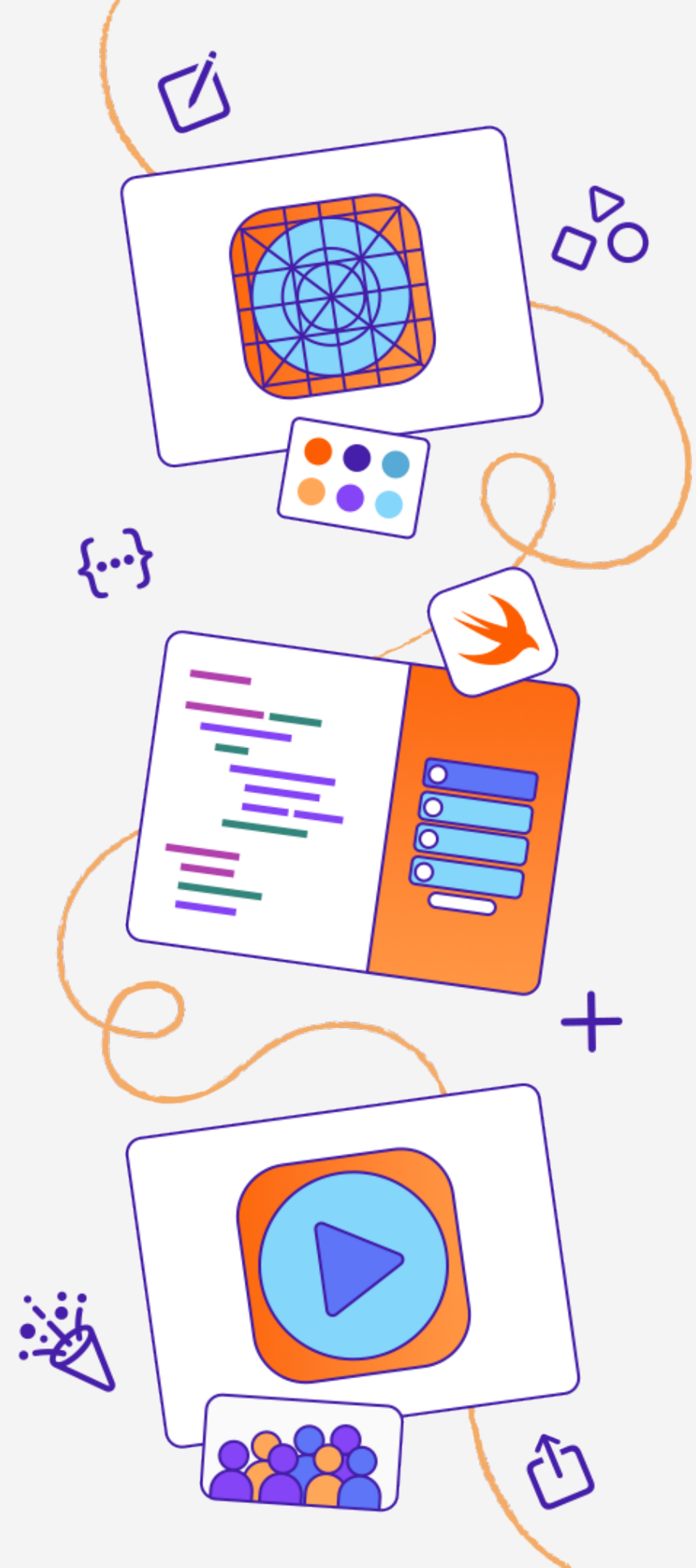
Everyone Can Code

Project Presentation



Build Custom Shapes

Everyone Can Code

This instructional presentation showcases iPadOS 16.3, Keynote for iPad version 13.0 and Swift Playgrounds for iPad version 4.3.
User interface elements may vary depending on the software version installed on your device.

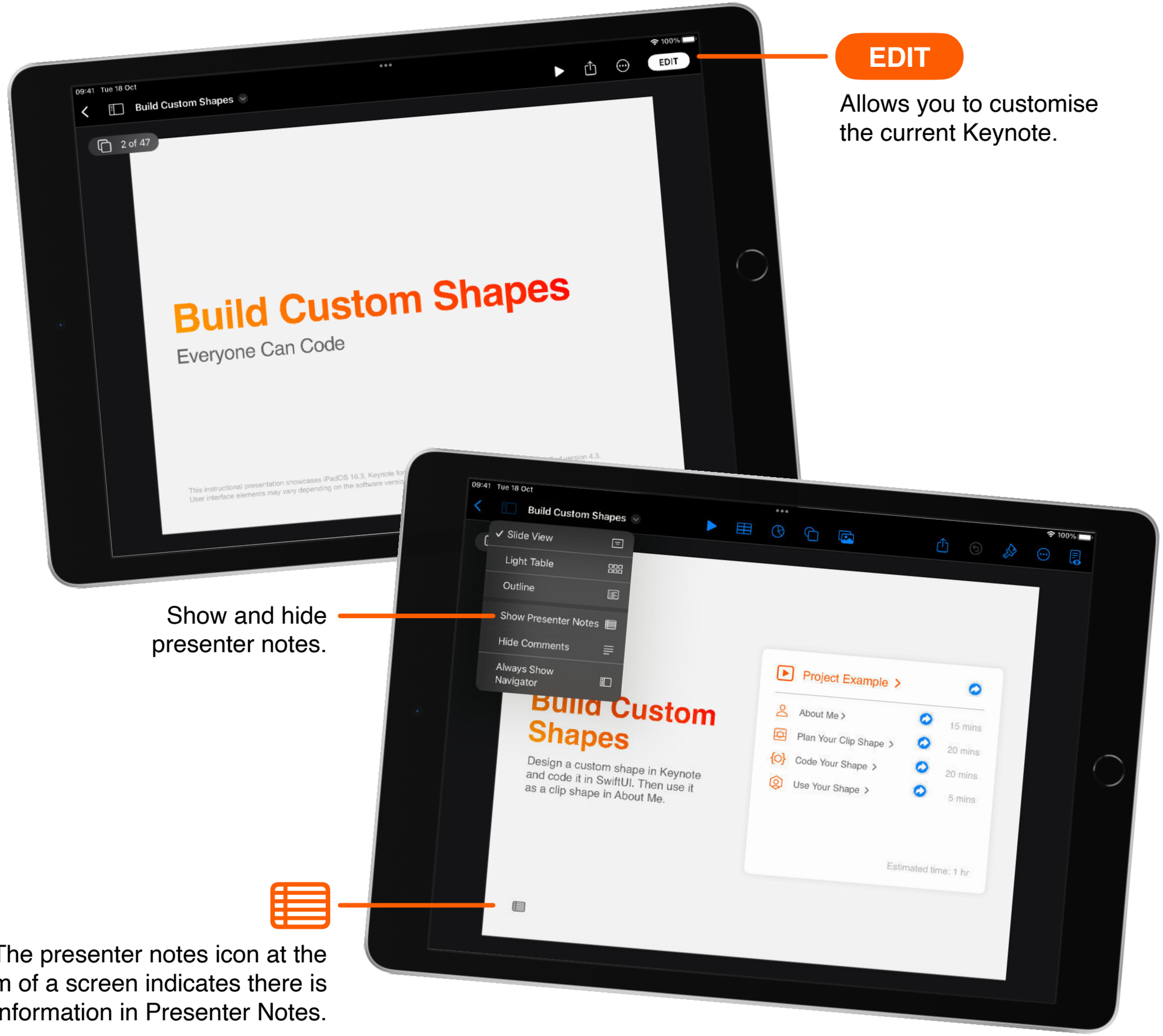
Build Custom Shapes

Design a custom shape in Keynote and code it in SwiftUI. Then use it as a clip shape in About Me.

Project Example >

About Me >	15 mins
Plan Your Clip Shape >	20 mins
Code Your Shape >	20 mins
Use Your Shape >	5 mins

Estimated time: 1 hr



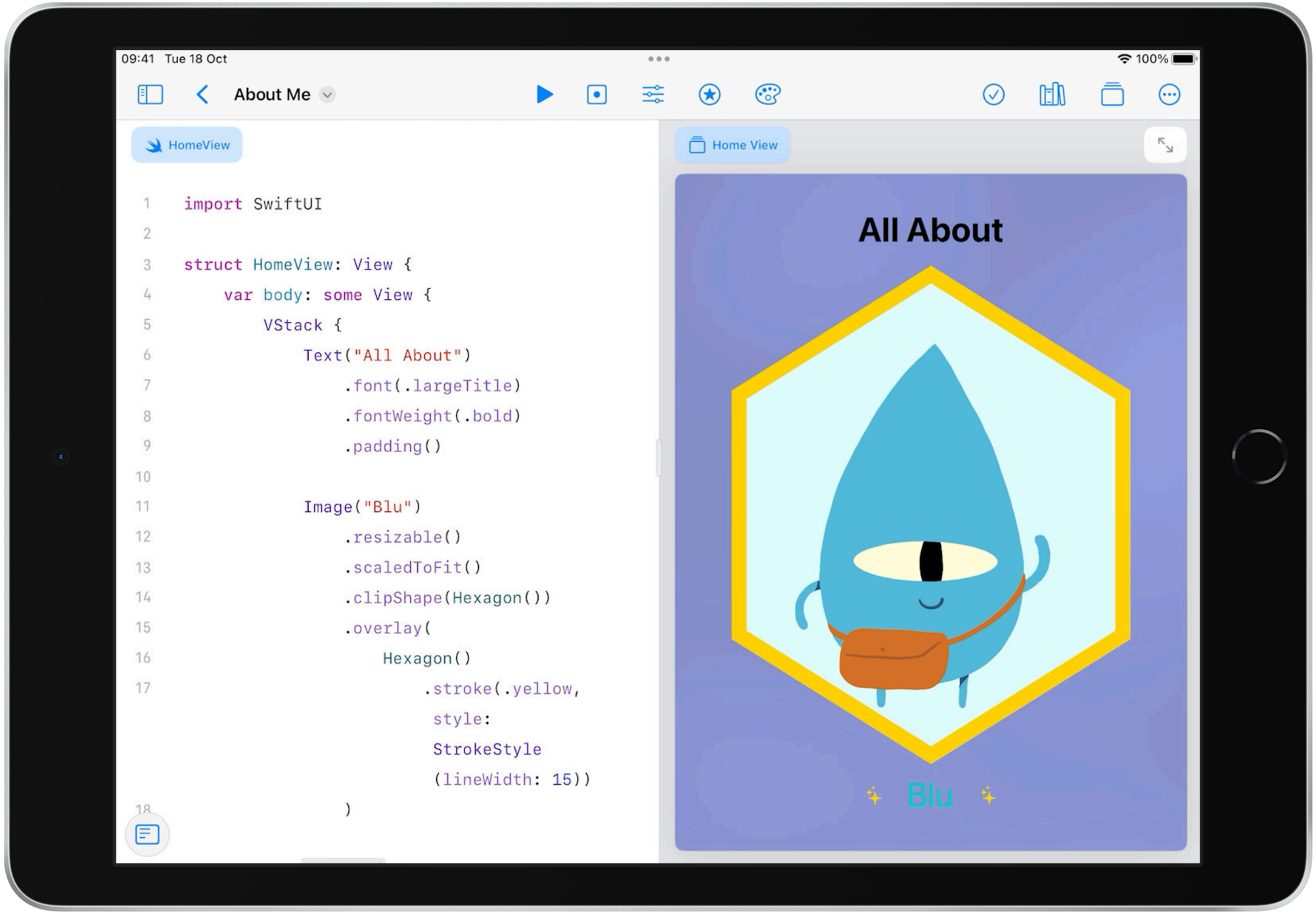
EDIT

Allows you to customise the current Keynote.

Swipe right from the left edge of the screen to show the slide navigator. Swipe left over the slide navigator to hide.

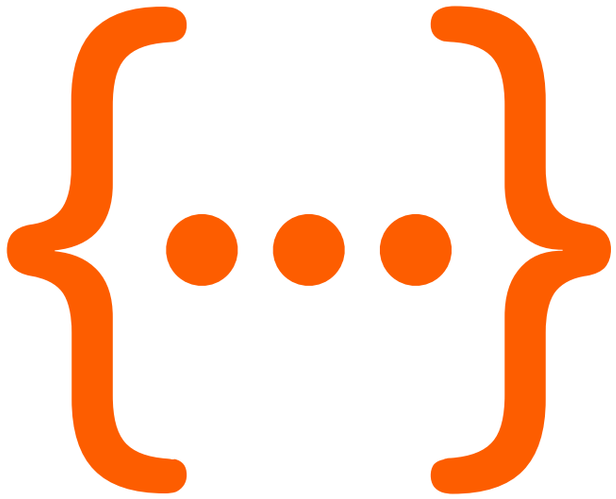
Show and hide presenter notes.

The presenter notes icon at the bottom of a screen indicates there is extra information in Presenter Notes.


09:41 Tue 18 Oct
100%
About Me
HomeView
Home View
import SwiftUI
struct HomeView: View {
var body: some View {
VStack {
Text("All About")
.font(.largeTitle)
.fontWeight(.bold)
.padding()
Image("Blu")
.resizable()
.scaledToFit()
.clipShape(Hexagon())
.overlay(
Hexagon()
.stroke(.yellow,
style:
StrokeStyle
(lineWidth: 15))
)
All About
Blu

What you'll learn:

- Customise the Home tab in About Me
- Plot the coordinates for a responsive, straight line, 2D shape
- Build a custom shape in SwiftUI

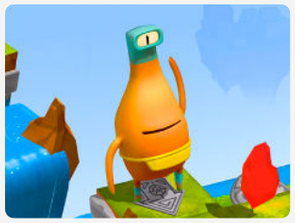


What you'll need:

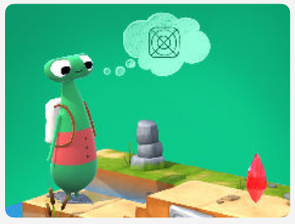
Keynote for iPad

Swift Playgrounds for iPad

Suggested prerequisites:



Get Started with Code >



Get Started with Apps >

Tutorial 1

About Me

Learn about the code in the Home tab,
then edit the code to make it your own.

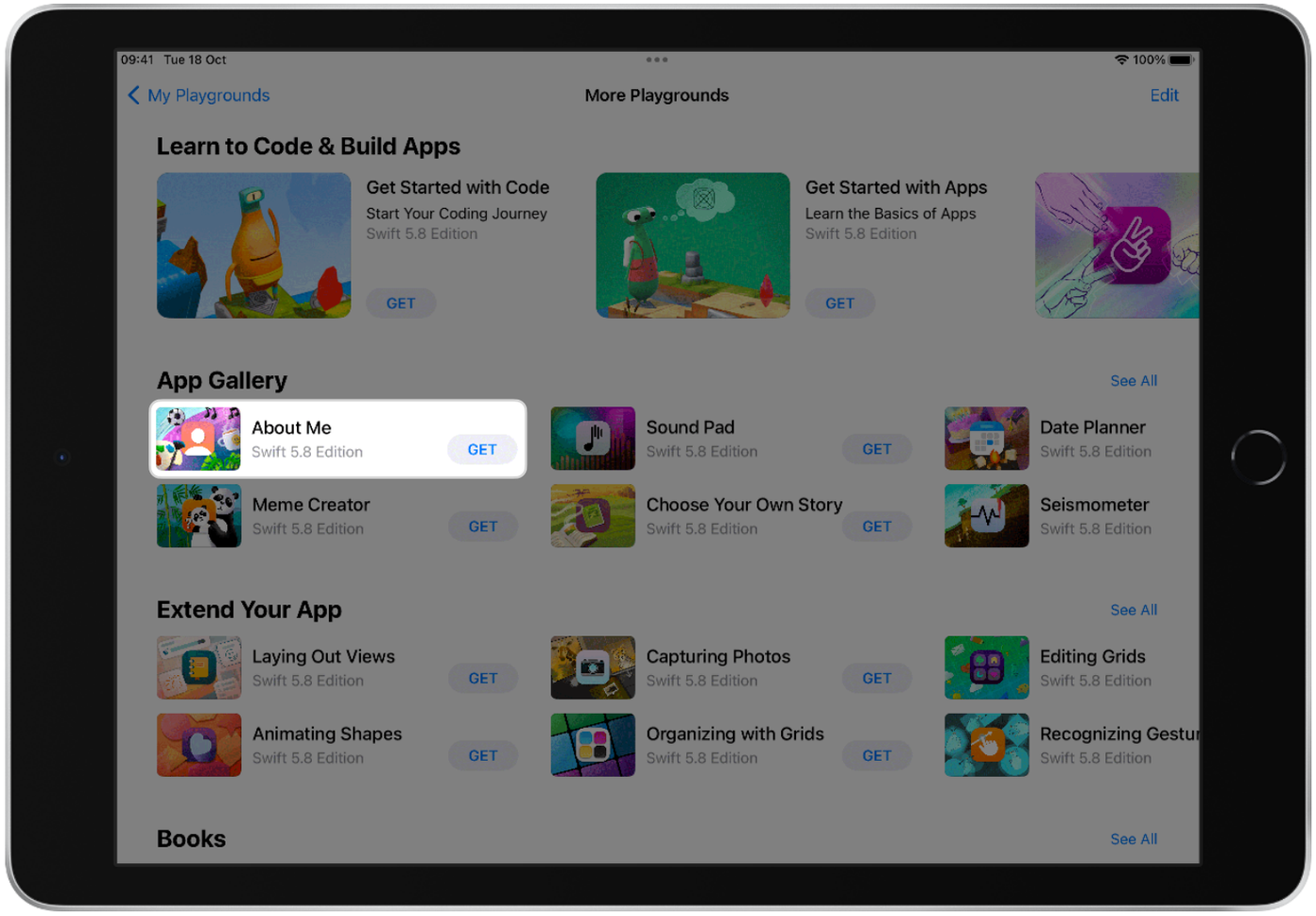
Estimated time:
15 mins

Step 1

Open Swift Playgrounds.

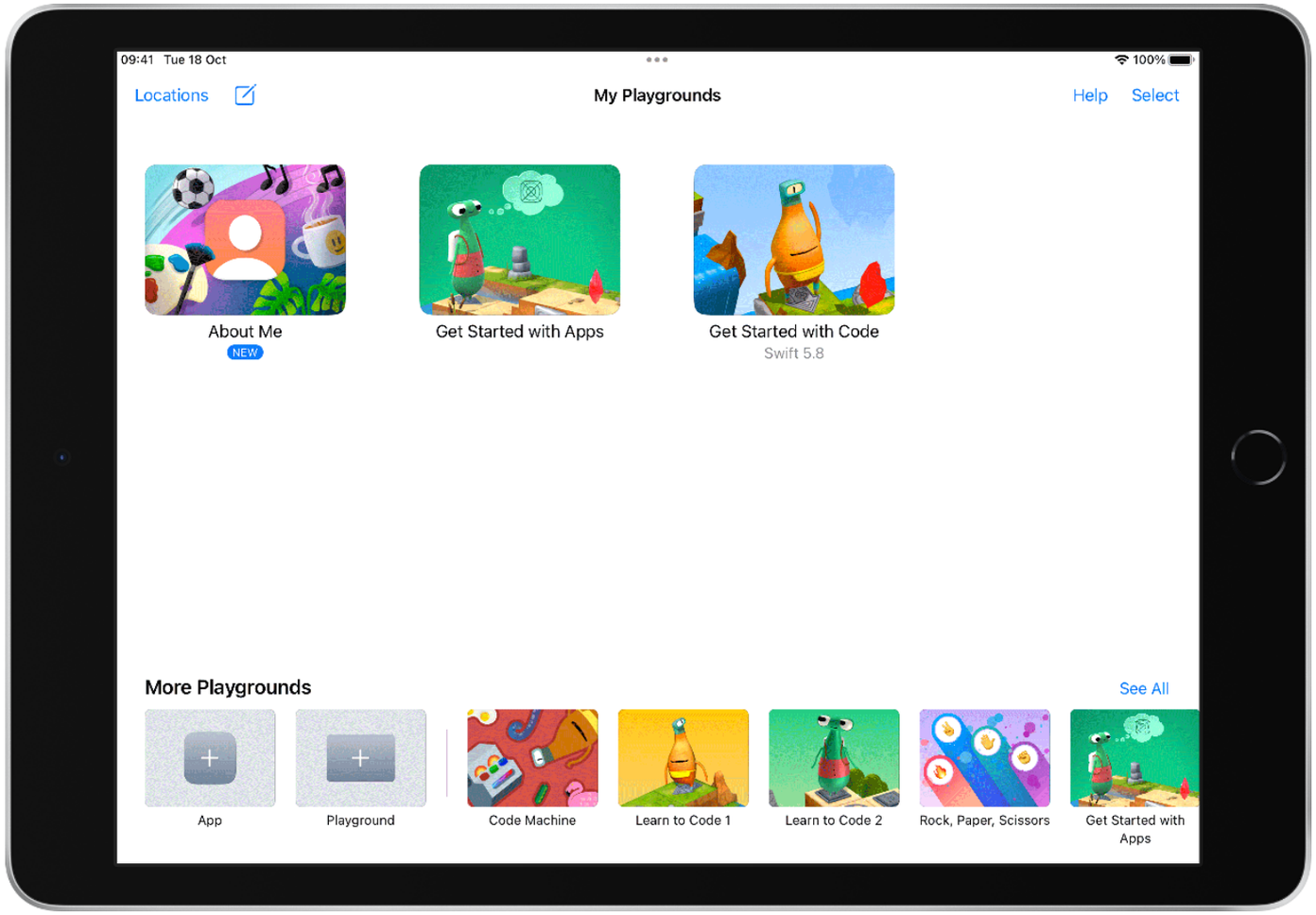
Step 2

Get About Me >



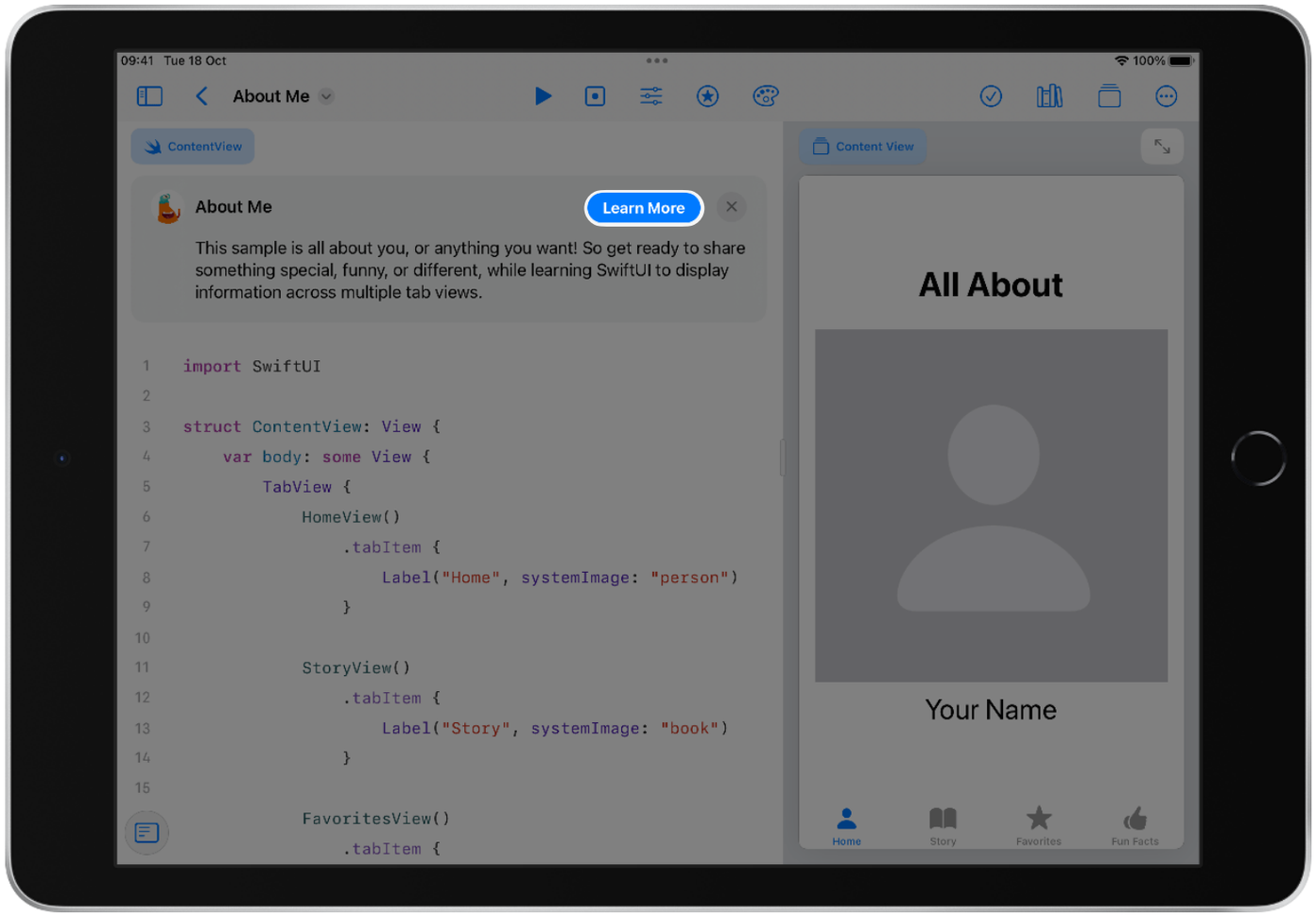
Step 3

Tap About Me to open it.



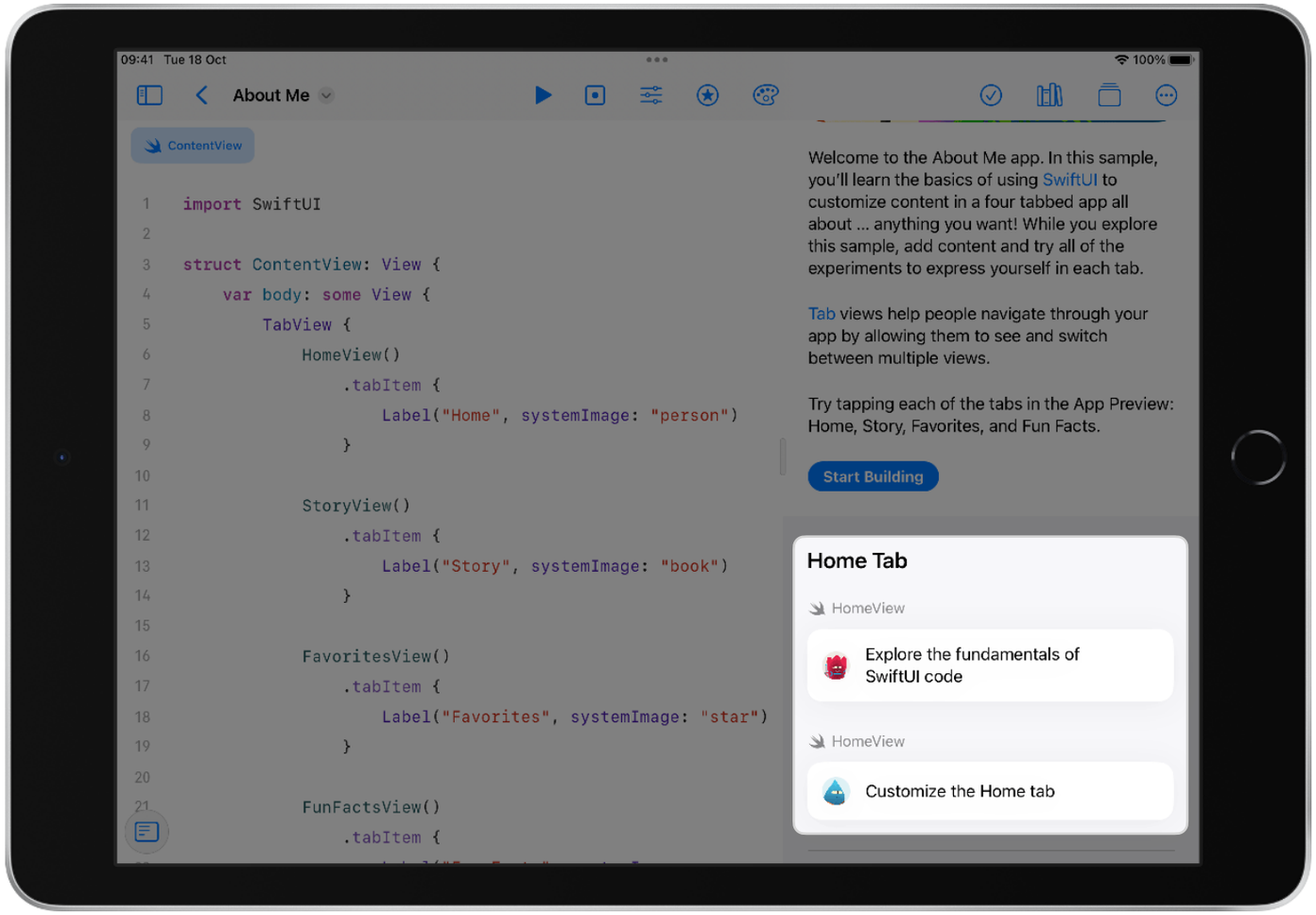
Step 4

Tap Learn More to open the guide.



Step 5

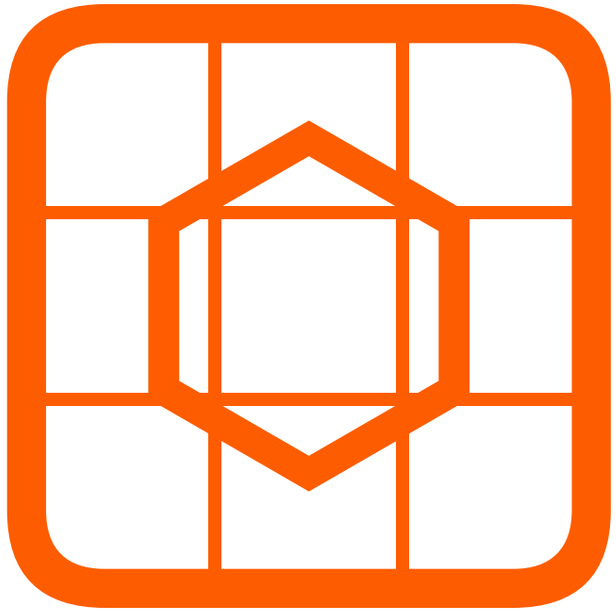
Scroll down to find Home tab. Complete the walkthrough and experiment in this section.



Tutorial 2

Plan Your Clip Shape

Design a shape in Keynote and calculate the coordinates.

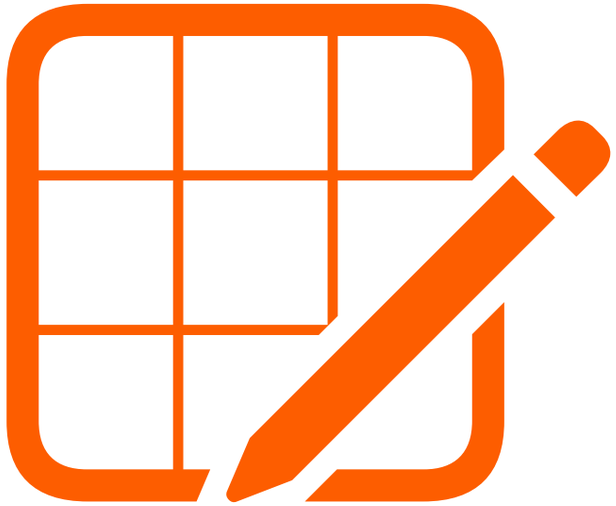


Estimated time:
20 mins

Section 1

Design your shape

Use a 1x1 grid to design a clip shape in Keynote.



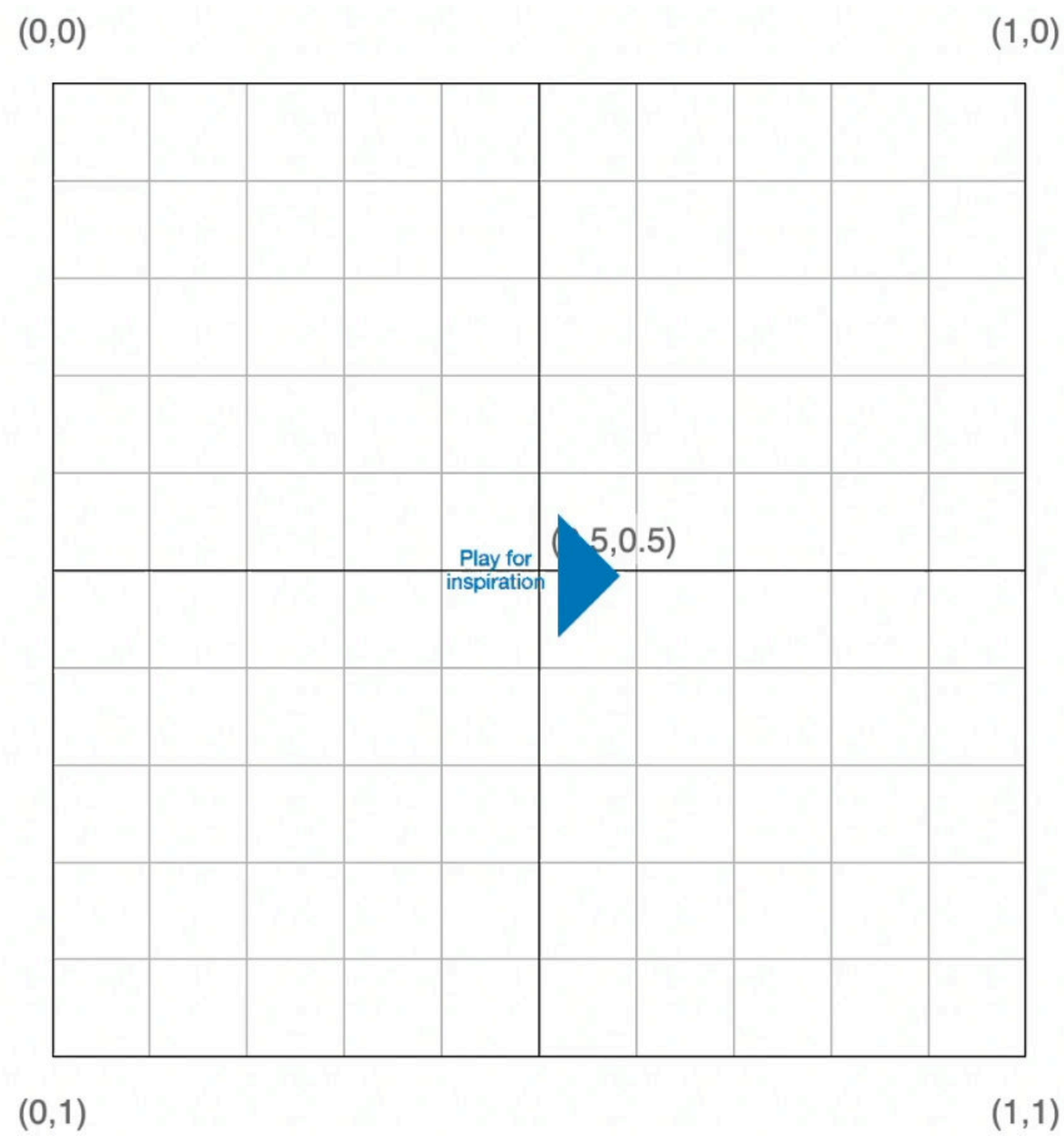
Step 1

Clip shapes are like cookie cutters. By applying a clipping shape to a view, you preserve the parts of the view covered by the shape, while eliminating the rest.

For best results, make a shape that’s open in the middle.

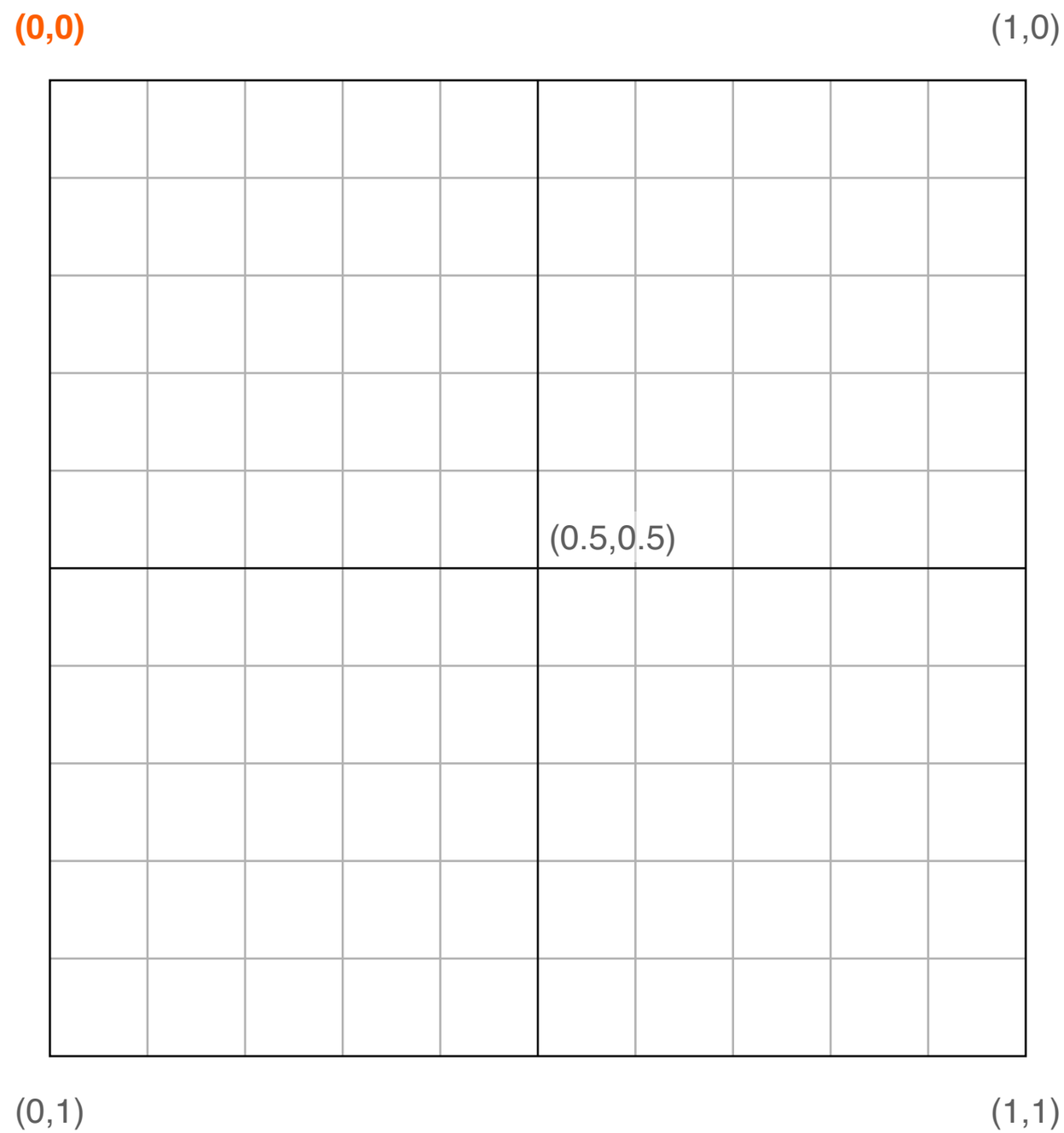
Step 2

Watch this video to preview how you'll use a 1x1 grid to plan your shape.



Step 3

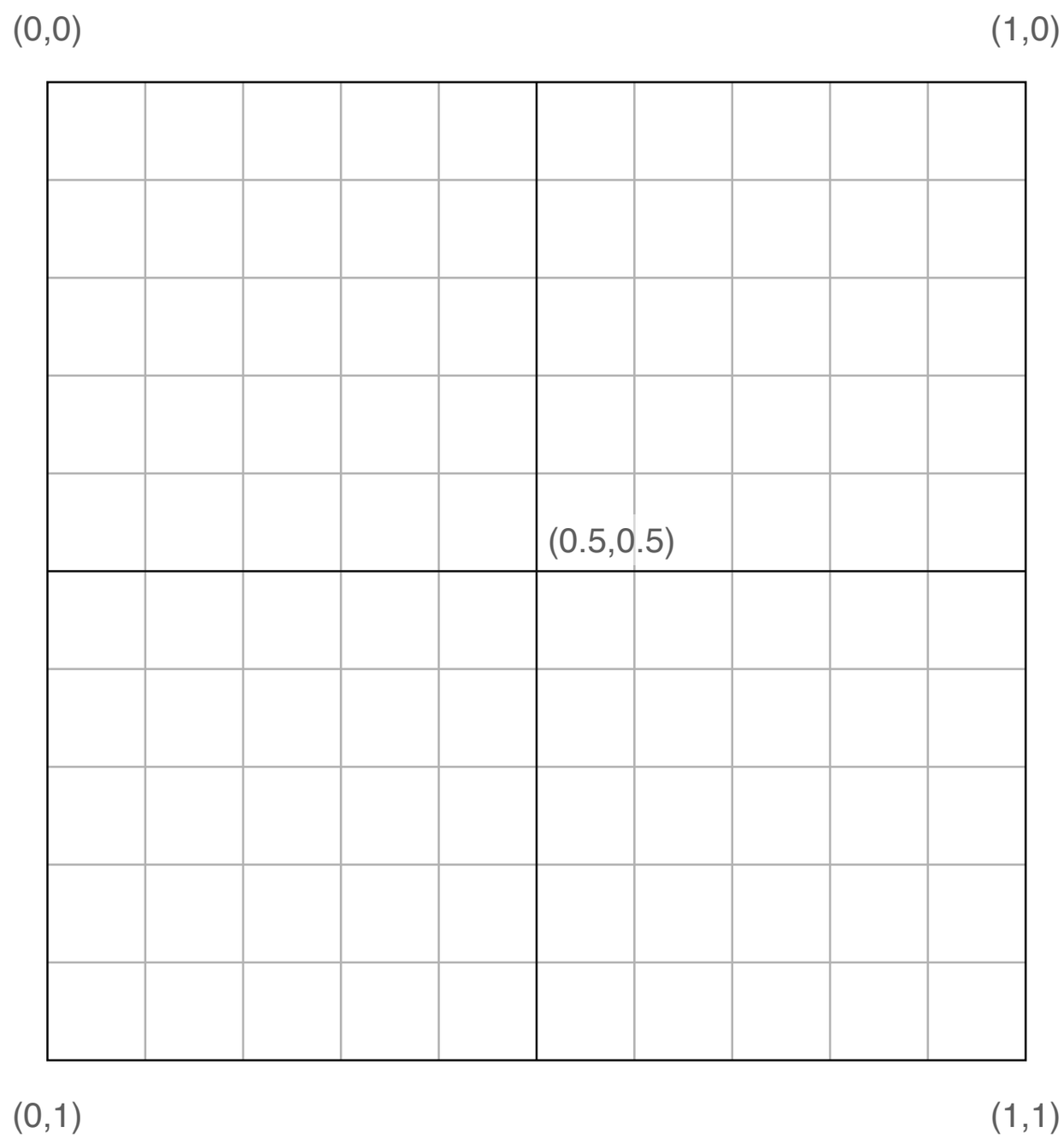
Notice where the origin, (0,0), is on the grid. When creating a shape the origin is at the top-left corner of the screen, unlike what you might see in a geometry class.



Step 4

Draw your shape.

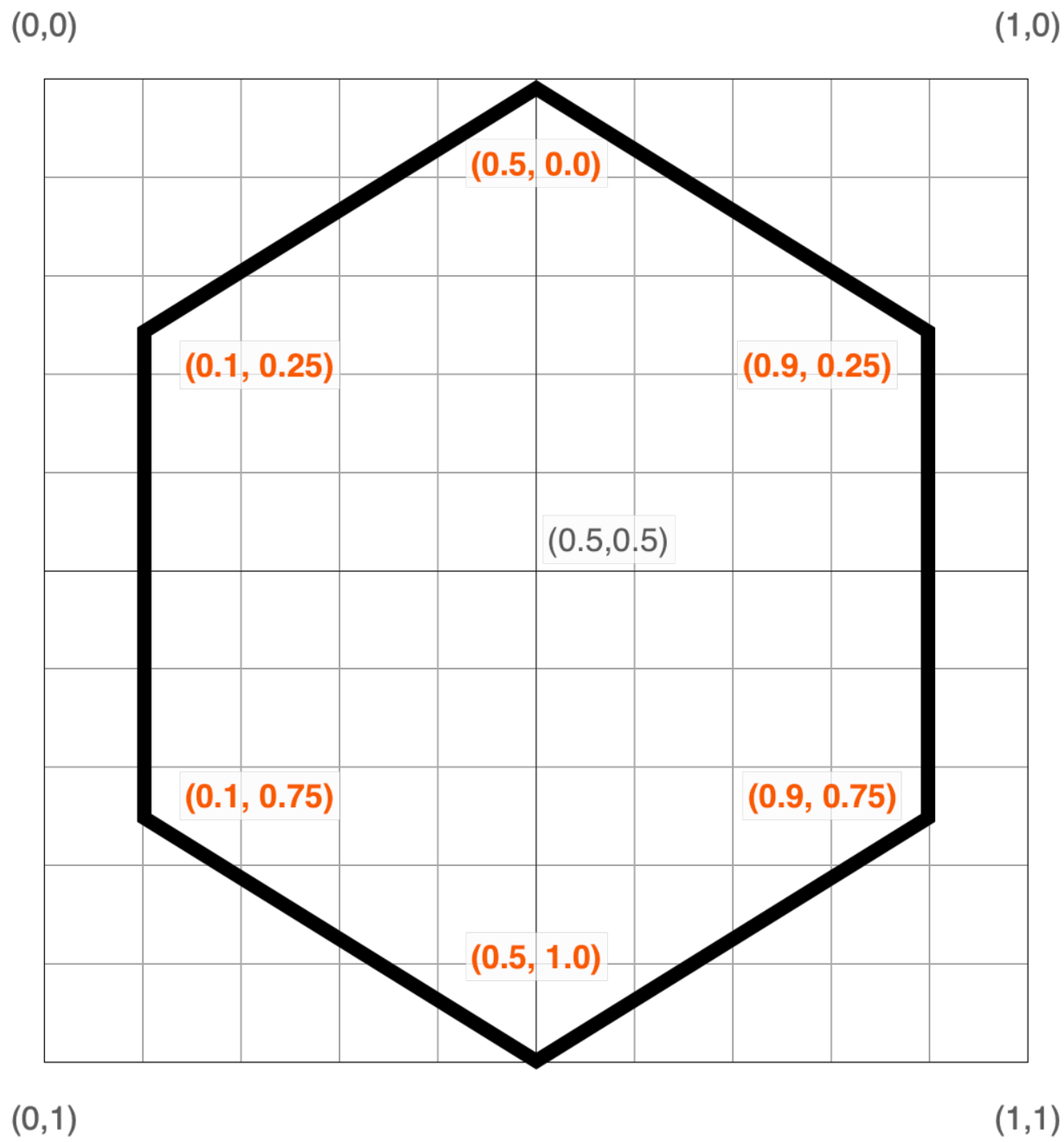
Tap Shapes to add a shape or build a shape using lines.



Step 5

Add the coordinates for each line segment, (x, y), to your shape.

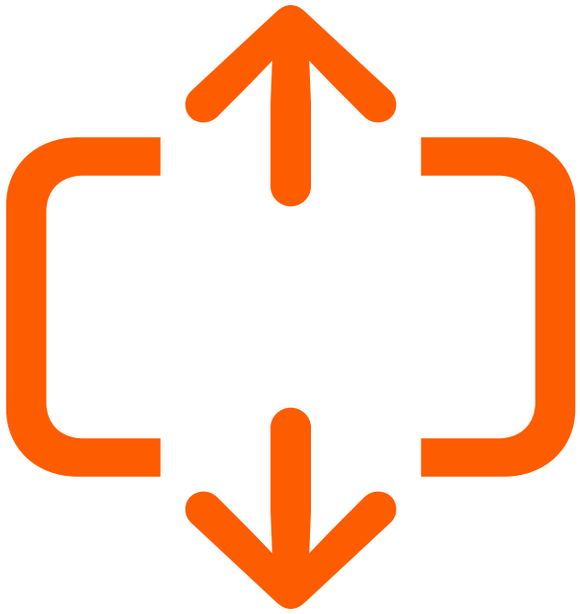
Previous slide >



Section 2

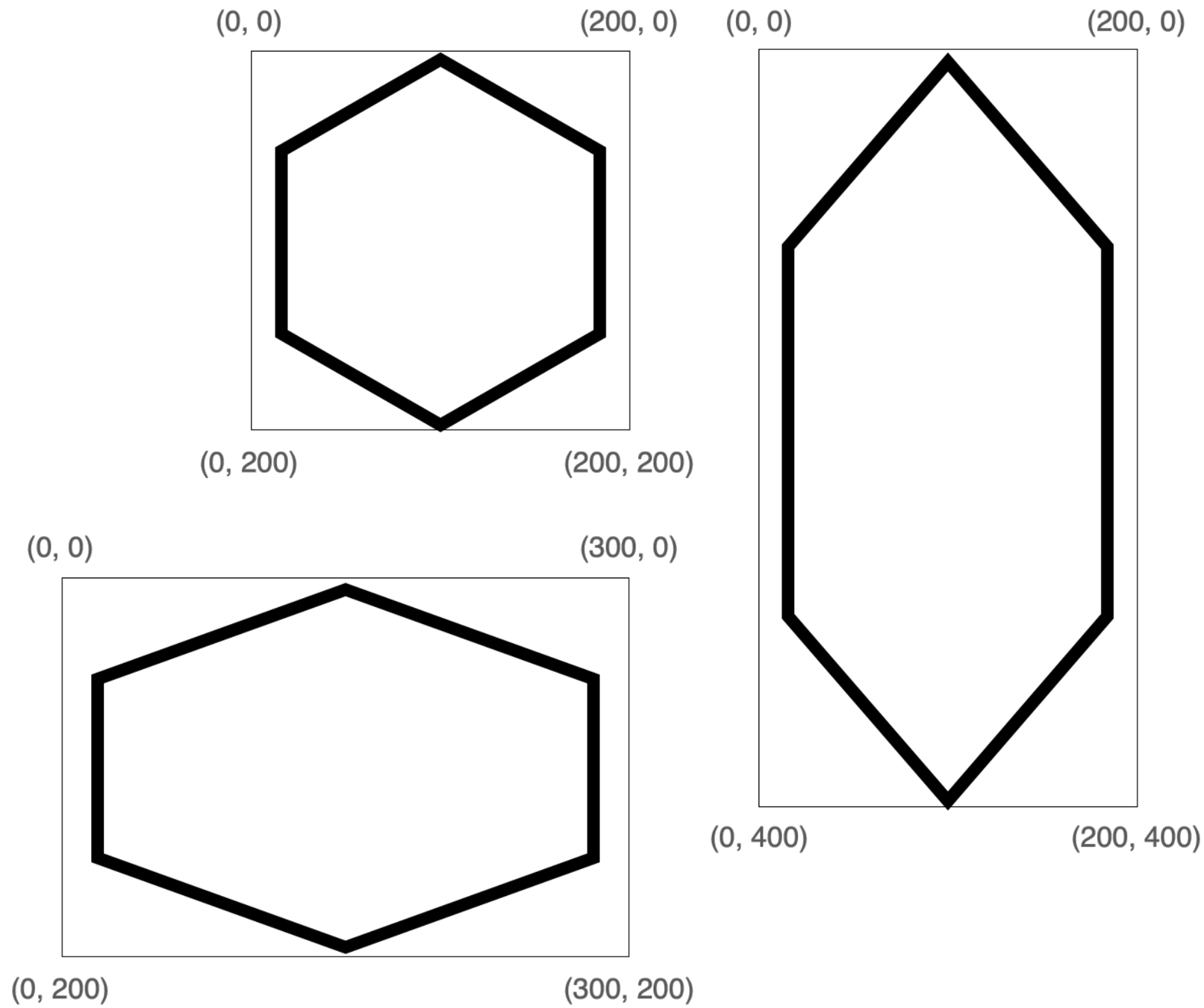
Create a responsive shape

Use your coordinates, and the width and height of the view, to create a shape that adjusts to different view sizes.



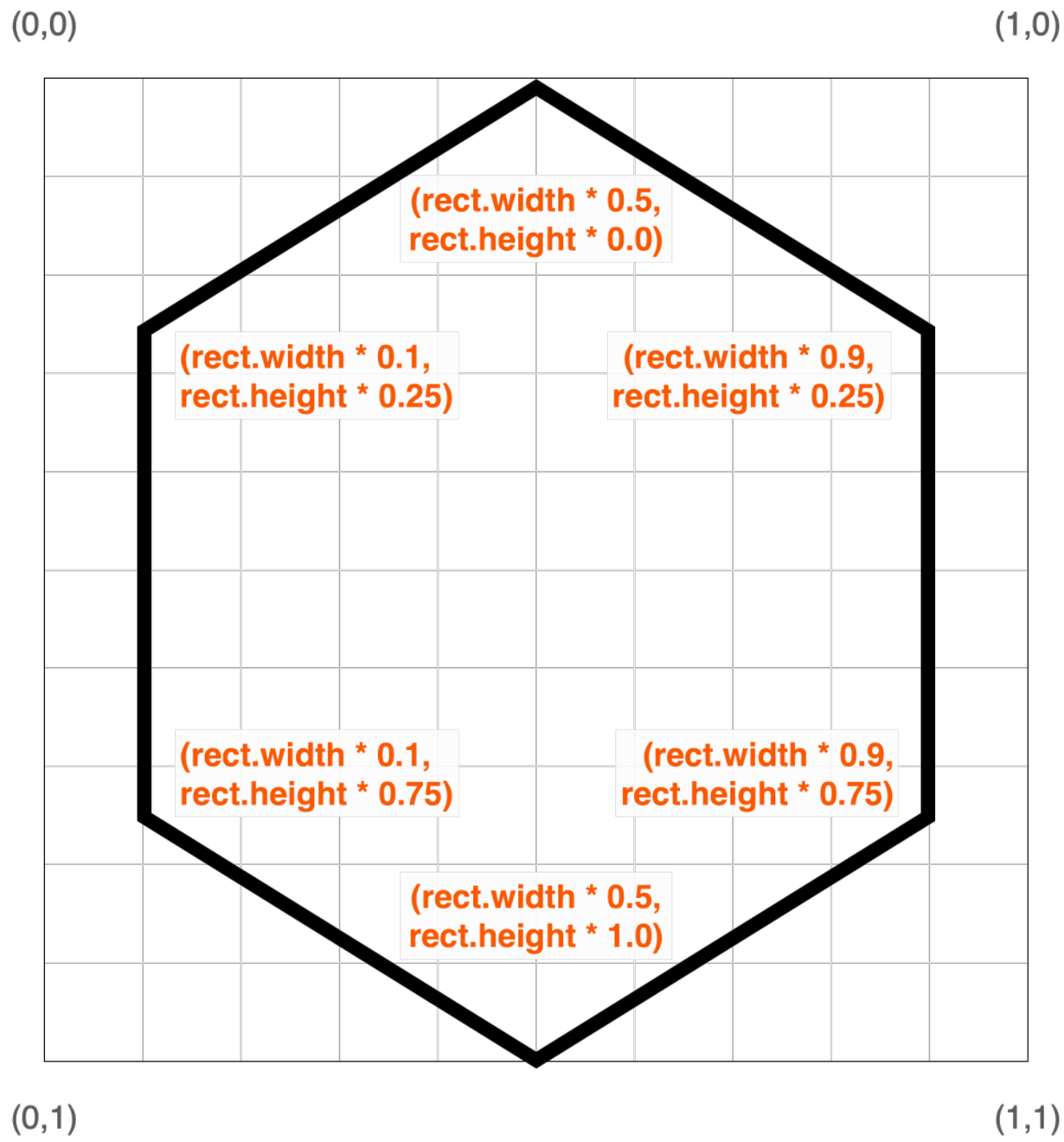
Step 1

Combining your coordinates, with the width and height of the view, ensures your shape will keep its relative dimensions, even when the size of the view it's in changes.



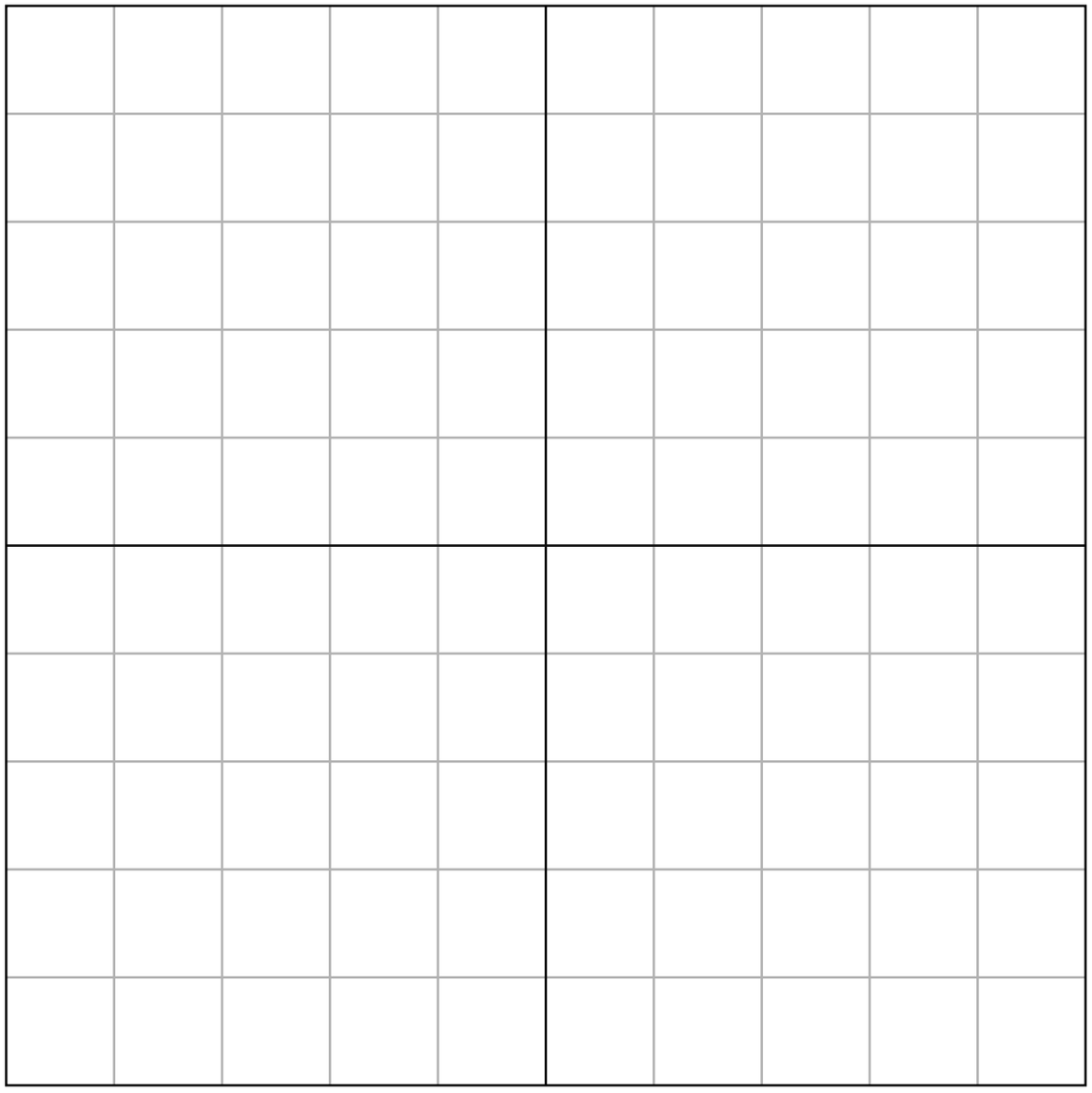
Step 2

Preview how to multiply your coordinates by `rect.width` for the x values and `rect.height` for the y values.



Name

(0,0) (1,0)



(0,1) (1,1)

Step 3

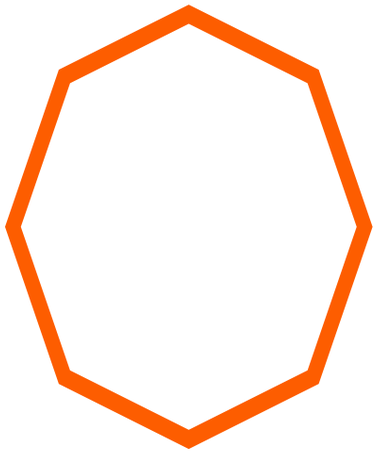
Copy and paste your shape and coordinates. Then edit your coordinates to multiply the x values by `rect.width`, and the y values by `rect.height`.

Step 4

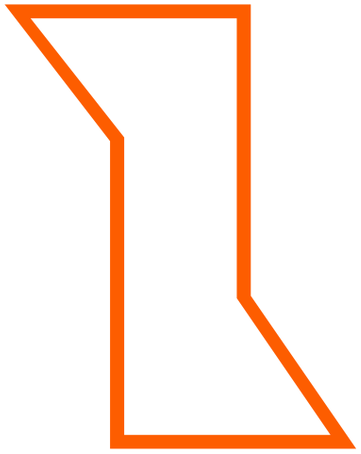
Give your shape a name.
Write the name at the top
of the previous slide.

Previous slide >

Star



Mirror



ModernHourGlass

Tutorial 3

Code Your Shape

Create a new file in About Me and give it a name. Then code your shape in SwiftUI.

Estimated time:
20 mins

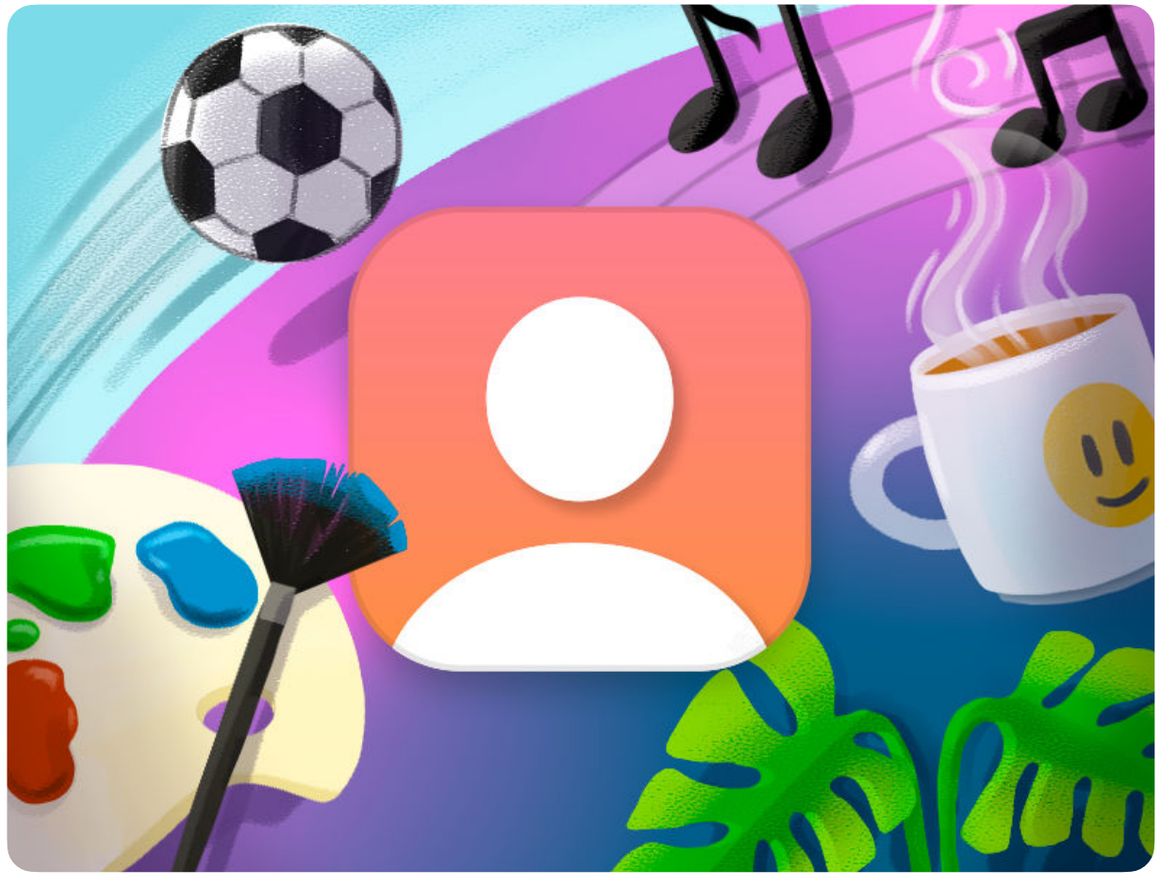
Section 1

Create a new file and begin your shape

Add a new file in About Me. Then create a new shape and give it a name.

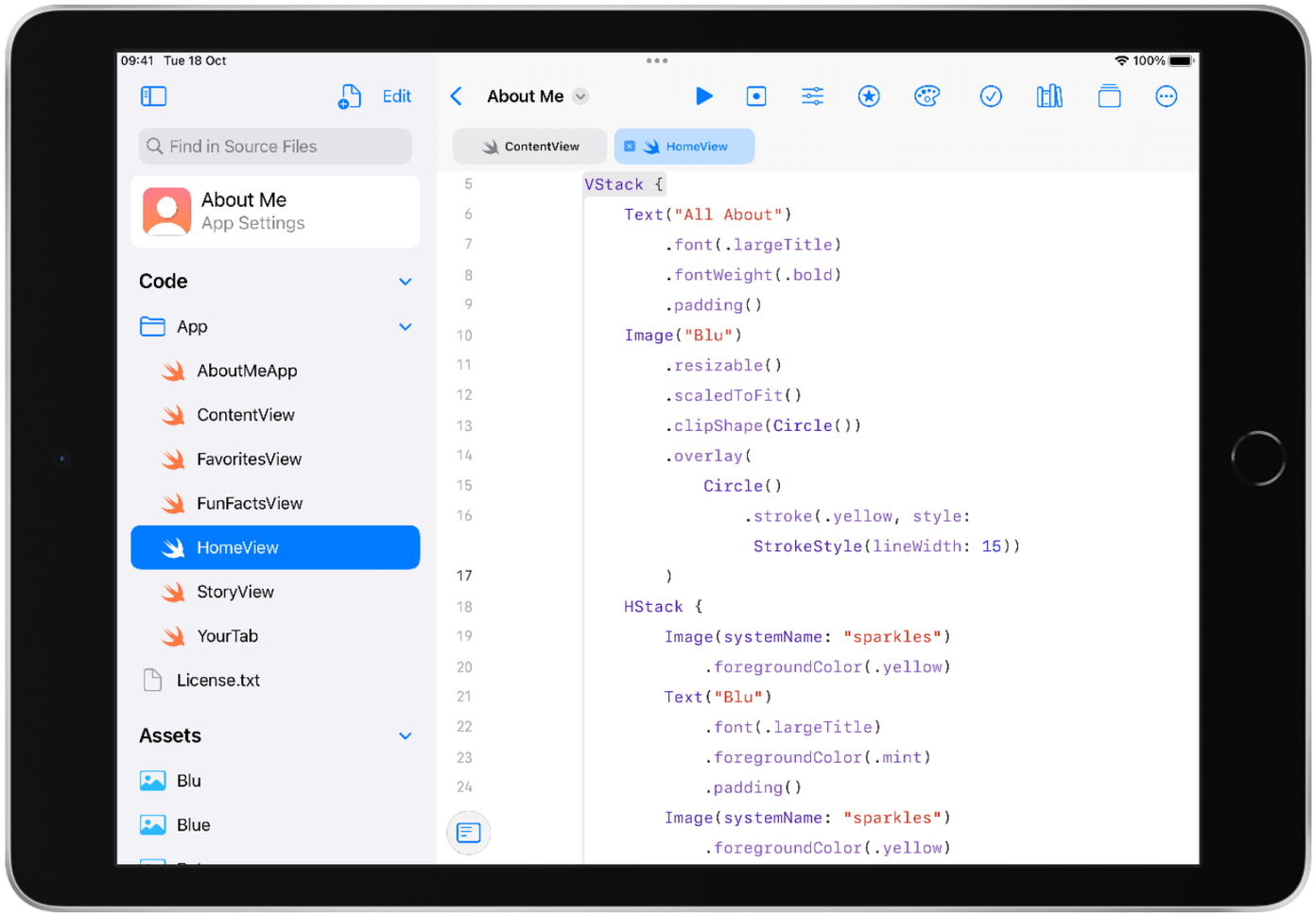
Step 1

Open About Me.



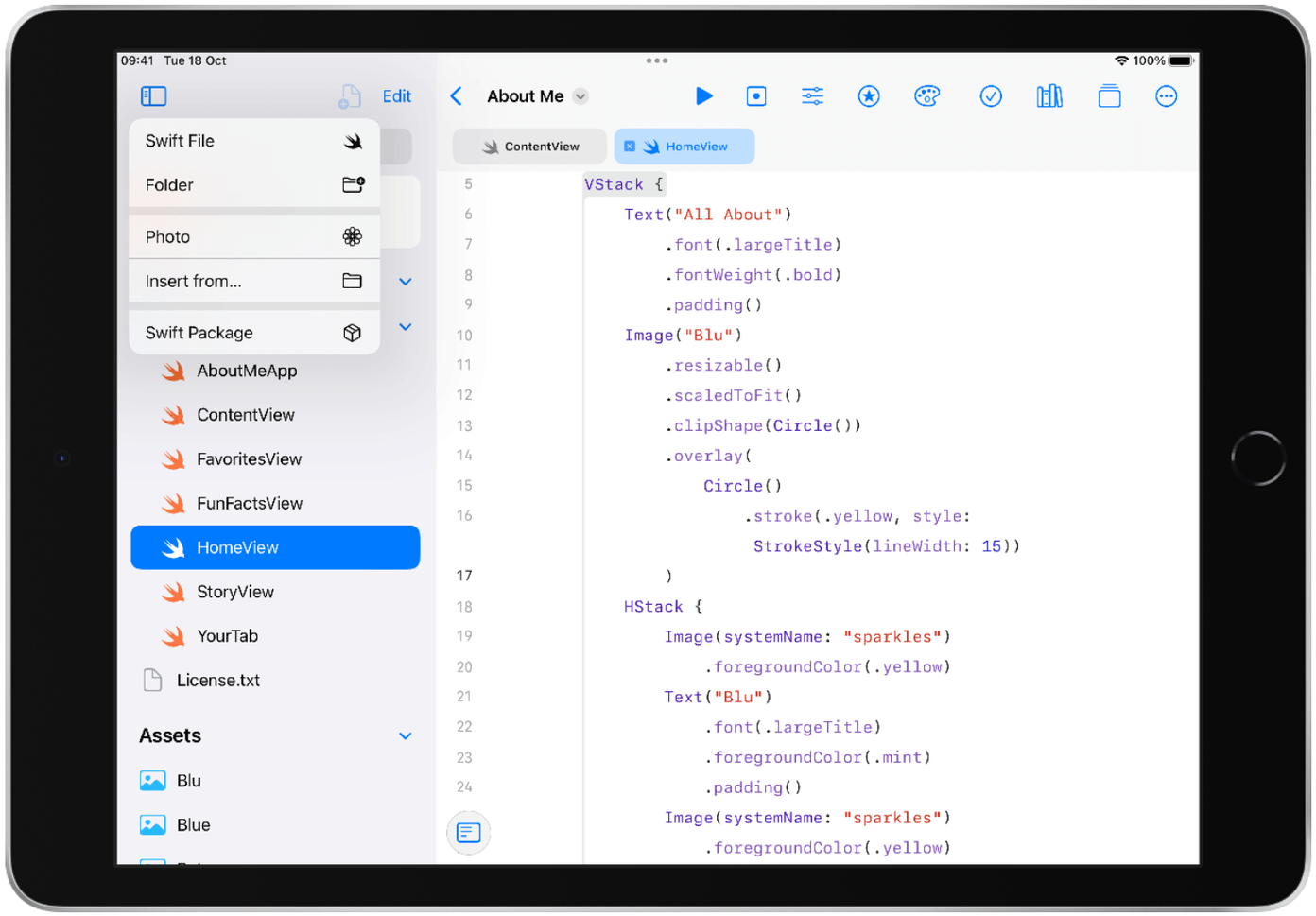
Step 2

Open the left-hand sidebar.



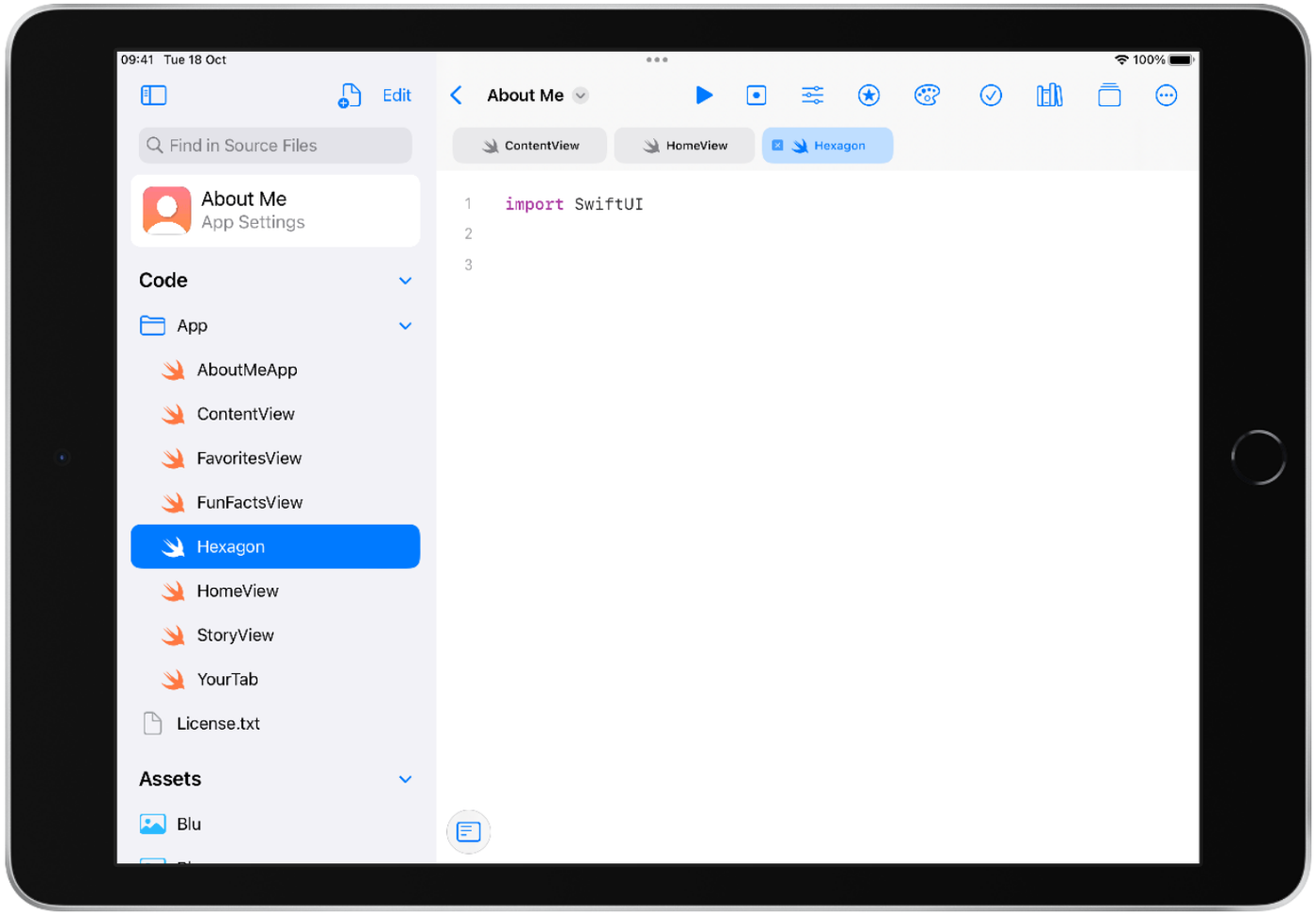
Step 3

Tap Add New Item and tap Swift File.



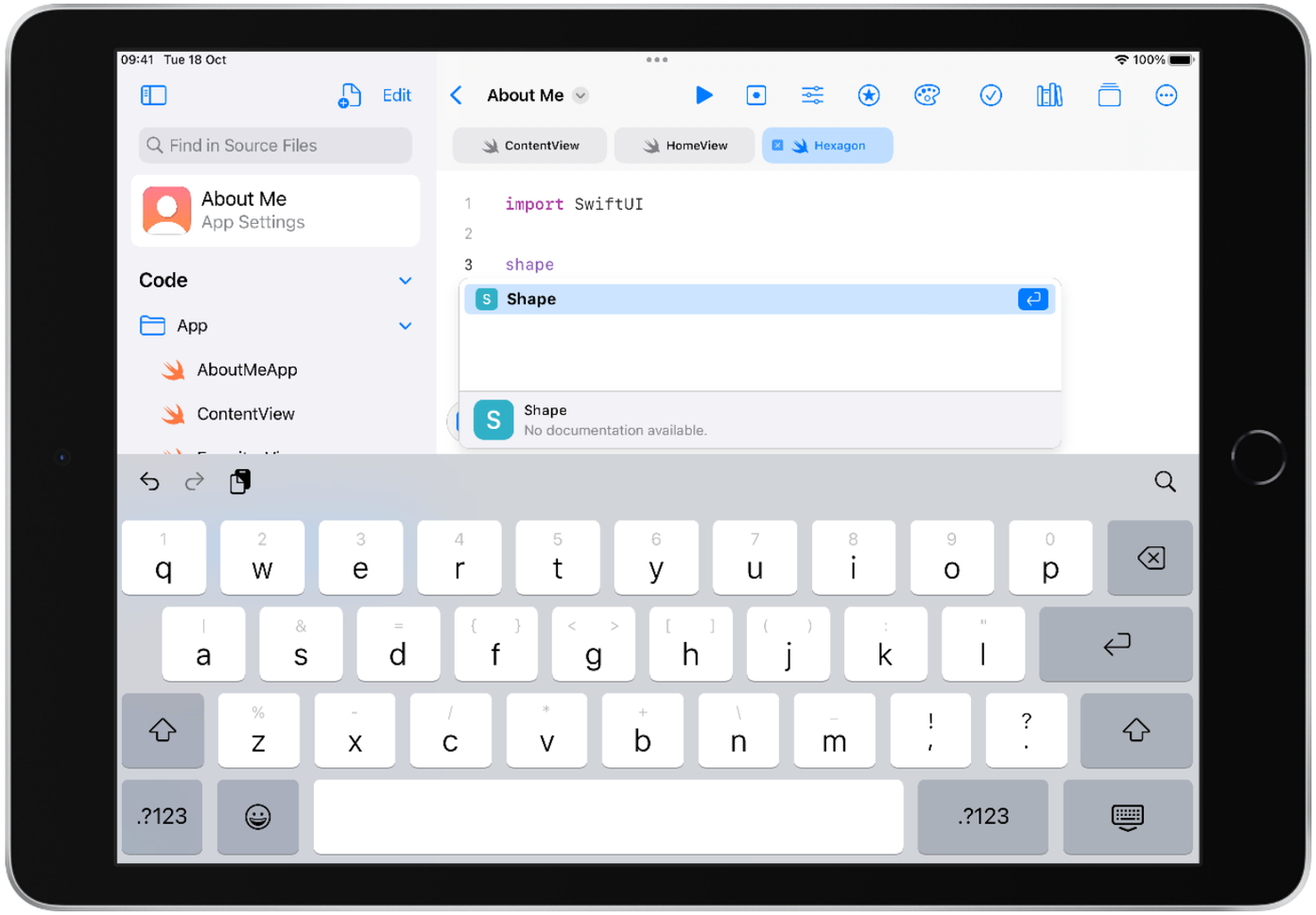
Step 4

Name your file the name of your shape.



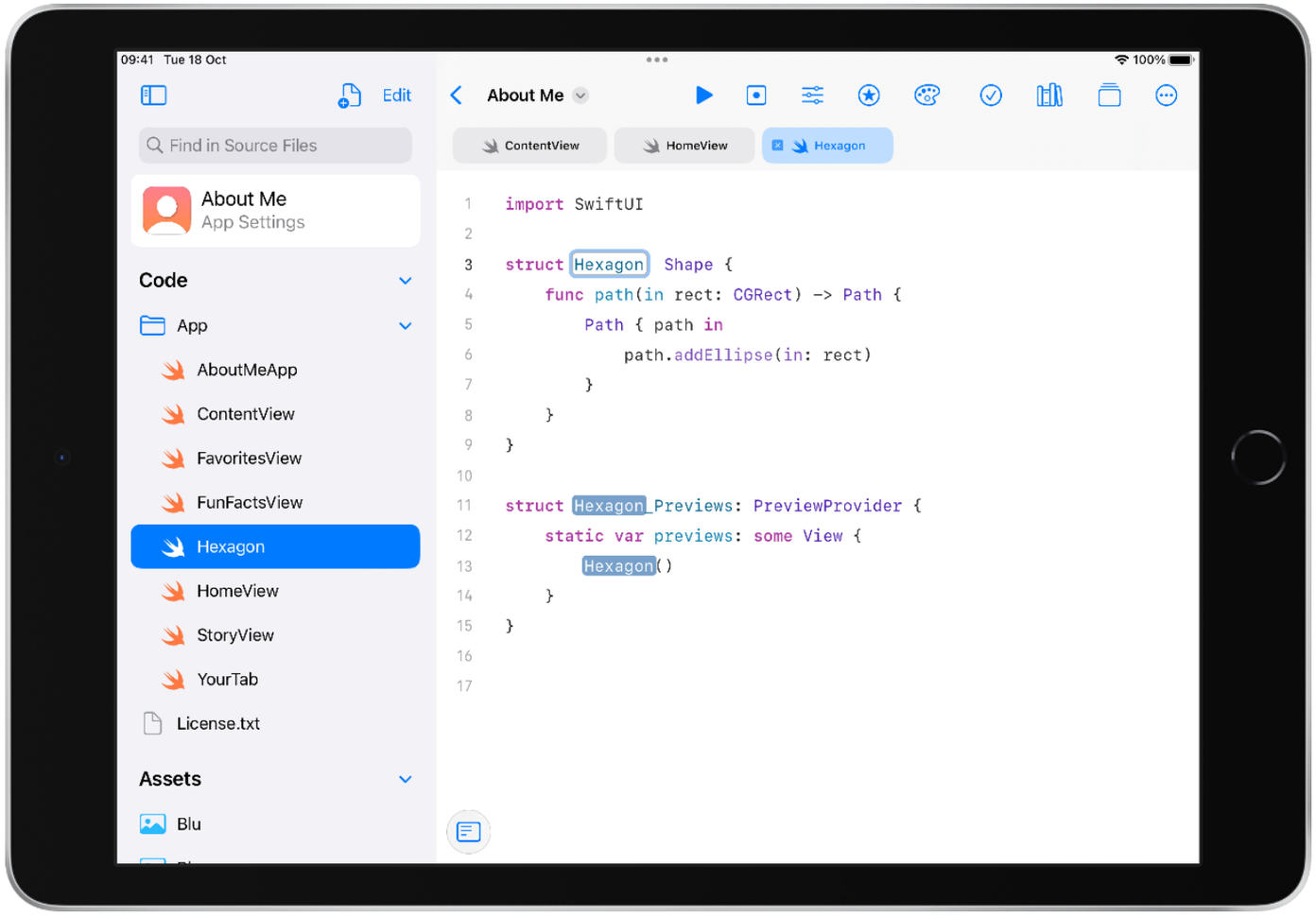
Step 5

Begin typing “shape” into the code editor. When code completions appears, tap the arrow or return on your keyboard to add the setup code for a shape.



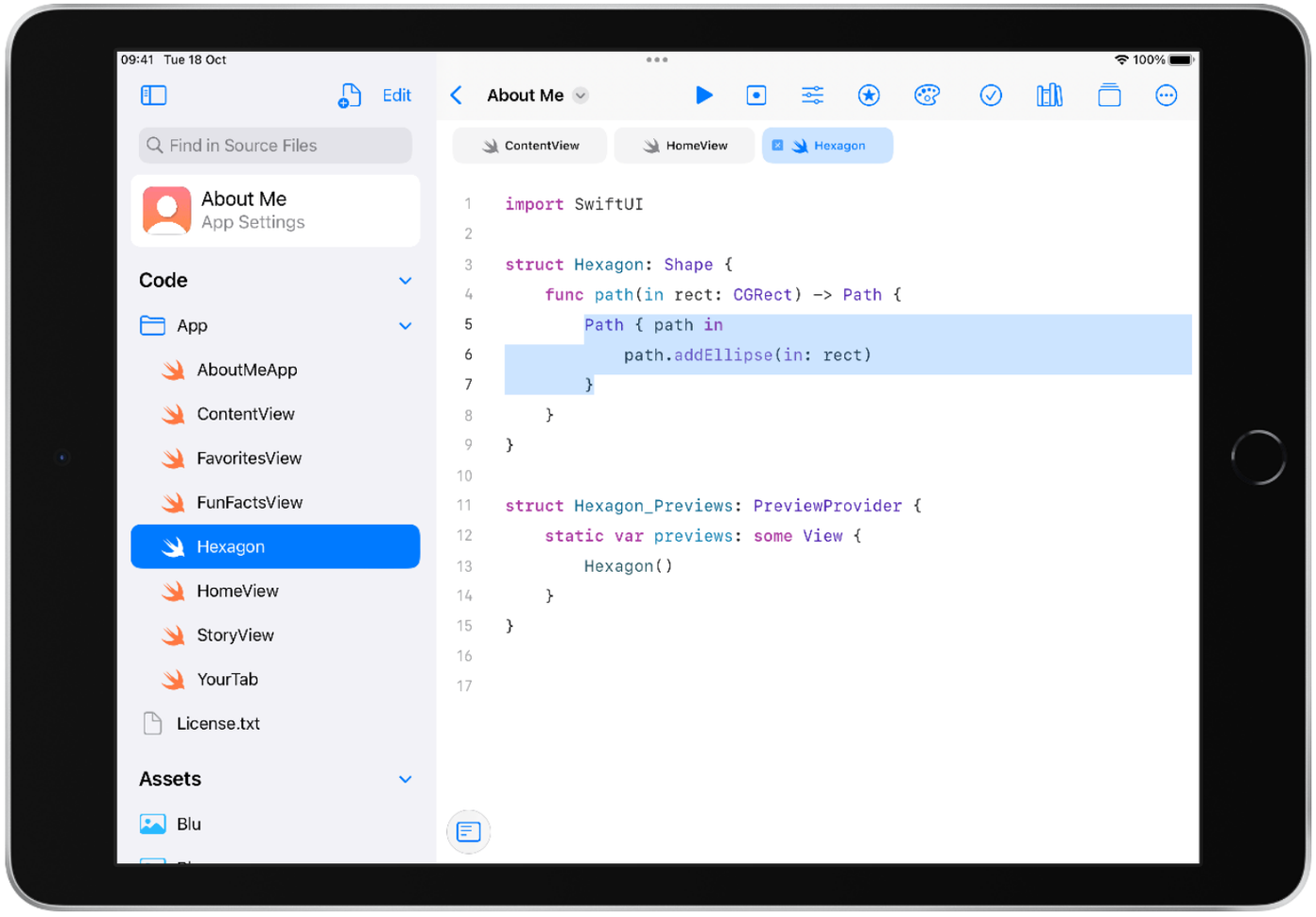
Step 6

Give your shape the same name as your file.



Step 7

Delete the code inside the function.



Step 8

Inside the function, create a variable named “path”.

```
struct Hexagon: Shape {
    func path(in rect: CGRect) -> Path {

        var path = Path()



    }
}
```

Step 9

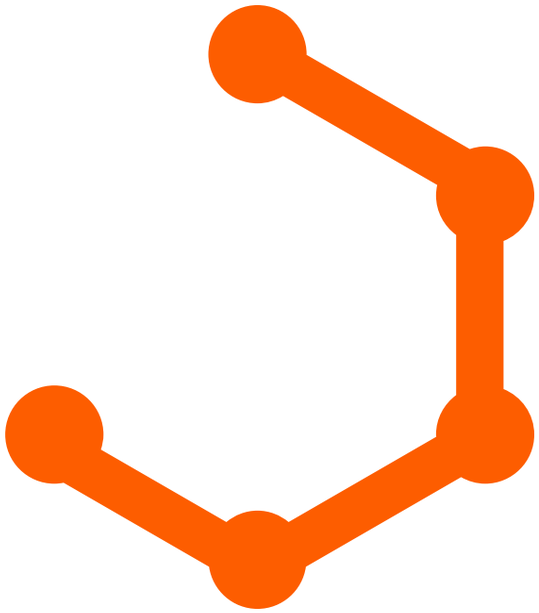
Add a few blank lines and return the path at the bottom of the function.

```
struct Hexagon: Shape {
    func path(in rect: CGRect) -> Path {

        var path = Path()


        return path
    }
}
```

Section 2

Add the lines

Create your shape using lines.



Step 1

Begin your shape using the **`move(to:)`** method and put in the coordinates for where your shape should start. Use the shape you designed to inform your coordinates.

```
struct Hexagon: Shape {
    func path(in rect: CGRect) -> Path {
        var path = Path()

        path.move(to: CGPoint(
            x: rect.width * 0.5,
            y: rect.height * 0))


        return path
    }
}
```

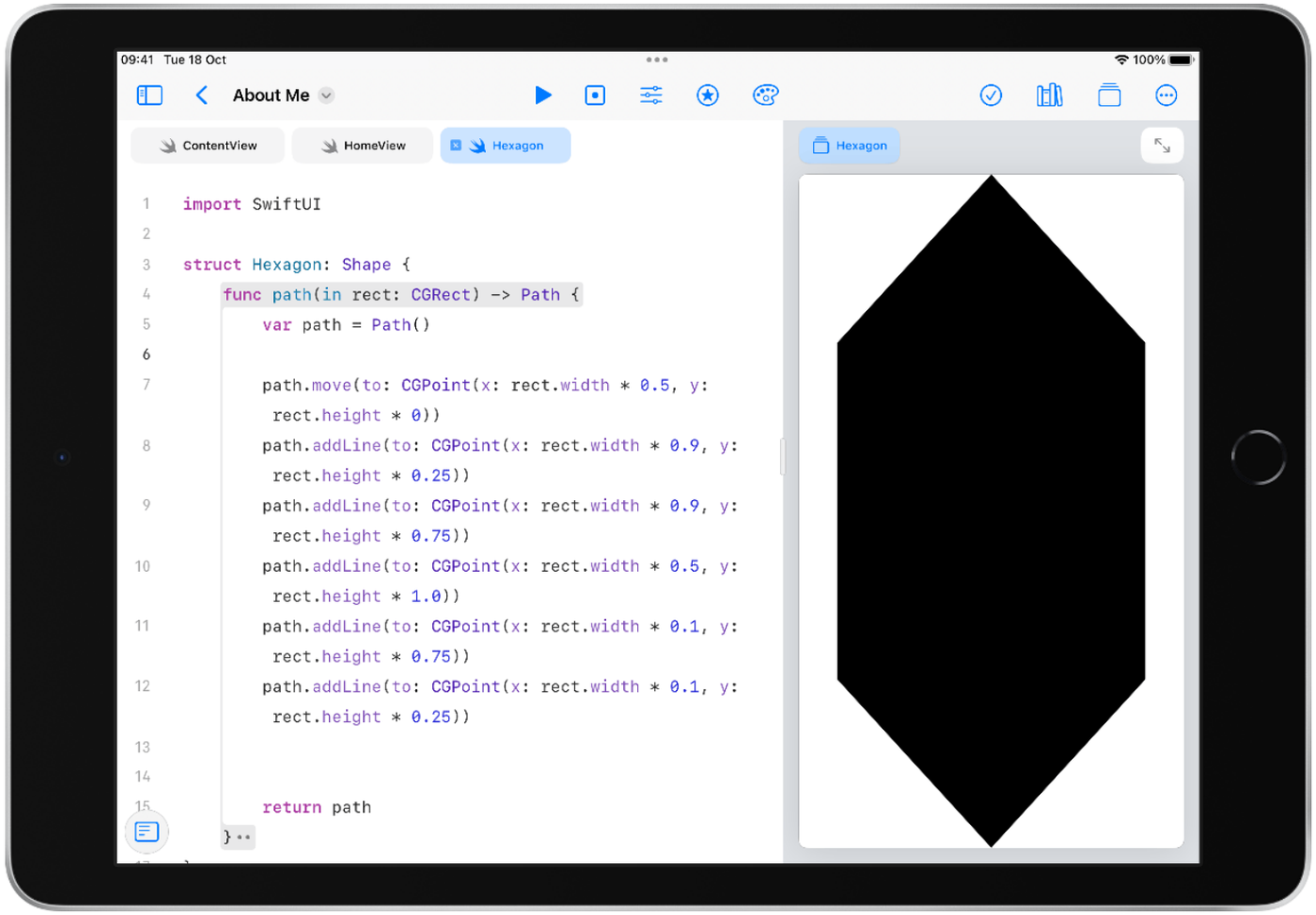
Step 2

Use the **`addLine(to:)`** method to add your first line segment.

```
struct Hexagon: Shape {
    func path(in rect: CGRect) -> Path {
        var path = Path()

        path.move(to: CGPoint(
            x: rect.width * 0.5,
            y: rect.height * 0))
        path.addLine(to: CGPoint(
            x: rect.width * 0.9,
            y: rect.height * 0.25))

        return path
    }
}
```

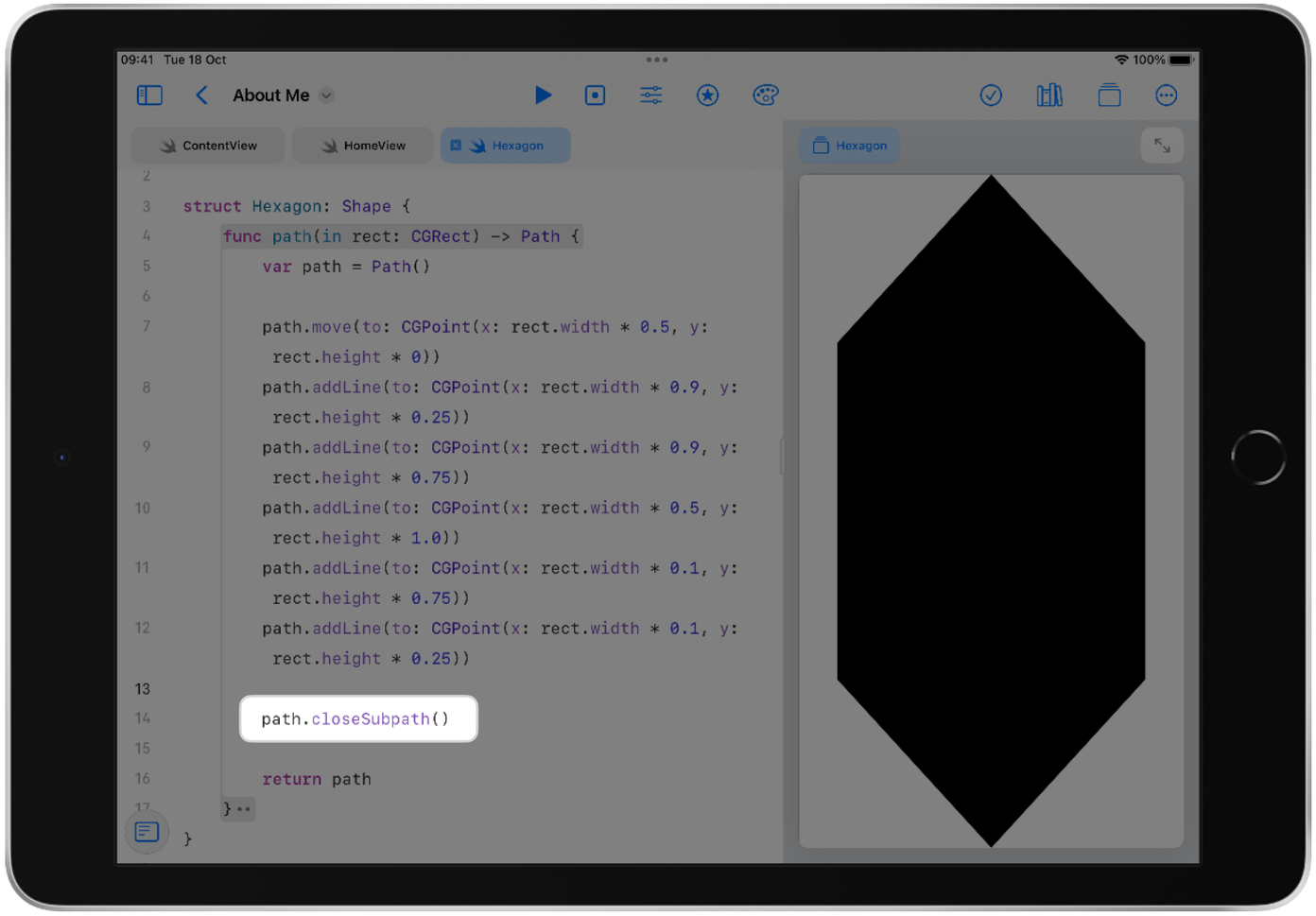
Step 3

Continue adding line segments to complete your shape.



Step 4

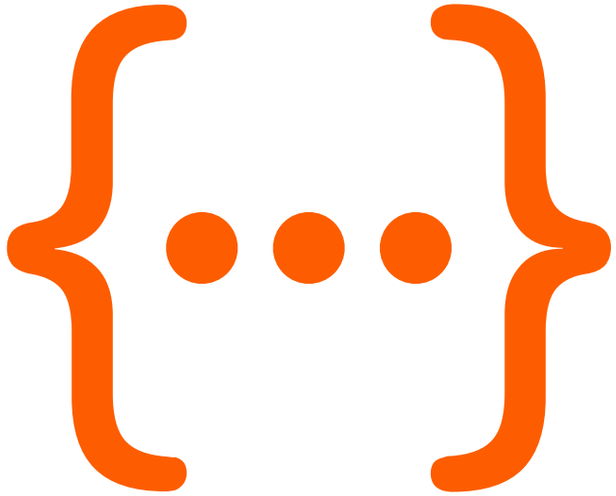
Close your shape using **`closeSubpath()`**.



Section 3

Code reference

Use the following reference to quickly edit your shape or create new ones.



Code reference

Start a shape

```
struct ShapeName: Shape {
    func path(in rect: CGRect) -> Path {
        var path = Path()
        // your code here
        return path
    }
}
```

Start a line

```
path.move(to: CGPoint(x: rect.width * 0.0, y: rect.height * 0.0))
```

Add lines

```
path.addLine(to: CGPoint(x: rect.width * 0.0, y: rect.height * 0.0))
```

Close the path

```
path.closeSubpath()
```

Tutorial 4

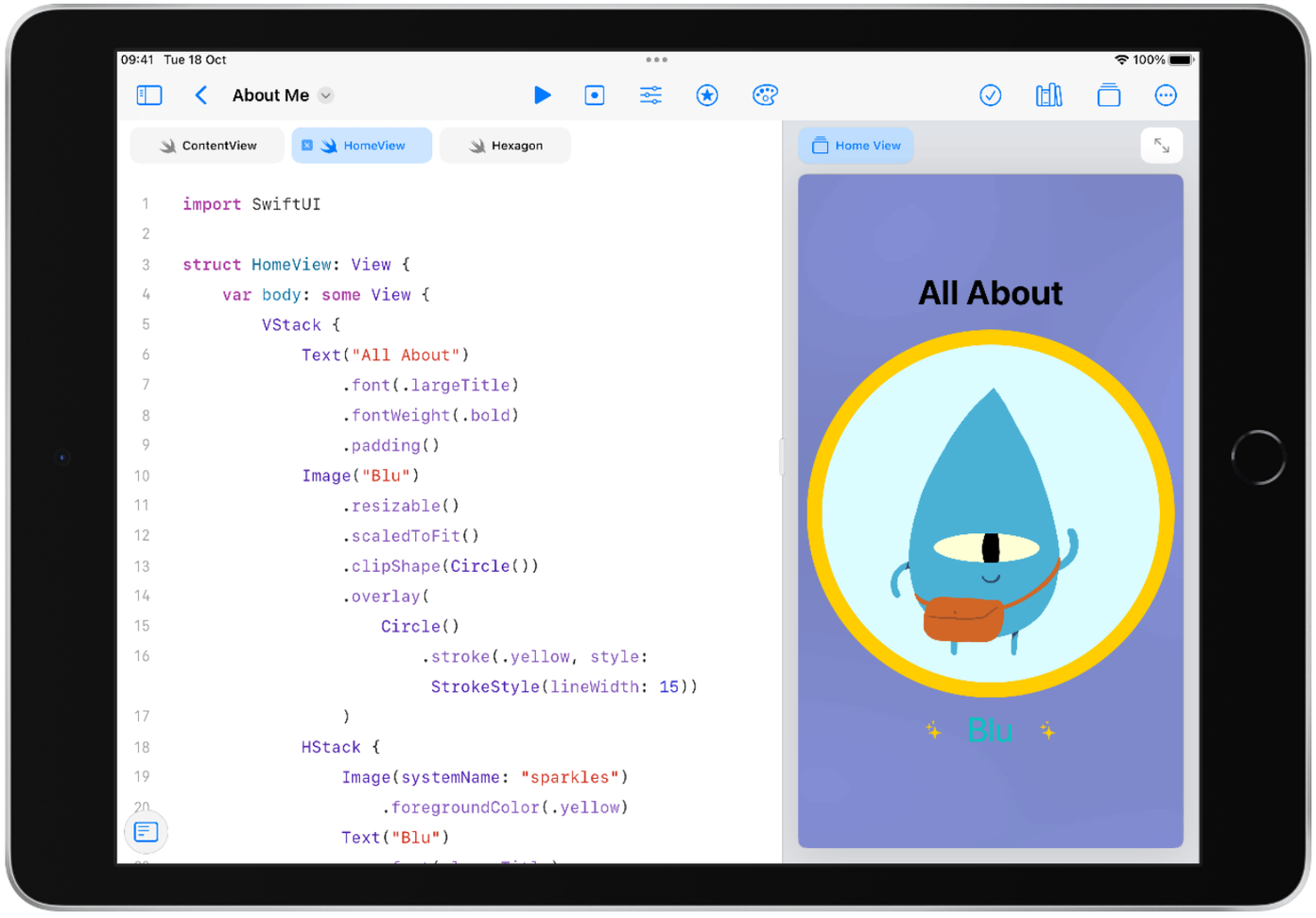
Use Your Shape

Use your clip shape in the Home tab of About Me.

Estimated time:
5 mins

Step 1

Select the HomeView tab above the code editor.



Step 2

Edit **`clipShape`** and **`overlay`** to use your custom shape.