Servlet入门

课程介绍

- 了解Browser-Server（B/S）模式
- 掌握Servlet开发技巧
- 掌握Servlet执行原理

软件结构发展史

- 单机时代-桌面应用
- 联机时代(Client-Server模式)
- 互联网时代（ Broswer-Server模式)

单机时代-桌面应用

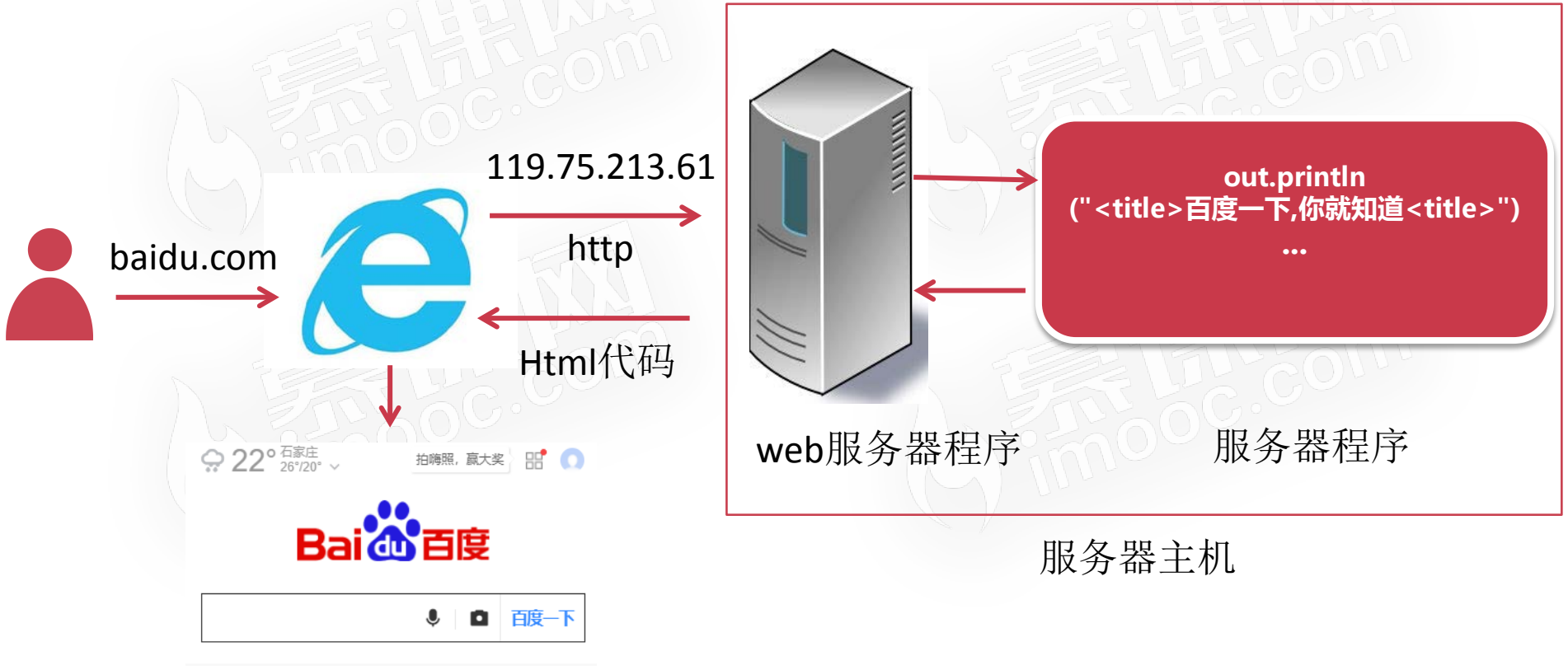
联机时代(Client-Server模式）

- Client/Server结构(C/S结构)是指客户端和服务器结构
- 优点：数据方便共享，安全性高
- 缺点：必须安装客户端，升级与维护困难

互联网时代(Broswer-Server模式）

- Broswer-Server(B/S)模式即浏览器和服务器架构模式
- 优点：开发简单，无需安装客户端，数据易于共享
- 缺点：相较于C/S模式，执行速度与用户体验相对较弱

B/S模式执行流程


baidu.com
119.75.213.61
http
Html代码
out.println
("<title>百度一下,你就知道<title>")
...
web服务器程序
服务器程序
服务器主机
22° 石家庄 26°/20°
拍晒照，赢大奖
Bai du 百度
百度一下

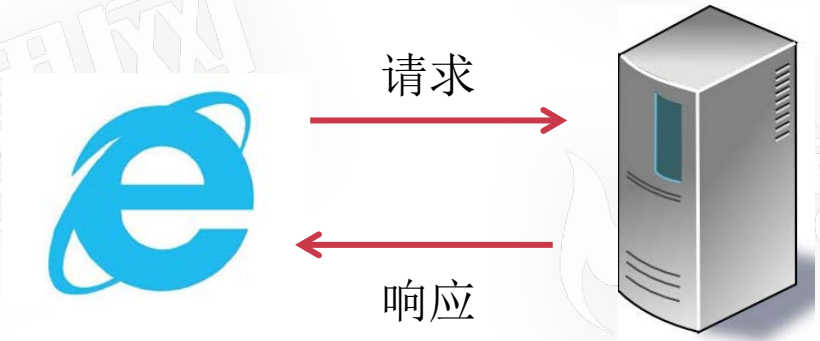
B/S模式执行流程


```
<html xmlns="http://www.w3.org/1999/xhtml">
 <head>
  <meta content="always" name="referrer" />
  <title>百度一下，你就知道 </title>
 </head>
 <body class=" s-manhattan-index">
  <div id="s_is_index_css" style="display:none;">
  </div>
  <div id="wrapper">
   <div class="s-skin-container s-isindex-wrap">
   </div>
   <div id="head" class="">
    <div id="s_top_wrap" class="s-top-wrap">
```

服务器主机

请求与响应

- 从浏览器发出送给服务器的数据包称为“请求(Request)”
- 从服务器返回给浏览器的结果称为“响应(Response)”



J2EE是什么

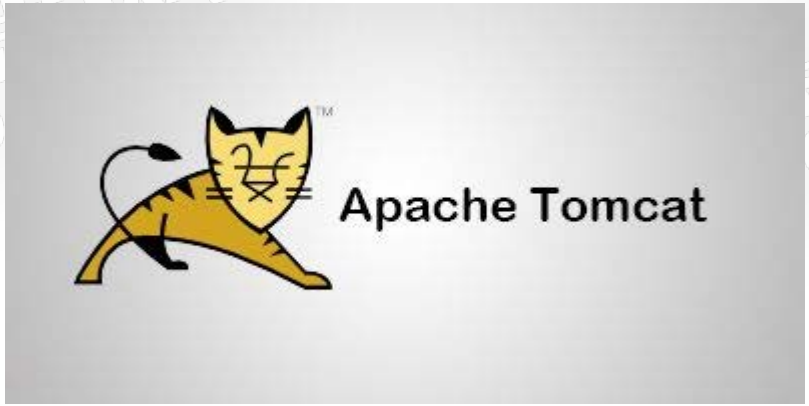
- J2EE（Java 2 Platform Enterprise Edition）是指“Java 2企业版”
- B/S模式开发Web应用就是J2EE最核心的功能
- J2EE由13个功能模块成

J2EE中13个功能模块

Servlet web服务器小程序	JSP 服务器页面	JDBC 数据库交互模块	XML XML交互模块
EJB 企业级Java Bean	RMI 远程调用	JNDI 目录服务	JMS 消息服务
JTA 事务管理	JavaMail 发送/接收Email	JAF 安全框架	CORBA CORBA集成
JTS CORBA事务监控			

Apache Tomcat

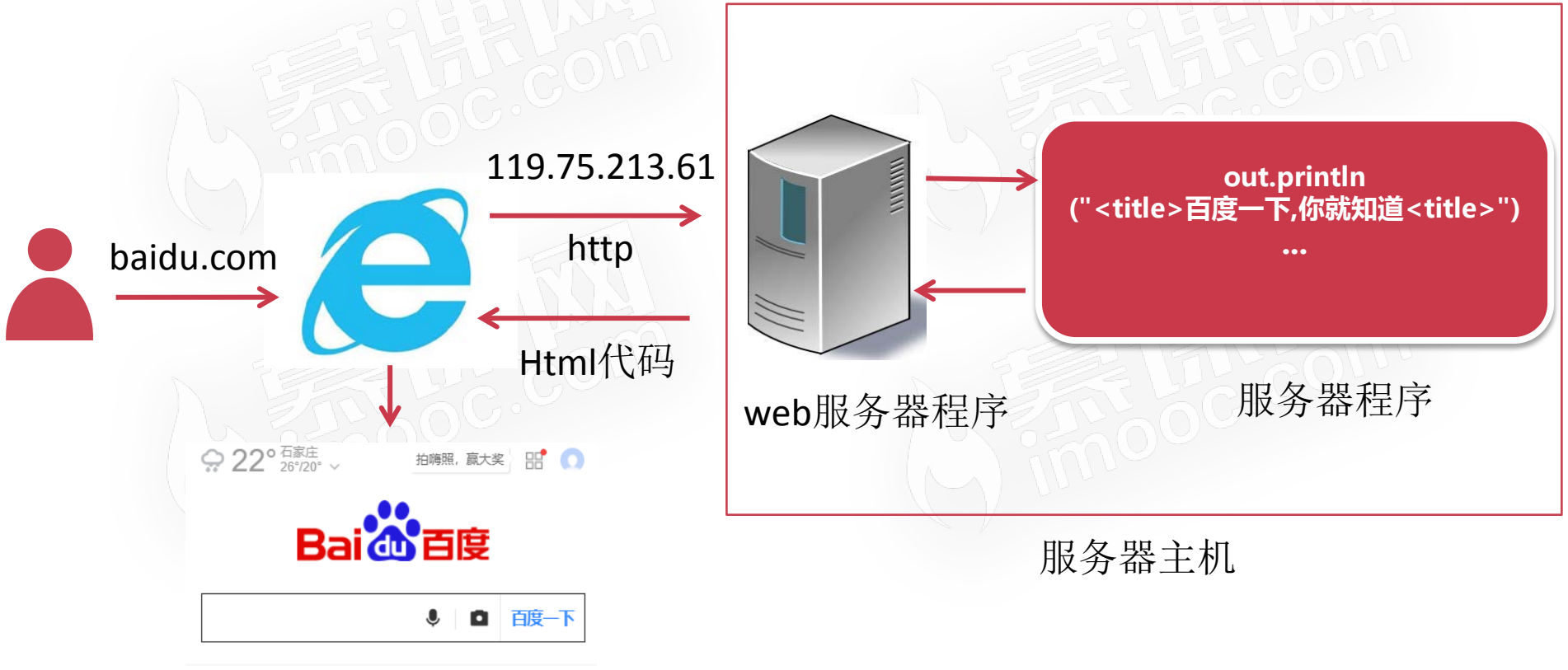
- Tomcat是Apache 软件基金会旗下一款免费的开放源代码的Web 应用服务器程序
- Tomcat是运行Servlet(服务器小程序)的容器



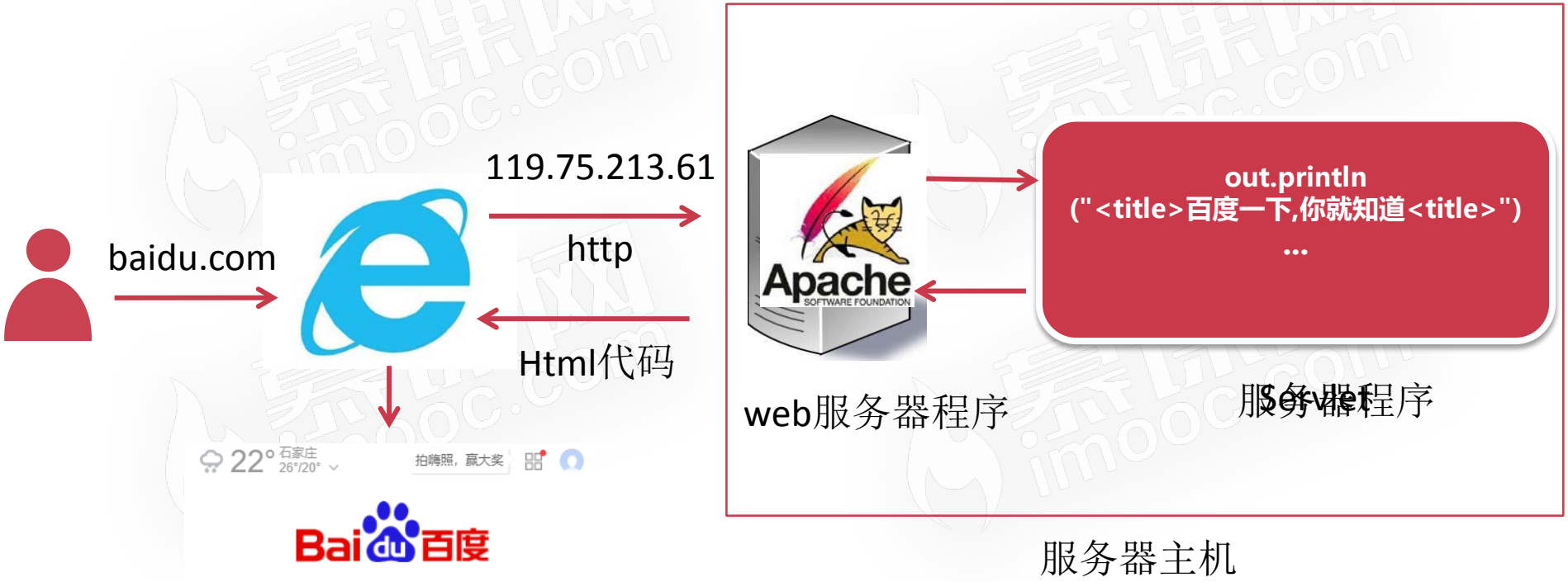
Servlet

- Servlet（Server Applet）服务器小程序，主要功能用于生成动态Web内容
- Servlet是J2EE最重要的组成部分，也是我们学习的重点

Tomcat与Servlet的关系



Tomcat与Servlet的关系



安装Tomcat

- 安装JDK 8
- 安装 Tomcat 8.x

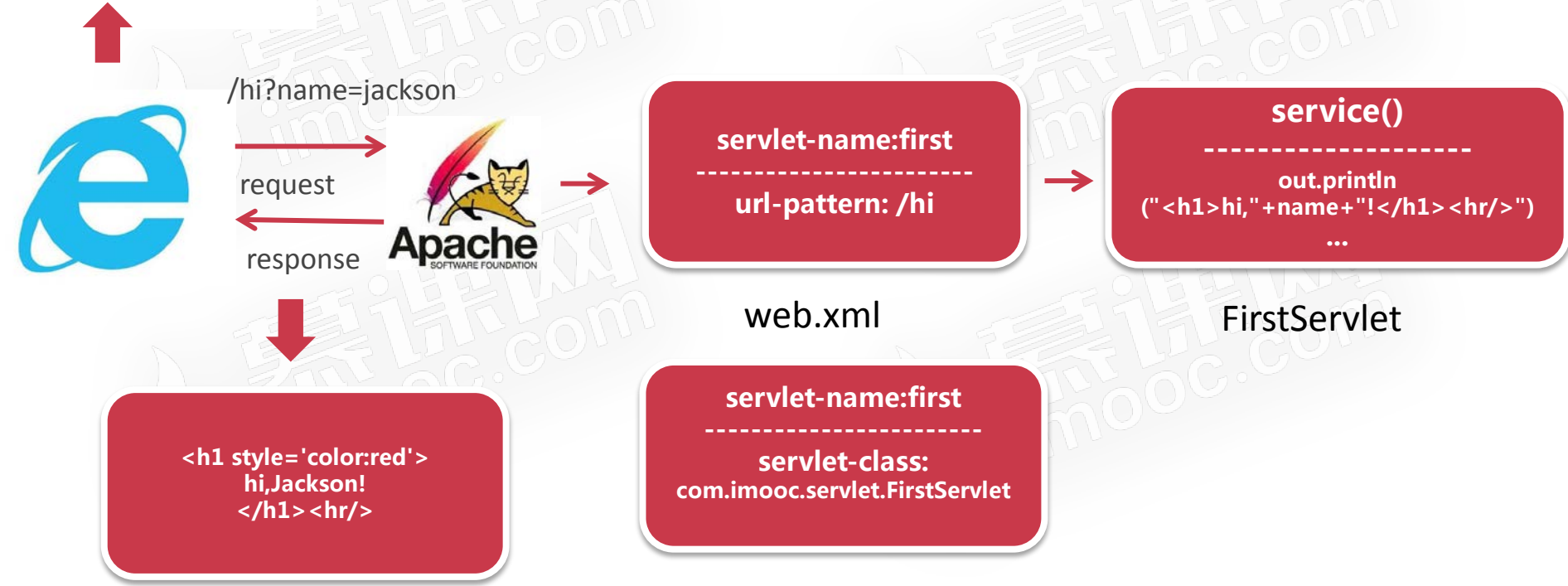
Eclipse For J2EE

- Eclipse 为J2EE提供了专用版本
- Eclipse J2EE最新版本为SimRel（齐美尔）

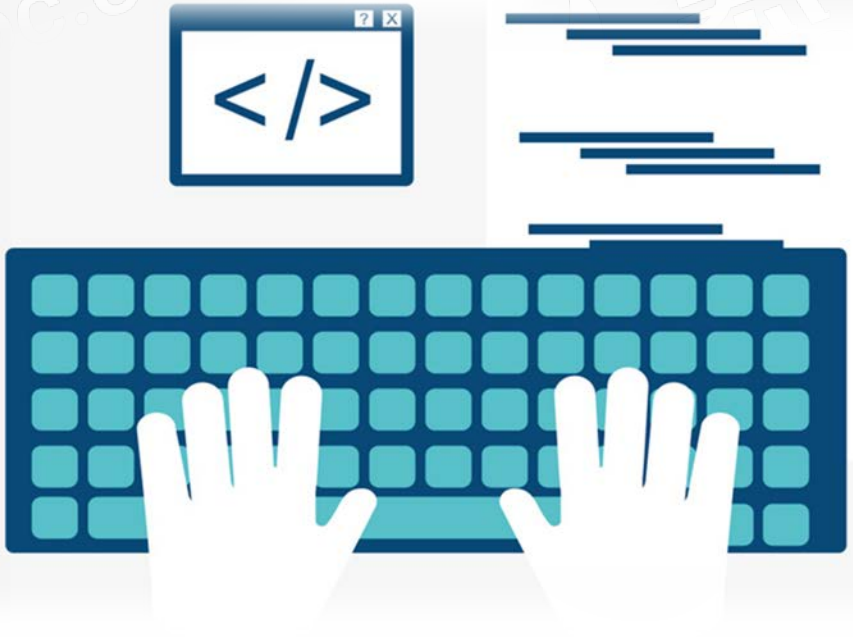
第一个Servlet

图解执行流程

hi,Jackson!



创建Java Web工程



标准Java Web工程结构

组织结构	描述
tomcat安装目录/webapps/	Tomcat应用根目录
/web应用目录/	Java Web应用目录
/web应用目录/index.html | .jsp	默认首页
/WEB-INF	WEB应用的安全目录，用于存放配置文件
/WEB-INF/web.xml	web.xml是“部署描述符文件”， 是该 Web 项目核心配置文件
/WEB-INF/classes	存放编译后的classes文件
/WEB-INF/lib	用于存放web应用依赖的jar文件
/META-INF/MANIFEST.MF	包含 Web 应用的版本等信息

Servlet开发步骤

- 创建Servlet类，继承HttpServlet
- 重写service方法，编写程序代码
- 配置web.xml，绑定URL

Servlet访问方法

- http://IP地址:端口/context-path/url-mapping
- 远程访问使用IP地址，本地访问localhost(127.0.0.1)
- context-path成为“上下文路径”，默认为工程名

Servlet访问方法

- http://IP地址:端口/context-path/url-mapping
- 远程访问使用IP地址，本地访问localhost(127.0.0.1)
- context-path成为“上下文路径”，默认为工程名

请求参数

- 请求参数是指浏览器通过请求向Tomcat提交的数据
- 请求参数通常是用户输入的数据，待Servlet进行处理
- 参数名1=值1&参数名2=值2&参数名n=...

Servlet接收请求参数

- request.getParameter() - 接收单个参数
- request.getParameterValues() - 接收多个同名参数

Get与Post请求方法

- Get方式是将数据通过在URL附加数据显性向服务器发送数据。
  - http://localhost:8080/FirstServlet/sample**?name=zhangsan**
- Post方式会将数据存放在”请求体”中隐性向服务器发送数据
  - http://localhost:8080/FirstServlet/sample
  - 请求体：**name=zhangsan**

Get与Post处理方式

- 所有请求 - service()方法
- Get请求 - doGet()方法
- Post请求 - doPost()方法

Get与Post应用场景

- Get常用于不包含敏感信息的查询功能
  - 例如:https://www.baidu.com/s?wd=imooc&rsv_spt=1

- Post用于安全性要求较高的功能或者服务器的”写“操作
  - 用户登录
  - 用户注册
  - 更新公司账目

Servlet生命周期

- 装载 - web.xml
- 初始化 - init()
- 销毁 - destory()
- 创建 - 构造函数
- 提供服务 - service()

使用注解配置Servlet

- Servlet 3.x 之后引入了“注解Annotation”特性
- 注解用于简化Web应用程序的配置过程
- Servlet核心注解：@WebServlet

启动时加载Servlet

- web.xml使用<load-on-startup>设置启动加载
- <load-on-startup>0~9999</load-on-startup>
- 启动时加载在工作中常用于系统的预处理