

# Welcome!

## Professional Development Seminar Series



[raywenderlich.com](https://raywenderlich.com)

Our mission is to create a world-class educational platform and community for mobile developers.



# Connect With Our Experts & Other Devs!

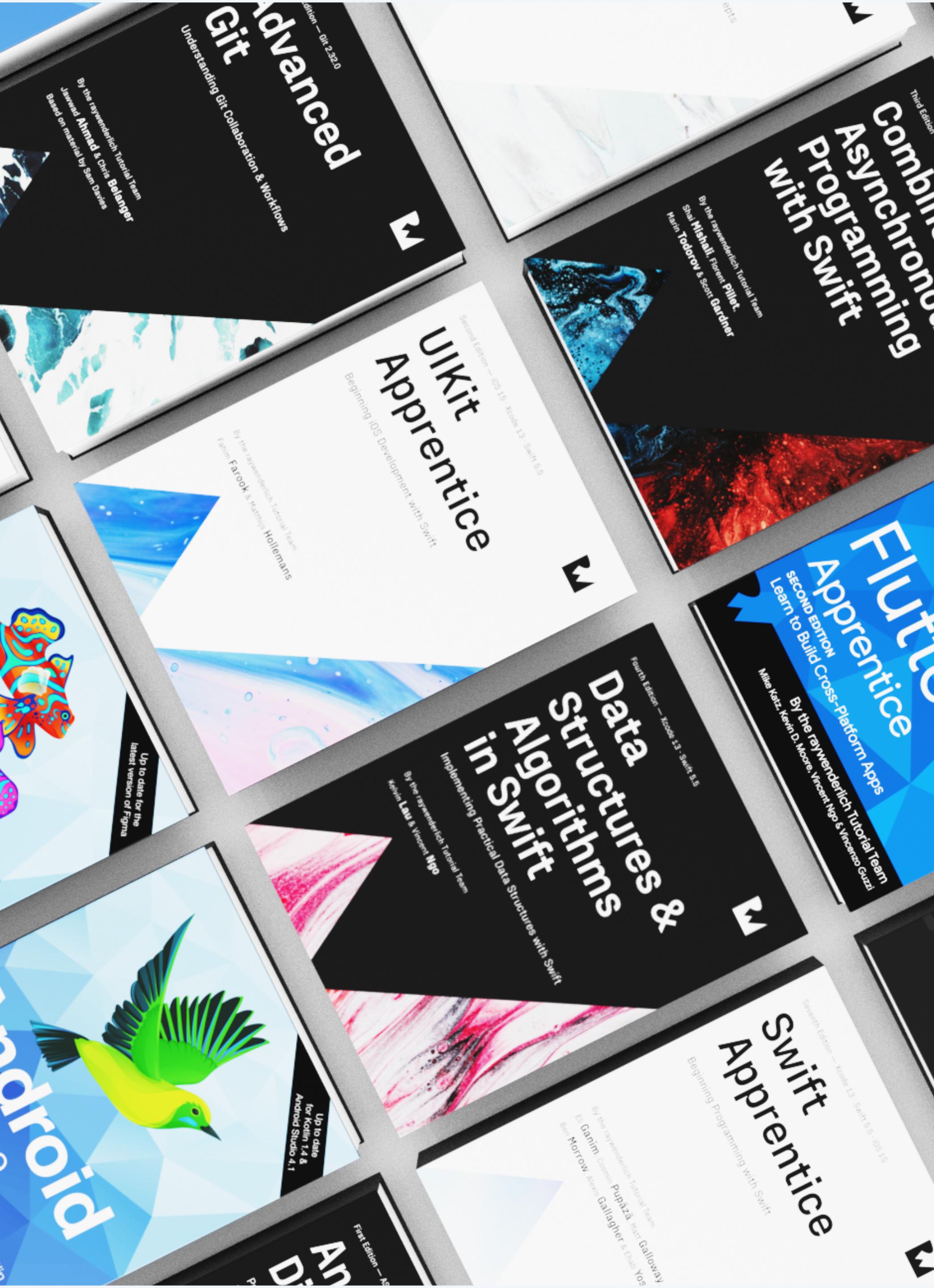
---



- \* Help with raywenderlich.com tutorials, books and videos
- \* Career advice and job opportunities
- \* Support for your apps and projects

-  [linkedin.com/company/rwenderlich/](https://linkedin.com/company/rwenderlich/)
-  [@rwenderlich](https://twitter.com/rwenderlich)
-  [discord.com/invite/ZTUmDxX](https://discord.com/invite/ZTUmDxX)
-  [facebook.com/raywenderlich](https://facebook.com/raywenderlich)
-  [instagram.com/rwenderlich](https://instagram.com/rwenderlich)



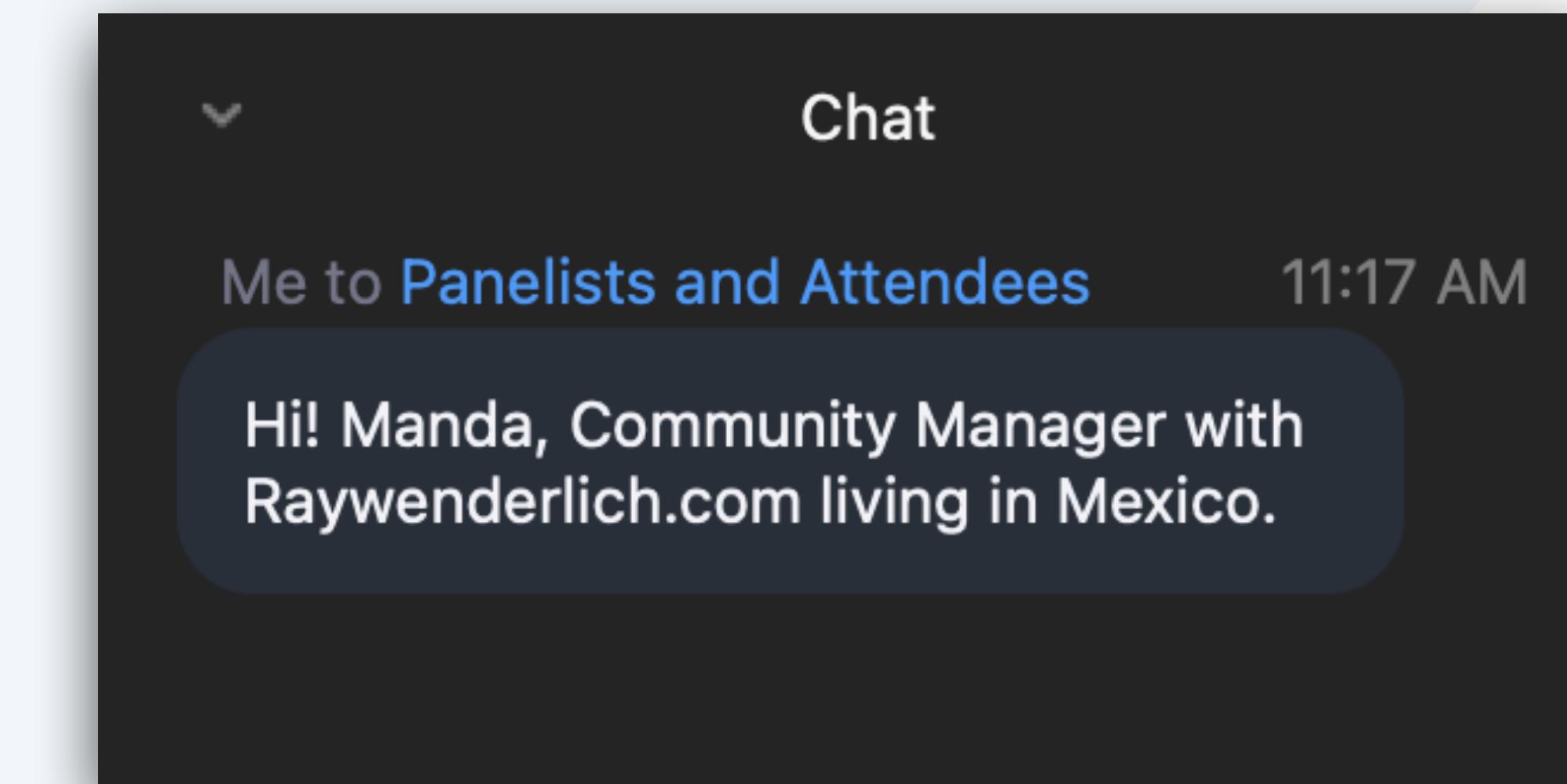


# Book Giveaway

In the Zoom Seminar Chat...

Say hello and where you're joining from!

That's it!



WELCOME!

# When to Write Your Own Tool

JANUARY 26, 2022 • 11AM EST



David Okun

PRINCIPAL IOS ENGINEER

 DOKUN24



[raywenderlich.com](https://raywenderlich.com)

# Agenda

- Who This Talk Is For
- Questions to Ask Yourself
- Question & Discussion



# What is a tool?



This talk is aimed at beginner developers who finally feel comfortable in their IDE, and are ready to start solidifying their skills





# Practice, practice, practice

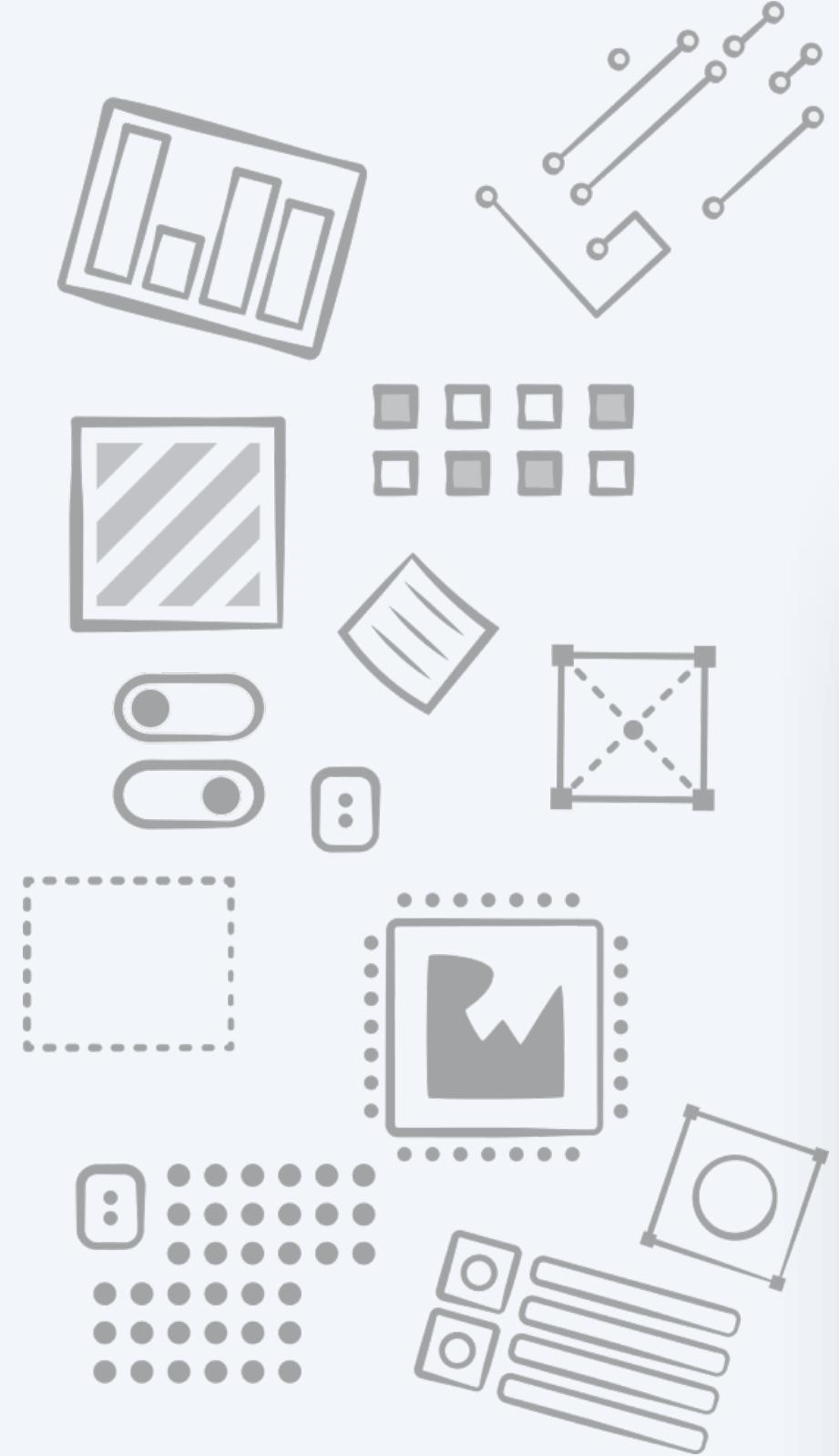


```
import SwiftUI

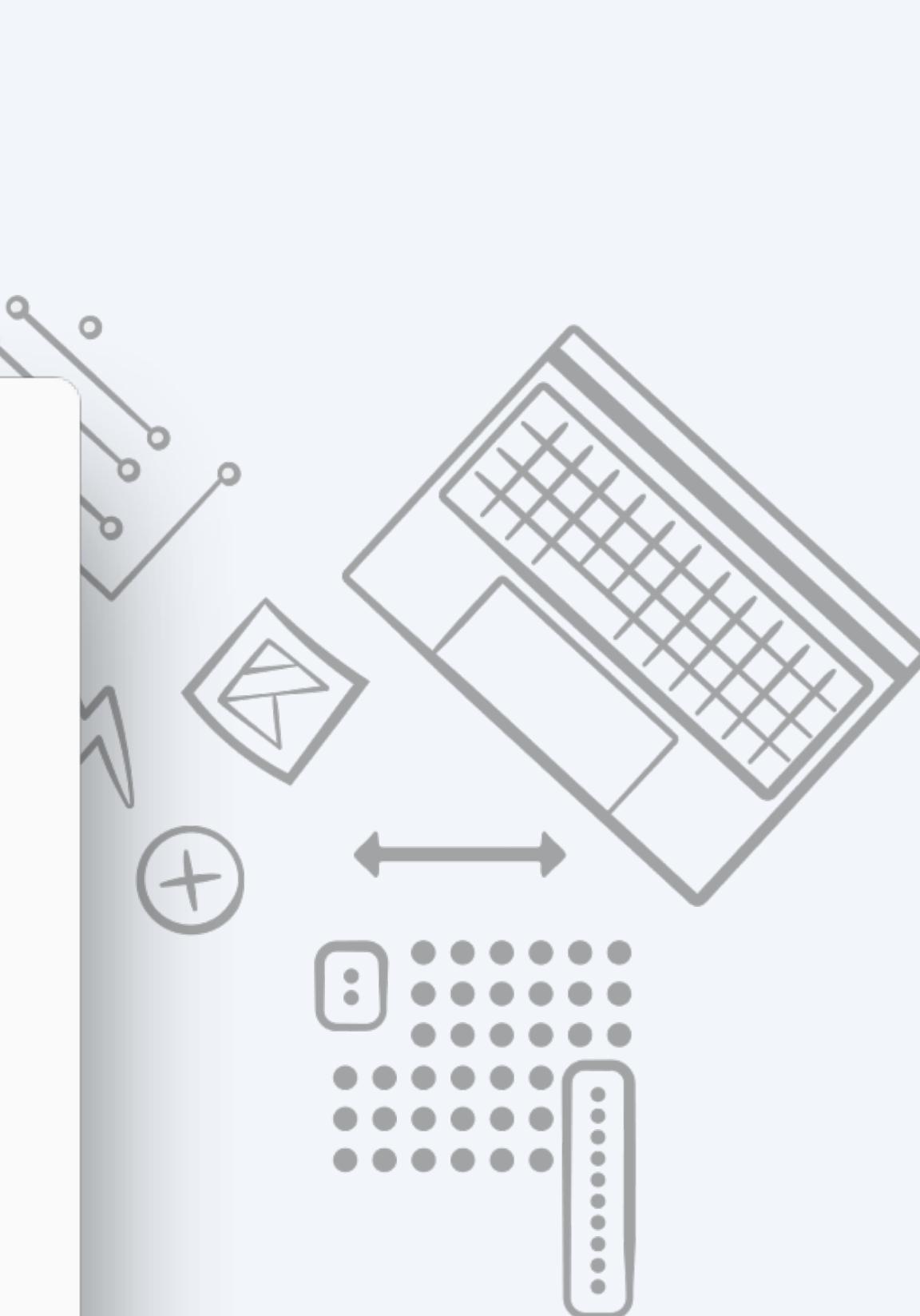
let primaryButton = Button("Primary") {
    print("You tapped a primary button!")
}.background(.blue)

let secondaryButton = Button("Secondary") {
    print("You tapped a secondary button!")
}.background(.green)

let tertiaryButton = Button("Tertiary") {
    print("You tapped a tertiary button!")
}.background(.gray)
```



```
struct MyView: View {  
    var body: some View {  
        VStack(alignment: .leading) {  
            Text("DRY")  
            Spacer()  
            primaryButton  
            secondaryButton  
            tertiaryButton  
        }  
    }  
}
```



# Questions to ask

- How many people are going to see this view?
- How many developers are going to use this view?
- Will I need to use those buttons elsewhere in my app?
- How many other kinds of buttons will I need to use?
- Who is depending on this code, and when?
- How much can “future you” tolerate?



**“Don’t repeat yourself.”**

-Andy Hunt, Dave Thomas, and every crusty senior developer you've ever talked to via code review





```
enum ButtonType: String {
    case primary
    case secondary
    case tertiary

    var backgroundColor: Color {
        switch self {
            case .primary: return .blue
            case .secondary: return .green
            case .tertiary: return .gray
        }
    }
}

@ViewBuilder func typedButton(with type: ButtonType) -> some View {
    Button(type.rawValue.capitalized) {
        print("You tapped a \(type.rawValue) button!")
    }.background(type.backgroundColor)
}
```



```
struct MyView: View {  
    var body: some View {  
        VStack(alignment: .leading) {  
            Text("DRY")  
            Spacer()  
            typedButton(with: .primary)  
            typedButton(with: .secondary)  
            typedButton(with: .tertiary)  
        }  
    }  
}
```

**“Don’t repeat yourself.”**

-Andy Hunt, Dave Thomas, and every crusty senior developer you've ever talked to via code review





```
enum ButtonType: String, CaseIterable {
    case primary
    case secondary
    case tertiary

    var backgroundColor: Color {
        switch self {
            case .primary: return .blue
            case .secondary: return .green
            case .tertiary: return .gray
        }
    }
}
```



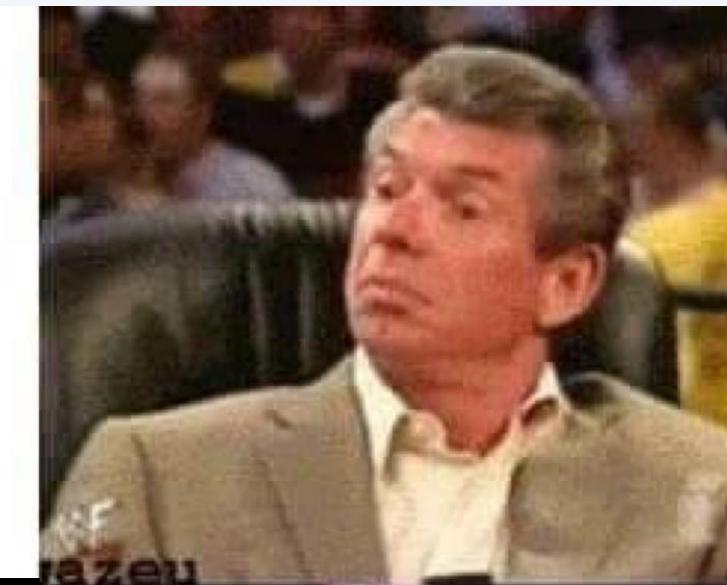
```
struct MyView: View {  
    var body: some View {  
        VStack(alignment: .leading) {  
            Text("DRY")  
            Spacer()  
            ForEach(ButtonType.allCases, id: \.self.rawValue) { type in  
                typedButton(with: type)  
            }  
        }  
    }  
}
```



Who doesn't love a  
good meme?

Writing a for loop to evaluate expressions  
for each item in an array

---



Writing a method that takes an array  
and evaluates each item individually

---



Creating a static function on an enum  
that serves as a utility for your array

---



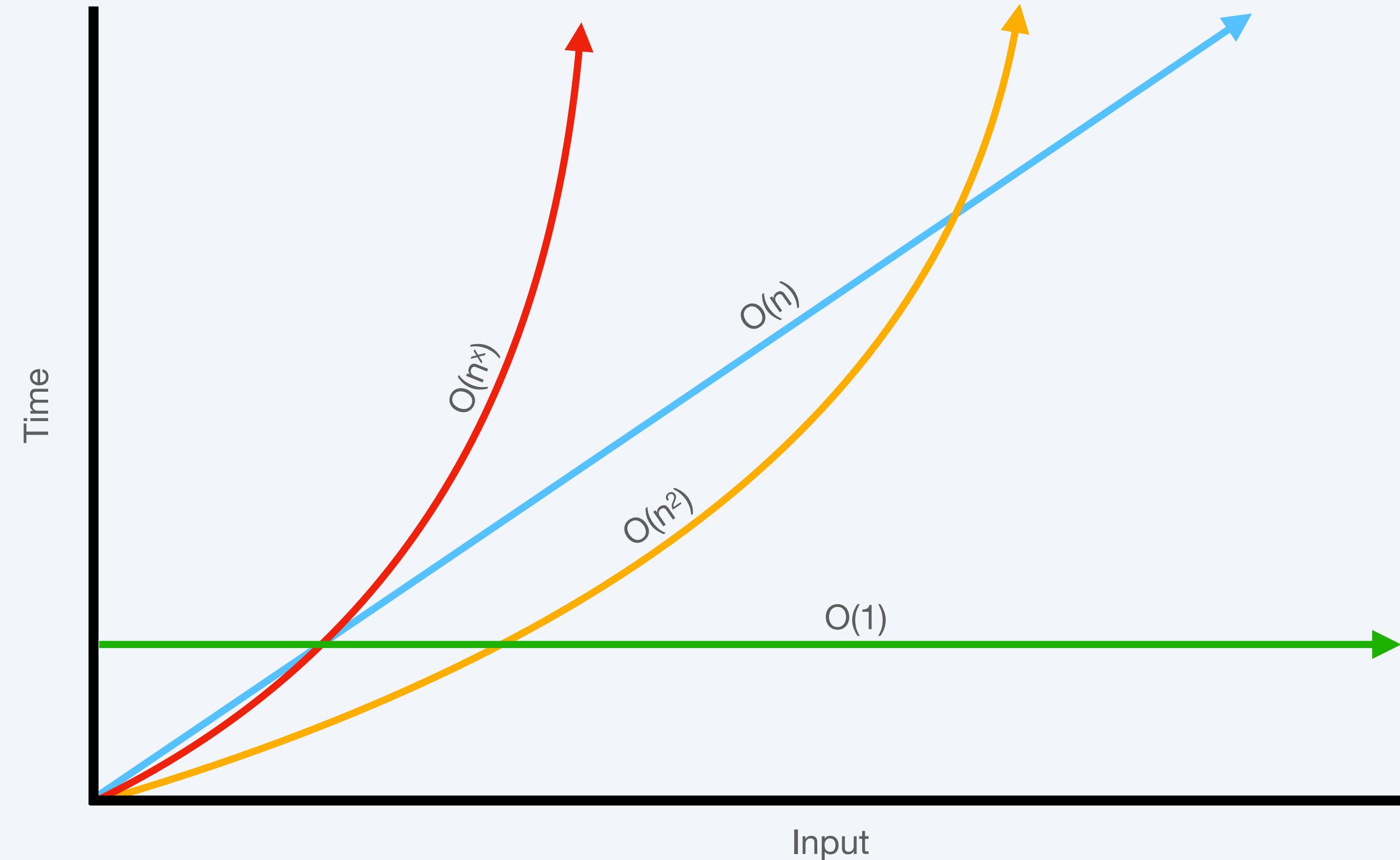
Writing a framework that can be imported  
into your project to evaluate the array



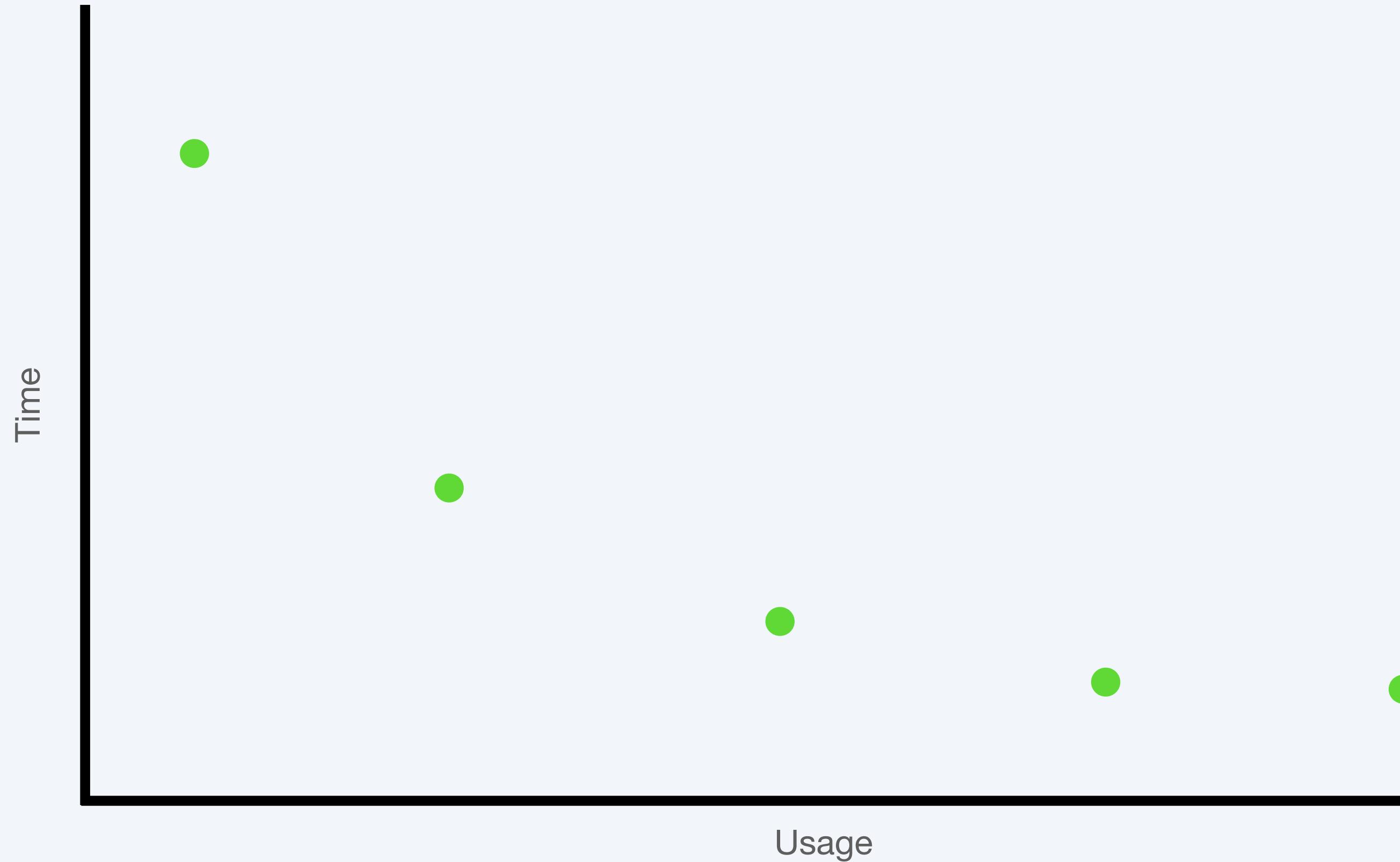
raywenderlich.com

- How many people are going to see this view?
- How many developers are going to use this view?

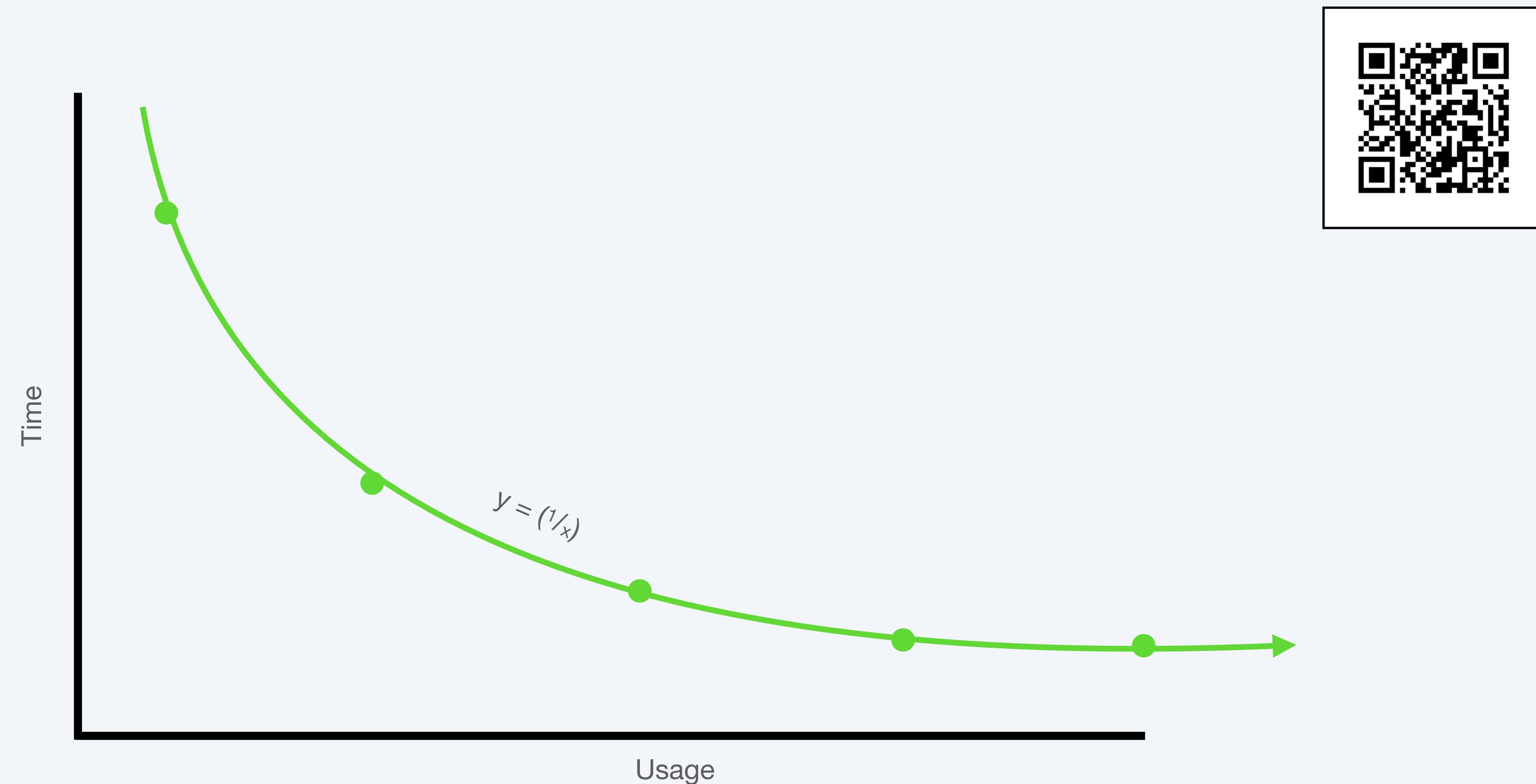




raywenderlich.com



[raywenderlich.com](https://raywenderlich.com)



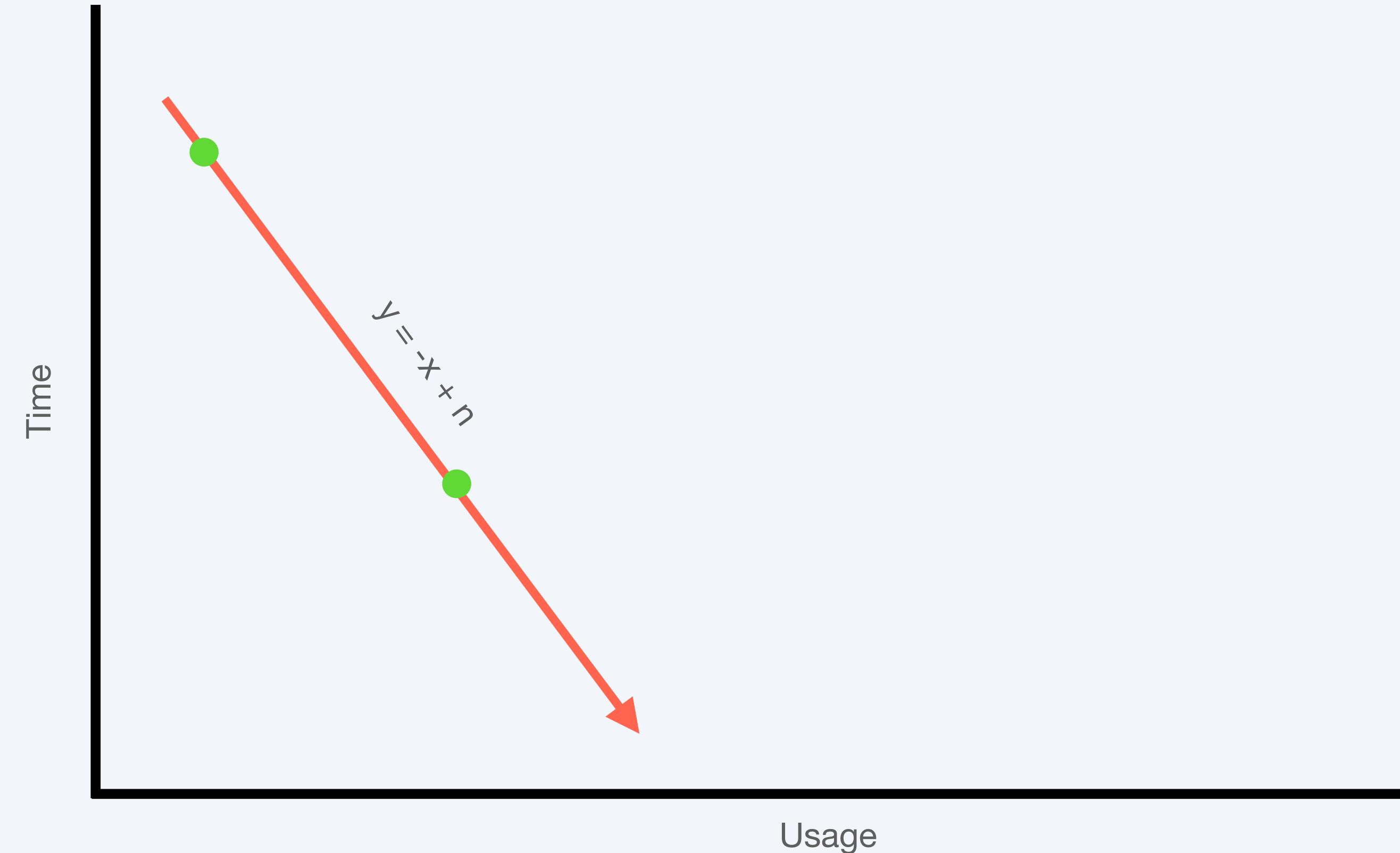
raywenderlich.com

- Will I need to use those buttons elsewhere in my app?
- How many other kinds of buttons will I need to use?



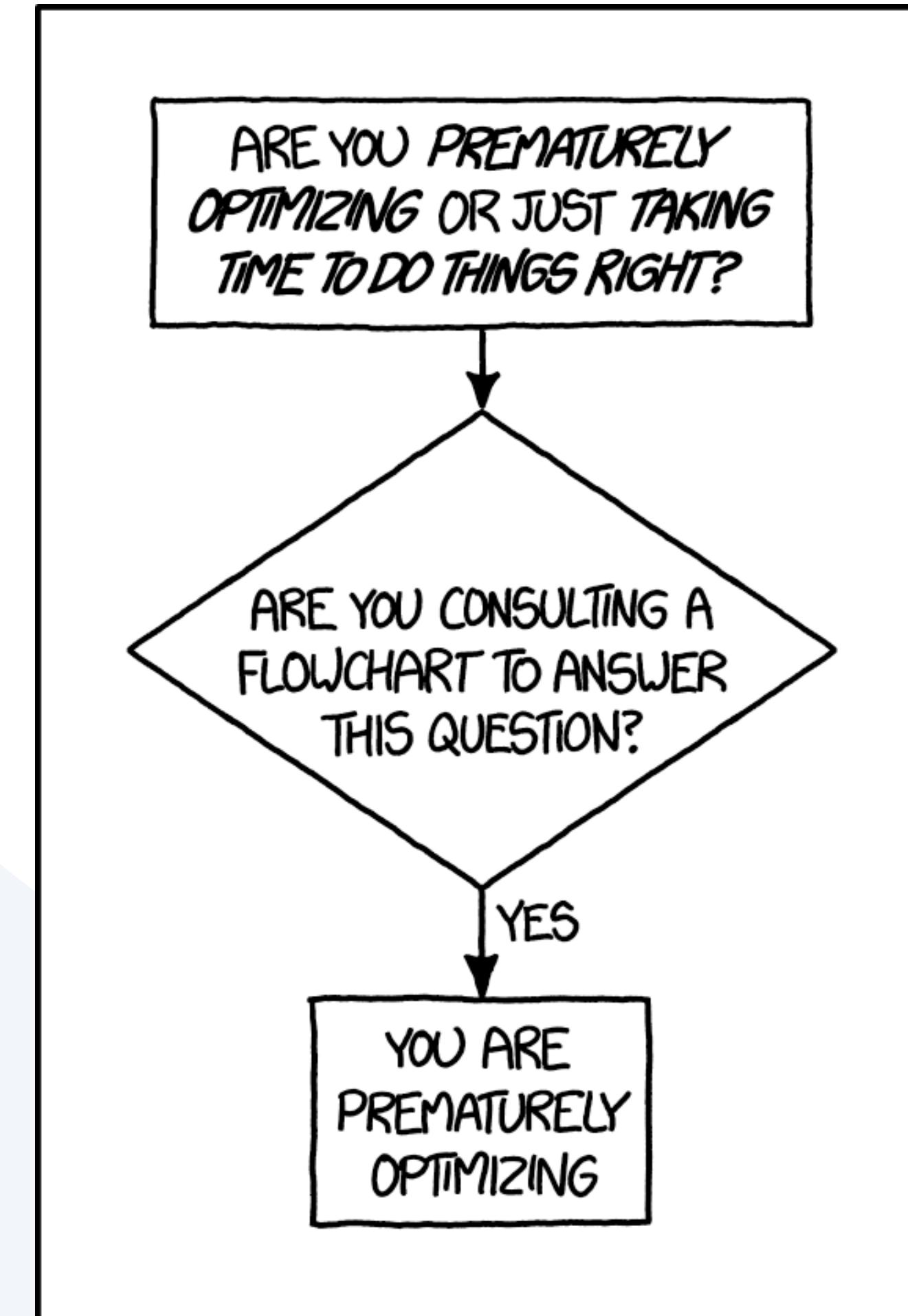


[raywenderlich.com](https://raywenderlich.com)



raywenderlich.com

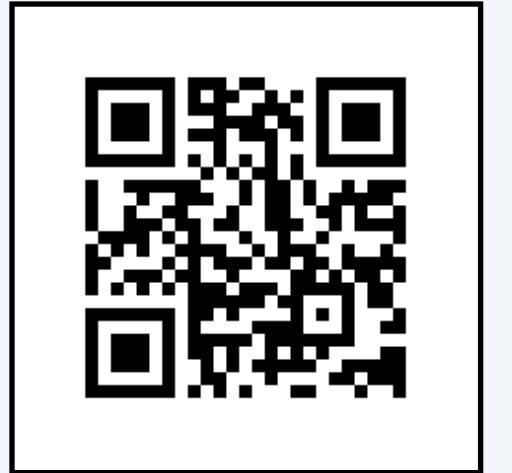
# Overthinking it



- Who is depending on this code, and when?



# Hyrum's Law



With a sufficient number of users of an API,  
it does not matter what you promise in the contract:  
all observable behaviors of your system  
will be depended on by somebody.



- How much can “future you” tolerate?







**Carla Notarobot**   
@CarlaNotarobot

there is nothing more permanent than a temporary  
workaround



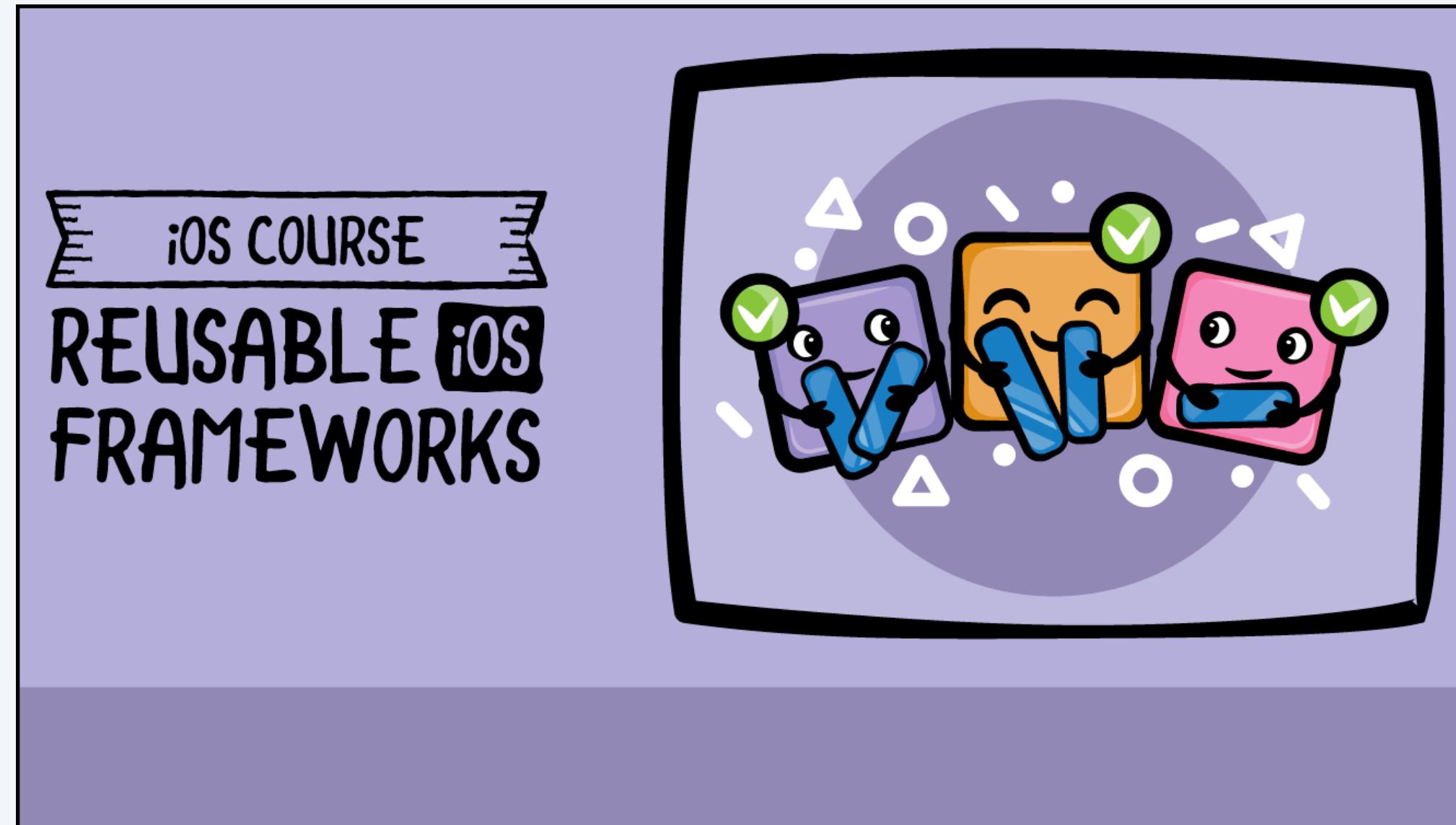
raywenderlich.com

# Takeaways

- Always think about the code “future you” wants to use
- If you’re stuck between “too much” or “not enough”, opt for too much
- The only way to get better at discerning these decisions is via practice
- It’s gonna be ok, y’all



# Shameless Plug



<https://www.raywenderlich.com/10318291-reusable-ios-frameworks>



raywenderlich.com

# Questions?

