[Gradle Commands] (CheatSheet)

1. Getting Started

- `gradle init`: Initializes a new Gradle project.
- `gradle wrapper`: Generates Gradle wrapper files.

2. Building Projects

- `gradle build`: Assembles and tests the project.
- `gradle assemble`: Assembles the outputs of the project.
- `gradle check`: Runs all checks.

3. Running Tests

- `gradle test`: Runs the unit tests.
- `gradle test --tests <test class or method>`: Run a single test class or method.
- `gradle jacocoTestReport`: Generates code coverage report.

4. Cleaning Projects

- `gradle clean`: Deletes the build directory.

5. Dependency Management

- `gradle dependencies`: Displays all the dependencies of the project.
- `gradle dependencyInsight --dependency <dependency>`: Provides insight into a specific dependency.

6. Listing Tasks

- `gradle tasks`: Displays all tasks available to run.
- `gradle tasks --all`: Displays all tasks in detail.

7. Running Specific Tasks

- `gradle <task>`: Executes the specified task(s).

By: Waleed Mousa

8. Project Information

- `gradle properties`: Displays the properties of the project.
- `gradle projects`: Displays the sub-projects of the project.

9. Running in Different Environments

- `gradle -Penv=prod build`: Uses a project property to define an environment.

10. Custom Scripts and Tasks

- `gradle -b <buildfile>`: Runs a different build file.

11. Daemon Management

- `gradle --stop`: Stops the Gradle daemon.

12. Configuration

- `gradle --configure-on-demand`: Configure necessary projects only.
- `gradle -g <path>`: Use specified Gradle user home directory.

13. Wrapper Configuration

- `gradle wrapper --gradle-version <version>`: Specifies a version for the Gradle wrapper.

14. Logging and Output

- `gradle -q`: Quiet logging.
- `gradle -i`: Info level logging.
- `gradle -s`: Print stack trace for errors.
- `gradle -d`: Debug level logging.

15. Parallel Execution

- `gradle --parallel`: Build/test projects in parallel.

16. Continuous Build

By: Waleed Mousa

- `gradle --continuous`: Continuously executes the task.

17. Task Execution Options

- `gradle --exclude-task <task>`: Excludes a specific task.
- `gradle --rerun-tasks`: Forces tasks to rerun.

18. Project Evaluation

- `gradle --dry-run`: Shows what tasks would run without actually executing them.

19. File Watch

- `gradle --watch-fs`: Enables file system watching.

20. Updating Wrapper

- `gradle wrapper --gradle-distribution-url <url>`: Specifies a new distribution URL for the Gradle wrapper.

21. Performance Monitoring

- `gradle --scan`: Creates a build scan with detailed performance data.

22. Build Cache

- `gradle --build-cache`: Enables the build cache.

23. Offline Mode

- `gradle --offline`: Executes the build without accessing the network.

24. Refreshing Dependencies

- `gradle --refresh-dependencies`: Refresh the state of dependencies.

25. Setting the Log Level

- `gradle -w`: Set log level to warn.
- `gradle -e`: Set log level to error.

By: Waleed Mousa

26. Gradle Profiling

- `gradle --profile`: Profiles build execution time.

27. System Properties

- `gradle -D<propertyName>=<value>`: Sets a system property.

28. Java Home and JVM Options

- `gradle -Dorg.gradle.java.home=<path_to_jdk>`: Sets the JDK path.
- `gradle -Dorg.gradle.jvmargs=-Xmx2048m`: Sets JVM arguments.

29. Environment Options

- `gradle -Dorg.gradle.daemon=false`: Disable the Gradle daemon.

30. Init Scripts

- `gradle --init-script <path>`: Specifies an initialization script.

31. Project Cache

- `gradle clean build --project-cache-dir <path>`: Specifies project cache directory.

32. Handling Large Projects

- `gradle --max-workers`: Specifies the maximum number of workers.

33. Managing Daemon

- `gradle --status`: Shows the status of Gradle daemons.

34. Gradle Version

- `gradle --version`: Print version info.

35. Building Specific Project in Multi-Project

By: Waleed Mousa

- `gradle :projectA:build`: Build a specific project in a multi-project build.

36. Excluding Tasks

- `gradle build -x test`: Execute the build but skip the test task.

37. Composite Builds

- `gradle --include-build <path>`: Includes another build.

38. File System Watching

- `gradle --watch-fs`: Turns on file system watching.

39. Incremental Builds

- `gradle assemble --no-rebuild`: Do not rebuild project dependencies.

40. Interactive Mode

- `gradle --console=rich`: Use rich console output.

41. Custom Build Scans

- `gradle build --scan --scan-tag <tag>`: Adds custom tags to build scans.

42. Task Avoidance

- `gradle assemble --exclude-task <task>`: Avoid certain tasks.

43. Resource Control

- `gradle --no-daemon`: Do not use the Gradle daemon for this build.

44. Build Comparisons

- `gradle build compareGradleBuilds`: Compare this build with another.

45. Generating Project Reports

By: Waleed Mousa

- `gradle projectReport`: Generates a report about your project.

46. Managing Daemon Memory

- `gradle --daemon-memory <size>`: Specifies memory for the Gradle daemon.

47. Forcing Rebuild

- `gradle clean build --rerun-tasks`: Force all tasks to rerun.

48. Custom Build Directory

- `gradle build -p <custom-dir>`: Specify the project directory.

49. Dependency Reports

- `gradle dependencyReport`: Generates a report of the project dependencies.

50. Custom Task Execution

- `gradle myCustomTask`: Execute a custom task defined in the build script.

51. Changing Build Behavior

- `gradle -Pflag=true`: Pass custom flags to change behavior.

52. Listing Dependency Updates

- `gradle dependencyUpdates`: Lists the dependencies that have updates.

53. Running Specific Test Classes

- `gradle test --tests *MyTestClass`: Run a single test class.
- `gradle test --tests *MyTestClass.myTestMethod`: Run a single test method.

54. Multi-Project Builds

- `gradle :subproject:task`: Run a task in a specific subproject.

55. Code Quality Checks

By: Waleed Mousa

- `gradle checkstyleMain`: Run Checkstyle for main source sets.
- `gradle checkstyleTest`: Run Checkstyle for test source sets.
- `gradle pmdMain`: Run PMD analysis for main source sets.

56. Handling Variants and Flavors

- `gradle assembleDebug`: Assemble the Debug variant (common in Android projects).
- `gradle assembleRelease`: Assemble the Release variant.

57. Publishing Artifacts

- `gradle publish`: Publishes all publications produced by this project.

58. Listing All Project Properties

- `gradle properties`: Lists the properties of a project.

59. Running a Daemon in the Background

- `gradle --background`: Starts a Gradle Daemon in the background.

60. Creating Documentation

- `gradle javadoc`: Generates Javadoc API documentation.

61. Gradle Extensions and Customizations

- `gradle tasks --all`: Shows all tasks, including those from plugins and custom tasks.

62. Customizing Console Output

- `gradle --console=plain`: Use plain console output.
- `gradle --console=auto`: Detect console type (auto, plain, rich).

63. Handling Signing

- `gradle signArchives`: Sign published archives.

By: Waleed Mousa

64. Running in Quiet Mode

- `gradle --quiet`: Log errors only.

65. Configuring Report Formats

- `gradle test -DjunitPlatform.reportsDir=<directory>`: Configure report directory.

66. Managing Gradle Daemon

- `gradle --status`: Show status of Gradle Daemons.
- `gradle --foreground`: Start Gradle in the foreground.

67. Inspecting Project Dependencies

- `gradle dependencyInsight --dependency <group:name>`: Provides insight into a specific dependency.

68. Using Different Gradle Versions

- `gradlew wrapper --gradle-version=6.0`: Change the Gradle wrapper to use a specific version.

69. Enabling Build Caching

- `gradle build --build-cache`: Enables build cache for the build.

70. Optimizing with Configuration Cache

- `gradle --configuration-cache`: Enables the configuration cache.

71. Configuring Test Logging

- `gradle test --info`: Show detailed test information.

72. Handling Project Archives

- `gradle assembleDist`: Assembles the main distribution.
- `gradle distZip`: Bundles the project as a distribution zip.

By: Waleed Mousa

73. Debugging Gradle Scripts

- `gradle --debug`: Enable debug-level logging.

74. Generating Source Sets

- `gradle generateSrc`: Generate source sets (custom task).

75. Applying Specific Gradle Plugins

- `gradle apply plugin: 'java'`: Apply a Java plugin.

76. Refreshing Dependencies

- `gradle --refresh-dependencies`: Force refresh of dependencies.

77. Handling Gradle Build Scans

- `gradle --scan`: Create a detailed build scan at scans.gradle.com.

78. Viewing Cached Outputs

- `gradle buildCache:view`: View what's currently stored in the build cache.

79. Customizing Daemon Usage

- `gradle --no-daemon`: Do not use the Gradle daemon to execute.

80. Incremental Build Features

- `gradle assemble --incremental`: Assemble incrementally if possible.

81. Setting Maximum Heap Size

- `gradle -Dorg.gradle.jvmargs=-Xmx2048m`: Set the max heap size for the JVM.

82. Filtering Tasks

- `gradle tasks --group <group name>`: Shows tasks from a specific group.

By: Waleed Mousa

83. Specifying Build File

- `gradle -b build.gradle.kts`: Specify a particular build file.

84. Defining System Properties

- `gradle -Dprop=value`: Define a system property.

85. Forcing Console Output

- `gradle --console=force`: Force console output.

86. Handling Input Files

- `gradle processResources`: Processes resources.

87. Optimizing Performance

- `gradle --parallel`: Execute tasks in parallel.

88. Customizing Output Directories

- `gradle -PbuildDir=some/other/dir`: Change the build directory.

89. Managing Dependencies and Publications

- `gradle components`: Displays the components produced by the project.
- `gradle model`: Displays the configuration model of the project.

90. Gradle Profiling for Performance

- `gradle --profile`: Profiles build performance.

91. Executing Tasks in Specific Order

- `gradle clean build`: Executes 'clean' then 'build'.

92. Generating Build Reports

- `gradle htmlDependencyReport`: Generates an HTML dependency report.

By: Waleed Mousa

93. Listing Gradle Daemon Details

- `gradle --status`: List running and recently stopped Gradle daemons.

94. Skipping Task Execution

- `gradle build -x test`: Build without executing the test task.

95. Specifying Project Properties

- `gradle -PmyProperty=value`: Set a project property.

96. Running with Info Level Logging

- `gradle --info`: Run with info level logging.

97. Configuring Default Tasks

- `gradle defaultTasks 'clean', 'run'`: Specify default tasks.

98. Custom Gradle Properties

- `gradle -I <init script>`: Specify an initialization script.

99. Handling Project Evaluations

- `gradle -m`: Perform a dry run (show what would happen).

100. Gradle Extensions

- `gradle extensions`: List the extensions added by plugins.

101. Setting Project Name and Version

- `gradle -PprojectName=MyProject`: Set the project name.
- `gradle -Pversion=1.0`: Set the project version.

By: Waleed Mousa