

WEB技术

操作系统有进程子系统，使用多进程就可以充分利用硬件资源。进程中可以多个线程，每一个线程可以被CPU调度执行，这样就可以让程序并行的执行。这样一台主机就可以作为一个服务器为多个客户端提供计算服务。

客户端和服务端往往处在不同的物理主机上，它们分属不同的进程，这些进程间需要通信。跨主机的进程间通信需要使用网络编程。最常见的网络编程接口是Socket。

Socket称为套接字，本意是插座。也就是说网络通讯需要两端，如果一端被动的接收另一端请求以提供技术和数据的称为**服务器端**，另一端往往只是发起计算或数据请求，称为**客户端**。

这种编程模式称为Client/Server编程模式，简称**CS编程**。开发的程序也称为CS程序。CS编程往往使用传输层协议（TCP/UDP），较为底层。

1990年，HTTP协议诞生，有了浏览器。在应用层使用文本跨网络在不同进程间传输数据，最后在浏览器中将服务器端返回的HTML渲染出来。由此，诞生了网页开发。

网页是存储在WEB服务器端的文件，浏览器发起HTTP请求后，到达WEB服务程序后，服务程序读取HTML文件并封装成HTTP响应报文返回给浏览器端。

起初网页开发主要指的是HTML、CSS等文件制作，目的就是显示文字或图片，通过超级链接跳转到另一个HTML并显示其内容。

后来，网景公司意识到让网页动起来很重要，傍着SUN的Java的名气发布了JavaScript语言，可以在浏览器中使用JS引擎执行的脚本语言，可以让网页元素动态变化，网页动起来了。

为了让网页动起来，微软使用ActiveX技术、SUN的Applet都可以在浏览器中执行代码，但都有安全性问题。能不能直接把内容直接在WEB服务器端组织成HTML，然后把HTML返回给浏览器渲染呢？

最早出现了CGI（Common Gateway Interface）通用网关接口，通过浏览器中输入URL直接映射到一个服务器端的脚本程序执行，这个脚本可以查询数据库并返回结果给浏览器端。这种将用户请求使用程序动态生成的技术，称为动态网页技术。先后出现了ASP、PHP、JSP等技术，这些技术的使用不同语言编写的程序都运行在服务器端，所以称为**WEB后端编程**。有一部分程序员还是要编写HTML、CSS、JavaScript，这些代码运行在浏览器端，称为**WEB前端编程**。合起来称为Browser/Server编程，即**BS编程**。

HTML

HTML（HyperText Markup Language）超文本标记语言，它不同于编程语言。

超文本就是超出纯文本的范畴，例如描述文本的颜色、大小、字体等信息，或使用图片、音频、视频等非文本内容。

HTML由一个个标签组成，这些标签各司其职。有的提供网页信息，有的负责图片，有的负责网页布局。

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>首页</title>
</head>
<body>
<h1>马哥教育欢迎您</h1>
</body>
</html>
```

超文本需要显示，就得有软件能够呈现超文本定义的排版格式，例如显示图片、表格，显示字体的大小、颜色，这个软件就是浏览器。

超文本的诞生是为了解决纯文本不能格式显示的问题，是为了好看，但是只有通过网络才能分享超文本的内容，所以制定了HTTP协议。

CSS (Cascading Style Sheets) 层叠样式表

HTML本身为了格式化显示文本，但是当网页呈现大家面前的时候，需求HTML提供更多样式能力。这使得HTML变得越来越臃肿。这促使了CSS的诞生。

1994年，W3C成立，CSS设计小组所有成员加入W3C，并努力研发CSS的标准，微软最终加入。

1996年12月发布CSS 1.0。

1998年5月发布CSS 2.0。

CSS 3采用了模块化思想，每个模块都在CSS 2基础上分别增强功能。所以，这些模块是陆续发布的。

不同厂家的浏览器使用的引擎，对CSS的支持不一样，导致网页布局、样式在不同浏览器不一样。因此，想要保证不同用户使用不同浏览器看到的网页效果一直非常困难。

浏览器

1980年代，Tim Berners-Lee为CERN（欧洲核子研究中心，当时欧洲最大的互联网节点）设计基于超文本思想的ENQUIRE项目，以促进科研人员之间的信息共享和更新。1989年他编写了《信息化管理：建议》一文，并构建基于Internet的Hypertext系统，并在CERN开发了World Wide Web项目，打造了世界上第一个网站，于1991年8月6日正式上线。

Tim Berners-Lee于1990年发明了第一个浏览器，还发明了HTTP协议。

1994年MIT他创建了W3C。W3C万维网联盟，负责万维网持续发展。他提出W3C的标准应该基于无专利权、无版税。

Marc Andreessen于1993年发明了Mosaic浏览器，看到了这个技术的前景，不久后他成立自己的公司——网景Netscape。1994发布了Netscape Navigator浏览器，席卷全球。1995年微软发布IE，开启第一次浏览器大战，最终后来居上。

1999年网景被AOL收购，收购后不久，Netscape公开了浏览器代码，并创建了Mozilla组织。Mozilla组织使用Gecko引擎重写浏览器。

Mozilla组织使用Gecko引擎发布了几款浏览器，最终于2004年更名为Firefox浏览器。

2003年5月，网景被解散。

AOL于2007年12月宣布停止支持Netscape浏览器。

Apple的Safari于2003发布第一个测试版。

2008年Google的Chrome浏览器带着 **V8** 引擎横空出世。

浏览器内两大核心：渲染引擎和JS引擎

JavaScript

JavaScript 简称JS，是一种动态的弱类型脚本解释性语言，和HTML、CSS并称三大WEB核心技术，得到了几乎主流浏览器支持。

1994年，网景Netscape公司成立并发布了Netscape Navigator浏览器，占据了很大的市场份额，网景意识到WEB需要动态，需要一种技术来实现。

1995年9月网景浏览器2发布测试版本发布了LiveScript，随即在12月的测试版就更名为**JavaScript**。同时期，微软推出IE并支持JScript、VBScript，与之抗衡。

1997年，网景、微软、SUN、Borland公司和其他组织在ECMA确定了ECMAScript语言标准。JS就成为ECMAScript标准的实现之一。

2008年后随着chrome浏览器的V8引擎发布，2009Nodejs诞生，从此，便可以在服务器端真正大规模使用JavaScript编程了。也就是说JavaScript也真正称为了服务器端编程语言了。

静态网页技术

早期的HTML设计之初，只能HTML，里面可以显示文字、图片，使用CSS来控制颜色、字体大小等。再后来引入了JavaScript就可以是网页可以人机交互、可以让元素动起来。但这都不是内容的动态变化。

动态网页技术

网页的内容是后端根据用户从浏览器端提交的请求不同，通过后台的程序将内容临时拼凑好，生成HTML，返回到浏览器端，通过浏览器端渲染呈现。常见的有ASP、JSP、PHP、Nodejs等。

开发语言

语言：人与人交流的沟通表达方式

计算机语言：人与计算机之间交互沟通的语言

语言分类

- 面向机器语言
 - 机器指令或对应的助记符，与自然语言差异太大
 - 汇编语言
- 面向过程语言
 - 做一件事情，排出个步骤，第一步干什么，第二步干什么，如果出现情况A，做什么处理，如果出现了情况B，做什么处理
 - 问题规模小，可以步骤化，按部就班处理
 - C语言
- 面向对象语言
 - 一种认识世界、分析世界的方法论。将万事万物抽象为各种对象
 - 类是抽象的概念，是万事万物的抽象，是一类事物的共同特征的集合

- 对象是类的具象，是一个实体
- 问题规模大，复杂系统

按照与自然语言的差异分类

- 低级语言
 - 机器语言、汇编语言都是面向机器的语言，都是低级语言。不同机器是不能通用的，不同的机器需要不同的机器指令或者汇编程序
- 高级语言
 - 接近自然语言和数学语言的计算机语言

常见语言

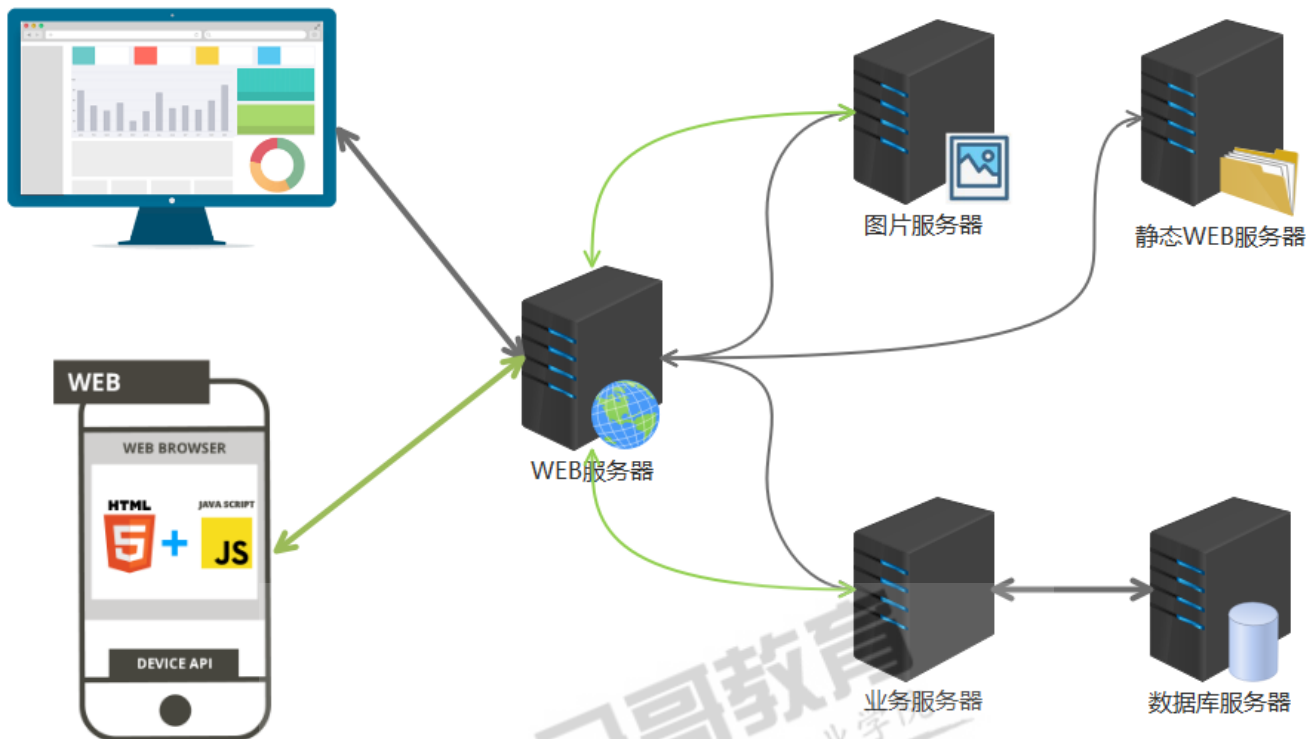
- C语言
 - 面向过程编程，只有函数
 - 操作系统编程、单片机编程等领域
- C++语言
 - 底层高效开发
 - 面向对象，学习难度极大，目前标准发展有点乱
- Java
 - WEB开发领域第一，延伸领域极多，库丰富
 - 大数据领域生态完整
- Python
 - 入门门槛低，非专业程序员容易接受，他们有丰富的专业知识，但计算机专业知识不够
 - Python简洁的语法，不需要让他们关注背后的细节，可以让他们较容易的掌握并开始编程
 - 运维开发
- Javascript
 - 网景公司发明的动态脚本语言，前端开发第一语言
 - JavaScript才是目前 前后端通吃的全栈语言
 - 前端执行的JS代码，需要从服务器端发送到浏览器端，在浏览器端使用JS引擎执行
- Go
 - C语言之父Ken Thompson亲自参与设计
 - 静态编译型语言，但结合了动态解释性语言的特点，例如GC
 - 充分利用多核，适合高并发场景

WEB架构

后端资源分类：

- 静态资源
 - 图片：一旦创建好，图片文件不再改变。图片数目多，占用磁盘空间多，一般使用单独的图片服务器

- HTML、CSS、JavaScript：这些文本是文本的，有前端工程师可以修改，但修改次数较少，一段时间都不变
- 动态资源
 - 内容有后台程序动态生成，比如查询数据库，将查询结果生成成为HTML



PC端或移动端浏览器访问

从静态服务器请求HTML、CSS、JS等文件发送到浏览器端，浏览器端接收后渲染在浏览器上

从图片服务器请求图片资源显示

从业务服务器访问动态内容，动态内容是请求后有后台服务访问数据库后得到的，最终返回到浏览器端

WEB App访问

内置了HTML和JS文件，不需要从静态WEB服务器下载JS或HTML。为的就是减少文件的发送，现代前端开发使用的JS文件太多或太大了

有必要就从图片服务器请求图片，从业务服务器请求动态数据

客户需求多样，更多的内容还是需要由业务服务器提供，业务服务器往往都是由一组服务器组成。

后台应用架构

- 单体架构
 - JSP、Servlet
 - 打包成一个jar、war部署
 - 服务器有开源的tomcat、jetty。商用的有Jboss、weblogic、websphere、glassfish商用的
- Dubbo
 - 分布式服务框架
 - 将单体程序分解成多个功能服务模块，模块间使用Dubbo框架提供的高性能RPC通信
 - 阿里开源贡献给了ASF

- 内部协调使用Zookeeper，实现服务注册、服务发现。有服务治理
- Spring cloud 微服务
 - 将单体应用拆分为粒度更小的单一功能服务
 - RPC通信
 - 需要更高的运维水平，服务太多了需要服务治理

不同的应用架构，部署方式也有不同。

Java

历史

最早就是印度尼西亚的爪哇岛，人口众多，盛产咖啡、橡胶等。

Java语言最早是在1991年开始设计的，期初叫Oak项目，它初衷是跑在不同机顶盒设备中的。

1993网景公司成立。Oak项目组很快他们发现了浏览器和动态网页技术这个巨大的市场，转向WEB方向。并首先发布了可以让网页动起来的Applet技术（浏览器中嵌入运行Java字节码的技术）。

在1995年，一杯爪哇岛咖啡成就了Java这个名字。

Sun公司第一个Java公开版本1.0发布于1996年。口号是“一次编写，到处运行”(Write once, Run anywhere)，跨平台运行。

1999年，SUN公司发布了第二代Java平台(Java2)。

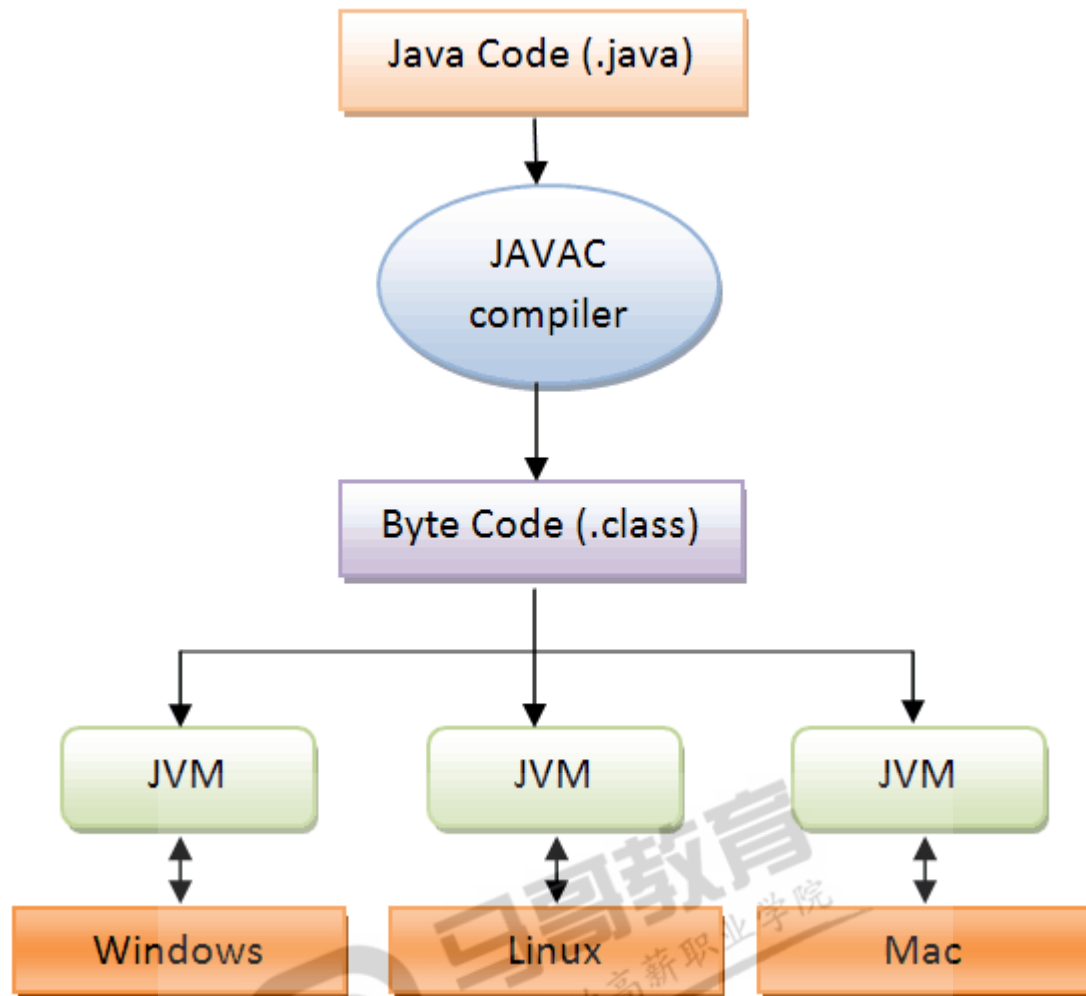
2009年4月20日，Oracle甲骨文公司宣布将以每股9.50美元，总计74亿美金收购SUN（计算机系统）公司。2010年1月成功收购。

2010年，Java创始人之一的James Gosling离开了Oracle，去了Google。

组成

Java包含下面部分：

- 语言、语法规则。关键字if、for、class等等
- 编写源代码source code
- 依赖库，标准库、第三方库。底层代码太难使用开发效率低，封装成现成的，好比净菜直接可以烧了，但是什么口味程序员自己定
- JVM虚拟机。字节码运行在JVM之上



由于操作系统ABI（应用程序二进制接口）不一样，采用编译方式，需要为不同操作系统编译二进制程序。

1995年，Java发布Applet技术，Java程序在后台编译成字节码，发送到浏览器端，在浏览器中运行一个Applet程序，这段程序是运行在另外一个JVM进程中的。

但是这种在客户端运行Java代码的技术，会有很大的安全问题。1997年CGI技术发展起来，动态网页技术开始向后端开发转移，在后端将动态内容组织好，拼成HTML发回到浏览器端。

Java动态网页技术

servlet

本质就是一段Java程序

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

public class HelloWorld extends HttpServlet {
    private String message;
    public void init() throws ServletException
    {
        message = "Hello World";
    }
}
```

```

public void doGet(HttpServletRequest request,
                  HttpServletResponse response)
    throws ServletException, IOException
{
    response.setContentType("text/html");

    PrintWriter out = response.getWriter();
    out.println("<h1>" + message + "</h1>");
}

public void destroy()
{
}
}

```

在Servlet中最大的问题是，HTML输出和Java代码混在一起，如果网页布局要调整，就是个噩梦。

jsp (Java Server Pages)

提供一个HTML，把它变成一个模板，也就是在网页中预留以后填充的空，以后就变成了填空了。

```

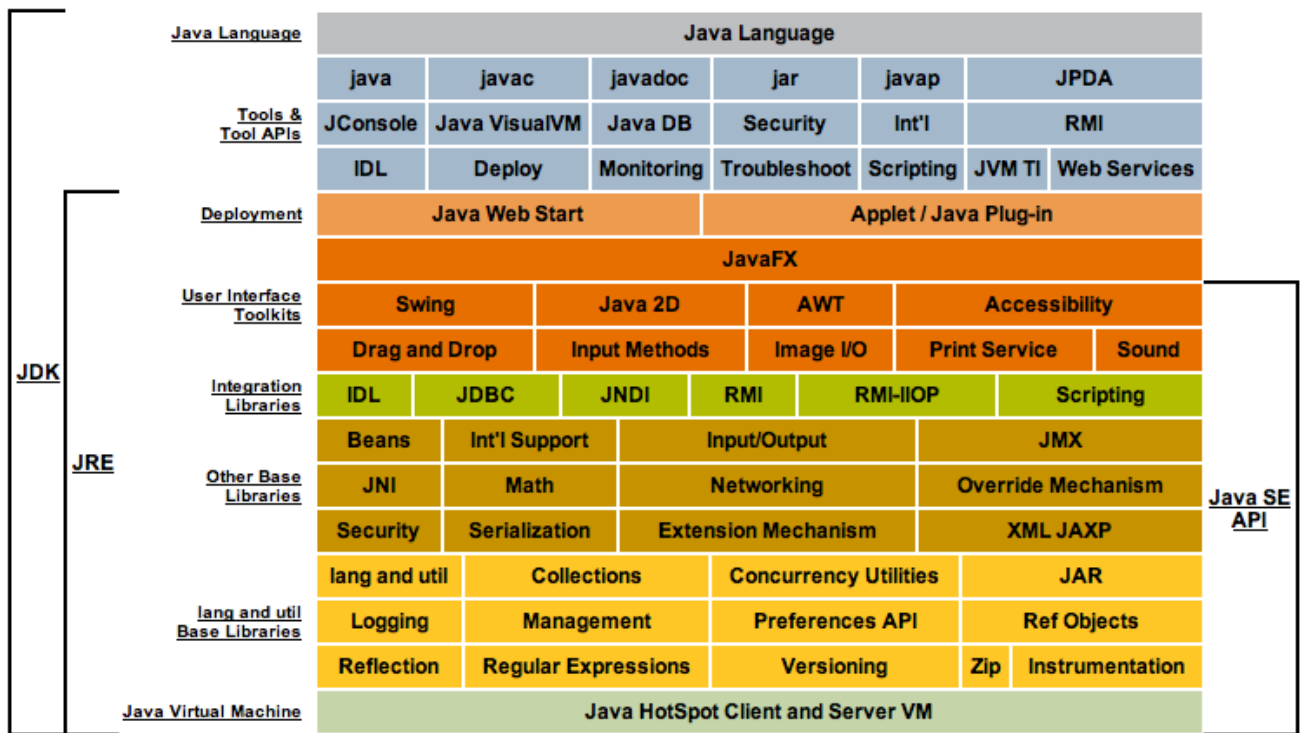
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>jsp例子</title>
</head>
<body>
后面的内容是服务器端动态生成字符串，最后拼接在一起
<%
out.println("你的 IP 地址 " + request.getRemoteAddr());
%>
</body>
</html>

```

JSP是基于Servlet实现，JSP将表现和逻辑分离，这样页面开发人员更好的注重页面表现力更好服务客户。

JSP 先转换为 Servlet的源代码.java文件（Tomcat中使用Jasper转换），然后再编译成.class文件，最后就可以在JVM中运行了。

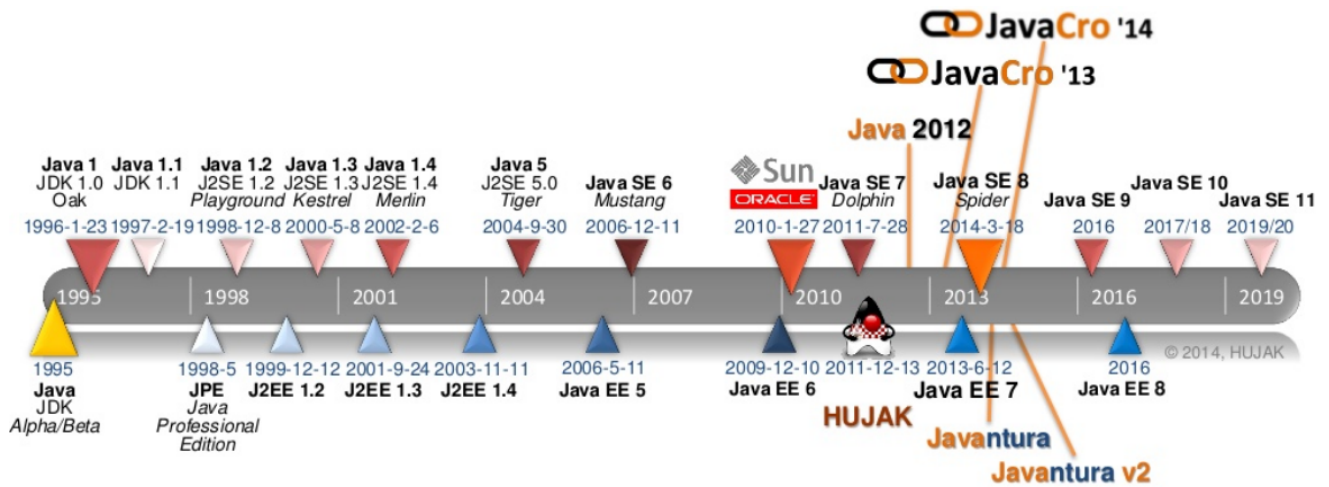
JDK



JRE：它是Java Runtime Environment缩写，指Java运行时环境， 包含 JVM + Java核心类库

JDK：它是Java Development Kit，即 Java 语言的软件开发工具包。





JDK也就是常说的J2SE，在1999年，正式发布了Java第二代平台，发布了三个版本：

J2SE：标准版，适用于桌面平台

J2EE：企业版，适用于企业级应用服务器开发

J2ME：微型版，适用于移动、无线、机顶盒等设备环境

2005年，Java的版本又更名为JavaSE、JavaEE、JavaME。

Servlet、Jsp都包含在JavaEE规范中。

JDK7、JDK8、JDK11是LTS（Long Term Support）

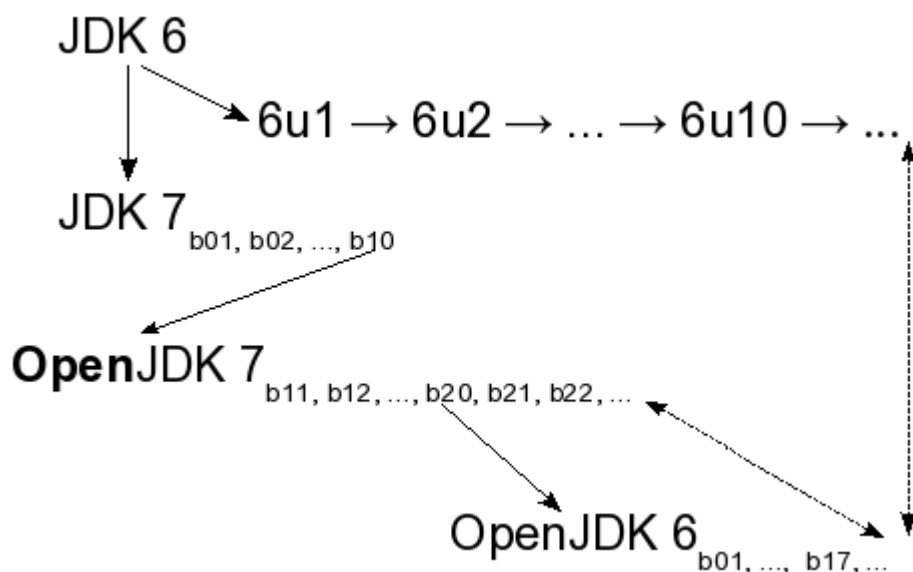


版本	项目名称	发行日期
JDK 1.1.4	Sparkler (宝石)	1997-09-12
JDK 1.1.5	Pumpkin (南瓜)	1997-12-13
JDK 1.1.6	Abigail (阿比盖尔-女子名)	1998-04-24
JDK 1.1.7	Brutus (布鲁图-古罗马政治家和将军)	1998-09-28
JDK 1.1.8	Chelsea (切尔西-城市名)	1999-04-08
J2SE 1.2	Playground (运动场)	1998-12-04
J2SE 1.2.1	none (无)	1999-03-30
J2SE 1.2.2	Cricket (蟋蟀)	1999-07-08
J2SE 1.3	Kestrel (美洲红隼)	2000-05-08
J2SE 1.3.1	Ladybird (瓢虫)	2001-05-17
J2SE 1.4.0	Merlin (灰背隼)	2002-02-13
J2SE 1.4.1	grasshopper (蚱蜢)	2002-09-16
J2SE 1.4.2	Mantis (螳螂)	2003-06-26
Java SE 5.0 (1.5.0)	Tiger (老虎)	2004-09-30
Java SE 6.0 (1.6.0)	Mustang (野马)	2006-04
Java SE 7.0 (1.7.0)	Dolphin (海豚)	2011-07-28
Java SE 8.0 (1.8.0)	Spider (蜘蛛)	2014-03-18
Java SE 9		2017-09-21
Java SE 10		2018-03-14 [3]

JDK协议是JRL(JavaResearch License)协议

OpenJDK

OpenJDK是Sun公司采用GPL v2协议发布的JDK开源版本，于2009年正式发布。



<https://openjdk.java.net/projects/jdk6/>

OpenJDK 7是基于JDK7的beta版开发，但为了也将Java SE 6开源，从OpenJDK7的b20构建反向分支开发，从中剥离了不符合Java SE 6规范的代码，发布OpenJDK 6。所以OpenJDK6和JDK6没什么关系。

OpenJDK使用GPL v2可以用于商业用途。

安装JDK

在Centos中，可以使用yum安装openjdk。

```
# yum install java-1.8.0-openjdk

# java -version
openjdk version "1.8.0_212"
OpenJDK Runtime Environment (build 1.8.0_212-b04)
OpenJDK 64-Bit Server VM (build 25.212-b04, mixed mode)
```

本次使用Oracle官网的JDK 8的rpm安装

```
# yum install jdk-8u191-linux-x64.rpm

# java
# java -version
java version "1.8.0_191"
Java(TM) SE Runtime Environment (build 1.8.0_191-b12)
Java HotSpot(TM) 64-Bit Server VM (build 25.191-b12, mixed mode)
```

安装目录为/user/java下

Java全局配置

```
# vi /etc/profile.d/jdk.sh
export JAVA_HOME=/usr/java/default
export PATH=$JAVA_HOME/bin:$PATH
# . /etc/profile.d/jdk.sh
```

