Learning Elixir

by **contributing**

Montreal Elixir - May 2017

HOWTO

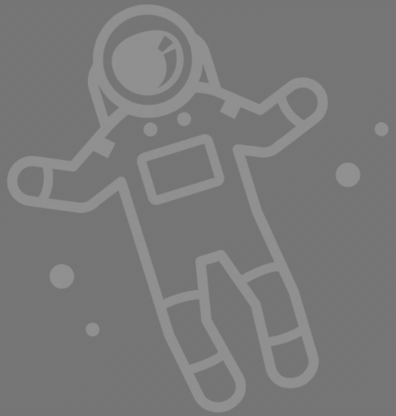
- Contribute to Elixir projects
- Get your projects noticed
- Navigate the compiler’s source code

—

The biggest **blockers** are **not technical**.

Documentation & UX

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

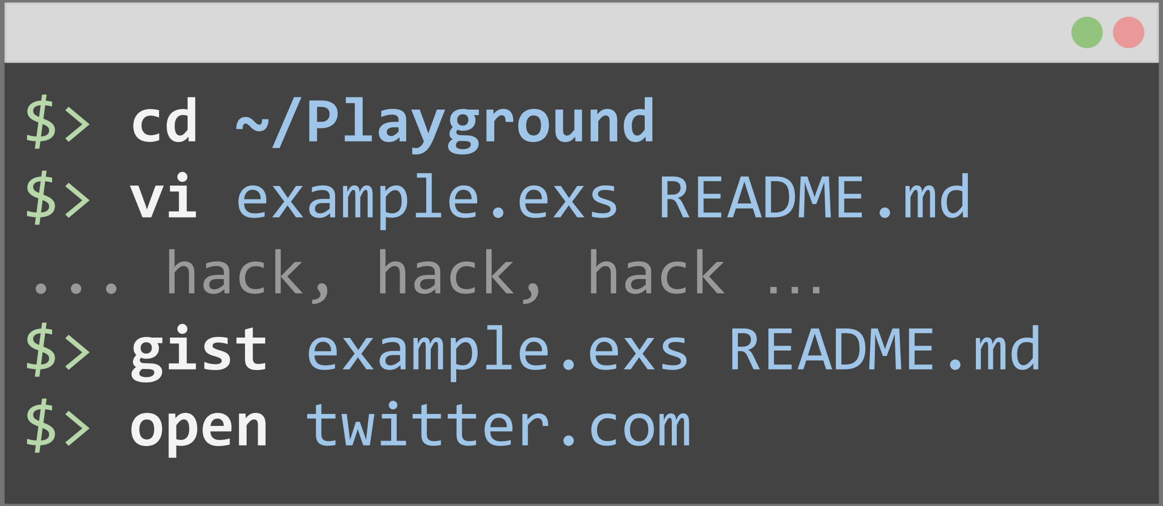


Remix.

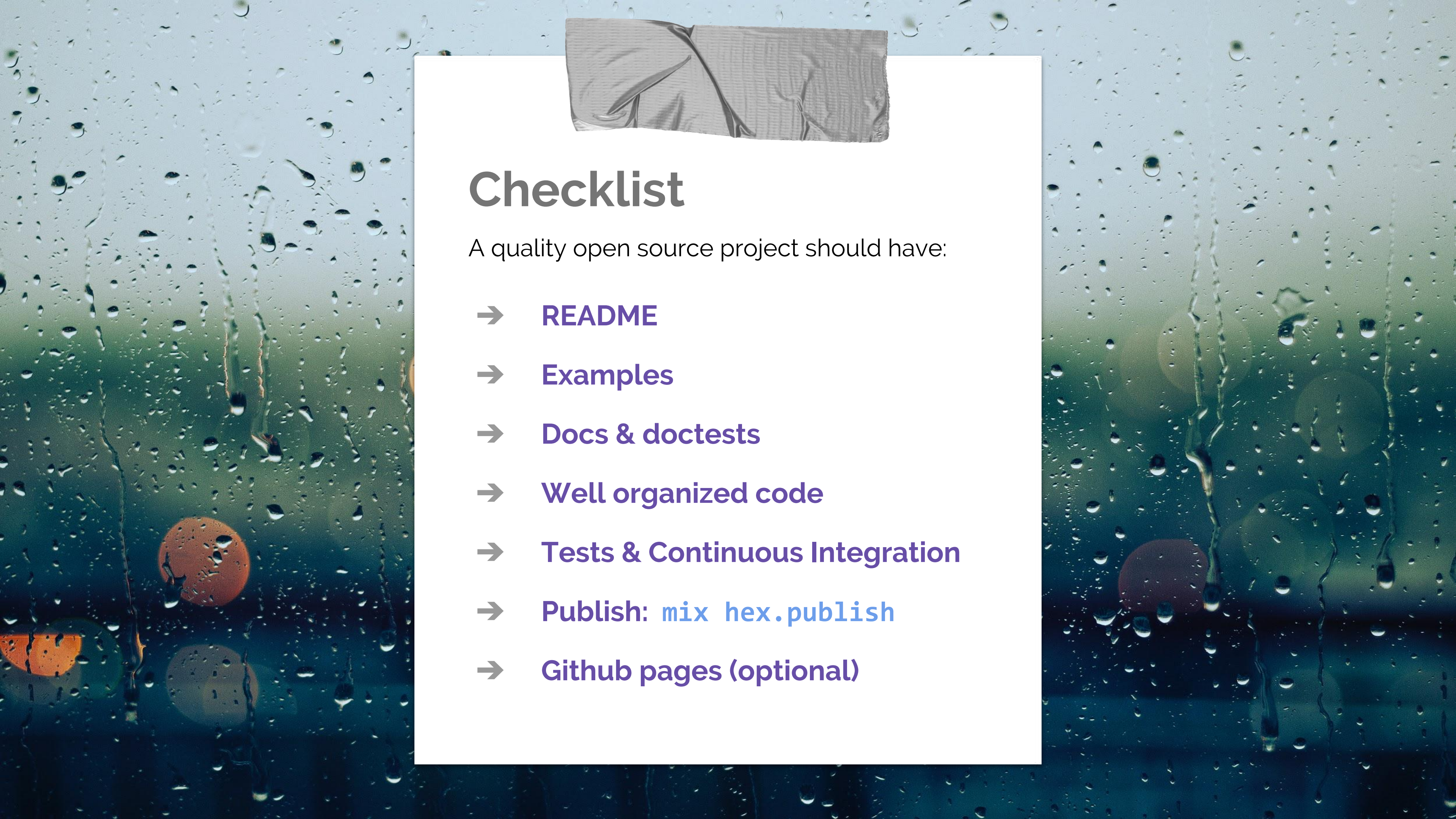
```
elixir = f(ruby) + f(erlang) + … + f(n)
```

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.

Got questions?

@joshnuss