Welcome to iOS Bootcamp

Hosted by App Team Carolina

Agenda

What can you expect this meeting?

1) Introductions
2) Intro to SwiftUI
3) Practice
4) Instagram Demo

Attendance

Please fill this out!

Disclaimer

Hardware Requirement

In order to participate, you must have access to a Mac laptop

Xcode, the IDE used to make iOS apps, is macOS-only

Come chat afterwards if you don’t currently have such a device

Intro to iOS Bootcamp

What is iOS Bootcamp?

And why should you participate?

1) Unlike traditional bootcamps, 100% free!
2) Bootcamp will teach you everything you need to know to create an app
3) Prepares you to join iOS Apprenticeship or a Production team

The Team

Who will be helping you this semester?

Alexandra Marum, Co-Lead
CS & Philosophy
Joined Fall 2023
Senior

Hussain Hassan, Co-Lead
CS & Mathematics
Joined Fall 2024
Sophomore

Tri Nguyen, LA
CS & Data Science
Joined Fall 2024
Junior

Alex Yang, LA
CS & Business Admin
Joined Fall 2024
Sophomore

Learning Philosophy

How do we approach learning?

1) This is not a class
2) The best teacher might be just ahead of you
3) Goofing around vs. winning championships
4) Safe, inclusive, and effective learning

Learning Philosophy

How do we approach learning?

1) **This is not a class**
   You’re here because you want to learn – not to graduate.
2) The best teacher might be just ahead of you
3) Goofing around vs. winning championships
4) Safe, inclusive, and effective learning

Learning Philosophy

How do we approach learning?

1) This is not a class
2) **The best teacher might be just ahead of you**
   We remember what it's like to not know iOS development.
3) Goofing around vs. winning championships
4) Safe, inclusive, and effective learning

Learning Philosophy

How do we approach learning?

1) This is not a class
2) The best teacher might be just ahead of you
3) **Goofing around vs. winning championships**
   There’s room for both here – but one is more rewarding
4) Safe, inclusive, and effective learning

Learning Philosophy

How do we approach learning?

1) This is not a class
2) The best teacher might be just ahead of you
3) Goofing around vs. winning championships
4) **Safe, inclusive, and effective learning**
   You can’t learn if you’re not comfortable. Let me know if you’re ever not.

Curriculum Overview

What will you learn?

1. Coding with Swift + SwiftUI
2. Building static user interfaces
3. Handling user input
4. Navigation and complex views
5. Networking with APIs

The Structure

How will you learn?

Meetings ~ 90 min.

- Icebreakers / announcements
- New content
- Coding practice
- Demos
- Resources

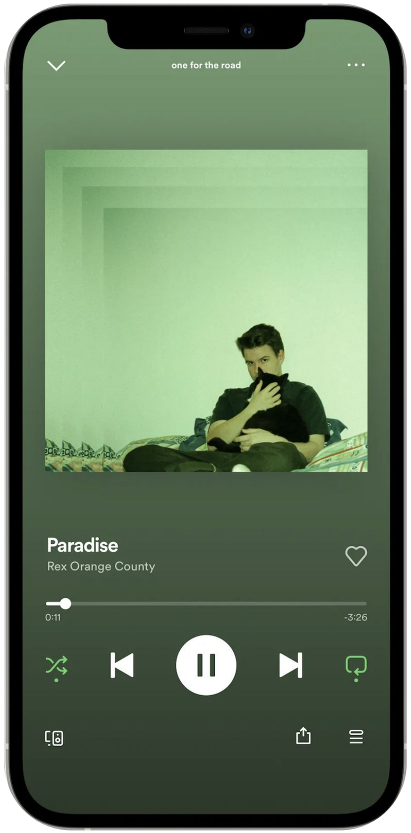
View meetings here!

The Structure

How will you learn?

Projects ~ 2 hrs.

- Small apps
- Completed between meetings
- Practice the week's concepts
- Build portfolio!!!



Spotify UI
Project 1

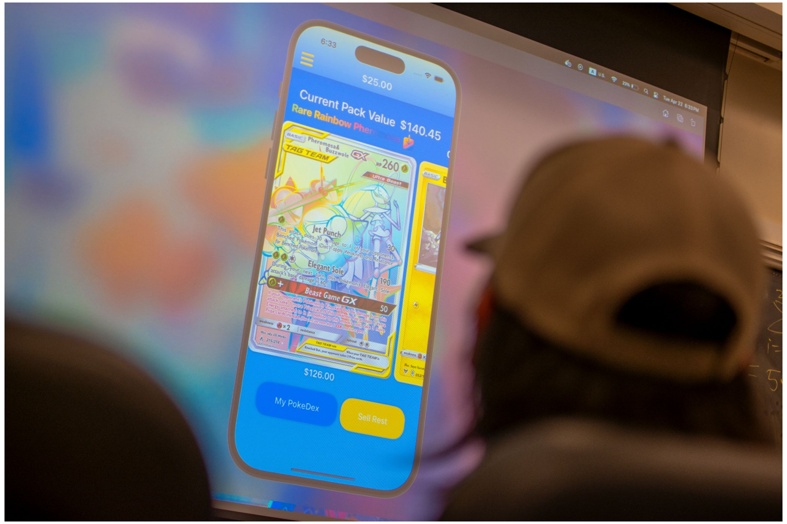
Our Expectations
“Graduation” requirements

Participate

- Complete at least **6 projects**
- Attend at least **10 meetings**

Present a final project

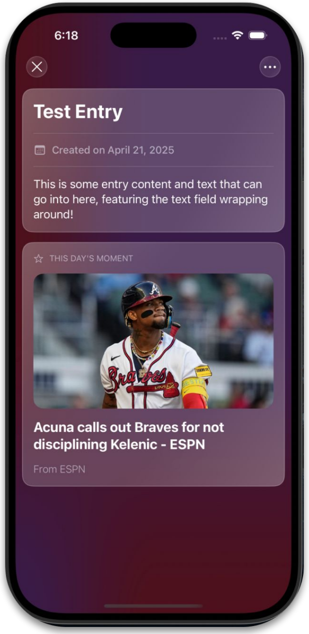
- **December 2nd**
- Show what you’ve learned



PokeGambe
Ian Forlemu

What Can You Expect?

A few final projects from spring…


6:18

Test Entry

Created on April 21, 2025

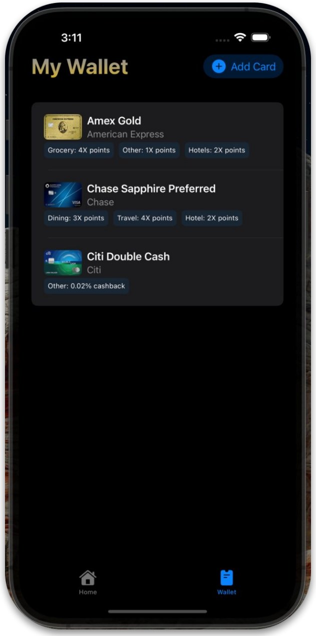
This is some entry content and text that can go into here, featuring the text field wrapping around!

THIS DAY'S MOMENT

Acuna calls out Braves for not disciplining Kelenic - ESPN

From ESPN

Moment
Connor Ruesch


3:11

My Wallet

Add Card

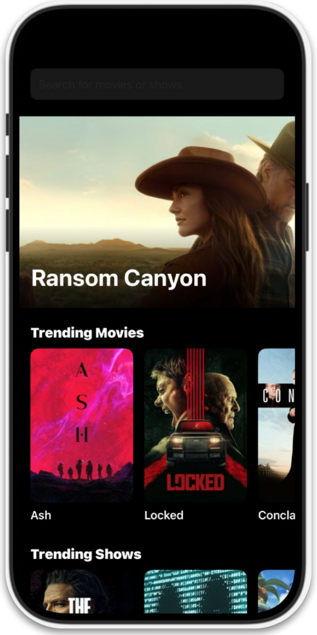
Amex Gold
American Express
Grocery: 4X points | Other: 1X points | Hotels: 2X points

Chase Sapphire Preferred
Chase
Dining: 3X points | Travel: 4X points | Hotel: 2X points

Citi Double Cash
Citi
Other: 0.02% cashback

Home | Wallet

Rewarded
Ethan Ahdout

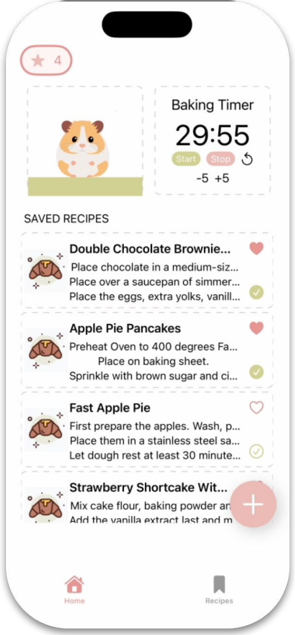

Ransom Canyon

Trending Movies

Ash | Locked | Concla

Trending Shows

MovieDB
Maxwell Hu


4

Baking Timer

29:55

Start | Stop

-5 +5

SAVED RECIPES

Double Chocolate Brownie...
Place chocolate in a medium-siz...
Place over a saucepan of simmer...
Place the eggs, extra yolks, vanill...

Apple Pie Pancakes
Preheat Oven to 400 degrees Fa...
Place on baking sheet.
Sprinkle with brown sugar and ci...

Fast Apple Pie
First prepare the apples. Wash, p...
Place them in a stainless steel sa...
Let dough rest at least 30 minute...

Strawberry Shortcake Wit...
Mix cake flour, baking powder an
Add the vanilla extract last and m

Home | Recipes

Bakemate
Olivia Kirby

Intro to SwiftUI

What is SwiftUI?

Overview

The tool we use to build iOS apps

- Displays views on screen
- Handle user interaction
- Manage state and data flow

User Interface Framework

- Swift != SwiftUI

What is SwiftUI?

Declarative Syntax

Declarative describes *what*

- Details handled under the hood

Imperative describes *how*

- UIKit, JavaFX, Android Native

Declarative is a popular UI paradigm

- React, SwiftUI, JetpackCompose (Android)

What is SwiftUI?

Simple list in SwiftUI

```swift
struct DeclarativeListView: View {
    private let data = ["Item 1", "Item 2", "Item 3"]

    var body: some View {
        List(data, id: \.self) { item in
            Text(item)
        }
    }
}
```

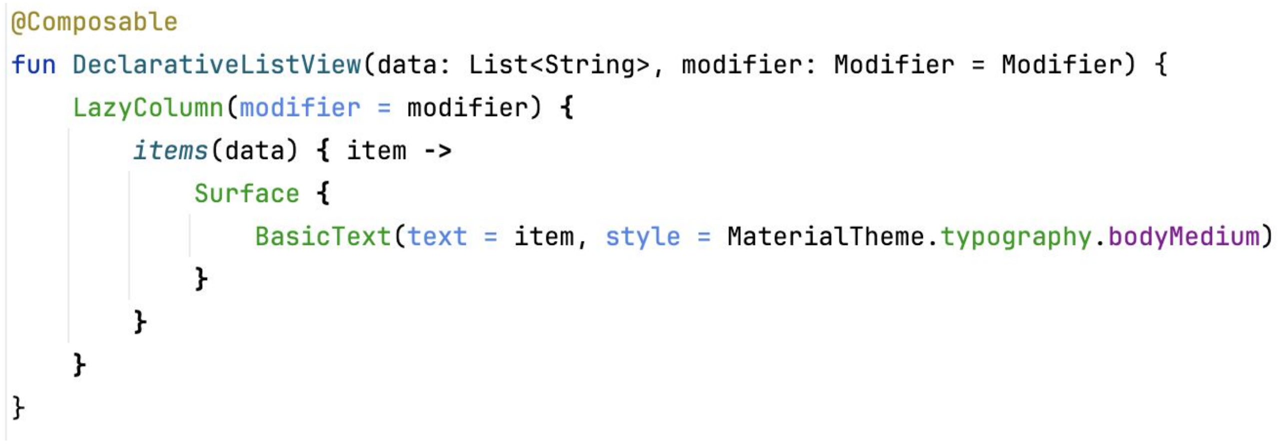
What is SwiftUI?

Simple list in UIKit

```swift
class ImperativeViewController: UIViewController, UITableViewDataSource, UITableViewDelegate {
    private let tableView = UITableView()
    private var data = ["Item 1", "Item 2", "Item 3"]

    override func viewDidLoad() {
        super.viewDidLoad()

        tableView.dataSource = self
        tableView.delegate = self
        tableView.frame = view.bounds
        view.addSubview(tableView)

        tableView.register(UITableViewCell.self, forCellReuseIdentifier: "Cell")
    }

    func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {
        return data.count
    }

    func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {
        let cell = tableView.dequeueReusableCell(withIdentifier: "Cell", for: indexPath)
        cell.textLabel?.text = data[indexPath.row]
        return cell
    }
}
```

Declarative Frameworks

Simple list in JetpackCompose



```
@Composable
fun DeclarativeListView(data: List<String>, modifier: Modifier = Modifier) {
    LazyColumn(modifier = modifier) {
        items(data) { item ->
            Surface {
                BasicText(text = item, style = MaterialTheme.typography.bodyMedium)
            }
        }
    }
}
```

Pretty similar to SwiftUI, right?

Declarative programming is a transferable skill

SwiftUI Basics

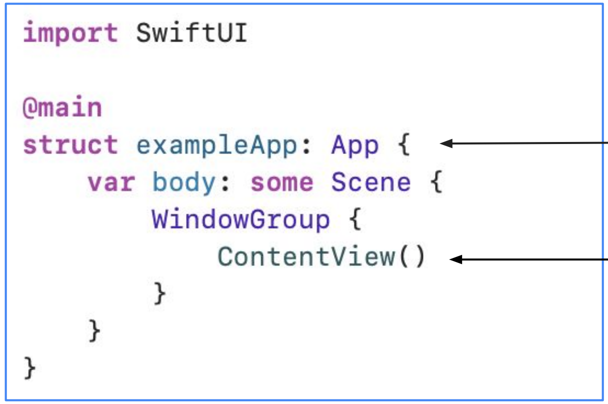
SwiftUI Basics

Example project

Your project starts with two files:

- exampleApp.swift (top)
- ContentView.swift (bottom)

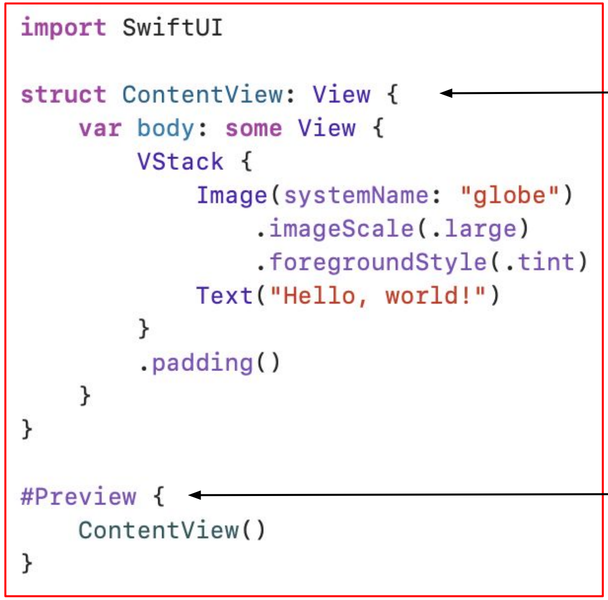
You won't need to change the App file this semester



```swift
import SwiftUI

@main
struct exampleApp: App {
    var body: some Scene {
        WindowGroup {
            ContentView()
        }
    }
}
```

App: entry-point

WindowGroup: View hierarchy container



```swift
import SwiftUI

struct ContentView: View {
    var body: some View {
        VStack {
            Image(systemName: "globe")
                .imageScale(.large)
                .foregroundStyle(.tint)
            Text("Hello, world!")
        }
        .padding()
    }
}

#Preview {
    ContentView()
}
```

View: UI component

Preview: Displays View while you work

SwiftUI Basics
Views

Building blocks of your app's UI

- Display content on screen

Broadly, there are two types:

- View *elements*: Visible content
- View *containers*: Manage subview layout

SwiftUI provides some essential Views to you

```swift
import SwiftUI

struct ContentView: View {
    var body: some View {
        VStack {
            Image(systemName: "globe")
                .imageScale(.large)
                .foregroundStyle(.tint)
            Text("Hello, world!")
        }
        .padding()
    }
}

#Preview {
    ContentView()
}
```

SwiftUI Basics

View Elements

Text: Displays text that you pass in

```swift
Text("text you'd like displayed")
```

Image: Displays custom images from the project Assets

```swift
Image("fileNameInAssets")
```

SF Symbols: Huge icon set built into SwiftUI. Accessed using systemName

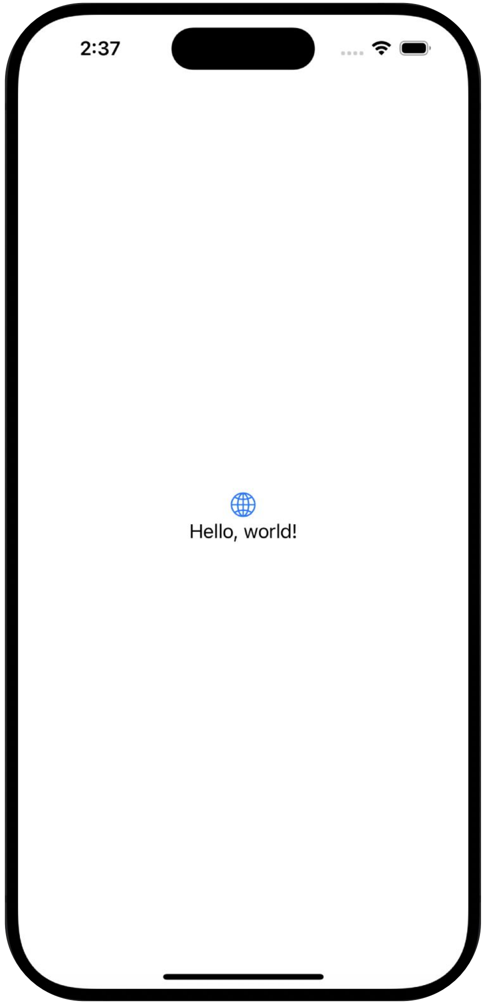
```swift
Image(systemName: "sun.max.fill")
```

→ ☀

SwiftUI Basics
VStack

```
VStack {
    Image(systemName: "globe")
        .imageScale(.large)
        .foregroundStyle(.tint)
    Text("Hello, world!")
}
```

$\rightarrow$



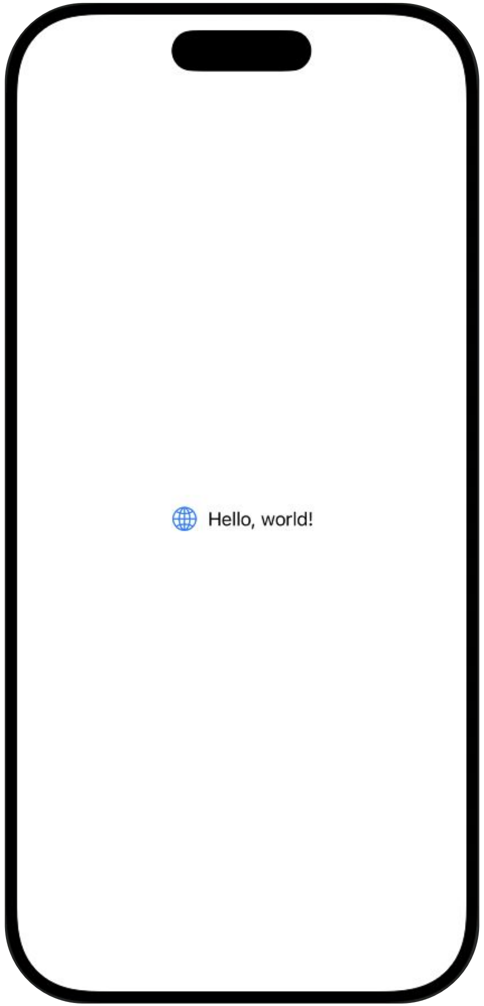
Arranges subviews **vertically**.

SwiftUI Basics

HStack

```swift
HStack {
    Image(systemName: "globe")
        .imageScale(.large)
        .foregroundStyle(.tint)
    Text("Hello, world!")
}
```

$\longrightarrow$



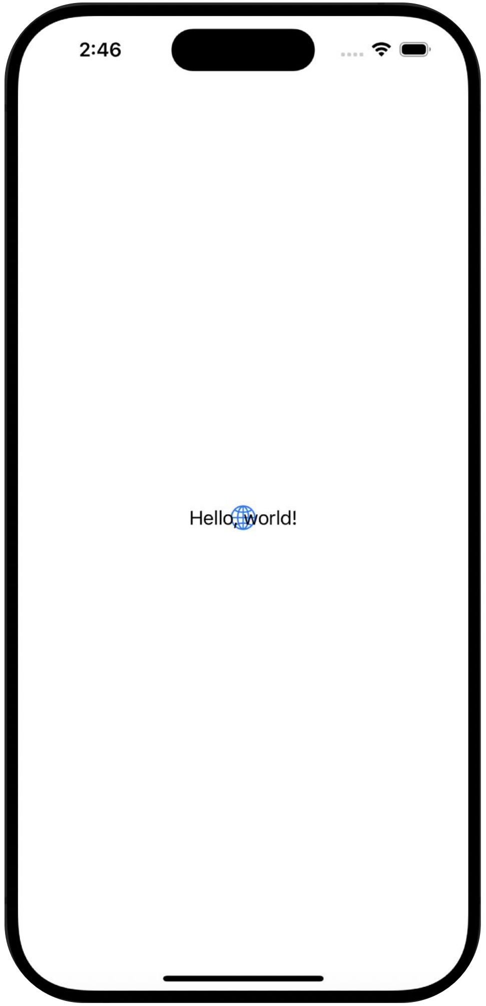
Arranges subviews **horizontally**.

SwiftUI Basics

ZStack

```
ZStack {
    Image(systemName: "globe")
        .imageScale(.large)
        .foregroundStyle(.tint)
    Text("Hello, world!")
}
```

$\longrightarrow$



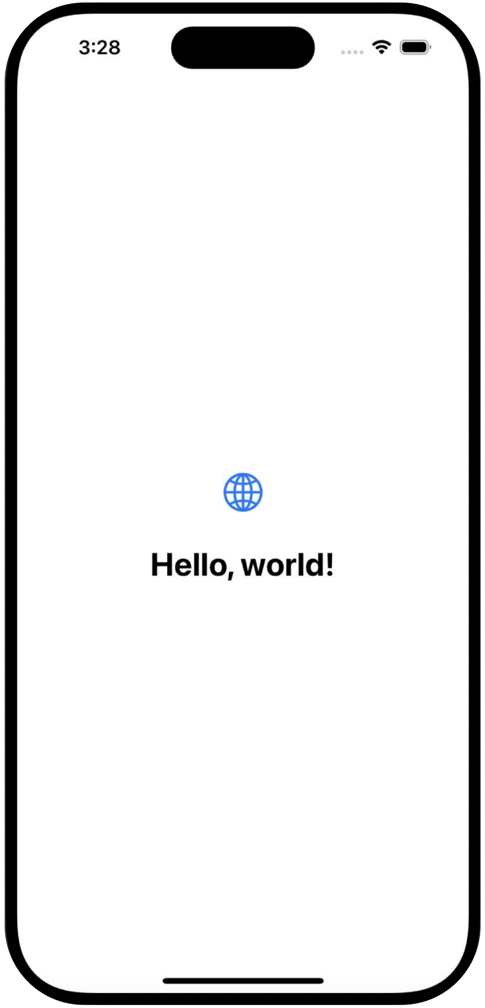
Arranges subviews **along the Z-axis**.

SwiftUI Basics

View Modifiers

```swift
VStack {
    Image(systemName: "globe")
        .imageScale(.large)
        .foregroundStyle(.tint)
        .padding()
    Text("Hello, world!")
        .font(.title)
        .bold()
}
```

$\longrightarrow$



Methods applied to views that alter their appearance.

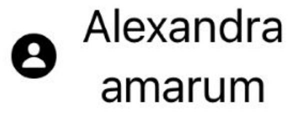
SwiftUI Basics

Layout

Container views can be **nested** to create more complex views.

```
HStack {
    Image(systemName: "person.circle.fill")
    VStack {
        Text("Alexandra")
        Text("amarum")
    }
}
```

$\longrightarrow$



SwiftUI Basics

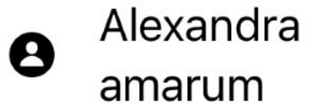
Layout

HStack and **VStack** each have two parameters: **alignment** and **spacing.**

They control… exactly those things

```
HStack(spacing: 15) {
    Image(systemName: "person.circle.fill")
    VStack(alignment: .leading) {
        Text("Alexandra")
        Text("amarum")
    }
}
```

$\longrightarrow$


Alexandra
amarum

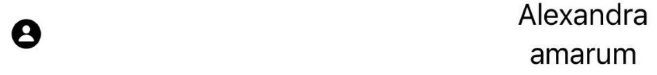
SwiftUI Basics

Layout

Spacers are empty view elements that are **greedy**. They take up as much space as possible.

```
HStack {
    Image(systemName: "person.circle.fill")

    Spacer()

    VStack {
        Text("Alexandra")
        Text("amarum")
    }
}
```

$\longrightarrow$



Practice creating a project

Return to Notion

Instagram recreation demo

Close your laptops and watch SwiftUI in action!