

Web 的特点

Web 是一种分布式超媒体系统；
Web 是多媒体化的和易于导航的；
Web 是与平台无关的；
Web 是动态的；
Web 是交互的；

TCP/IP 每一层的作用

网络接口层：负责将 IP 分组封装成适合在具体的物理网上传输的帧
网络层：解决网络互联中的 IP 分组的寻址问题
传输层：负责维护信息段的完整性，提供端到端的通信
应用层：提供了网上计算机之间的各种应用服务

HTTP 处理事务的4个步骤

客户端浏览器与 WEB 服务器建立连接；
客户端浏览器向 WEB 服务器提出请求，在请求中指明所要求的特定文件；
WEB 服务器响应客户端浏览器的请求；
客户端浏览器与 WEB 服务器断开连接；

应用 CSS 样式的4种方法

直接定义 HTML 标记中的 Style 属性

```
<HTML标记名称 style="样式属性1:属性值1;...">
```

在 HTML 文档内定义内部样式表

```
<style type="text/css">  
    选择符A{样式属性1:属性值1;...}  
</style>
```

嵌入外部样式表

```
<style type="text/css">  
    @import url("外部样式表文件名")  
</style>
```

链接外部样式表

```
<link type="text/css" rel="stylesheet href="外部样式表文件名">
```

CSS 布局理念

将页面用 Div 分块；
设计各块的位置；
用 CSS 定位；

CSS 的选择器及格式

id 选择器

#标识{} 标记#类名{}

class 选择器

*.类名{} 标记.类名{}

标签选择器

标记{}

JavaScript 的基本特点

是一种脚本语言；
基于对象；
安全性；
跨平台性；
动态；

JSP 的特点

一次编译，多次、多处运行，代码的执行效率高；
组件的重用性；
将内容的生成与显示进行分离；

JSP 程序的执行过程

客户端向服务器发出请求；
JSP 服务器将 JSP 翻译成 Servlet 源代码；
将生成的 Java 代码进行编译，得到 .class 文件，将 .class 文件加载到内存中执行；
把执行结果（标准的 HTML 文本）作为响应发送至客户端由浏览器解释显示；

JSP 元素

JSP 元素分为3种类型，脚本元素、指令元素、动作元素
JSP 脚本元素主要包括注释、声明、表达式、脚本程序；
JSP 指令包括 include 指令、page 指令、taglib 指令；
JSP 动作包括 <jsp:include> 、 <jsp:forward> 、 <jsp:plugin> 、 <jsp:useBean> 、 <jsp:setProperty> 、 <jsp:getProperty> 、 <jsp:params>

JSP 内置对象包括

request、response、session、out、application、pageContext、page、config、exception

JSP 内置对象的作用域

application：对象可以在与创建它的 JSP 页面 属于相同的 Web 应用程序 的任意一个 JSP 中被访问；

session：对象可以在与创建它的 JSP 页面 共享相同的 HTTP 会话 的任意一个 JSP 中被访问；

request：对象可以在与创建它的 JSP 页面 监听的 HTTP 请求相同 的任意一个 JSP 中被访问；

page：对象只能在创建它的 JSP 页面中被访问；

include 动作与 include 指令的区别

include 指令静态包含其他文件的内容，组合成一新文件，然后编译成一个 class 文件再执行，include 动作动态包含其他文件，包含页面与被包含页面是两个文件，JSP 编译器分别对这两个文件进行编译，生成两个 class 文件

forward 动作标记和 response 对象的 sendRedirect() 方法的区别

forward 动作标记和 response 对象的 sendRedirect() 方法都可以使页面重定向，但 `<jsp:forward>` 动作只能在本网站内转向，而 response 对象的 sendRedirect() 方法可以跳转到任何页面

Session 对象

使用 session 对象保存客户访问网站期间，在多个页面之间跳转浏览的信息。从客户打开浏览器并连接到服务器开始，到客户关闭浏览器离开服务器为止，称为一个会话（session）

Application 对象

application 对象是服务器运行期间所有客户共享的对象。它用于客户之间的数据共享，类似于服务器运行期间的全局变量

Cookies

是由 WEB 服务器管理的存放在客户计算机中的一个数据集合，里面包含了浏览器与多个网站的相关信息

Out 对象

out 对象是向客户端输出流进行写操作的对象，用于各种数据的输出

Exception 对象

exception 对象用来发现、捕获和处理异常

Page 对象

page 对象是指向当前 JSP 程序本身的对象

Session 对象的特点

当用户与服务器连接时，服务器会为每个用户创建一个 session 对象，并设定其中内容。创建的 session 对象之间相互独立，服务器可以借此来辨别用户信息，进而提供个别服务。session 常用于跟踪用户的会话信息，其属性可以在多个页面之间共享

application 对象与 session 对象的不同

每个客户拥有自己的 session 对象，保存客户的自有信息，所有客户共享一个 application 对象，保存服务器运行期间所有客户的共享信息；

session 对象生命期从客户打开浏览器并连接到服务器开始，到客户关闭浏览器离开服务器为止，在客户的多个请求期间持续有效，application 对象生命期从服务器启动开始，到服务器关闭为止；

可以使用 session 对象存储某个客户在一个会话期间的数据，使用 application 对象存储服务器运行期间所有客户共享的变量；

JDBC

JDBC 是一种访问数据库的技术标准，是一种能通过 Java 语言访问数据库的应用程序接口，由一组用 Java 语言编写的类和接口组成

JDBC API 最大的特点是

对下，JDBC API 封装了各种底层数据源之间的差异，对上，JDBC API 提供标准的 SQL 界面

JDBC API 包括了两个包

java.sql ; javax.sql ;

编写 Java 数据库应用的基本过程 / 连接数据库的基本过程

建立数据源；

导入用到的包；

加载驱动程序；

创建与数据库的连接；

创建语句对象；

编写 sql 语句；

执行 sql 语句；

处理得到的结果集；

关闭相关对象；
处理异常；

Servlet

servlet 是用 Java 语言编写的服务器端程序，它担当客户请求与服务器响应的中间层，属于 JavaEE 中间层技术，是由服务器端调用和执行，可以处理客户端传来的 HTTP 请求，并返回一个响应

JavaBean

JavaBean 是 JavaWeb 程序的重要组成部分，是一个可以重复使用的软件组件，是一个遵循特定写法的 Java 类，它封装了数据和业务逻辑，供 JSP 或 Servlet 调用，完成数据封装和数据处理功能

Filter

Servlet 过滤器（Filter）是在服务器上运行的，且位于请求与响应中间起过滤功能的程序，它是一个 Java 组件，是能改变 HTTP 请求、响应与头信息中内容的可重用代码

Filter 的工作原理大致是

在 servlet 作为过滤器使用时，它可以对客户请求进行处理，处理完毕后它会交给下一个过滤器处理，这个客户的请求在过滤器链中逐个处理，直到请求发送到目标为止

Servlet 的运行机制

servlet 由 servlet 容器来负责 servlet 实例的查找、创建以及整个生命周期的管理，servlet 整个生命周期分为4个阶段：

加载、初始化、调用和销毁。servlet 具体生命周期如下：

servlet 容器加载 servlet 类并实例化一个 servlet 实例对象；

servlet 容器调用该实例对象的 init() 方法进行初始化；

如果 servlet 容器收到对该 servlet 的请求，则调用它的 service() 方法处理请求返回响应结果；

servlet 容器卸载该 servlet 前调用它的 destroy() 方法；

MVC

MVC 是软件工程中的一种软件架构模式，MVC 模式将整个应用系统划分为三个基本部分：模型（Model）、视图（View）和控制器（Controller）

Model 2

所谓模式2（Model 2）就是指基于 MVC 模式来构建 JavaWeb 应用的开发模式，通常将 Web 应用架构划分为三个层次：

视图层：主要充当用户的操作接口，负责数据的输入与结果的输出；

控制层：控制整个应用的流程，将视图层提交的数据交付给业务逻辑层进行处理，并将结果返回至视图层；

业务逻辑层：应用的核心，主要功能包括数据处理、数据维护以及业务逻辑的实现；

在 Model 2 中，系统的行为时序如下

servlet 充当控制器角色，负责接收客户端请求并处理请求；

根据请求类型，servlet 可以直接存取数据库中的数据，也可以创建 JavaBean 并将请求的结果作为初始化参数传入 JavaBean 中；

servlet 将请求传送给适当的 JSP，用于显示结果；

JSP 从 JavaBean 中读取数据，由 JavaBean 与数据库进行交互；

JSP 返回客户端；

「CSS定位 / 基本的 HTML 页面（链接、图像、表格）」

「Servlet / Filter」

对 Web 的理解，对 Web 前景的看法？