Life Cycle of Maven

Maven has 3 Life Cycle.Each Life Cycle contains number of Phases

- Clean
- default
- site

Clean:-This Life Cycle contains 3 phases

Pre-clean	Executes processes needed prior to the actual project cleaning
Clean	Remove all the file generated in previous build
Post-clean	Executes processes needed to finalize the project cleaning

Default: This Life Cycle contains 23 phases

1	validate	validate the project is correct and all necessary information is available.
2	initialize	initialize build state, e.g. set properties or create directories.
3	generate-sources	generate any source code for inclusion in compilation.
4	process-sources	process the source code, for example to filter any values.
5	generate-resources	generate resources for inclusion in the package.
6	process-resources	copy and process the resources into the destination directory, ready for packaging.
7	compile	compile the source code of the project.
8	process-classes	post-process the generated files from compilation,for example to do bytecode enhancement on Java classes.
9	generate-test-sources	generate any test source code for inclusion in compile
10	process-test-sources	process the test source code, for example to filter any values.
11	generate-test-resources	create resources for testing.
12	process-test-resources	copy and process the resources into the test destination directory.
13	test-compile	compile the test source code into the test destination directory
14	process-test-classes	post-process the generated files from test compilation, for example to do bytecode enhancement on Java classes. For Maven 2.0.5 and above.
15	test	run tests using a suitable unit testing framework.These tests should not require the code be packaged or deployed.
16	prepare-package	perform any operations necessary to prepare a package before the actual packaging. This often results in an unpacked, processed version of the package. (Maven 2.1 and above)
17	package	take the compiled code and package it in its distributable format, such as a JAR.
18	pre-integration-test	perform actions required before integration tests are executed. This may involve things such as setting up the required environment.
19	integration-test	process and deploy the package if necessary into an environment where integration tests can be run.
20	post-integration-test	perform actions required after integration tests have been executed. This may including cleaning up the environment.
21	verify	run any checks to verify the package is valid and meets quality
22	install	install the package into the local repository, for use as a dependency in other projects locally.
23	deploy	done in an integration or release environment, copies the final package to the remote repository for sharing with other developers and projects.

Site: It contains 4 phases

Pre-site	executes processes needed prior to the actual project site generation
Site	generates the project's site documentation
Post-site	executes processes needed to finalize the site generation, and to prepare for site deployment
Site-deploy	deploys the generated site documentation to the specified web server

Below are some of the List of Important Plugins and goals

Plugin Name	Goals Available	Purpose
maven-compiler-plugin	compile,testCompile	Compile java file and testCase file
exec-maven-plugin	exec,java	Compiles and executes java classes
maven-antrun-plugin	run	Execute ant tasks
tomcat-maven-plugin	Start,stop,deploy, redeploy,undeploy	Deploys the project into tomcat server
jboss-maven-plugin	--do--	Deploys the project into jboss