



# SwiftUI

Declarative. Powerful. Revolutionary.

# SwiftUI vs UIKit

Declarative

Imperative

Reactive

Event driven

Less Code

More Control

# UIKit vs SwiftUI



## UIKit

Storyboards and constraints



## SwiftUI

Casting UI spells with SwiftUI

SwiftUI: Where UI challenges disappear faster than you can say "Abracadabra!" 🎩✨

## Building blocks of SwiftUI

```
struct ContentView: View {  
    var body: some View {  
        Text("Hello, SwiftUI!")  
            .font(.title)  
            .foregroundColor(.blue)  
    }  
}
```

## Reactive UI with @State and @Binding

```
struct ContentView: View {  
    @State private var isOn = false  
  
    var body: some View {  
        Toggle("Switch", isOn: $isOn)  
    }  
}
```

# SwiftUI Layouts

- VStack
- HStack
- ZStack

# Dynamic Lists and Navigation

```
NavigationView {  
    List(items, id: \.self) { item in  
        NavigationLink(destination: DetailView(item:  
            item)) {  
            Text(item)  
        }  
    }  
}
```

# Rapid Development with Previews

```
#Preview {  
    TransactionListView()  
        .previewDevice("iPhone 15")  
        .preferredColorScheme(.dark)  
}  
  
#Preview {  
    TransactionListView()  
        .previewDevice("iPhone 15")  
        .preferredColorScheme(.light)  
}
```