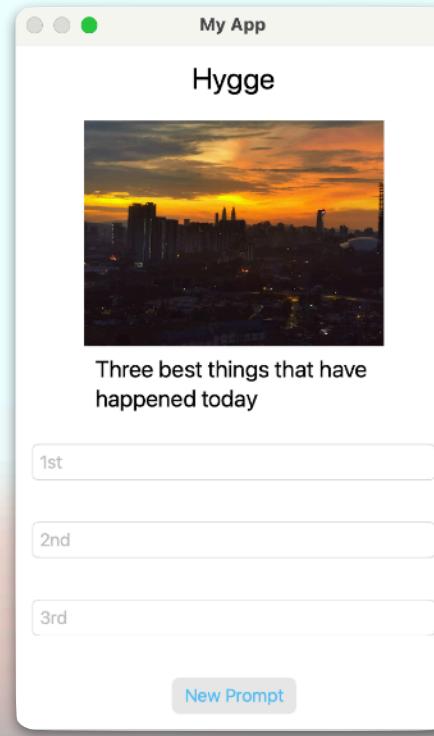




# Beginners Guide to Creating Your First App



By James Abela

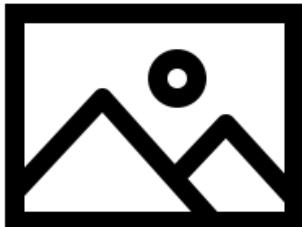


Distinguished  
Educator

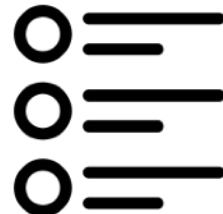
# Today's Coding will include:



Displaying Text



Pictures & icons



List handling

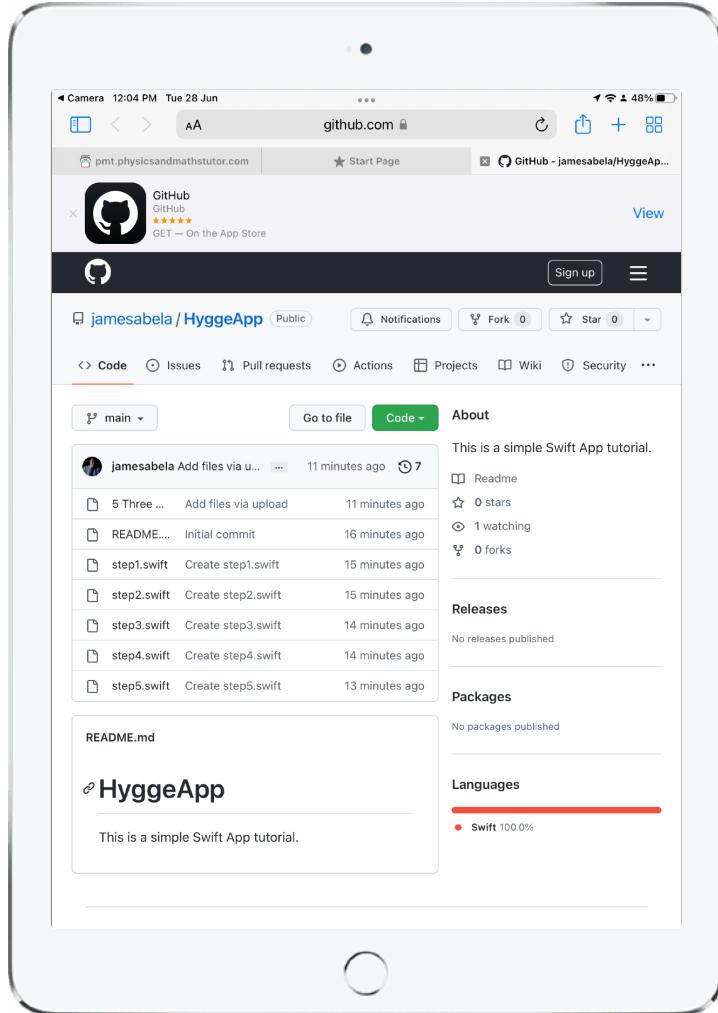


Inputting text

Can use iPad or Mac OS with Swift Playgrounds 4.1

Files at: <https://github.com/jamesabela/HyggeApp>





	jamesabela	Add files via u...	...	11 minutes ago	 7
	5 Three ...	Add files via upload		11 minutes ago	
	README....	Initial commit		16 minutes ago	
	step1.swift	Create step1.swift		15 minutes ago	
	step2.swift	Create step2.swift		15 minutes ago	
	step3.swift	Create step3.swift		14 minutes ago	
	step4.swift	Create step4.swift		14 minutes ago	
	step5.swift	Create step5.swift		13 minutes ago	



Files at: <https://github.com/jamesabela/HyggeApp>

## About Playgrounds

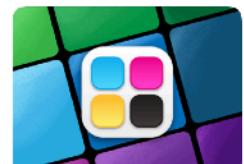
[Preferences...](#)[Services](#)[Hide Playgrounds](#)[Hide Others](#)[Show All](#)[Quit Playgrounds](#)

Filter

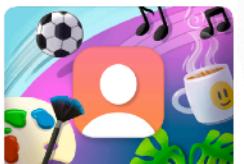
Basic Three Good Things



My App copy 3



Organizing with Grids



About Me



My App copy



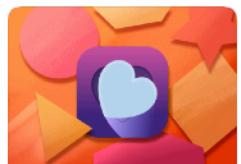
Three Good Things Tidied



Fonts You Can Use



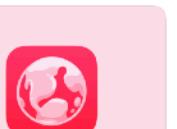
Meme Creator



Animating Shapes



More Playgrounds



See All



+



+



Shapes



Blu's Adventure



Spirals



Answers



Sonic Create



Choose Your Own Story



Puzzle World

## About Playgrounds

- Preferences... ⌘ ,
- Services >
- Hide Playgrounds ⌘ H
- Hide Others ⌘ ⌥ H
- Show All
- Quit Playgrounds ⌘ Q

Basic Three Good Things



Organizing with Grids



Fonts You Can Use



# Playgrounds

Version 4.1 (1674.8)

Copyright © 2016-2022 Apple Inc.  
All rights reserved.

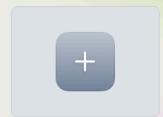


Animating Shapes



See All

## More Playgrounds



App



Playground



Shapes



Blu's Adventure



Spirals



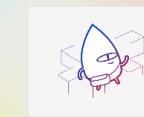
Answers



Sonic Create



Choose Your Own Story



Puzzle World

## My Playgrounds



Basic Three Good Things



My App copy 3



Fathers Day Card



Box Ga



About Me



My App copy



Notepad 3



Meme Ci



More Playgrounds



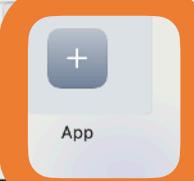
Pseudocode →



Pseudocode →



Spirals



+

App



Playground



Shapes



Blu's Adventure



Spirals

11:31 AM Mon 27 Jun

59%

Locations

My Playgrounds

Help Select



My App

NEW



Basic Three Good Things



AR Create  
Swift 5.3



Answers  
Swift 5.3

More Playgrounds



App



Playground



Sonic Create



Sensor Create



Answers

See All

## My Playgrounds

Filter



My App  
NEW



Basic Three Good Things



Fathers Day Card



Box Game



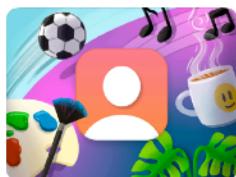
Three Good Things Tidied



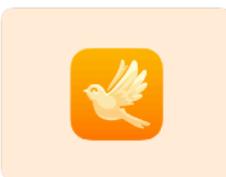
Fonts You Can Use



Organizing with Grids



About Me



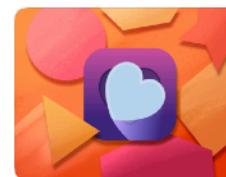
My App copy



Notepad 3



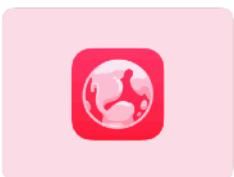
Meme Creator



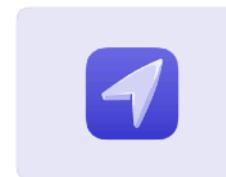
Animating Shapes



Choose Your Own Story co...



My Map



## More Playgrounds

See All



App



Playground



Shapes



Blu's Adventure



Spirals



Answers



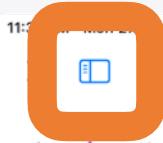
Sonic Create



Choose Your Own Story



Puzzle World



\*\*\*

58%

My App



```
1 import SwiftUI
2
3 struct ContentView: View {
4     var body: some View {
5         VStack {
6             Image(systemName: "globe")
7                 .imageScale(.large)
8                 .foregroundColor(.accentColor)
9             Text("Hello, world!")
10        }
11    }
12 }
13
```



11:31 AM Mon 27 Jun

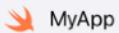


Find in Source Files



My App  
App Settings

Code



...  
My App



```
1 import SwiftUI
2
3 struct ContentView: View {
4     var body: some View {
5         VStack {
6             Image(systemName: "globe")
7                 .imageScale(.large)
8                 .foregroundColor(.accentColor)
9             Text("Hello, world!")
10        }
11    }
12 }
```



11:31 AM Mon 27 Jun



My App

...



App Preview

58%

```
1 import SwiftUI
2
3 struct ContentView: View {
4     var body: some View {
5         VStack {
6             Image(systemName: "globe")
7                 .imageScale(.large)
8                 .foregroundColor(.accentColor)
9             Text("Hello, world!")
10        }
11    }
12 }
13
```

Hello, world!



## My App copy 3



App Preview ▾



Find in Source Files



My App

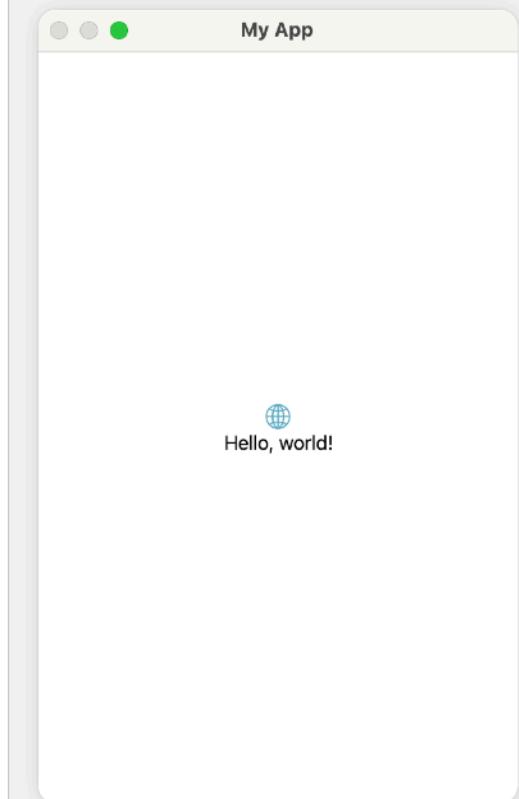
App Settings

Code

ContentView

MyApp

```
1 import SwiftUI
2
3 @main
4 struct MyApp: App {
5     var body: some Scene {
6         WindowGroup {
7             ContentView()
8         }
9     }
10 }
11
```





## My App copy 3



App Preview ▾

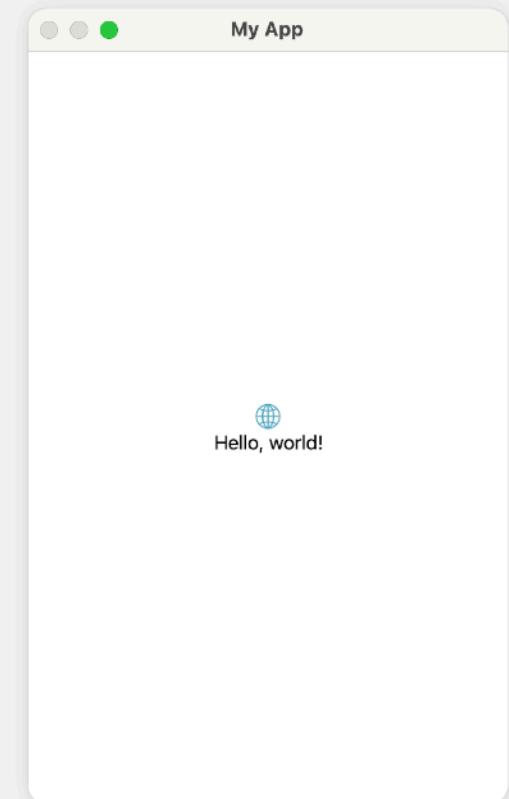


Find in Source Files



ContentView

```
1 import SwiftUI
2
3 struct ContentView: View {
4     var body: some View {
5         VStack {
6             Image(systemName: "globe")
7                 .imageScale(.large)
8                 .foregroundColor(.accentColor)
9             Text("Hello, world!")
10        }
11    }
12 }
13
```



# My App copy 3

```
1 import SwiftUI
2
3 struct ContentView: View {
4     var body: some View {
5         VStack {
6             Image(systemName: "globe")
7                 .imageScale(.large)
8                 .foregroundColor(.accentColor)
9             Text("Hello, world!")
10        }
11    }
12}
13
```



Hello, world!

# My App copy 3



Find in Source Files

My App  
App Settings

Code

ContentView

MyApp

```
1 import SwiftUI
```

V accentColor	V indigo
V black	V mint
V blue	V orange
V brown	V pink
V clear	V primary
V cyan	V purple
V gray	V red
V green	V secondary
V teal	V teal
V white	V white
V yellow	V yellow

```
/view: View {
    some View {
        Image(systemName: "globe")
            .imageScale(.large)
            .foregroundStyle(.accentColor)
        text("Hello, world!")
    }
}
```

U large
U medium
U small

Change some settings. See what it does

# Vertical Stack

```
1 1 import SwiftUI
2 2
3 3 @ContentView: View {
4 4     var body: some View {
5 5         VStack {
6 6             Image(systemName: "globe")
7 7                 .imageScale(.large)
8 8                 .foregroundColor(.accentColor)
9 9             Text("Hello, world!")
10 10 }
11 11
12 12
13 13
```



Predict

```
1 import SwiftUI
2
3 struct ContentView: View {
4     var body: some View {
5         VStack {
6             Text("Hello, world!")
7             Image(systemName: "globe")
8                 .imageScale(.large)
9                 .foregroundColor(.green)
10
11         } ...
12     }
13 }
14
```

# Spot the difference?

ContentView

MyApp

```
1 import SwiftUI
2
3 struct ContentView: View {
4     var body: some View {
5         VStack {
6             Text("Hello, world!")
7             Image(systemName: "globe")
8                 .imageScale(.large)
9                 .foregroundColor(.green)
10
11     } ...
12 }
13
14 }
```

My App

Hello, world!

The image shows a split-screen view of Xcode. On the left, the 'ContentView' file is open, displaying Swift code for a SwiftUI view. The code defines a struct ContentView with a body containing a VStack. Inside the VStack, there is a Text view with the string "Hello, world!" and an Image view with the system name "globe". The image has its scale set to large and its foreground color set to green. On the right, a screenshot of the 'My App' window is shown, which displays the text "Hello, world!" and a green globe icon. The title bar of the window also says "My App".

# Spot the difference?

ContentView

MyApp

```
1 import SwiftUI
2
3 struct ContentView: View {
4     var body: some View {
5         VStack {
6             Text("Hello, world!")
7             Image(systemName: "globe")
8                 .imageScale(.large)
9                 .foregroundColor(.green)
10
11     } ...
12 }
13
14 }
```

My App

Hello, world!

The image shows a split-screen view of Xcode. On the left, the 'ContentView' file is open, displaying Swift code for a SwiftUI view. The code defines a struct 'ContentView' with a 'body' property containing a 'VStack' that holds a 'Text' view with the string 'Hello, world!' and an 'Image' view with the system name 'globe'. The image has its 'imageScale' set to '.large' and its 'foregroundColor' set to '.green'. On the right, a preview window titled 'My App' shows the resulting interface: a white screen with the text 'Hello, world!' in black and a large green globe icon below it.

# Spot the difference?

```
1 import SwiftUI
2
3 struct ContentView: View {
4     var body: some View {
5         VStack {
6             Image(systemName: "globe")
7                 .imageScale(.large)
8                 .foregroundColor(.accentColor)
9             text("Hello, world!")
10        }
11    }
12 }
13
```

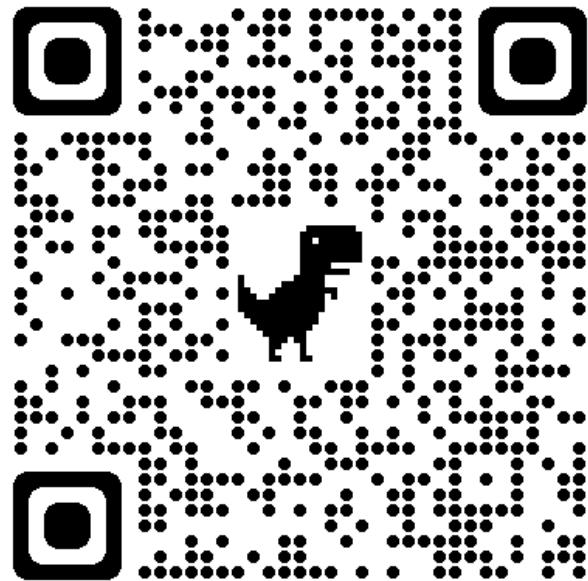
Spot the bug?

```
1 import SwiftUI
2
3 struct ContentView: View {
4     var body: some View {
5         VStack {
6             Image(systemName: "globe")
7                 .imageScale(.large)
8                 .foregroundColor(.accentColor)
9             text("Hello, world!")
10        }
11    }
12 }
13
```

Watch out - Swift is cASE sENSITIVE

# Step 1

Files at: <https://github.com/jamesabela/HyggeApp>





ContentView

MyApp

```
1 import SwiftUI
2
3 struct ContentView: View {
4     var body: some View {
5         VStack {
6             Text("Hello, world!")
7             Image(systemName: "globe")
8                 .imageScale(.large)
9                 .foregroundColor(.green)
10
11     }
12 }
13
14 }
```

1. Delete everything inside the Vstack

My App

ContentView

MyApp

Find in Source Files

My App  
App Settings

Code

ContentView

MyApp

```
1 import SwiftUI
2
3 struct ContentView: View {
4     var body: some View {
5         VStack {
6             Text("Hello, world!")
7             Image(systemName: "globe")
8                 .imageScale(.large)
9                 .foregroundColor(.green)
10        }
11    }
12 }
13 }
14 }
```

App Preview

My App

Hello, world!

🌐

The screenshot shows the Xcode interface with the following details:

- Top Bar:** Standard Xcode icons for file, edit, search, and navigation.
- Search Bar:** "Find in Source Files" search bar.
- Sidebar:** "My App" icon, "App Settings" link, "Code" dropdown menu with "ContentView" selected, and "MyApp" link.
- Editor:** A code editor window titled "ContentView". It contains the following Swift code:

```
1 import SwiftUI
2
3 func RandomPrompt() -> String{
4     let things = ["Three people you enjoyed talking to today", "Three things you
5         are grateful for today."]
6     return String(things.randomElement()!)
7 }
8 struct ContentView: View {
9     var body: some View {
10         VStack {
11             ...
12         }
13     }
14 }
```
- Compiler Error:** A red callout bubble at the bottom right of the code editor says "Cannot find 'GoodThingPrompt' in scope".
- Preview:** An "App Preview" window titled "My App" shows the output of the code: "Three things you are grateful for today."
- Bottom Bar:** Standard Xcode navigation icons.

```
import SwiftUI

func RandomPrompt() -> String{
    let things = ["Three people you enjoyed talking to today","Three things you are grateful for today."]
    return String(things.randomElement()!)
}

struct ContentView: View {
    @State var GoodThingPrompt: String = RandomPrompt()
    var body: some View {
        VStack {
            Text(GoodThingPrompt)
        }
    }
}
```

Random list function

Variable monitored in the view

Text

What else could you make with this code?

## Magic 8 Ball

## Random Name Generator

```
import SwiftUI

func RandomPrompt() -> String{
    let things = ["Three people you
enjoyed talking to today","Three things
you are grateful for today."]
    return String(things.randomElement()!)
}

struct ContentView: View {
    @State var GoodThingPrompt: String =
RandomPrompt()
    var body: some View {
        VStack {
            Text(GoodThingPrompt)

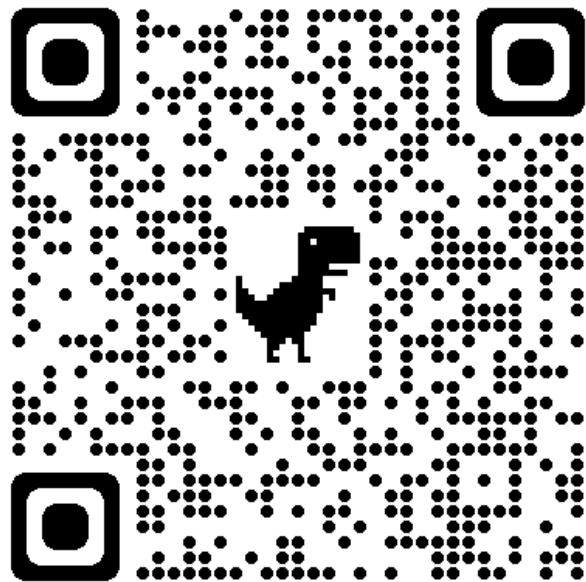
        }
    }
}
```

## Dice

## Story Prompts

# Step 2

Files at: <https://github.com/jamesabela/HyggeApp>



# TextFields

1. We need to declare a text string for each item

```
@State var GoodThing1: String = ""
```

2. We need to add textfields

```
TextField("1st", text: $GoodThing1)
```

Prompt text

Variable

Three moments that made you smile today

1st

2nd

3rd



```
import SwiftUI

func RandomPrompt() -> String{
    let things = ["Three people you enjoyed talking to today",
    "Three things you are grateful for today", "Three moments that
    made you smile today", "Three best things that have happened
    today"]
    return String(things.randomElement()!)
}

struct ContentView: View {
    @State var GoodThing1: String = ""
    @State var GoodThing2: String = ""
    @State var GoodThing3: String = ""
    @State var GoodThingPrompt: String = RandomPrompt()

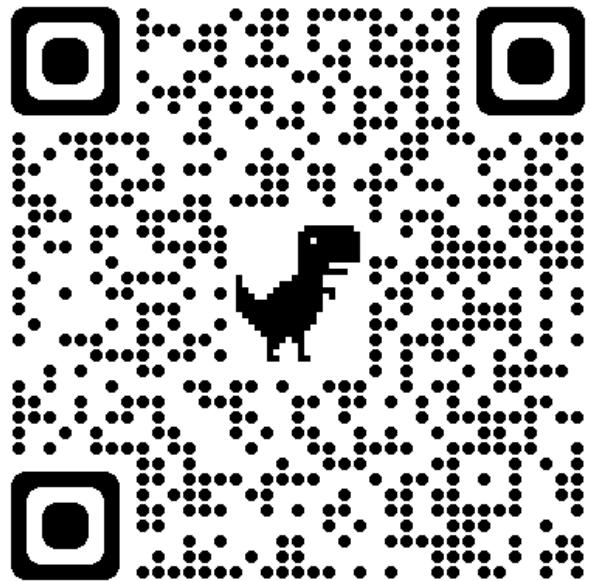
    var body: some View {
        VStack {
            Text(GoodThingPrompt)
            TextField("1st", text: $GoodThing1)
            TextField("2nd", text: $GoodThing2)
            TextField("3rd", text: $GoodThing3)
        }
    }
}
```

Three moments that made you smile today

1st  
2nd  
3rd

# Step 3

Files at: <https://github.com/jamesabela/HyggeApp>

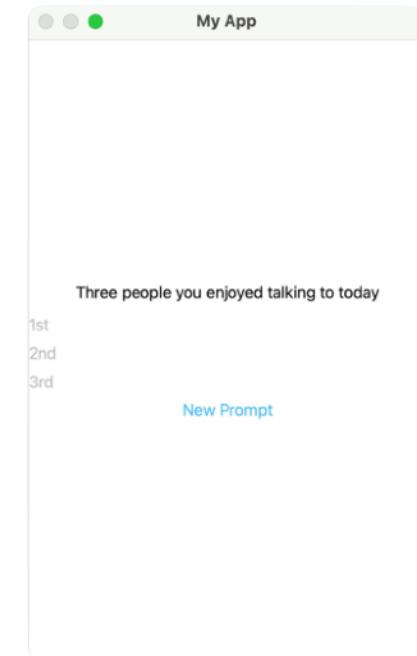


# Add a button

Button Prompt

```
Button("New Prompt"){\n    GoodThingPrompt = RandomPrompt()\n}
```

Code to call  
function

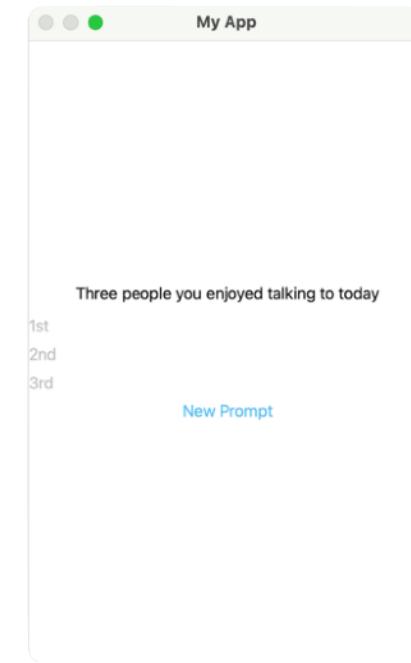


```
import SwiftUI
```

```
func RandomPrompt() -> String{
    let things = ["Three people you enjoyed talking to today",
    "Three things you are grateful for today", "Three moments that made
    you smile today", "Three best things that have happened today"]
    return String(things.randomElement()!)
}
```

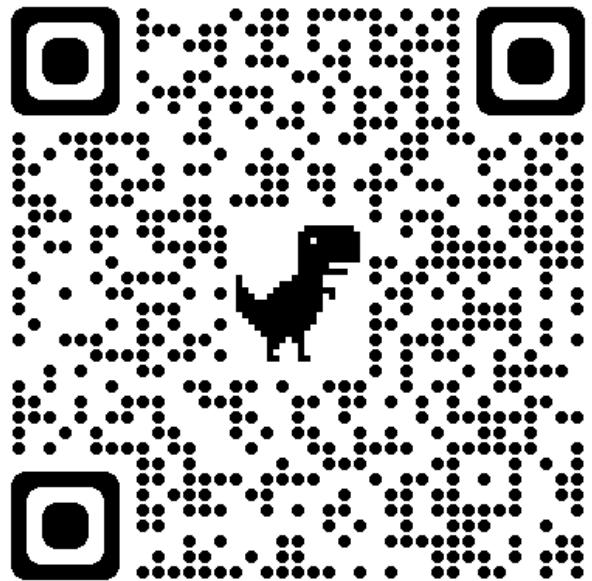
```
struct ContentView: View {
    @State var GoodThing1: String = ""
    @State var GoodThing2: String = ""
    @State var GoodThing3: String = ""
    @State var GoodThingPrompt: String = RandomPrompt()

    var body: some View {
        VStack {
            Text(GoodThingPrompt)
            TextField("1st", text: $GoodThing1)
            TextField("2nd", text: $GoodThing2)
            TextField("3rd", text: $GoodThing3)
            Button("New Prompt") {
                GoodThingPrompt = RandomPrompt()
            }
        }
    }
}
```

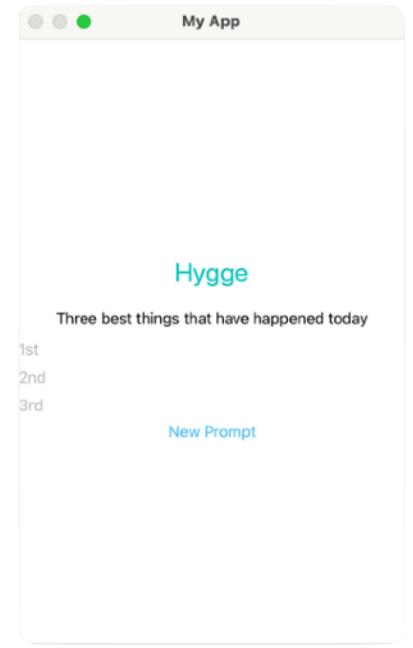
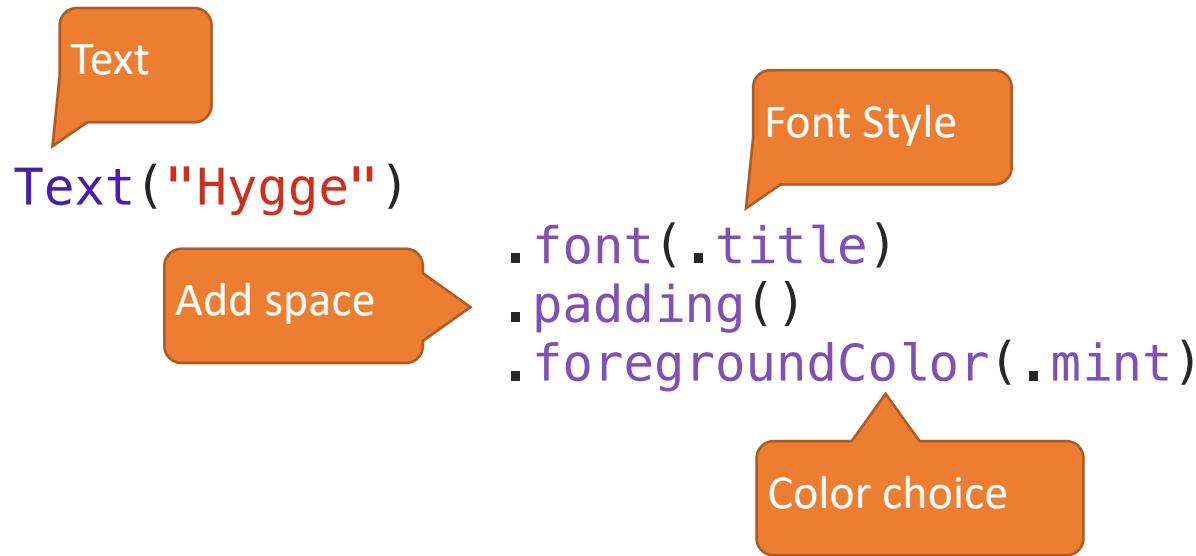


# Step 4

Files at: <https://github.com/jamesabela/HyggeApp>



# Add a Title



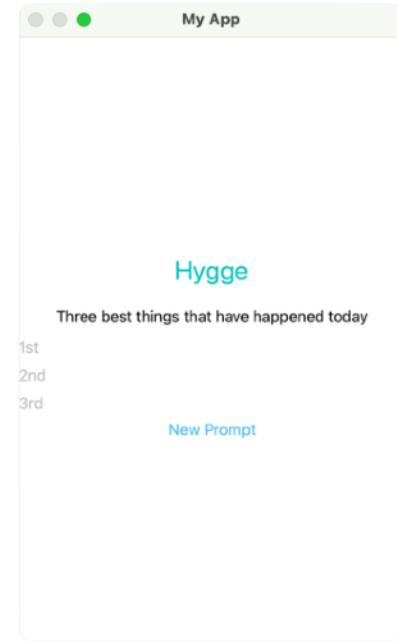
```
import SwiftUI

func RandomPrompt() -> String{
    let things = ["Three people you enjoyed talking to today", "Three things you are
grateful for today", "Three moments that made you smile today", "Three best things that have
happened today"]
    return String(things.randomElement()!)
}
```

```
struct ContentView: View {
    @State var GoodThing1: String = ""
    @State var GoodThing2: String = ""
    @State var GoodThing3: String = ""
    @State var GoodThingPrompt: String = RandomPrompt()

    var body: some View {
        VStack {
            Text("Hygge")
                .font(.title)
                .padding()
                .foregroundColor(.mint)

            Text(GoodThingPrompt)
            TextField("1st", text: $GoodThing1)
            TextField("2nd", text: $GoodThing2)
            TextField("3rd", text: $GoodThing3)
            Button("New Prompt"){
                GoodThingPrompt = RandomPrompt()
            }
        }
    }
}
```



# Font Choices

```
.font(.largeTitle)  
.font(.title)  
.font(.title2)  
.font(.title3)  
.font(.headline)  
.font(.subheadline)  
.font(.body)  
.font(.callout)  
.font(.caption)  
.font(.caption2)
```

```
.bold()
```

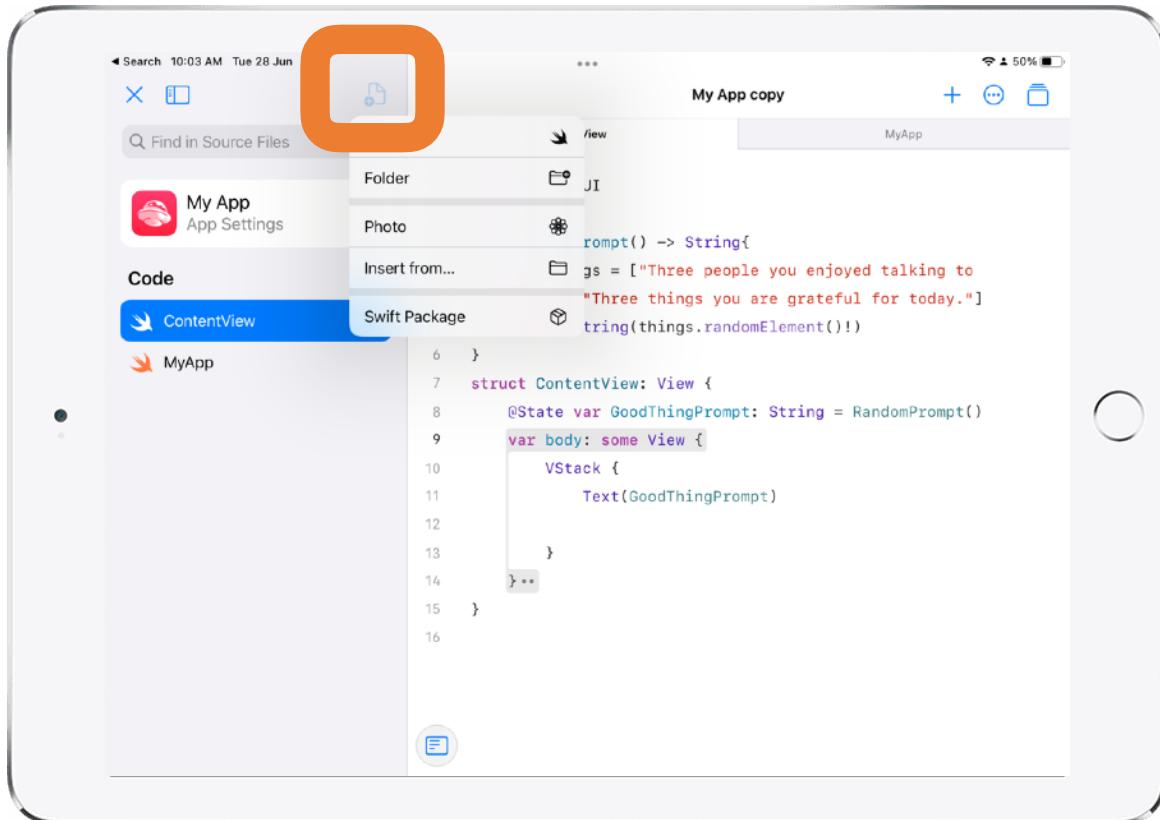
```
.italic()
```

```
.font(.system(.caption, design: .monospaced))
```

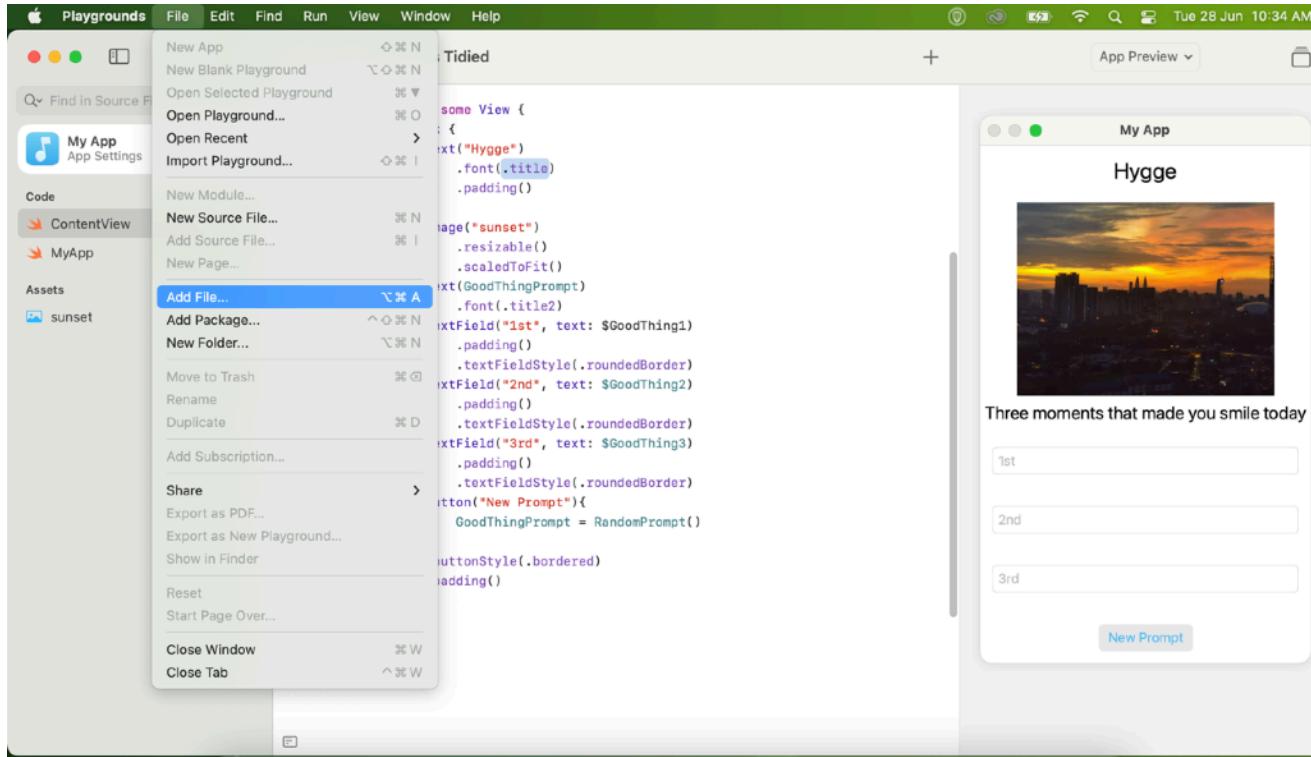
# Colors

V accentColor	V indigo
V black	V mint
V blue	V orange
V brown	V pink
V clear	V primary
V cyan	V purple
V gray	V red
V green	V secondary
V teal	V white
V yellow	

# Add an image (iPad)

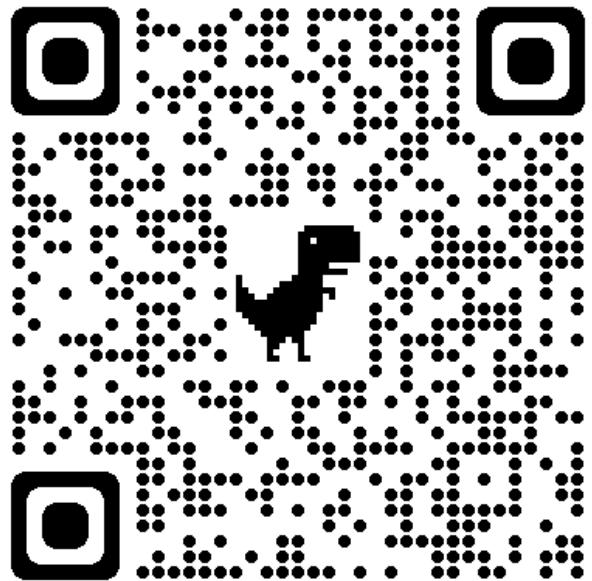


# Add an image (Mac OS)



# Step 5

Files at: <https://github.com/jamesabela/HyggeApp>



# Code for image

```
Image("sunset")  
    .resizable()  
    .scaledToFit()
```

Image file name

Can change size

Fits in size given

```
import SwiftUI

func RandomPrompt() -> String{
    let things = ["Three people you enjoyed talking to today", "Three things you are grateful for today", "Three moments that made you smile today", "Three best things that have happened today"]
    return String(things.randomElement()!)
}

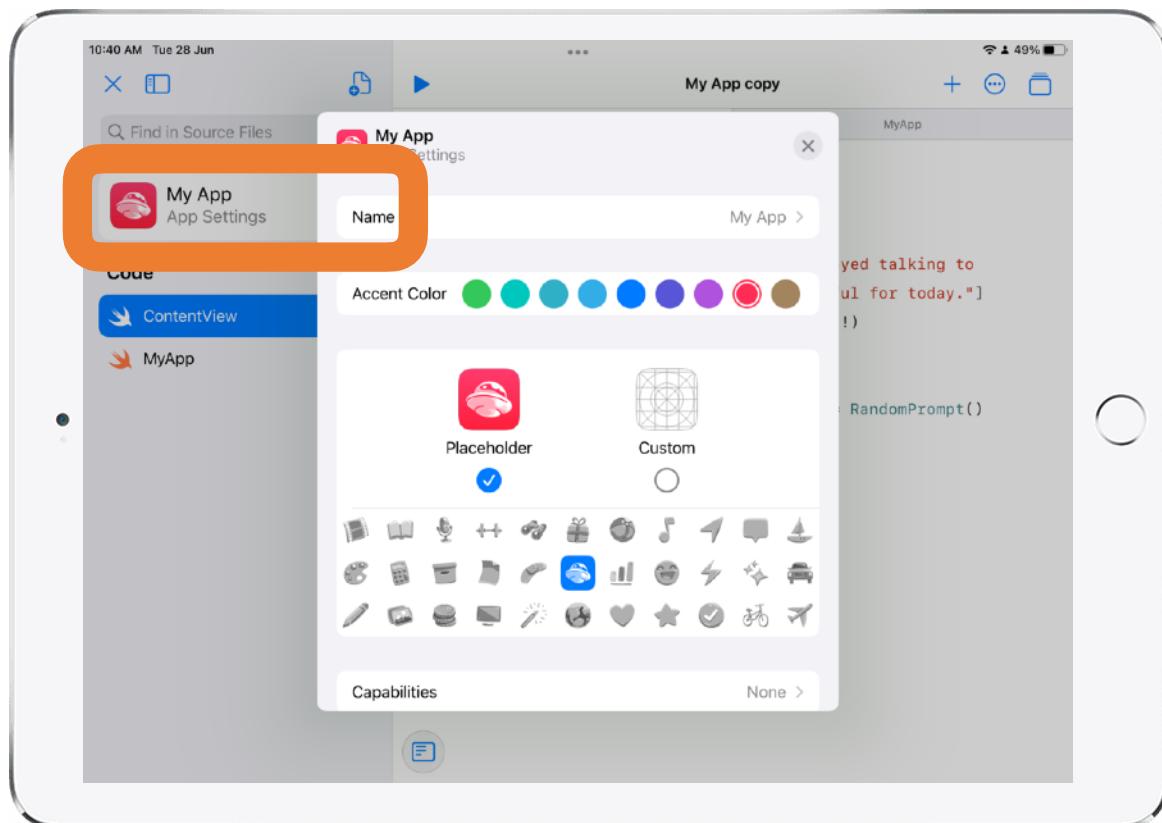
struct ContentView: View {
    @State var GoodThing1: String = ""
    @State var GoodThing2: String = ""
    @State var GoodThing3: String = ""
    @State var GoodThingPrompt: String = RandomPrompt()

    var body: some View {
        VStack {
            Text("Hygge")
                .font(/*@START_MENU_TOKEN@*/.title/*@END_MENU_TOKEN@*/)
                .padding()

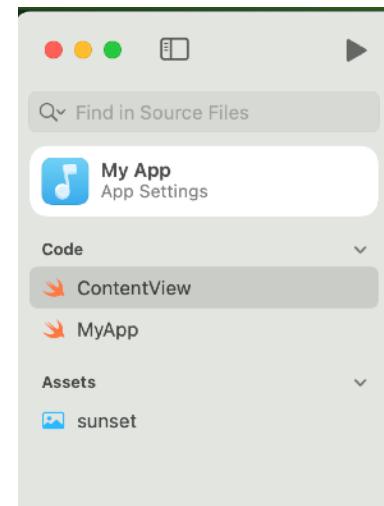
            Image("sunset")
                .resizable()
                .scaledToFit()

            Text(GoodThingPrompt)
                .font(.title2)
            TextField("1st", text: $GoodThing1)
                .padding()
                .textFieldStyle(.roundedBorder)
            TextField("2nd", text: $GoodThing2)
                .padding()
                .textFieldStyle(.roundedBorder)
            TextField("3rd", text: $GoodThing3)
                .padding()
                .textFieldStyle(.roundedBorder)
            Button("New Prompt"){
                GoodThingPrompt = RandomPrompt()
            }
            .buttonStyle(.bordered)
            .padding()
        }
    }
}
```

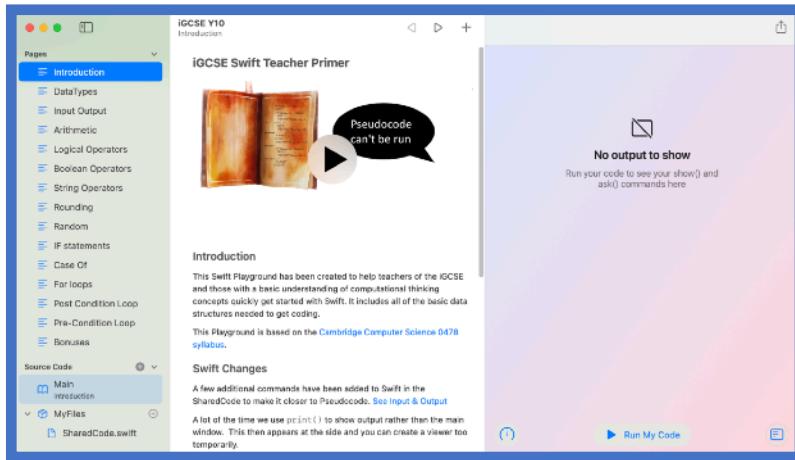
# Name your app & make an icon



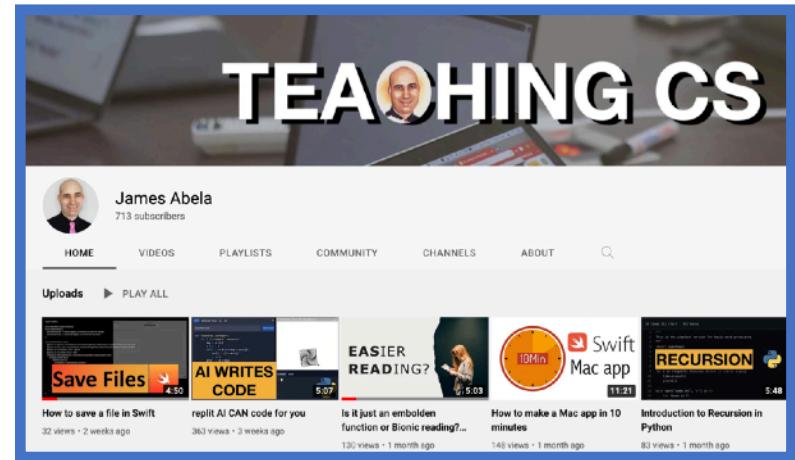
Mac OS



# Next Steps



<https://bit.ly/swiftiGCSE>



<https://www.youtube.com/c/JamesAbela>

Too easy? <https://www.hackingwithswift.com/100>