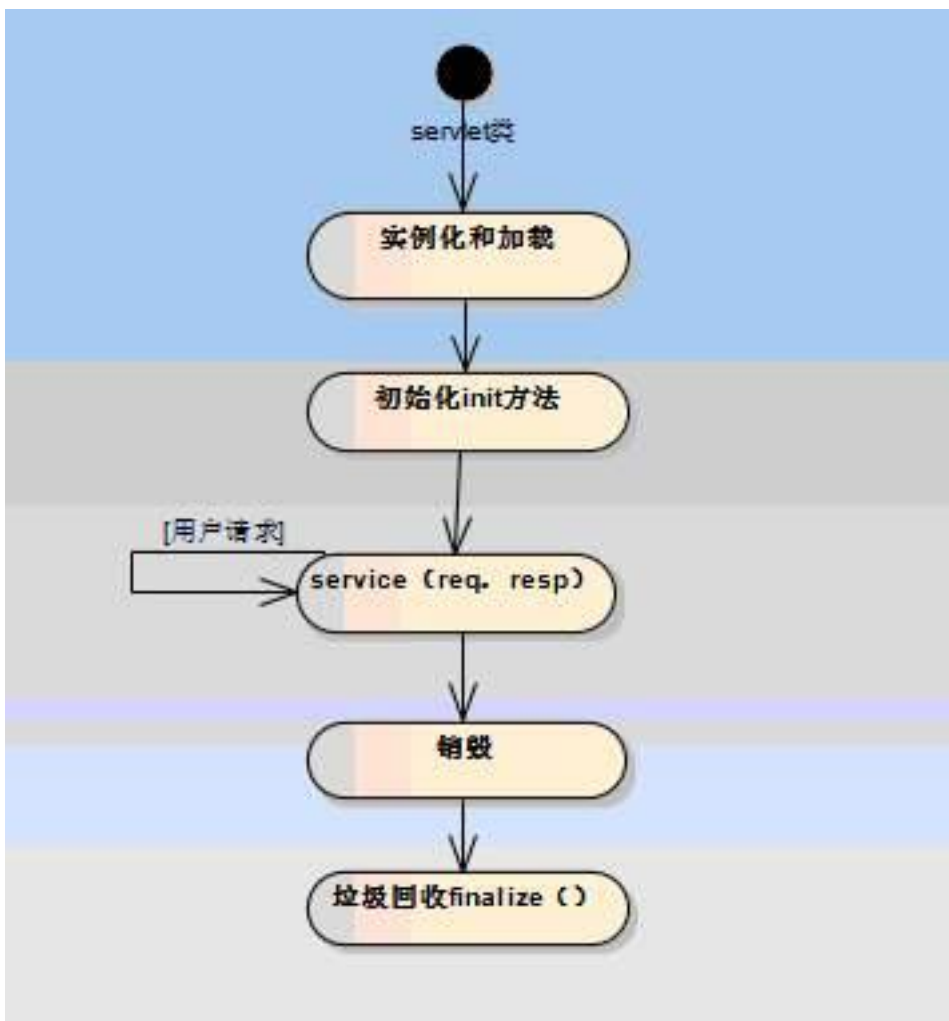
```
    public void destroy(){
```

```
        System.out.println("=====系统调用destroy（）方法=====");
```

```
    }
```

```
}
```

# 生命周期流程图



次

service()执行多次

destroy()仅执行一次

# 最新servlet4.0 api下载地址

<http://tomcat.apache.org/tomcat-9.0-doc/servletapi/index.html>

Servlet 4.0 - Apache Tomcat 9.0.29

OVERVIEW PACKAGE CLASS TREE DEPRECATED INDEX HELP

PREV NEXT FRAMES NO FRAMES

## Servlet 4.0 API - Apache Tomcat 9.0.29

**Packages**

Package	Description
<a href="#">javax.servlet</a>	The javax.servlet package contains a number of classes and interfaces that describe and define the contracts between a servlet class and the runtime environment provided for an instance of such a class by a conforming servlet container.
<a href="#">javax.servlet.annotation</a>	
<a href="#">javax.servlet.descriptor</a>	
<a href="#">javax.servlet.http</a>	The javax.servlet.http package contains a number of classes and interfaces that describe and define the contracts between a servlet class running under the HTTP protocol and the runtime environment provided for an instance of such a class by a conforming servlet container.

Servlet 4.0 - Apache Tomcat 9.0.29

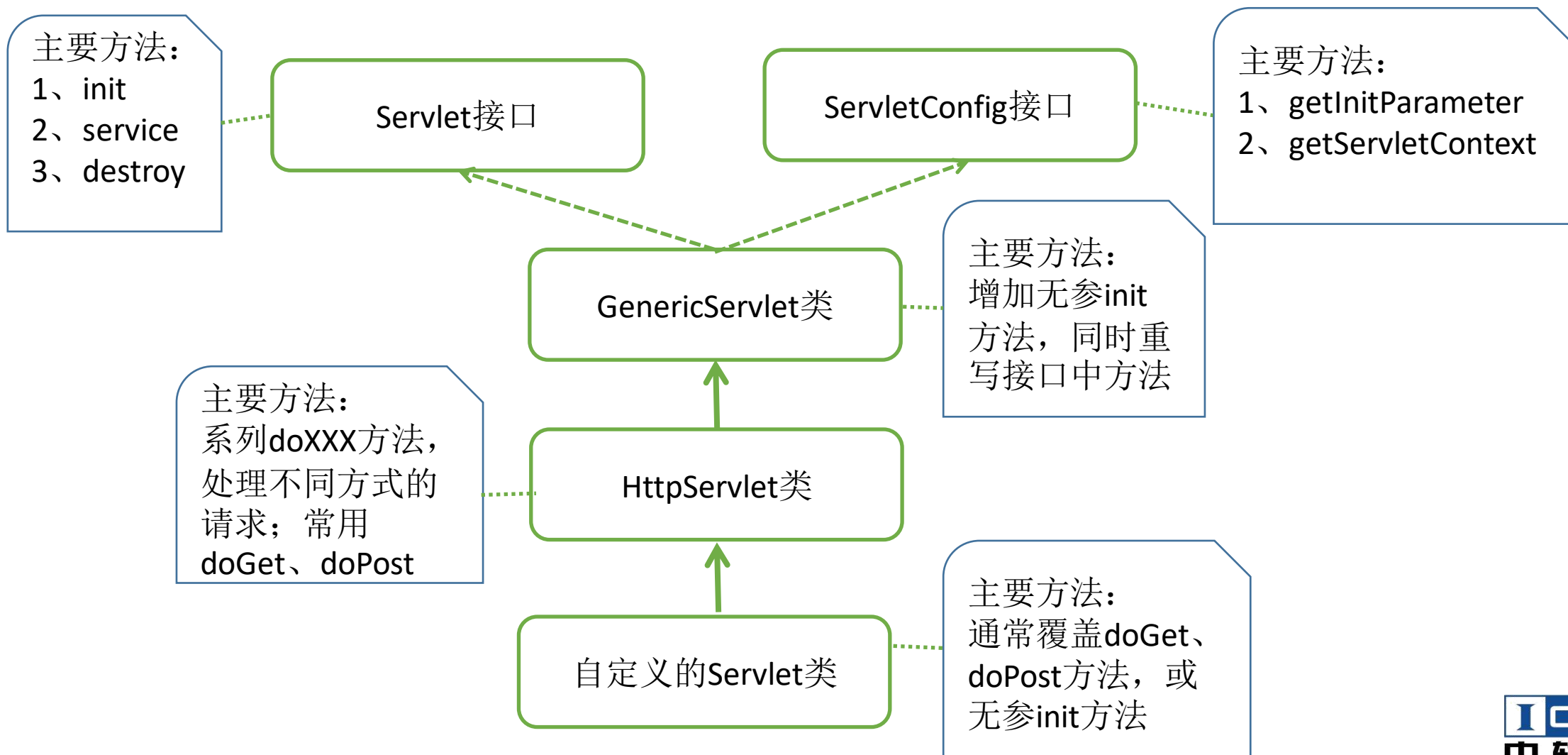
OVERVIEW PACKAGE CLASS TREE DEPRECATED INDEX HELP

PREV NEXT FRAMES NO FRAMES

Copyright © 2000-2019 Apache Software Foundation. All Rights Reserved.

## 知识点6 【Servlet的生命周期】 -1

- 一个Servlet类总是继承HttpServlet，它的“家谱”如下：





## 知识点6 【Servlet的生命周期】 -2

- 当用户第一次从浏览器请求访问LifeServlet时，对应的LifeServlet的生命周期简略描述如下：

1

Tomcat调用LifeServlet的构造方法，创建该类的对象；

2

Tomcat调用JavaEE API中的初始化方法：先调用有参数的init方法，再调用无参的init方法，进行初始化工作；

3

初始化成功后，调用service服务方法，通过判断请求方式，调用相应的doXXX方法，如doGet、doPost等方法；

4

doXXX方法正常返回后，即提供服务结束；

5

Tomcat根据使用情况，在适当的时机销毁Servlet对象，销毁前调用destroy方法；

# Web程序调试和排错

- Web程序常见错误：
  - **404错误**---找不到访问的页面或资源
    - 运行时，URL输入错误
    - 将页面放在WEB-INF下
    - 外部启动Tomcat，未部署项目
  - **500错误**-----JSP页面代码有误
    - JSP页面代码有错误
  - **页面无法显示**
    - 未启动Tomcat

## 本节总结提问【第一个Web应用】

- Servlet需要在web.xml中配置什么信息?
- 自定义Servlet类的继承关系如何?
- 自定义Servlet类中一般写什么方法?
- Servlet的生命周期如何?

## 本节总结 【第一个Web应用】

- 在web.xml中，需要对Servlet配置<servlet>以及<servlet-mapping>，其中url-pattern是访问Servlet使用的逻辑地址；
- 自定义Servlet类都继承于HttpServlet，HttpServlet的父类是GenericServlet；
- GenericServlet实现了Servlet和ServletConfig两个顶级接口；
- 自定义Servlet类中一般重写doGet或doPost方法，需要的话可以重写无参init方法进行初始化操作；
- 请求Servlet后，容器会创建Servlet对象并进行初始化，调用doXXX方法响应特定的HTTP请求，不再使用时容器将销毁Servlet。

# 本章总结

- 本章主要学习Web应用开发的基本概念，了解C/S及B/S，Web站点及Web应用，HTTP服务器及Web应用服务器等；
- 明确课程目标，学习JavaEE中的Servlet以及JSP技术开发Web应用；
- JSP的本质是Servlet，所以从Servlet开始学习；
- 使用IDE创建一个简单的Web应用，编写简单的Servlet，部署应用，并通过浏览器访问应用；
- 了解自定义Servlet类的继承关系，生命周期。

# 本章作业

- 作业1:

题目：创建Web应用，编写一个Servlet，实现功能：在页面中输出个人简历。

难度：低

作业2：学会查询Servlet API

