1. [Practice1 Servlet](http://localhost:8080/JavaWebPractices/UrlServletPractice1)  … 1
2. [Practice2 Request](http://localhost:8080/JavaWebPractices/UrlServletPractice2)  2
3. [Practice3 Response](http://localhost:8080/JavaWebPractices/UrlServletPractice3)  2
4. [Practice4 Session](http://localhost:8080/JavaWebPractices/UrlServletPractice4)  2
5. [Practice5 Context](http://localhost:8080/JavaWebPractices/UrlServletPractice5)  3
6. [Practice6 RequestDispatcher](http://localhost:8080/JavaWebPractices/UrlServletPractice6)  3
7. [Practice7 Request Attributes](http://localhost:8080/JavaWebPractices/UrlServletPractice7)  3
8. [Practice7 Session Attributes](http://localhost:8080/JavaWebPractices/UrlServletPractice72)  3
9. [Practice7 ContextAttributes](http://localhost:8080/JavaWebPractices/UrlServletPractice73)  3
10. [Practice8 Request Parameters](http://localhost:8080/JavaWebPractices/UrlServletPractice8)  3
11. [Practice8 Init Parameters](http://localhost:8080/JavaWebPractices/UrlServletPractice82)  3
12. [Practice9 Filters](http://localhost:8080/JavaWebPractices/UrlServletPractice2)  4
13. [Practice10 ServletContextListener](http://localhost:8080/JavaWebPractices/UrlServletPractice5)  4
14. [Practice11 ServletContextAttributeListener](http://localhost:8080/JavaWebPractices/UrlServletPractice5)  4
15. [Practice12 HttpSessionListener](http://localhost:8080/JavaWebPractices/UrlServletPractice12)  5
16. [Practice13 HttpSessionAttributeListener](http://localhost:8080/JavaWebPractices/UrlServletPractice13)  5
17. [Practice14 HttpSessionActivationListener](http://localhost:8080/JavaWebPractices/UrlServletPractice14)  5
18. [Practice15 HttpSessionBindingListener](http://localhost:8080/JavaWebPractices/UrlServletPractice15)  5
19. [Practice16 ServletRequestListener](http://localhost:8080/JavaWebPractices/UrlServletPractice16)  6
20. [Practice17 ServletRequestAttributeListener](http://localhost:8080/JavaWebPractices/UrlServletPractice17)  6
21. [Practice18 ServletInputStream](http://localhost:8080/JavaWebPractices/UrlServletPractice18)  6
22. [Practice19 ServletOutputStream](http://localhost:8080/JavaWebPractices/UrlServletPractice19)  7
23. [Practice20 ServletException](http://localhost:8080/JavaWebPractices/UrlServletPractice20)  7
24. [Practice21 UnavailableException](http://localhost:8080/JavaWebPractices/UrlServletPractice21)  7
25. [Practice22 Java Server Page JSP](http://localhost:8080/JavaWebPractices/jspPractice22)  7
26. [Practice23 JSP path](http://localhost:8080/JavaWebPractices/jspPractice23)  8
27. [Practice24 JSP Directive Page](http://localhost:8080/JavaWebPractices/jspPractice24)  8
28. [Practice25-1 JSP Directive Include](http://localhost:8080/JavaWebPractices/jspPractice251)  9
29. [Practice25-2 JSP Directive TagLib](http://localhost:8080/JavaWebPractices/jspPractice252)  9
30. [Practice26 JSP Implicit Objects](http://localhost:8080/JavaWebPractices/jspPractice21)  9
31. [Practice26-1 JSP Implicit Object request](http://localhost:8080/JavaWebPractices/jspPractice261)  9
32. [Practice26-2 JSP Implicit Object response](http://localhost:8080/JavaWebPractices/jspPractice262)  9
33. [Practice26-3 JSP Implicit Object application](http://localhost:8080/JavaWebPractices/jspPractice263)  9
34. [Practice26-4 JSP Implicit Object session](http://localhost:8080/JavaWebPractices/jspPractice264)  9
35. [Practice26-5 JSP Implicit Object out](http://localhost:8080/JavaWebPractices/jspPractice265)  9
36. [Practice26-6 JSP Implicit Object page](http://localhost:8080/JavaWebPractices/jspPractice266)  9
37. [Practice26-7 JSP Implicit Object pageContext](http://localhost:8080/JavaWebPractices/jspPractice267)  9
38. [Practice26-8 JSP Implicit Object exception](http://localhost:8080/JavaWebPractices/jspPractice268)  9
39. [Practice27 JSP Standard Actions](http://localhost:8080/JavaWebPractices/jspPractice27)  9
40. [Practice28 JSP Custom Actions](http://localhost:8080/JavaWebPractices/jspPractice28)  10
41. [Practice28-1 JSP Custom Action BodyTagSupport](http://localhost:8080/JavaWebPractices/jspPractice281)  10
42. [Practice28-2 JSP Custom Action SimpleTagSupport](http://localhost:8080/JavaWebPractices/jspPractice282)  10

[**Practice1 Servlet**](http://localhost:8080/JavaWebPractices/UrlServletPractice1)

Create a simple web application with a valid J2EE structure.

Definition del DTD web.xml and the valid structure of a J2EE application

Steps

1.- Create a Dynamic web Project.

2.- Create a class that extends from HttpServlet.

3.- Set the configuration of the DTD web.xml for this servlet.

4.- Run and confirm the life cycle of a servlet.

[**Practice2 Request**](http://localhost:8080/JavaWebPractices/UrlServletPractice2)

Create a servlet that confirms the existence of the servlet request and all the objects on it.

Steps

1.-Create a servlet.

2.-Print out the servlet request.

[**Practice3 Response**](http://localhost:8080/JavaWebPractices/UrlServletPractice3)

Create a servlet that confirms the existence of the servlet response and all the objects on it.

Steps

1.-Create a servlet.

2.-Print out the servlet response.

[**Practice4 Session**](http://localhost:8080/JavaWebPractices/UrlServletPractice4)

Create a servlet that confirms the existence of the session object and all the objects on it.

Steps

1.-Create a servlet.

2.-Print out the session object.

[**Practice5 Context**](http://localhost:8080/JavaWebPractices/UrlServletPractice5)

Create a servlet that confirms the existence of the servlet context and all the objects on it.

Steps

1.-Create a servlet.

2.-Print out the servlet context.

[**Practice6 RequestDispatcher**](http://localhost:8080/JavaWebPractices/UrlServletPractice6)

Create a servlet that confirms the existence of the request dispatcher and all the objects on it.

Steps

1.-Create a servlet.

2.-Print out the request dispatcher.

[**Practice7 Attributes**](http://localhost:8080/JavaWebPractices/UrlServletPractice7)

Create a servlet that confirms the objects that supports attributes.

Steps

1.-Create a servlet.

2.-Print out the context attributes.

3.-Print out the request attributes.

4.-Print out the session attributes.

[**Practice8 Parameters**](http://localhost:8080/JavaWebPractices/UrlServletPractice8)

Create a servlet that confirms the objects that supports attributes.

Steps

1.-Create a servlet.

2.-Print out the request parameters attributes.

3.-Print out the init parameters.

[**Practice9 Filters**](http://localhost:8080/JavaWebPractices/UrlServletPractice2)

Create a Filter that confirms the doFilter method.

Steps

1.-Create a filter.

2.-Executes doFilter method.

3.-Print out the result.

[**Practice10 ServletContextListener**](http://localhost:8080/JavaWebPractices/UrlServletPractice5)

Create a servlet that confirms the SevletContextListener.

Steps

1.-Create a servlet.

2.-Print out the result of the listener methods.

[**Practice11 ServletContextAttributeListener**](http://localhost:8080/JavaWebPractices/UrlServletPractice5)

Create a servlet that confirms the ServletContextAttributeListener.

Steps

1.-Create a servlet.

2.-Add some attributes.

3.-Print out the result of the listener methods.

[**Practice12 HttpSessionListener**](http://localhost:8080/JavaWebPractices/UrlServletPractice12)

Create a servlet that confirms the HttpSessionListener.

Steps

1.-Create a servlet.

2.-Add some attributes.

3.-Print out the result of the listener methods.

[**Practice13 HttpSessionAttributeListener**](http://localhost:8080/JavaWebPractices/UrlServletPractice13)

Create a servlet that confirms the HttpSessionListener.

Steps

1.-Create a servlet.

2.-Add some attributes.

3.-Print out the result of the listener methods.

[**Practice14 HttpSessionActivationListener**](http://localhost:8080/JavaWebPractices/UrlServletPractice14)

Create a servlet that confirms the HttpSessionActivationListener.

Steps

1.-Create a servlet.

2.-Add some attributes.

3.-Print out the result of the listener methods.

[**Practice15 HttpSessionBindingListener**](http://localhost:8080/JavaWebPractices/UrlServletPractice15)

Create a servlet that confirms the HttpSessionBindingListener.

Steps

1.-Create a servlet.

2.-Add some attributes.

3.-Print out the result of the listener methods.

[**Practice16 ServletRequestListener**](http://localhost:8080/JavaWebPractices/UrlServletPractice16)

Create a servlet that confirms the ServletRequestListener.

Steps

1.-Create a servlet.

2.-Add some attributes.

3.-Print out the result of the listener methods.

[**Practice17 ServletRequestAttributeListener**](http://localhost:8080/JavaWebPractices/UrlServletPractice17)

Create a servlet that confirms the ServletRequestAttributeListener.

Steps

1.-Create a servlet.

2.-Add some attributes.

3.-Print out the result of the listener methods.

[**Practice18 ServletInputStream**](http://localhost:8080/JavaWebPractices/UrlServletPractice18)

Create a servlet that confirms the ServletInputStream with a file.

Steps

1.-Create a servlet.

2.-create a getMethod to dispatch a jsp.

3.-the jsp will send an image file to the method post of the servlet.

4.-method doPost will receive a file and save it.

[**Practice19 ServletOutputStream**](http://localhost:8080/JavaWebPractices/UrlServletPractice19)

Create a servlet that confirms the ServletOutputStream with an image.

Steps

1.-Create a servlet.

2.-Read an image file.

3.- Print out the result.

[**Practice20 ServletException**](http://localhost:8080/JavaWebPractices/UrlServletPractice20)

Create a servlet that throws a ServletException.

Steps

1.-Create a servlet.

2.- throw an ServletException.

[**Practice21 UnavailableException**](http://localhost:8080/JavaWebPractices/UrlServletPractice21)

Create a servlet that throws an UnavailableException.

Steps

1.-Create a servlet.

2.- throw an UnavailableException.

[**Practice22 Java Server Page JSP**](http://localhost:8080/JavaWebPractices/jspPractice22)

Create a JSP file .

Steps

1.-Create a JSP.

[**Practice23 JSP path**](http://localhost:8080/JavaWebPractices/jspPractice23)

Create a Servlet that forwards a request and show the request in a JSP file .

Steps

1.-Create a servlet.

2.-Use the request dispatcher to load a JSP file.

3.-Print the request in the JSP file.

[**Practice24 JSP Directive Page**](http://localhost:8080/JavaWebPractices/jspPractice24)

Create a JSP that confirms the usage of directive page.

Steps

1.-Create a JSP file.

2.- Confirms the usage of directive page.

[**Practice25-1 JSP Directive Include**](http://localhost:8080/JavaWebPractices/jspPractice251)

Create a JSP that confirms the usage of directive include.

Steps

1.-Create a JSP file.

2.- Confirms the usage of directive include.

[**Practice25-2 JSP Directive TagLib**](http://localhost:8080/JavaWebPractices/jspPractice252)

Create a JSP that confirms the usage of directive TagLib.

Steps

1.-Create a JSP file.

2.- Confirms the usage of directive TagLib.

[**Practice26 JSP Implicit Objects**](http://localhost:8080/JavaWebPractices/jspPractice21)

Create a JSP that confirms the implicit objects.

Steps

1.-Create a JSP file.

2.- Print out the Implicit object request.

3.- Print out the Implicit object response.

4.- Print out the Implicit object application.

5.- Print out the Implicit object session.

6.- Print out the Implicit object out.

7.- Print out the Implicit object pageContext.

8.- Print out the Implicit object request.

9.- Print out the Implicit object exception.

[**Practice27 JSP Standard Actions**](http://localhost:8080/JavaWebPractices/jspPractice27)

Create a JSP that confirms the standard actions.

Steps

1.-Create a JSP file.

2.- Print out the usage of standard actions.

[**Practice28 JSP Custom Actions**](http://localhost:8080/JavaWebPractices/jspPractice28)

Create a JSP that confirms custom actions by creating a custom tag.

Steps

1.-Create a JSP file.

2.- Create a custom tag using BodyTagSupport.

3.- Create a custom tag using SimpleTagSupport.

4.- Print out the result of the tags.