**参数式查询**：PreparedStatement **设置编码，以防乱码：**setChraracterEncoding() **通常命名为：**applicationContext.xml **获取映射器的方法名：**getMapper() **文件上传：**multipart/form-data **根路径EL表达式：**${pageContext.request.contextPath} **获取请求转发对象：**getRequestDispatcher() **配置 Servlet 的访问路径及名称：**<url-pattern> **与XX等效的 EL 表达式：**${sessionScopy. username} **XXX必须使用属性**items**和 var。**

**简述 JSP 与 Servlet 的关系？**在 Web 服务器端，每个 JSP 页面会被转译为一个 Servlet。

**如何在 web.xml 里配置 Servlet？**使用标签<servlet>和<servlet-mapping>分别配置 Servlet 名和 Servlet 映射。

**简述使用 Servlet 过滤器的好处？**用于身份认证；统一网站的请求编码和响应编码。

**简述使用 MyBatis 框架的主要步骤？**在 MyBatis 框架配置文件里，通过标签<dataSource>配置数据源的相关信息，如数据库类型、服务器 url、数据库名、用户名及密码等；通过标签<mapper>及属性 resource 或 class 配置 xml 映射文件或映射接口文件。

**MyBatis 的 xml 映射文件中，不同的 xml 映射文件，SQL id 是否可以重复？**在不同的 xml 映射文件里，SQL id 可以重复，它通过在引用 SQL id 名称前加上 xml 文件的命名空间名称来区别。

**如何使用 PageHelper 组件？**在 MyBatis 框架配置文件里，使用标签<pluging>及属性 interceptor 引入分页组件 PageHelper；使用 PageHelper 的静态方法 startPage(pageIndex,pageSize)分页；做 MyBatis 查询后创建的 PageInfo<Entity> 对象就包含了分页的相关信息。

**简述 Spring MVC 的工作原理？**在 web.xml 里配置前端控制器 DispatcherServlet，拦截用户的所有请求，并查找相应的后端控制器 Controller；Controller 完成数据处理后，返回一个 ModelAndView 对象给前端控制器；前端控制器调用视图解析器并根据模型数据进行视图渲染后返回给客户端用户。

**简述使用 Spring MVC 框架的一般步骤？**在 pom.xml 文件里定义 Spring MVC 框架依赖；在 web.xml 里配置 DispatcherServlet，拦截用户的所有请求；在 web.xml 里加载 Spring MVC 框架配置文件；在 Spring MVC 框架配置文件里，依次配置注解驱动、组件扫描基础包、资源视图解析器和静态资源映射等。

**比较 MVC 模式开发与 MVC 框架开发的异同点？**Spring MVC 是对 Servlet 的再封装，通过框架约定来简化 Web 项目开发。MVC 框架开发需要引入 MVC 框架依赖，能更加方便地实现 MVC 模式开发。使用 MVC 框架开发时，一个控制器可以包含多个相关功能的方法。

**简述 Spring MVC 配置文件的主要内容？**配置注解驱动、组件扫描基础包、资源视图解析器和静态资源映射等。

**简述在 Spring MVC 项目里使用 Ajax 的一般步骤？**除了引入 Spring MVC 框架依赖外，还需要引入 Ajax 依赖包，以实现数据格式的转换（转为JSON 数据格式）；控制器方法使用注解@ResponseBody；视图页面的 JS 脚本可以使用 jQuery 的$.ajax()方法，通过 url 参数请求控制器的 Ajax 方法。

**简述使用分页组件 PageHelper 的一般步骤?** 在 pom.xml 里引入 PageHelper 插件依赖并在 MyBatis 配置文件里通过标签<plugin>及属性interceptor 引入分页组件；使用 PageHelper 的静态方法 startPage(pageIndex,pageSize)分页；做 MyBatis 查询后创建的 PageInfo<Entity>对象就包含了分页的相关信息。未使用 Ajax 请求的 Web 项目，一般还要设计一个分页导航实体类（参见第 7 章综合项目 MemMana5 的后台会员信息页面）。

**简述 DI 与 IoC 的关系?** IoC 表示控制反转，即一个对象 a 所需的协作对象 b 不是由 a 使用 new 运算符创建，而是由IoC 容器创建。DI 表示依赖注入，它是 IoC 的一种实现方式。

**简述 Spring 框架的两种使用方式?**  使用 Spring 框架，有注解方式和非注解方式两种。非注解方式是指：（1）使用类 ClassPathXmlApplicationContext 加载 Spring 配置文件；（2）通过调用 Spring核心接口的方法 getBean()。 注解方式是指：（1）使用@RunWith(SpringJUnit4ClassRunner.class)注解加载 Spring 配置文件；（2）通过@Autowired 等注解获取 Spring 容器管理的对象。

**比较 JUnit 与 Spring 单元测试的异同点?** 单元测试 JUnit 用于测试普通的 Java 类的方法，只需要引入单元测试依赖。Spring 单元测试除了需要引入 Java 单元测试依赖外，还需要引入 Spring 单元测试依赖；通过注解@ContextConfiguration 加载 Spring 配置文件；通过注解@RunWith 引用 Spring 单元测试机。

**简述 AOP 的作用?** AOP 是指面向切面的编程，它将业务逻辑从应用服务中分离出来，实现高内聚开发，应用对象只关注主要业务逻辑，不再负责其它系统问题（如日志、事务等）。

**简述使用 Spring 整合的 Java 项目与 Web 项目在配置文件上的区别?** Spring 整合的 Java 项目不包含 Spring MVC 框架依赖，而 Web 项目不然。

**简述 SSM 项目开发的一般步骤?** （1）创建 Maven 项目，添加 Web 模块和 Web 项目所需要的各种依赖。 （2）在 web.xml 里配置 Spring MVC 字符编码过滤器。（3）在程序文件夹里创建一个用于存放 DAO 层、服务层和控制层的包，作为组件扫描基础包。（4）在组件扫描基础包里，分别创建用于存放 DAO 层实体类和映射接口的子包，并分别编写实体类和映射接口文件。（5）在组件扫描基础包里，分别创建用于存放服务层接口及其实现类的子包，并分别编写接口和实现类文件。（6）在组件扫描基础包里，创建用于存放控制器的子包，编写控制器。（7）在资源文件夹里分别编写 Spring 整合配置文件和 Spring MVC 配置文件，并在 Spring MVC 配置文件里导入 Spring 整合配置文件。（8）在 web.xml 里配置主页（主控制器）。

**比较 SSM 整合项目与非 SSM 整合项目开发的区别?** SSM 整合项目需要编写 Spring 对 MyBatis 的整合配置文件，不再单独编写 Spring 和 MyBatis框架的配置文件；非 SSM 整合项目开发没有使用 Spring 框架，分别编写 Spring MVC 和 MyBatis 框架的配置文件。

**IDEA 主要提供了 Project 和 Packages 两种视图，试比较它们的特点?** Project 视图是默认使用的视图，可以查看到文件 pom.xml，具有多层结构；Packages 是包视图，不可以查看 pom.xml 文件，很清晰地显示了程序文件夹、静态资源、视图模板和项目配置文件。

**简述模板引擎 Thymeleaf 的优点（相对 JSP 模板引擎而言）?** 使用模板引擎 Thymeleaf 的视图文件以.html 作为扩展名，对 HTML 标签进行扩展。

**试比较 SSM 项目与 Spring Boot Web 项目的开发效率?** SSM 项目需要开发者手工添加较多的依赖，需要注意版本的匹配，效率相对较低；Spring Boot Web 项目使用父工程依赖，默认了许多配置，实现了统一的版本控制，项目创建只需选择特定的功能，不必手工添加依赖，因而开发效率留言较高。