1.JSON生成，JSON解析

2.Servlet和JSP其实是完全统一的，二者在底层的运行原理是一样的，实际上，JSP必须被Web服务器编译成Servlet，真正在Web服务器内运行的是Servlet。从这个意义上来看，我们可以把JSP当成一个“草稿”文件，Web服务器内运行的是Servlet。真正提供HTTP服务的是Servlet。因此广义的Servlet包括了JSP和Servlet.

3..将已存在的项目导入到Eclipse中有两种方法，第一种是用Import导入，另一种是将src中的内容和WebContent中的内容分别都复制到已建好的空项目的目录中。

4.Servelet类中包含三个方法：init():初始化JSP/Servlet的方法；destroy():销毁JSP/Servlet之前的方法；service():对用户请求生成响应的方法。

5.JSP文件必须在JSP服务器中运行；JSP文件必须生成Servlet才能运行；每个JSP页面的第一个访问者速度很慢，因为必须等待JSP编译成Servlet；JSP页面的访问者无须安装任何客户端，甚至不需要可以运行Java的运行环境，因为JSP页面输送到客户端的是标准的HTML页面。

6.JSP的注释的格式为：<%--JSP的注释部分-->;HTML的注释的格式为：<!—HTML的注释部分-->.HTML的注释可以通过源代码查看到，但JSP的注释是无法通过源代码查看到的。这表明JSP注释不会被发送到客户端。

7.JSP生明的语法格式：<%!声明部分%>,JSP声明将会转换成对应Servlet的成员变量或成员方法。JSP中不能使用abstract修饰声明部分的方法，因为抽象方法将导致JSP对应Servlet变成抽象类，从而导致无法实例化。

8.JSP表达式的格式为：<%=表达式%>，输出表达式将转换成Servlet中的输出语句，输出表达式语法后不能有分号。

9.在JSP小脚本部分也可以声明变量，但在JSP脚本中声明的变量是局部变量，但不能用private、public等修饰符修饰，也不可使用static修饰。由于JSP脚本将转换成\_jspService方法里的可执行代码，而Java语法不允许在方法里定义方法，所以JSP脚本里不能定义方法。

10.JAR文件放在Tomcat的lib路径下所有的Web应用都可使用，JAR文件放在Web应用的WEB-INF/lib路径下，只有该Web应用可以使用。

11.jsp的编译指令是通知JSP引擎的消息，它不直接生成输出，常见的编译指令有3个：page：该指令是针对当前页面的指令；include：用于指定包含另一个页面；taglib：用于定义和访问自定义标签。Include指令包含页面在编译时将完全包含了被包含页面的代码，需要指出的是，静态包含还会将被包含页面的编译指令也包含进来。

12.动作指令与编译指令不同，编译指令是通知Servlet引擎的处理消息，而动作指令只是运行时的动作。编译指令在将JSP编译成Servlet时起作用；而处理指令通常课替换成JSP脚本，它只是JSP脚本的标准化写法。JSP动作执行主要有如下7个：jsp:forword:执行页面跳转，将请求的处理转发到下一个页面。Jsp:param:用于传递参数，必须与其他支持参数的标签一起使用。jsp:include:用于动态引入一个JSP页面。jsp:plugin:用于下载JavaBean或Applet到客户端指定。jsp:useBean:创建一个JavaBean实例。jsp:setProperty:设置JavaBean实例的属性值。jsp：getProperty:输出JavaBean实例的属性值。

13.执行forward指令转发请求时，客户端的请求参数不会丢失。从表面上看，<jsp:forward…/>指令给人一种感觉：它是将用户请求“转发”到了另一个新的页面，但实际上，<jsp:forward…/>并没有重新向新页面发送请求，它只是完全采用了新页面来对用户生成响应——请求依然是一次请求，所以请求参数、请求属性都不会丢失。

14.归纳起来，静态导入与动态导入有如下三点区别：

静态导入是将被导入页面的代码完全融入，两个页面融合成一个整体Servlet；而动态导入则在Servlet中使用include方法来引入被导入页面的内容；静态导入是被导入页面的编译指令会起作用；而动态导入时被导入页面的编译指令则失去作用，只是插入被导入页面的body内容；动态包含还可以增加额外的参数。

15.由于只有JSP脚本、输出表达式才会对应于\_jspService()方法里的代码，所以这两个部分的代码无需处理checked异常。但JSP声明部分依然需要处理checked异常，JSP的异常处理机制对JSP声明不起作用。

16.