
Learning Elixir

by contributing

Montreal Elixir - May 2017

HOWTO

- Contribute to Elixir projects
- Get your projects noticed
- Navigate the compiler's source code



—

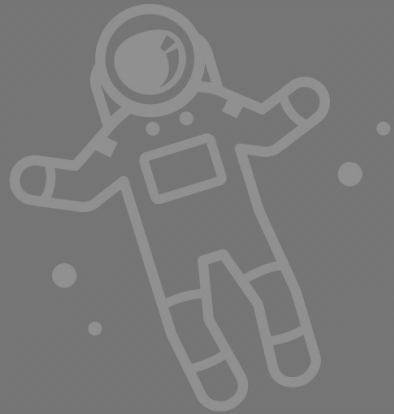
The biggest **blockers**
are not technical.



—

Documentation & UX

are the **best** ways to **start**.



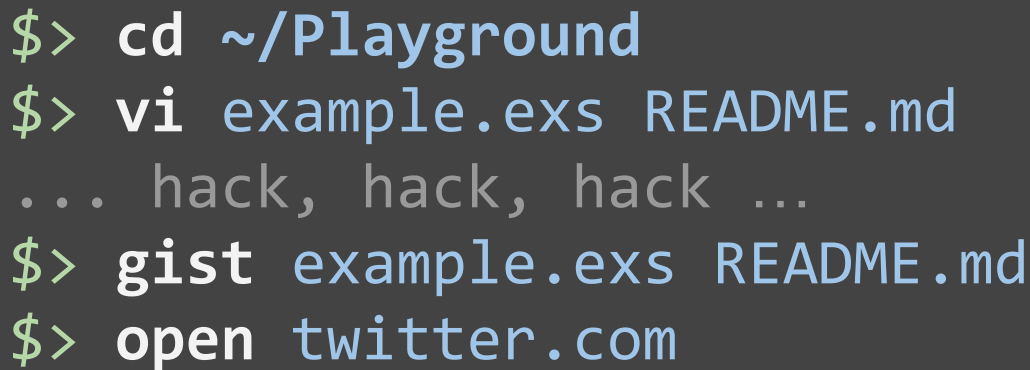
A close-up photograph of a turntable. A white vinyl record is spinning on the platter. A silver tonearm is positioned over the record, with its black headshell and stylus assembly resting on the surface. The tonearm has a silver-colored tube and a black headshell with two silver screws. The background is dark, and the turntable's platter is white with concentric grooves.

Remix.

elixir = $f(\text{ruby}) + f(\text{erLang}) + \dots + f(n)$

<http://everythingisaremix.info>

Lots of **micro** experiments.



```
$> cd ~/Playground  
$> vi example.exs README.md  
... hack, hack, hack ...  
$> gist example.exs README.md  
$> open twitter.com
```



Checklist

A quality open source project should have:

- **README**
- **Examples**
- **Docs & doctests**
- **Well organized code**
- **Tests & Continuous Integration**
- **Publish: `mix hex.publish`**
- **Github pages (optional)**

—

Open source *requires* marketing.



#elixirlang
#myelixirstatus

code

<https://github.com/elixir-lang/elixir>



Take aways

- **Micro progressions**
Begin ridiculously small.
- **Sizzle**
Marketing is vital.
- **Remix**
Re-arrange things that already work.
- **Practice**
Improve by reading & writing more code.
- **Share**
Early and often.

@joshnuss



Got questions?