sbt Scala build tool

CONCEPTS

> Start sbt once and stay in its shell: later commands skip the JVM startup and reuse compiled state.

> Tasks take a scope prefix: Test/compile compiles test sources, subproj/test runs one subproject.

RUN AND BUILD

`sbt`
Start the interactive shell

`sbt` *`<command>`*
Run one command and exit (batch mode)

`compile`
Compile main sources

`run` *`<args>`*
Run the main class in sbt's JVM

`runMain` *`<class>`* *`<args>`*
Run a specific main class

`clean`
Delete target directories

`exit`
Leave the shell (or Ctrl+D)

`help` *`<command>`*
Show help for a command

TESTING

`test`
Run all tests

`testOnly` *`<Suite>`*
Run named suites, * works

`testOnly` *`<Suite>`* **`-- -z "`***`<text>`***`"`**
Run tests matching a name

`testQuick`
Run new, failed or affected tests

WATCH MODE

`~compile`
Recompile on source change

WATCH MODE (cont.)

`~test`
Re-run tests on change

`~testQuick`
Re-run only affected tests

`Enter`
Stop a ~ watch and return to shell

PROJECTS

`projects`
List all subprojects

`project` *`<id>`*
Switch the current project

*`<id>`***`/test`**
Run a task in one subproject

BUILD DEFINITION

`reload`
Reload build files after editing

`show` *`<key>`*
Print a setting or task value

CROSS BUILDING AND REPL

`console`
Scala REPL with project classpath

`+test`
Run for every crossScalaVersions entry

PUBLISH AND OPTIONS

`publishLocal`
Publish to the local Ivy repository

BUILD.SBT BASICS

Minimal build.sbt

```
ThisBuild / scalaVersion := "3.3.4"

lazy val root = (project in file("."))
  .settings(name := "app")
```

Add a dependency

```
libraryDependencies += "org.scalatest" %
% "scalatest" % "3.2.19" % Test
```