

- WEB 和 Javaweb 有和异同
 - WEB 是用来展示 互联网多媒体程序的一套服务
 - 常见的有 Apache Nginx Tomcat 都是 web服务器
 - 那么 Javaweb 就是用 Java 语言来处理 web 程序的一套基础服务
- B/S 架构, C/S架构
 - 早些年的都是 CS架构, 因为那个年月啊, 网速太慢了, 基本上都是 2兆, 5兆, 8 兆 那样的网速! 支持不起来 高性能的应用
 - 后来 网速上来了 300兆, 100兆起步! 然后就是 计算机性能也好了, JS 效率高了, 所以 BS架构就兴盛起来了
 - B/S 代码在服务器端, 代码可控! 非常的有效率
 - B/S 端一般, 电信系统, 电力系统, 很多服务系统都是B/S端
 - C/S 端 一般用于保密性比较强的, 实时性比较高的 业务场景, 比如说 微信, QQ 都是CS端
- 静态网页
 - 静态网页, 就是不经过服务处理, 直接读取查看的静态资源
 - 静态网页优点, 图片, 音频, 视频, 特点是速度快, 并发高
 - 静态网页缺点, 每一个页面都是手动生成, 或者程序生成的, 实时性很差
 - 交互性差, 网页留言, 视频弹幕 都是不能实现的, 静态网页只能查看静态的资源(html, js, css, mp3, mp4)
- 动态网页
 - 动态网页是由服务器处理过的请求, 主要的页面都是动态生成的, 数据是可变的, 实时的, 所以交互性更好
 - 主要处理动态网页的语言有 (Java , PHP, C#, Python) 等等
 - 当没有用户访问我们站点的时候, 网页是不存在的, 有请求的时候, 是动态生成出来的
- 数据库
 - 既然动态网页是动态生成的, 那么我们依据数据库的资源生成网页
 - 或者将数据存储到 缓存中
- HTTP协议
 - 浏览器 URL 协议
 - <http://draw.80boys.com/index.html>
 - 域名级数不同 也是不同站点
 - img.baidu.com pic.baidu.com 不是一个站点
 - <http://img.baidu.com> <https://img.baidu.com> 也不是一个站点

▼ 常规

```

请求网址: http://192.168.3.21:5000/webapi/entry.cgi
请求方法: POST
状态代码: ● 200 OK
远程地址: 192.168.3.21:5000
引荐来源网址政策: strict-origin-when-cross-origin

```

- web 协议
 - 基于TCP层, 需要三次握手
 - 传输的是 文本流

- 传输的内容 分相应头和响应体, 中间用两个换行符分割
- 响应头是 KEY VALUE 格式
- 响应体是 完整的 HTML 代码
- 响应头里面 包含了很多渲染细节
- HTTP 状态相应码
 - 2 0 0 系列 200, 201, 202 都是成功
 - 3 0 0 系列 301, 302 跳转系列
 - 4 0 0 系列 403 404 鉴权或者查不到
 - 5 0 0 系列 500 503 都是服务器异常
- TOMCAT
 - 我们刚才说了, 网页需要一个服务器作为服务, 那么我们Java web 承担服务器角色的是 Tomcat
 - Tomcat 是一个用Java 语言开发的 web服务器, 主要用来处理 web 静态资源请求的, 如果遇到了 动态懂球
 - 则 转发给 Servlet, 那么 Servlet 也是一个用 Java语言开发的库, 专门用来解析 Tomcat 转发的请求, 并封装 web资源
 - 运行Tomcat
 - 由于Tomcat 是Java 语言开发的, 所以我们运行Tomcat 就必须有 Java环境!
 - Tomcat 运行的时候, 会检查 JAVA_HOME 环境变量, 所以我们运行Tomcat之前要配置一下 JAVA_HOME 变量
 - JAVA_HOME 一般情况下, 我们安装 Java JDK 的时候, JDK 会帮我们处理 这个变量
 - 如果 没有这个变量, 需要我们自己 设置一下
 - 也就是说 我们必须明确知道 自己的 Java JDK 安装在哪里了
- Tomcat 的使用
 - 我们可以自己下载 Tomcat 然后通过配置档 关联 编辑器使用
 - 我们也可以 使用 Maven 插件 引入Tomcat 编写代码
 - 我们在 开发 Servlet JSP 过程中 会用到很多 插件, 需要从 Tomcat 里面取
 - Tomcat 一个核心文件是 web.xml 它是站点 处理请求和转发的核心配置档
 - 我们在这个 xml 文件内 配置 Servlet 对象和 URL 匹配从而把 请求转发到 我们自己的类上面
 - 我们自己的业务类 需要 实现 Servlet 接口, 或者继承 HttpServlet 类
 - Servlet 有五个 接口

```

1 //Servlet 每个Servlet 只初始化一次
2 void init(ServletConfig var1) throws ServletException;
3 ServletConfig getServletConfig();
4 //每一个请求都会调用
5 void service(ServletRequest var1, ServletResponse var2) throws ServletException,
  IOException;
6 String getServletInfo();
7 //Servlet 在注销的时候 才会调用一次

```

8 `void destroy();`

- 一般情况下 我们使用 `HttpServlet`, 因为 这个类已经帮我们处理好了 请求分类
- `HttpServlet` 在 `Service` 函数内对方法进行了 判断和转发