# 抛开书本不谈，你行我也行——Java Web复习简要

本提纲为个人整理，只适用于有一定html、css、javascript、servlet、jsp基础的学习，**如有出错，纯属意外**

## HTML+CSS+Java Script

### 名词解释

**HTML:**(HyperText Markup Language, **超文本标记语言**)**一种含有一套语法规则的文本标记语言**

**CSS:**(Cascading Style Sheets，**级联样式表**)样式是HTML的扩展，可把页面显示样式与显示内容分开。

**Java Script:** JavaScript是一种基于对象(Object)和事件驱动(Event driven)的、嵌在HTML文档中的**脚本语言**。

**HTML5:** HTML5是一种**网络标准**，相比现有的HTML4.01和XHTML1.0，可以实现更强的页面表现效果

**Dom**：DOM (Document Object Model) 译为**文档对象模型**，是 HTML 和 XML 文档的编程接口。

ps：

* javascript是一种解释型语言，而非编译型语言
* H5是一种规范标准，与HTML5不等价

### 简答题

### HTML5的新特性（重点）

1）语义特性

2）本地存储特性

3）连接特性

4）网页多媒体特性

5）三维、图形及特效特性

### css选择器的类别及表示方式

1）标签选择器（标签{}）

2）class选择器（.类名{}）

3）id选择器（#.id表识{}）

### javascript特点

1）是一种**脚本语言**

2）是一种**解释型语言**，在程序运行过程中会被逐行解释

3）是一种**基于对象的语言**

4）是一种**安全性语言**。它不允许访问本地的硬盘，也不允许对网络文档进行修改和删除。而只能通过浏览器实现信息浏览和动态交互。

5）**跨平台性**。JavaScript的执行只依赖英语浏览器本身，而与操作环境无关

6）**动态**。只用HTML开发的页面是静态的，而使用JavaScript开发的页面是动态的，可以控制页面对象，可以与客户进行交互，满足客户的进一步请求。

### java与JavaScript的区别（重点）

1）JavaScript是基于对象和事件驱动的脚本语言，java面向对象，是一种编程语言

2）在浏览器中的执行方式不一样：JavaScript是解释型语言；java是编译型语言

3）采用变量时不一样：js是弱类型；java是强类型

4）代码格式不一样：JS需要嵌入到HTML中；java是单独的类

### 编程题

题量过大，这里只给出几个需要注意的

* JavaScript的外部引入，使用**src**

<script src="script.js" async></script>

css的外部引用，使用**href**

<link rel="stylesheet" href="styles.css">

* JavaScript中："=="表示数值相等或字符串相同，"==="不仅要求数值相等，而且类型也要一致，值得注意的是，这里的一致包括对象与类型的区别：

string a = "abc";  
String b = new String("abc");  
//a === b 结果为false

* 在HTML文档中，必须加上头文件<!DOCTYPE>
* CSS优先级：内联样式（行内样式）>内部样式>外部样式

## Servlet + JSP

### 名词解释

**Servlet** : Servlet是用java语言编写的服务器程序，可以接受HTTP请求，并返回一个响应

**JSP**：（Java Server Pages），以servlet为基础，是一种运行在服务器端的脚本语言，是一种动态网页开发技术。它使用JSP标签在HTML网页中插入Java代码。

**HTTP**：超文本传输协议，是专门为Web设计的一种网络通信协议

**Tomacat** : Tomacat 服务器是一个免费的开放源代码的Web应用服务器，属于轻量级应用服务器。

**Fitter**：（过滤器）是一个java组件，是能改变HTTP请求、响应及头信息中内容的可重用代码

**request**：request对象主要用于**接受**客户端通过HTTP协议连接传输到Web服务器端的数据

**response**：response对象用于响应客户请求，由服务器向客户端发送数据，当服务器向客户端传送数据时，JSP容器会自动创建response对象并将信息封装，当JSP容器处理完请求后response对象会被销毁。

### 简答

### HTTP报文结构

HTTP请求报文：请求行、请求头部、空行和请求数据

HTTP响应报文：状态行、信息报头、响应正文

### 简述什么是会话技术

会话技术指一个客户端（浏览器）与Web服务器之间连续发生一系列请求和响应的过程，在Servlet中，提供了两个用于保存会话数据的对象，分别是Cookie和session

### 什么是Servlet？Servlet的运行机制？

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JAVA EE的中间技术，是由服务器端调用和执行的，可以处理客户端传来的HTTP请求，并返回一个响应。

Servlet是由Servlet容器来负责实例的查找、创建以及整个生命周期的管理

Servlet的生命周期分为4个阶段：加载-初始化-调用-销毁

1）Servlet容器加载Servlet类并实例化Servlet实例对象

2）Servlet容器调用该实例对象的init（）方法进行初始化

3）如果Servlet容器收到该Servlet的请求，则调用此实例对象的service（）处理请求，返回响应结果

4）Servlet容器卸载该Servlet前调用它的destroy（）方法，最后由JVM回收

### 什么是Fitter？其作用是什么？接口及作用？（重点）

Fitter(过滤器)是一个java组件，是能改变HTTP请求、响应以及头信息中内容的可重用代码

1）Fitter可在请求到达Servlet之前，拦截客户的请求，也可以根据需要检查请求，修改请求头信息和数据

2）Fitter还可以再响应送至客户端之前，拦截响应，根据需要检查响应，修改响应头信息和数据

Fitter主要功能：通过某种拦截机制对请求进行**预处理**或对响应进行**后处理**，借此修改或者调整请求和响应的资源

* **void init（FilterConfig config）**：Fitter初始化执行的方法，容器调用此方法完成过滤的初始化（只调用一次）
* **void doFilter（ServletRequest request, ServletResponse response, FilterChain chain）:**实现过滤功能，该方法就是对每个请求及响应增加额外的处理，并且通过FitterChain对象的doFitter方法来调用下一个Fitter，如果没有其他的Fitter，则调用最初的servlet ，chain指向下一个过滤器
* **void destroy（）：**Fitter生命周期结束时，自动调用该方法，用于释放过滤器所占用资源

### 什么是JSP？JSP有什么特点？

JSP是一种运行在服务器端的脚本语言，它继承了Java的所有优点，是用来开发Web程序的重要技术。

JSP具有如下特点：

1）一次编译，多次、多处运行,代码的执行效率高

2）组件的重用性

3）将内容的生成和显示进行分离

### 简述JSP的运行过程

1）客户端（浏览器）向服务器（Tomacat）发出请求（Request）

2）JSP服务器将JSP翻译成Servlet源码（.java)

3）将产生的java代码进行编译，称为.class文件，将.class文件加载到内存中执行

4）把执行结果（标准的HTML文本）作为响应（Response）发送至客户端由浏览器解释显示

### JSP的元素有哪些？

1）脚本元素：嵌入到JSP页面中的java代码，包括JSP注释、声明、表达式和脚本段

2）指令元素：是针对JSP引擎设计的，它控制JSP引擎如何处理代码。包括include、page、taglib三大指令

3）动作元素：用于连接所要使用的组件，另外还可以控制JSP引擎的动作。主要由**include**和**forward**动作

### JSP中include动作与include指令的区别（重点）

（动作<jsp:include page = "URL"/> 指令：<%@include file = "URL" %>

include静态指令包含其他文件的内容，组合成一新文件，然后编译成class文件再执行

而include动态则是动态包含其他文件，包含页面和被包含页面是两个文件，JSP编译器分别对两个文件进行编译，生成两个class文件

### JSP中的forward动作指令与response对象中的sendRedirect（）方法的功能与区别是什么？

两者的功能都是实现页面重定向

<jsp:forward>动作只能在本网站内实现跳转（同文件夹之内）

而response.sendRedirct()方法可以跳转到**任意页面**（例如百度）

### get和post方法的区别（重点）

（1）Get是用来从服务器上获取数据，Post是用来向服务器上传数据

（2）Get将表单中的数据按照variable = value的形式，添加到action所指向的URL后面，并且两者使用"?"连接，而各个变量之间使用”&“连接；Post是将表单中的数据放在form的数据体中，按照变量和值对应的方式，传递到action所指的URL

（3）Get是**不安全的**，因为在传输过程中，数据被放在请求的URL中（可以被第三发爬取）；Post是**安全的**，所有操作对用户都不可见G

（4）Get传输的**数据量小**，这主要是因为受URL长度限制；而Post可以**传输大量的数据**，所以在上传文件时只能使用Post

（5）Get限制Form表单的数据集必须为ASCII字符；而Post支持整个ISO10646字符集

（6）Get是Form默认方法

### 请求转发与重定向的区别

重定向：(两次请求，地址栏看得到变化，数据丢失)

response.sendRedirect("相对路径");

请求转发：（一次请求，地址栏看不到改变，数据不丢失，可以getParameter数据，也可以getAttribute数据）

request.getRequestDispatcher("相对路径").forward(request,response);

请求转发与重定向的异同点如下：

请求转发和重定向都可以实现访问一个资源时转向当前应用资源  
请求转发是一次请求一次响应，而重定向为两次请求两次响应  
3）一般情况下应该使用请求转发，减少浏览器对服务器的访问，减轻服务器压力

4）如果需要改变浏览器的地址栏，或更改浏览器的刷新按钮的功能时需要使用重定向

### （servlet）JSP的内置对象

1）**out**：主要用于页面输出；

2）**request**：用于得到用户请求信息；

3）**response**：表示服务器向客户端的回应信息；

4）config：包含服务器的配置信息，可以使用该对象获得Servlet初始化参数；

5）**session**：主要用来保存用户信息；

6）**application**：包含所有用户的共享信息；

7）page：指当前页面转换后的Servlet类的实例；

8）pageContext：表示JSP页面容器，它提供了对JSP页面内所有的对象及名字空间的访问；

9）exception:：表示JSP页面所发生的异常，在错误页中才起作用。

### application对象与session对象的不同之处（重点）

（1）前者是共享的，后者是私有的。每个客户都拥有自己的seesion对象，保存客户各自的信息，如果有十万个访问客户，就有十万个session对象。而十万个客户只共享一个application对象，保存服务器运行所有客户的共享信息

（2）生命周期不同：session对象生命周期从客户打开浏览器与服务器建立连接开始，到客户关闭浏览器为止，在客户的多个请求期间持续有效、application对象生命周期从服务器启动开始，到服务器关闭结束

（3）可以使用seesion对象存储某个客户一个会话期间的数据，例如纪录某个客户的姓名、密码等，使用application对象存储服务器运行期间所有客户共享的变量，例如记录所有客户的访问次数等

### Cookie与session的区别

1）Cookie和HttpSession是保存会话相关数据的技术，其中Cookie将信息存储在浏览器端，是客户端技术，Session将数据保存在服务器端，是服务器端技术

2）Cookie是基于HTTP协议中的Set-Cookie响应头和Cookie请求头进行工作的

3）默认情况下HttpSession是基于一个名称为JSESSIONID 的特殊Cookie工作的

4）浏览器对Cookie具有严格的限制，一个网站能在浏览器中保存多少Cookie是有限制的

5）HttpSession默认是基于Cookie运作的。

### 比较JSP与Servlet的异同（重点）

（1）JSP技术是Servlet技术的扩展，本质上是servlet的简易方式，更加强调应用的外表表达

（2）JSP编译之后就是“类servlet”

（3）Serlet的应用逻辑是在java文件中，并且完全从表示层中的HTML里分离开来。而JSP的情况是Java和HTML可以组合成为一个扩展名为.jsp的文件。JSP更侧重于视图，Servlet主要用于控制逻辑

## JDBC技术

### 名词解释

**JDBC**：JDBC是一种访问数据库的技术标准，它是一种能通过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

**B/S**：web浏览器/web服务器工作模式

**C/S**：客户（client）/服务（server）工作模式

**DBMS**：数据库管理系统

### 简答

### 连接数据库的过程

（1）建立数据源 ：建立数据库，建表

（2）导入用到的包：（java.sql javax.sql)

（3）加载驱动程序

（4）创建与数据库的连接

（5）创建语句对象

（6）编写SQL语句

（7）执行SQL语句

（8）处理得到的结果集

（9）关闭相关对象

（10）处理异常

### JSP中导入数据库需要的包

在JSP中使用JDBC语句访问数据库，正确导入SQL类库的语句是：

<%@ page import="java.sql.\*" %>

### 编程

### 连接到数据库，并实现查询功能（具体情况具体考虑，这里给出示例）

String className = "com.mysql.jdbc.Driver";   
String url = "jdbc:mysql://localhost:3306/bookstore";   
String username = "root";   
String password = "123456";   
String tableName = "book";   
String sql = "select \* from";   
   
try{   
 class.forName(className).newInstance();   
 Connection con = DriverManager.getConnection(url,username,password);   
 Statement stmt = con.creatStatement();   
 ResultSet rst = stmt.excuteQuery(sql + tableName);   
 while(rst.next()){   
 for(int i =0 ; i < rst.getMetaDate().getColumnCount();i++)   
 System.out.print(rst.getString(i+1));   
 System.out.println();   
 }   
}catch(Exception e ){   
 System.out.println(e.toString());   
}

## Model 1 与 Model 2 模式

### 名词解释

**JavaBean** ：JavaBean实际上是一个遵循特定写法的java类，它封装了数据和业务逻辑，提供JSP或servlet调用，完成数据封装和数据处理功能（当然在java中也是可以使用的，简单来说，除去主类外的所有类都可以称为JavaBean）

**MVC**：MVC（Model-View-Controller，模型-视图-控制器模式）是软件工程的一种软件架构模式

### 简答题

### 什么是Javabean，其优点（作用）是什么？

JavaBean实际上是一个遵循特定写法的java类，它封装了数据和业务逻辑，提供JSP或servlet调用，完成数据封装和数据处理功能

使用javabean的优点：

（1）一次性编写

（2）任何地方执行

（3）任何地方重写

### JSP中如何调用javabean？

在JSP中调用JavaBean是通过三个标签来完成的：

（1）**<jsp:useBean>** :声明一个具有一定生存周期以及一个唯一id的JavaBean对象

（2）**<jsp:setProperty>** ：对JavaBean的属性赋值

（3）**<jsp:getProperty>** :获取JavaBean属性的值

### model1是什么？优缺点？行为时序？

Model 1 就是JSP +javabean设计模式

优点：架构简单，适合**小型项目**开发

缺点：JSP的职责不单一，职责过重，不变维护

行为时序：

（1）Request

（2）取得页面参数

（3）调用页面逻辑方法

（4）渲染页面

（5）Response

### model2是什么？优缺点？行为时序？（重点）

Model 2 就是 JSP + severlet + JavaBean设计模式，它是将控制层（Servlet）单独划分出来，专门负责业务流程的控制，接收页面的请求，创建所需的JavaBean实例，并将处理后的数据返回到JSP

优点：职责清楚，比较适合大型项目开发，提高开发效率

缺点：不适合小型项目开发

行为时序：

（1）Servlet充当控制器角色，负责接收客户端请求并处理请求

（2）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并将请求结果作为初始化参数传入JavaBean中

（3）Servlet将请求传送给合适的JSP

（4）JSP从JavaBean中读取数据，由JavaBean与数据库进行交互

（5）JSP返回客户端

### 什么是MVC模式？主要功能？

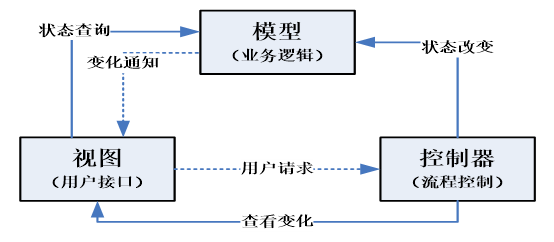
MVC（Model-View-Controller，模型—视图—控制器模式）是软件工程中的一种软件架构模式。

**MVC**模式将整个应用系统分为三个基本部分：模型（Model）、视图（View）和控制器（Controller），其目的就是实现一种动态的程序设计，能使程序的修改和扩展简化，并且使程序模块的重复利用成为可能。

模型：体现应用程序的功能，用于封装应用程序的业务逻辑

视图：主要用于描述模型，视图能够实现模型的有目的显示，允许一个模型对应多个视图

控制器：用于控制应用程序的流程。控制器处理事件并做出响应，事件包括用户的行为和数据模型的改变。用户与视图交互时，可以通过控制器来更新模型状态，通知视图刷新显示。



### Model 2 的三个层次及其作用是什么？（重点）

Model 2就是指基于MVC架构来构建Java Web应用的一种开发模式。 Model 2通常将Web应用的架构根据MVC模式也能划分为三个层次：

1. **视图层**：视图层包括前端的HTML、XML、JSP及Applet等，主要充当用户的操作接口，负责数据的输入及结果的输出。该层的功能对应于MVC模式中的视图部分。
2. **控制层**：控制层的主要工作是控制整个应用的流程。控制层将视图层提交的数据，交付给业务逻辑层处理，并将结果返回至视图层。控制层的角色是介于视图层和业务逻辑层之中，该层的功能相当于MVC模式中的控制器部分。
3. **业务逻辑层**：业务逻辑层是应用的核心部分，它的主要功能包括：数据处理、数据的维护及业务逻辑的实现。该层的功能对应于MVC模式中的模型部分。

在开发Web应用的Model 2中，通常

* 由JSP负责数据显示逻辑；
* 由JavaBean负责业务逻辑；
* 由Servlet负责流程控制

## 框架

### 名词解释

**AOP**：面向切面编程

**IOC**：控制反转

**XML**：可扩展标记语言，标准通用标记语言的子集，是一种用于标记电子文件使其具有结构性的标记语言

**struts**：Struts框架是用于开发基于Java Web应用的开源框架

### 简答

### spring和springMVC的含义

（1）spring是一个一站式的轻量级java开发框架，核心是**控制反转（IOC）**和**面向切面编程（AOP）**

（2）springMVC是spring基础上的一个MVC框架，主要处理web开发的路径映射和视图渲染，属于spring框架中web开发层的一部分

### 什么是cookie？

cookie是小段文本信息，通过cookie可以标识用户身份、记录用户名及密码、跟踪重复用户。cookie在服务器端生成并发送给浏览器，；浏览器将cookie的key/value保存到某个指定的目录中，服务器的名称与值可以由服务器端定义。

### session的原理是什么？

Session对象的原理在于，服务器可以为客户端创建并维护一个所谓的Session对象，用于存放数据。

在创建Session对象的同时，服务器将会为该Session对象产生一个唯一编号，这个编号称之为SessionID，服务器以Cookie的方式将SessionID存放在客户端。当浏览器再次访问该服务器时，会将SessionID作为Cookie信息带到服务器，服务器可以通过该SessionID检索到以前的Session对象，并对其进行访问。

需要注意的是，此时的Cookie中仅仅保存了一个SessionID，而相对较多的会话数据保存在服务器端对应的Session对象中，由服务器来统一维护，这样一定程度保证了会话数据安全性，但增加了服务器端的内存开销。存放在客户端的用于保存SessionID的Cookie会在浏览器关闭时清除。我们把用户打开一个浏览器访问某个应用开始，到关闭浏览器为止交互过程称为一个“会话”。在一个“会话”过程中，可能会向同一个应用发出了多次请求，这些请求将共享一个Session对象，因为这些请求携带了相同的SessionID信息。Session对象的正常使用要依赖于Cookie。如果考虑到客户端浏览器可能出于安全的考虑禁用了Cookie，应该使用URL重写的方式使Session在客户端禁用Cookie的情况下继续生效。

### 说明IOC和AOP在spring框架中的应用（重点）

Spring 核心容器（Core）提供Spring框架的基本功能。核心容器的主要组件是BeanFactory，它是工厂模式的实现。BeanFactory使用控制反转（Ioc）模式将应用程序的配置和依赖性规范与实际的应用代码程序分开。Spring的声明式事务基于AOP实现，却并不需要程序开发者成为AOP专家，亦可轻易使用Spring的声明式事务管理。

## 简述web部署Tomacat的过程

Web部署Tomcat的过程大致如下：

* [下载并解压Tomcat压缩包，例如apache-tomcat-8.5.32.tar.gz1](https://blog.csdn.net/u011578734/article/details/112851672)
* 确保已经安装并配置好JDK环境，设置好JAVA\_HOME和PATH变量
* 进入Tomcat的bin目录，运行startup.sh或startup.bat启动Tomcat服务
* 在浏览器中输入<http://localhost:8080/>访问Tomcat的默认页面，如果看到Tomcat的欢迎页面，说明部署成功
* [可以在Tomcat的webapps目录下放置自己的Web应用，或者在conf目录下修改server.xml文件配置虚拟主机或其他参数](https://blog.csdn.net/weixin_46902396/article/details/116454366)

### 编程题

### 建立一个cookie保存用户信息

String name = "username";   
String mail = "tom@126.com";   
String brithday = "19901010";   
Cookie myCook = newCookie("myCookInfo",name+"#"+birthday+"#"+mail);   
myCook.setMaxAge(60\*60\*24\*365);//设置cookie有效期   
response.addCookie(myCook)

## WEB综合

### 名词解释

**web**：web是分布在全世界的、基于HTTP通信协议、存储在web服务器中的所有互相连接的超文本集

### 简答

### Web的特点

（1）web是一种分布式超媒体系统

（2）web是多媒体化的和易于导航的

（3）web与平台无关

（4）web是动态的

（5）web是交互的

### TCP/IP协议的层次结构和作用

**TCP/IP** ：TCP/IP（传输控制协议/网际协议）是Internet最底层的核心通信协议，它定义了电子设备如何接入Internet

层次结构（从上到下）

（1）应用层

（2）传输层

（3）网络层

（4）网络接口层

## HTTP及事务处理的流程

**HTTP**：超文本传输协议，是专门为web设计的一种网络通信协议，位于TCP/IP的应用层

（1）客户端浏览器与web服务器建立连接

（2）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

（3）web服务器响应客户端浏览器的请求

（4）客户端浏览器与web服务器断开连接

### 请描述一个常用的web应用，分析其技术难点，盈利模式和可以改进的地方（往年题目）

一个常用的web应用是在线购物平台，它可以让用户通过网站或手机应用浏览、搜索、比较和购买各种商品和服务。它的技术难点有：

* 如何保证网站的安全性、稳定性和可扩展性，应对大量的用户访问和交易请求
* 如何提供高效的搜索引擎，根据用户的输入和偏好，返回相关的商品和推荐
* 如何优化用户体验，提高转化率和留存率，增加用户的信任和满意度
* 如何处理复杂的物流、支付、退货、售后等业务流程，与第三方合作伙伴进行数据交换和协调
* 如何利用大数据分析，挖掘用户的行为和需求，提供个性化的服务和营销策略

它的盈利模式有：

* 通过向商家收取佣金或服务费，获取交易的一定比例
* 通过向用户提供付费会员或增值服务，获取订阅或使用费
* 通过向广告商出售广告位或流量，获取广告收入
* 通过向金融机构提供信用评估或贷款服务，获取利息或分成

它可以改进的地方有：

* 提高网站的性能和可靠性，减少故障和延迟，提升用户的满意度
* 增加网站的功能和互动性，提供更多的选择和便利，增加用户的黏性
* 创新网站的设计和内容，提供更多的价值和体验，增加用户的忠诚度
* 扩大网站的覆盖和影响力，进入更多的市场和领域，增加用户的规模

（以上答案为new bing给出）

### 请描述json的定义及其数据格式（往年题目）

\*\*JSON的定义及其数据格式如下：

* JSON是JavaScript Object Notation的缩写，是一种轻量级的数据交换格式¹²
* JSON是基于JavaScript语言的一个子集，但是也独立于语言和平台，可以被多种编程语言支持¹²
* JSON的数据格式是由两种结构组成的：对象（object）和数组（array）²
* 对象（object）是一个无序的“名称/值”对的集合，用{ }括起来，每个“名称”后跟一个:冒号，每个“名称/值”对之间用,逗号分隔²
* 数组（array）是值（value）的有序集合，用[ ]括起来，每个值之间用,逗号分隔²
* 值（value）可以是字符串（string）、数值（number）、布尔值（true或false）、空值（null）、对象（object）或数组（array）²
* 字符串（string）是由双引号包围的任意数量Unicode字符的集合，使用反斜线转义²
* 数值（number）与C或Java的数值非常相似，除去未使用的八进制和十六进制格式²

JSON的数据格式示例：

{  
 "name": "Alice",  
 "age": 25,  
 "gender": "female",  
 "hobbies": ["reading", "writing", "singing"],  
 "address": {  
 "city": "Beijing",  
 "country": "China"  
 }  
}

### 结合所学知识，请描述目前常用改的基于Java 的web程序后端开发框架，并简要描述各部分（往年题目）

目前常用的基于Java的web程序后端开发框架有以下几种：

* Spring Boot：是一个基于Spring框架的快速开发平台，提供了自动配置、依赖管理、嵌入式容器、监控等功能，简化了Java web开发的复杂性
* Spring MVC：是一个基于MVC模式的web层框架，提供了灵活的URL映射、数据绑定、视图解析、异常处理等功能，支持RESTful风格的web服务
* MyBatis：是一个持久层框架，提供了对象与SQL语句的映射，支持动态SQL、存储过程、分页等功能，可以与Spring框架集成
* Hibernate：是一个对象关系映射（ORM）框架，提供了对象与数据库表的映射，支持事务管理、缓存、查询语言等功能，可以与Spring框架集成
* Struts2：是一个基于MVC模式的web层框架，提供了拦截器、标签库、验证器等功能，支持RESTful风格的web服务
* 以上框架各有优劣，可以根据具体的项目需求和场景进行选择和组合。一般来说，Spring Boot + Spring MVC + MyBatis 是比较流行和实用的组合

、

### 谈谈web发展的前景（重点）

Web 技术是在互联网技术的发展产物，尤其Internet 技术的广泛使用，改变了人们的工作、学习、生活等方式。Web 技术从初期的静态Web，发展到现在动态的Web，尤其现在互联网+时代，电子商务行业的兴起与发展，促使Web技术成为人们生活中的一部分，Web技术成为人们生活的焦点，其关注率很高，必须提高Web技术性能，满足用户需求  
从更长远的方向去看，Web Page会往Web App的方向去发展，浏览器内核也会往操作系统的方向去发展。在将来，很可能浏览器内核就变成了操作系统的外壳，页面通过浏览器内核能够使用越来越多的系统底层基础能力。那个时候的Web可能就是现在的Native了，只不过它还继续保持着Web跨平台容易传播等天然的优点。  
这样演进的优点是什么？Web页面能具备Native应用的性能体验，而且不需要下载安装包，能跑在所有的平台，能快速分享传播，多么美好的事情！开发者只需遵循W3C标准去开发页面，就能跑在所有平台所有客户端，不需要关心页面是跑在客户端上，还是跑在操作系统上，这就是标准技术的魅力！  
无论 Web 技术如何发展，它必将会继续保持着自由平等开放共享的互联网精神，这些才是Web

（开放回答）