Level 5–1

The Mix Tool

Running Tasks & Organizing Projects

Benefits of a Well-structured Project

Keeping a well-organized project and adopting a standard for project organization can help in many ways. Here are three major benefits:

- Easier to navigate project files.
- Facilitates collaboration from other developers on the team.
- Facilitates onboarding new members.

* MIXING IT UP *
with
ELIXIR

Using Mix to Create a New Project

Mix is a **build tool** installed with Elixir that provides tasks for creating, compiling, and testing Elixir projects, managing its dependencies, and more.

Name of the project

```
$ mix new budget
```

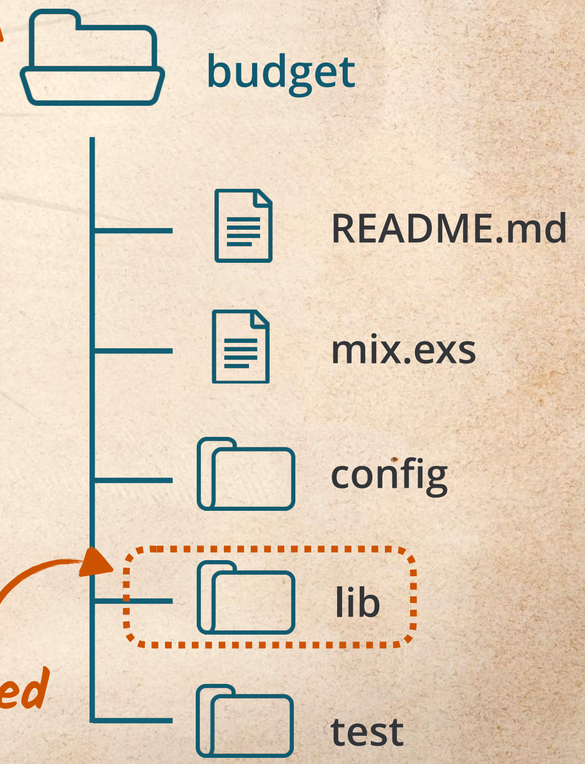
```
* creating README.md
...

Your Mix project was created successfully.
You can use "mix" to compile it, test it, and more:

    cd budget
    mix test

Run "mix help" for more commands.
```

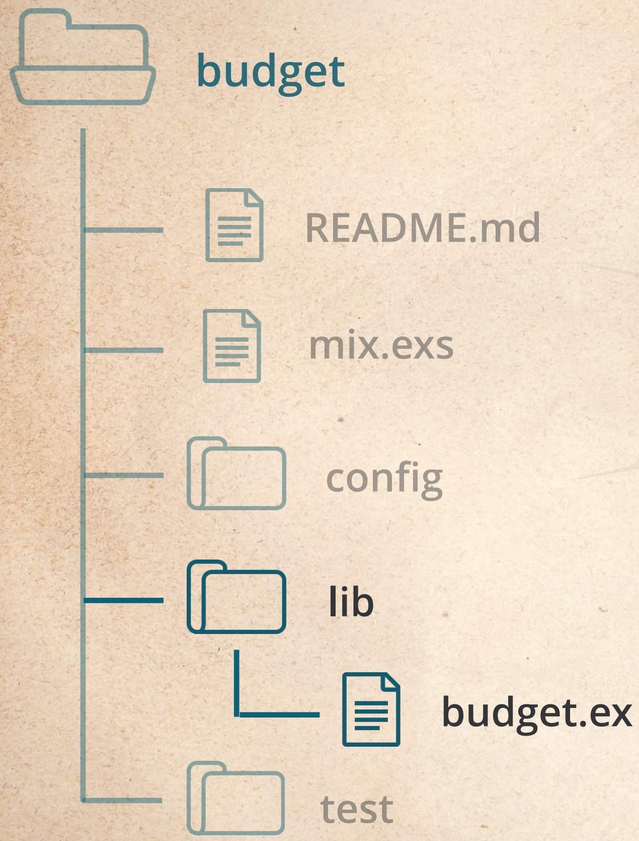
Directories and files created for us!



The only folder we need to access for now

Writing a New Function

We'll define current_balance **as part of the** Budget **module, created for us by the** mix new **command.**



Created by Mix

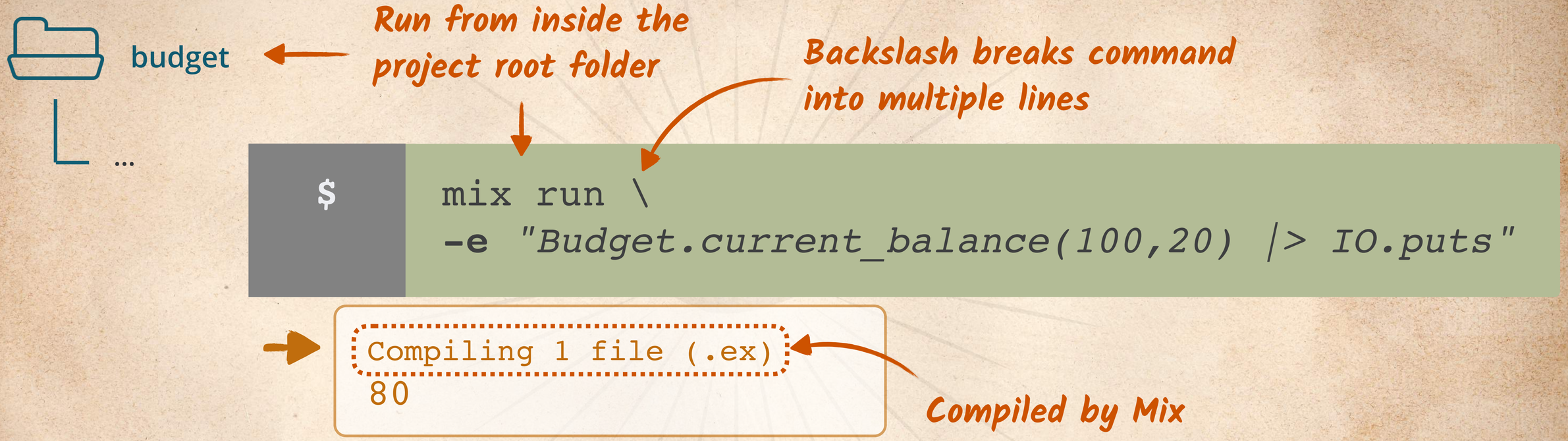
lib/budget.ex

```
defmodule Budget do
  def current_balance(initial, spending) do
    initial - spending
  end
end
```

New function defined by us

Running Programs With mix run

The -e option tells the mix run command to evaluate a given code in the context of the application.



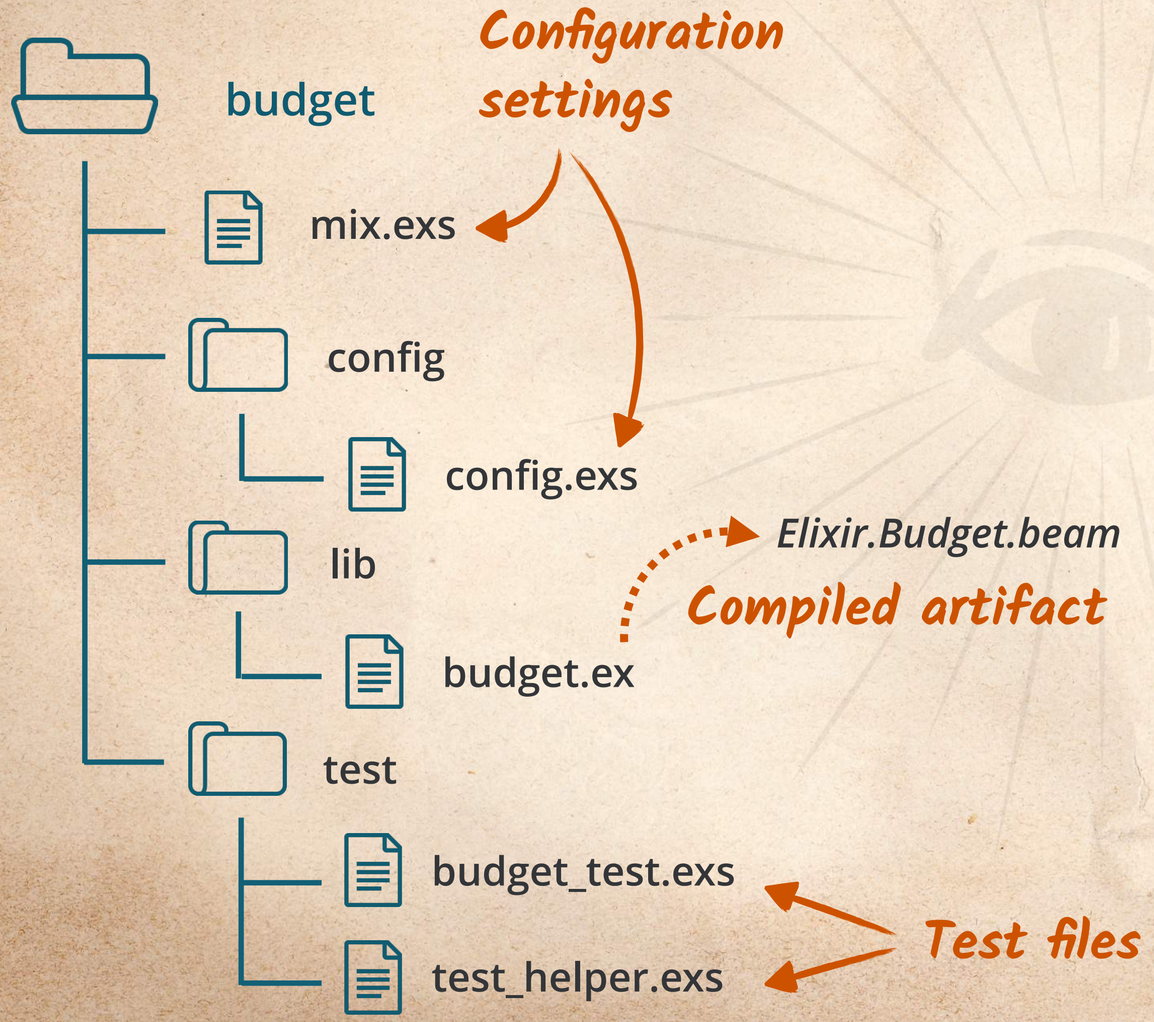
What the mix run command does:

1. Compiles the budget application.
2. Loads the generated bytecode into the Erlang Virtual Machine.
3. Detects the -e option and evaluates the argument as code.

* MIXING IT UP * with *ELIXIR*

The Difference Between File Extensions

Both .ex and .exs file extensions are treated **the same way**. The difference is intention: .ex files are meant to be **compiled** while .exs files are used for **scripting.**



.ex files

- Generates production artifacts (*.beam* files)
- Examples: lib files

.exs files

- Does NOT generate production artifacts
- Examples: configuration files, test files

Mix Help!

We can run the **mix help** command to see the list of all available tasks.

```
$ mix help
```

```
mix                   # Runs the default task (current: "mix run")
mix app.start         # Starts all registered apps
mix app.tree          # Prints the application tree
mix archive           # Lists installed archives
mix archive.build     # Archives this project into a .ez file
mix archive.install   # Installs an archive locally
mix archive.uninstall # Uninstalls archives

...
```