

# Building Cross-platform GUI apps

**FOSDEM - 4 February 2024**

# About Me

- Software Engineer, Entrepreneur, Author
- Enlightenment, Maven contributor
- Founder of Fyne project
- Go developer since 2018
- CEO Fyne Labs



fyne  
labs

# How Go makes GUI simple

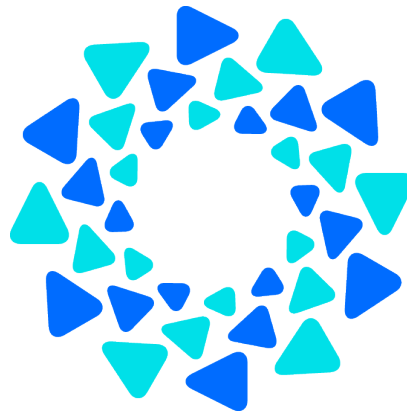
- Write once, run anywhere
- Apps that just work, do not require libraries or setup
- Native performance, without duplicating code
- Lower barrier of entry to building GUI apps
- Modern language standards and techniques
- Promote good engineering principles too

# Fyne Project

"

Fyne aims to be the simplest toolkit for developing beautiful and usable native graphical applications for desktop, mobile and beyond

"

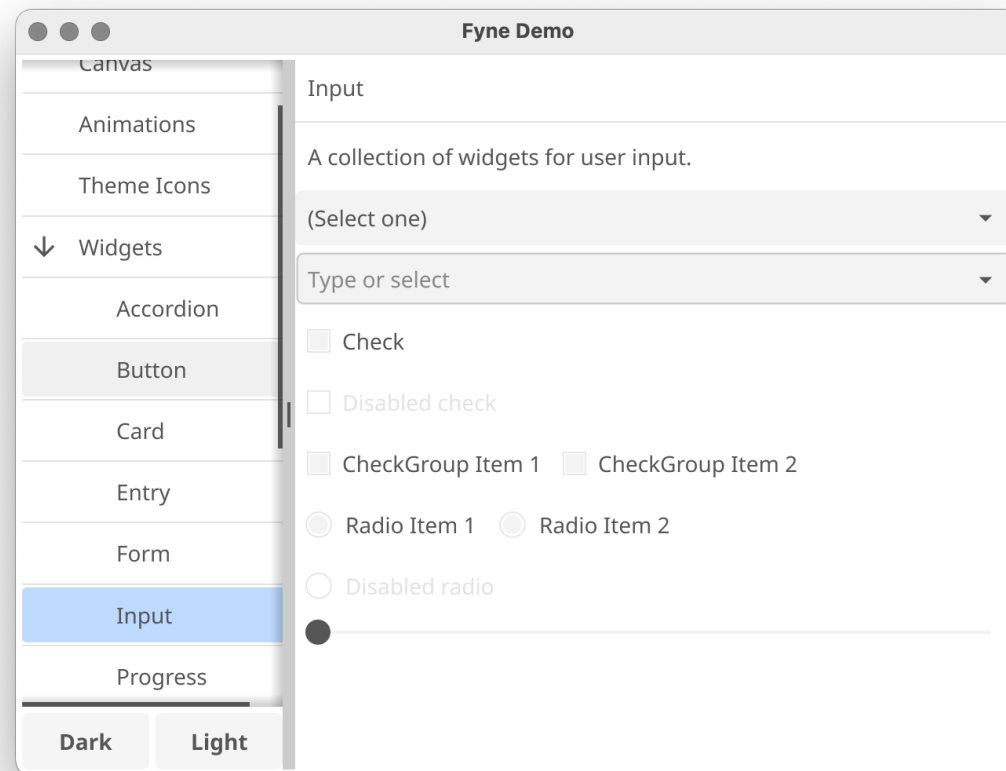


**fyne**

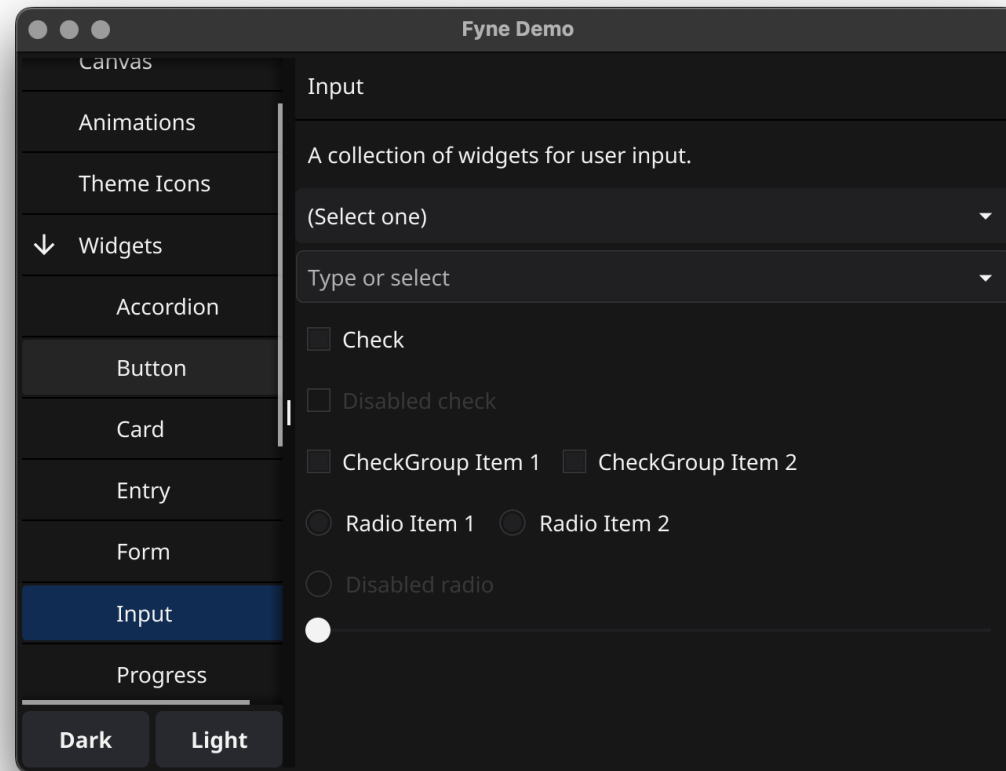
# Fyne Stats

- 6 years old
- Most popular GUI toolkit for Go
- Ranked 6th of all cross-platform GUI tools by @OSSInsight
- In GitHub 1000 most popular repositories
- Community of >2000 on Slack, Discord, Matrix

# Screenshot



# Screenshot (dark)



# Build our first app

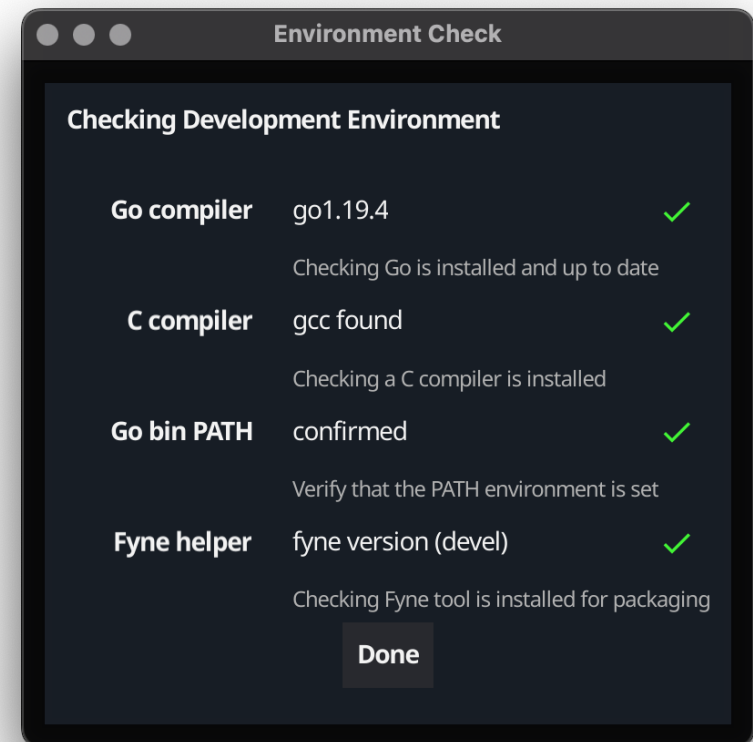
(install Go, gcc/clang)



# Build our first app

(install Go, gcc/clang)

Double check with "Fyne Setup"



# Build our first app

(install Go, gcc/clang)

```
$ mkdir myproject; cd myproject
```

```
$ go mod init myproject
```

```
$ go get fyne.io/fyne/v2
```

```
$ vim ui.go
```

# The code...

```
package main

import (
    "fyne.io/fyne/v2/app"
    "fyne.io/fyne/v2/widget"
)

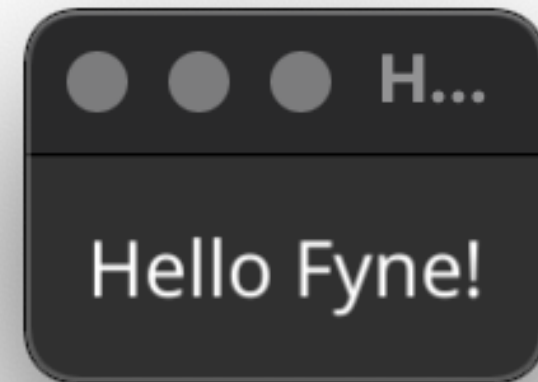
func main() {
    a := app.New()
    w := a.NewWindow("Hello")

    w.SetContent(widget.NewLabel("Hello Fyne!"))

    w.ShowAndRun()
}
```

# Build our first app

```
$ go run .
```



# Let's make a Markdown editor!

- Editor widget for input
- RichText widget for output
- HSplit container for layout
- Update data with OnChanged

# The code...

```
package main

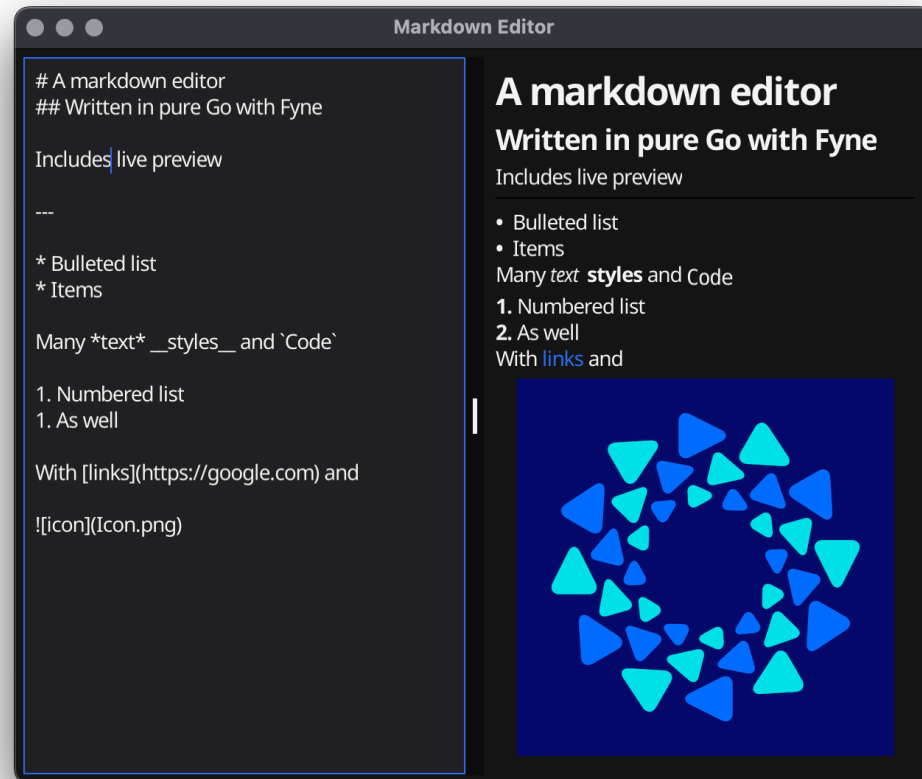
import (
    "fyne.io/fyne/v2/app"
    "fyne.io/fyne/v2/container"
    "fyne.io/fyne/v2/widget"
)

func main() {
    a := app.New()
    w := a.NewWindow("Markdown Editor")

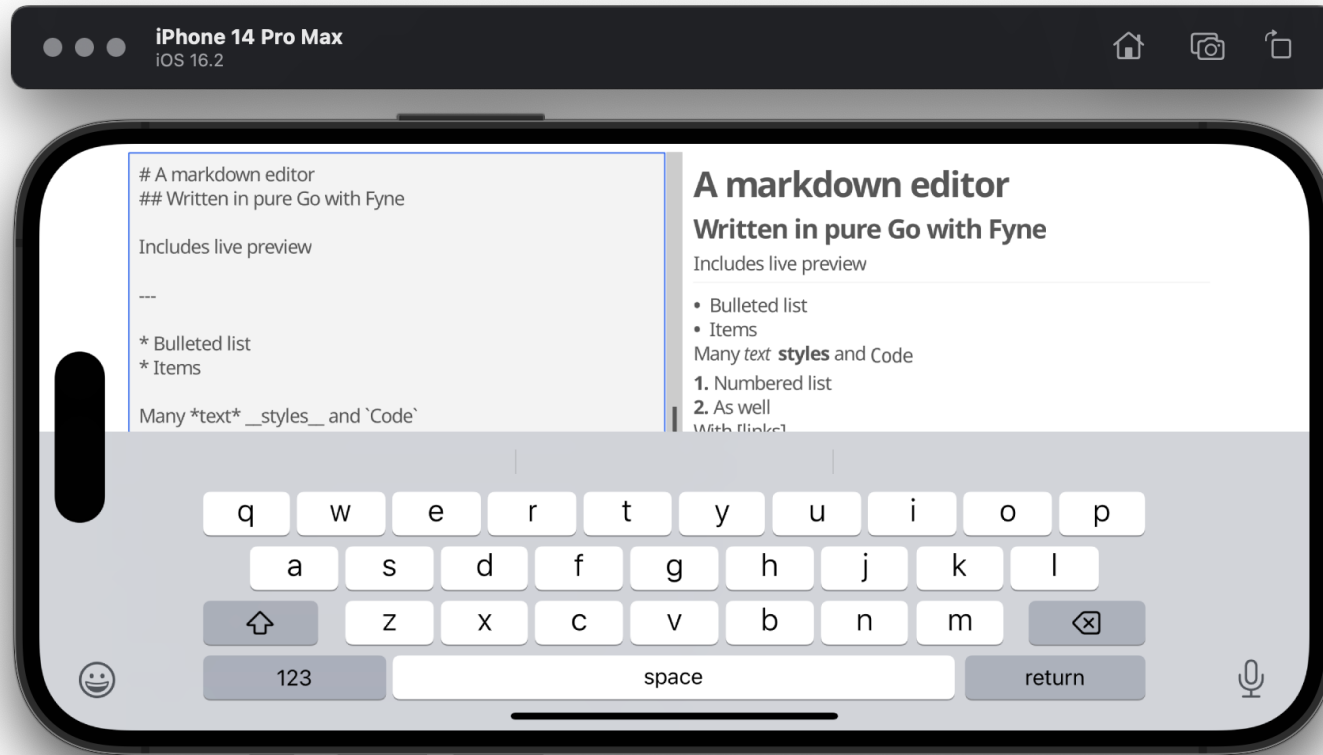
    edit := widget.NewMultiLineEntry()
    preview := widget.NewRichTextFromMarkdown("")
    edit.OnChanged = preview.ParseMarkdown

    w.SetContent(
        container.NewAdaptiveGrid(2, edit, preview))
    w.ShowAndRun()
}
```

# Let's make a Markdown editor!



# Let's make a Markdown editor!





# Compiling for other targets

```
$ go get fyne.io/fyne/v2/cmd/fyne@latest
```

```
$ fyne install
```

```
$ fyne package -os windows
```

```
$ fyne package -os android -appID com.company.myapp
```

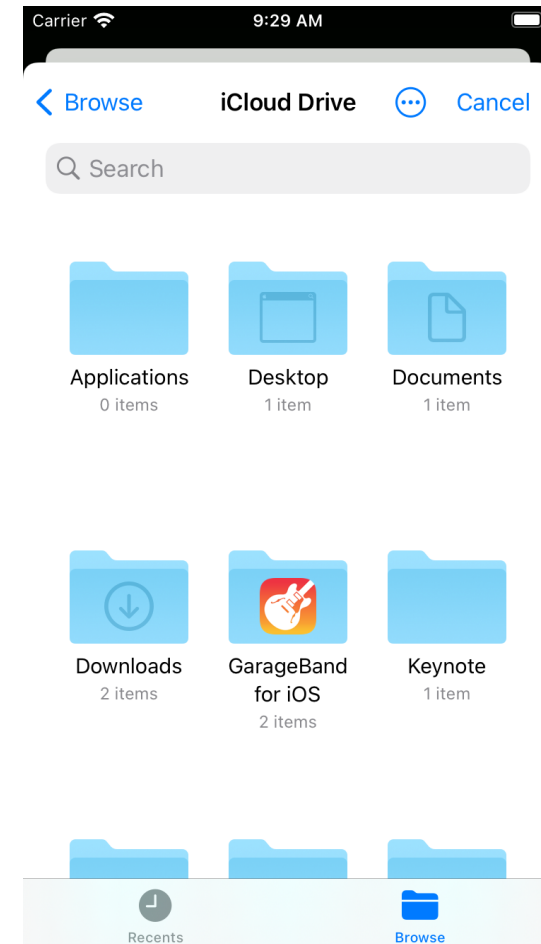
```
--
```

Also fyne-cross project

- <https://github.com/fyne-io/fyne-cross>

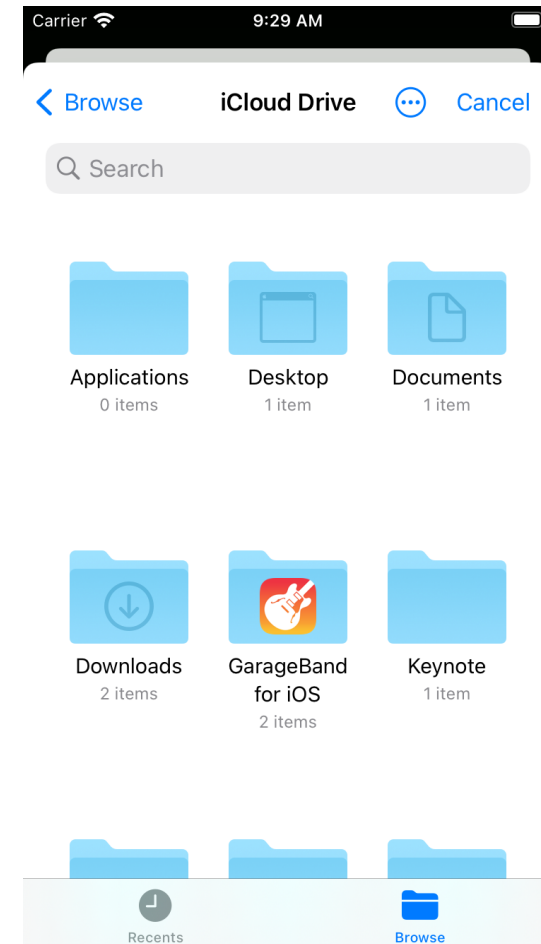
# Data input and binding

- `dialog.ShowFileOpen`
- `dialog.ShowFileSave`
- storage package abstraction



# Data input and binding

- `dialog.ShowFileOpen`
- `dialog.ShowFileSave`
- storage package abstraction
- 
- `binding.NewString`
- `widget.NewLabelWithData`



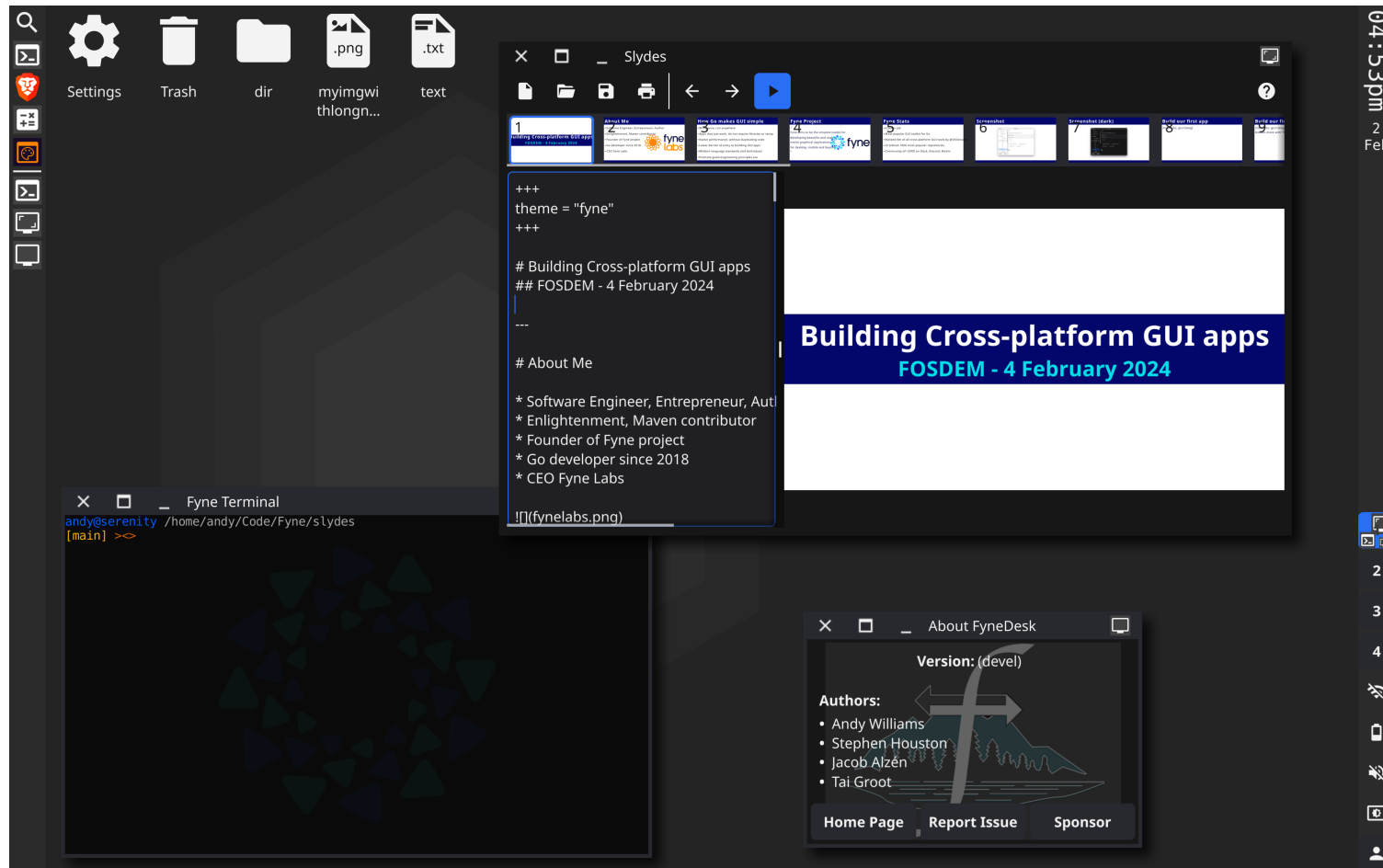
# But there is more!

- Full Widget library
- Dialogs, Forms
- Menus, and containers
- Notifications
- System Tray
- Native access (through CGo)

**But there is EVEN MORE!**

**This entire presentation, and desktop, is Fyne!**

# Screenshot



# Learn more

- Documentation: <https://docs.fyne.io>
- Videos: <https://www.youtube.com/@fyneio>
- Read: "Building Cross-platform GUI Applications with Fyne"
- Contribute: <https://github.com/fyne-io/fyne/>
- Sponsor! <https://fyne.io/sponsor/>

# Questions?

[@andydotxyz](#) / [andy@fynelabs.com](mailto:andy@fynelabs.com)