Mix Elixir build tool

mix.exs deps, aliases → `deps.get` → **mix.lock** pinned versions → `compile` → **_build/<env>** beams per MIX_ENV → `release` → **Release** self-contained

CONCEPTS

Mix is task-driven: mix <task> [args]. Run mix help to see every task, including those from deps.

MIX_ENV picks the environment: dev (default), test (used by mix test) or prod. Each gets its own _build directory.

Commit mix.lock for applications so every machine builds the same dependency versions.

PROJECTS

mix new *<name>*
Create a new project

mix new *<name>* **--sup**
Project with a supervision tree

mix phx.new *<name>*
Phoenix project (needs the phx_new archive)

COMPILE & RUN

mix compile
Compile the project

COMPILE & RUN (cont.)

mix compile --force
Recompile everything

mix compile --warnings-as-errors
Fail when there are warnings

mix run *<file.exs>* **[args]**
Run a script inside the app

mix run -e "*<expr>***"**
Evaluate an expression

mix run --no-halt
Keep the app running

iex -S mix
Interactive shell with the project

mix clean
Remove compiled files

mix do *<task>* **+** *<task>*
Run several tasks in one call

TEST

mix test
Run the whole suite

mix test *<file>*
Run one test file

mix test *<file>*:*<line>*
Run the test at a line

mix test --failed
Re-run last failures

TEST (cont.)

mix test --stale
Only tests for changed modules

mix test --only *<tag>*
Run tests with a tag

mix test --exclude *<tag>*
Skip tests with a tag

mix test --trace
One test at a time, verbose

mix test --max-failures *<n>*
Stop after n failures

mix test --cover
Report code coverage

DEPENDENCIES

mix deps.get
Fetch and lock dependencies

mix deps.update *<dep>*
Update one dependency

mix deps.update --all
Update every dependency

mix deps
List deps and their status

mix deps.tree
Show the dependency tree

mix deps.clean *<dep>*
Remove a dependency's build

DEPENDENCIES (cont.)

mix deps.unlock *<dep>*
Remove a lock entry

Dependency forms in mix.exs

```
{:plug, "~> 1.16"}
{:ex_doc, "~> 0.34",
only: :dev, runtime: false}
{:lib, github: "<user>/
<repo>", tag: "<tag>"}
{:local, path: "../local"}
```

FORMAT & QUALITY

mix format
Format files per .formatter.exs

mix format --check-formatted
Fail if anything is unformatted

mix format *<file>*
Format specific files

mix hex.outdated
Show deps with newer versions

RELEASES

MIX_ENV=prod mix release
Build a production release

MIX_ENV=prod mix release *<name>*
Build a named release

RELEASES (cont.)

_build/prod/rel/*<name>***/bin/***<name>* **start**
Run the release in the foreground

ENVIRONMENTS & TOOLING

MIX_ENV=*<env>* **mix** *<task>*
Run a task in another env

mix local.hex
Install or update the Hex client

RECIPES

Fresh clone to green tests

```
mix deps.get
mix compile --warnings-as-
errors
mix test
```

CI gate

```
mix format --check-formatted
mix compile --warnings-as-
errors
mix test --cover
```

HELP

mix help
List available tasks

mix help *<task>*
Docs for one task

mix --version
Show Elixir and Mix versions