

#WWDC18

# Getting the Most out of Playgrounds in Xcode

Session 402

Tibet Rooney-Rabdau, Xcode Engineer

Alex Brown, Core OS Engineer

TJ Usiyan, Xcode Engineer

Fundamentals

Running step by step

Advanced techniques

Fundamentals

Running step by step

Advanced techniques

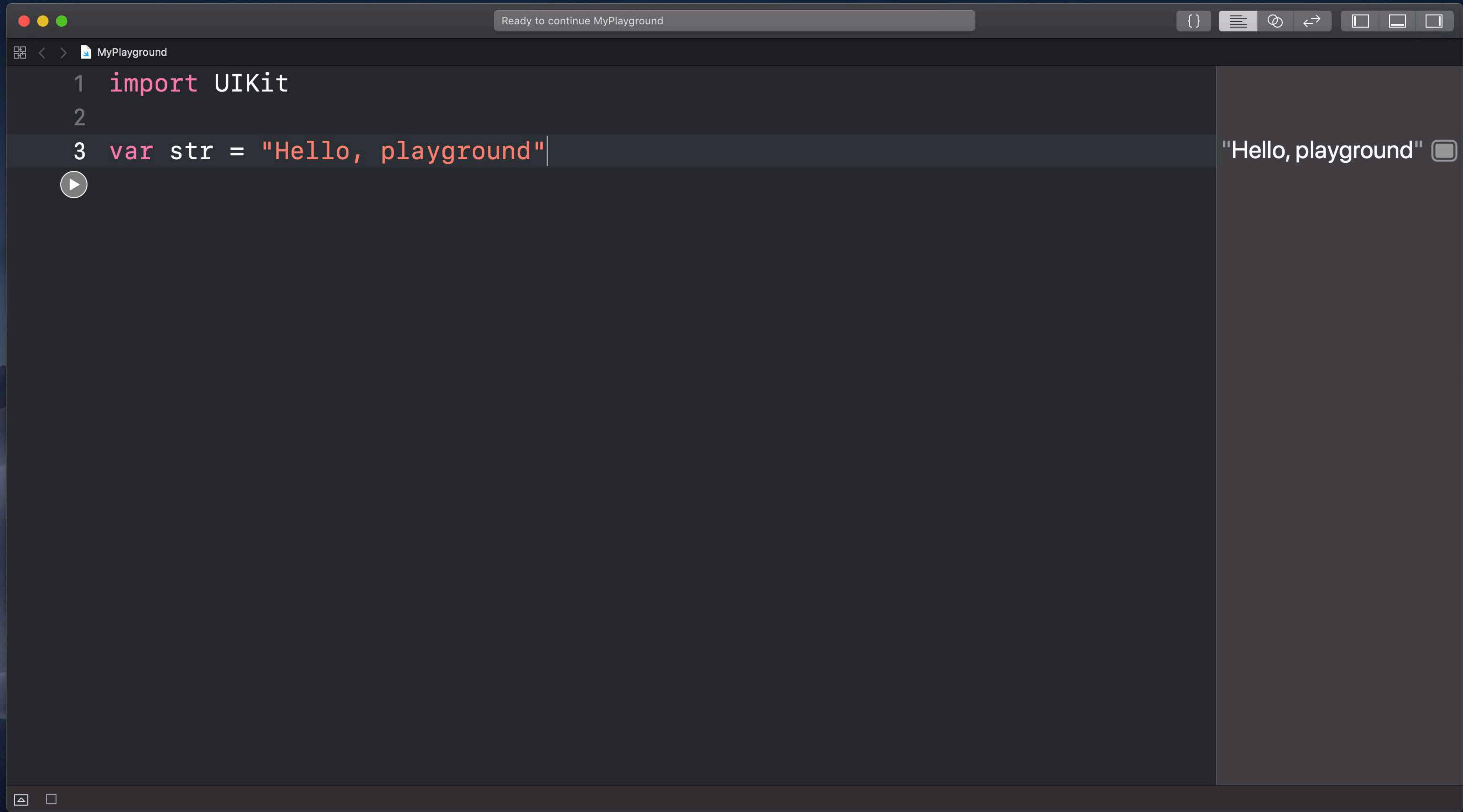
# A Quick Tour

Ready to continue MyPlayground

MyPlayground

```
1 import UIKit
2
3 var str = "Hello, playground"
```

"Hello, playground"

The image shows a screenshot of the Xcode playground interface. The top bar contains the text "Ready to continue MyPlayground" and several icons for source control, search, and window management. Below the bar, the playground name "MyPlayground" is displayed. The main area is split into two panes. The left pane is a code editor with three lines of Swift code: "1 import UIKit", "2", and "3 var str = \"Hello, playground\"". A play button icon is located at the bottom left of the code editor. The right pane is the output area, which displays the string "Hello, playground" in white text on a dark background, followed by a copy icon.


Ready to continue MyPlayground

MyPlayground

```
1 import UIKit
2
3 var str = "Hello, playground"
```

▶

"Hello, playground"



Ready to continue MyPlayground

MyPlayground

```
1 import UIKit
2
3 var str = "Hello, playground"
```

Hello, playground

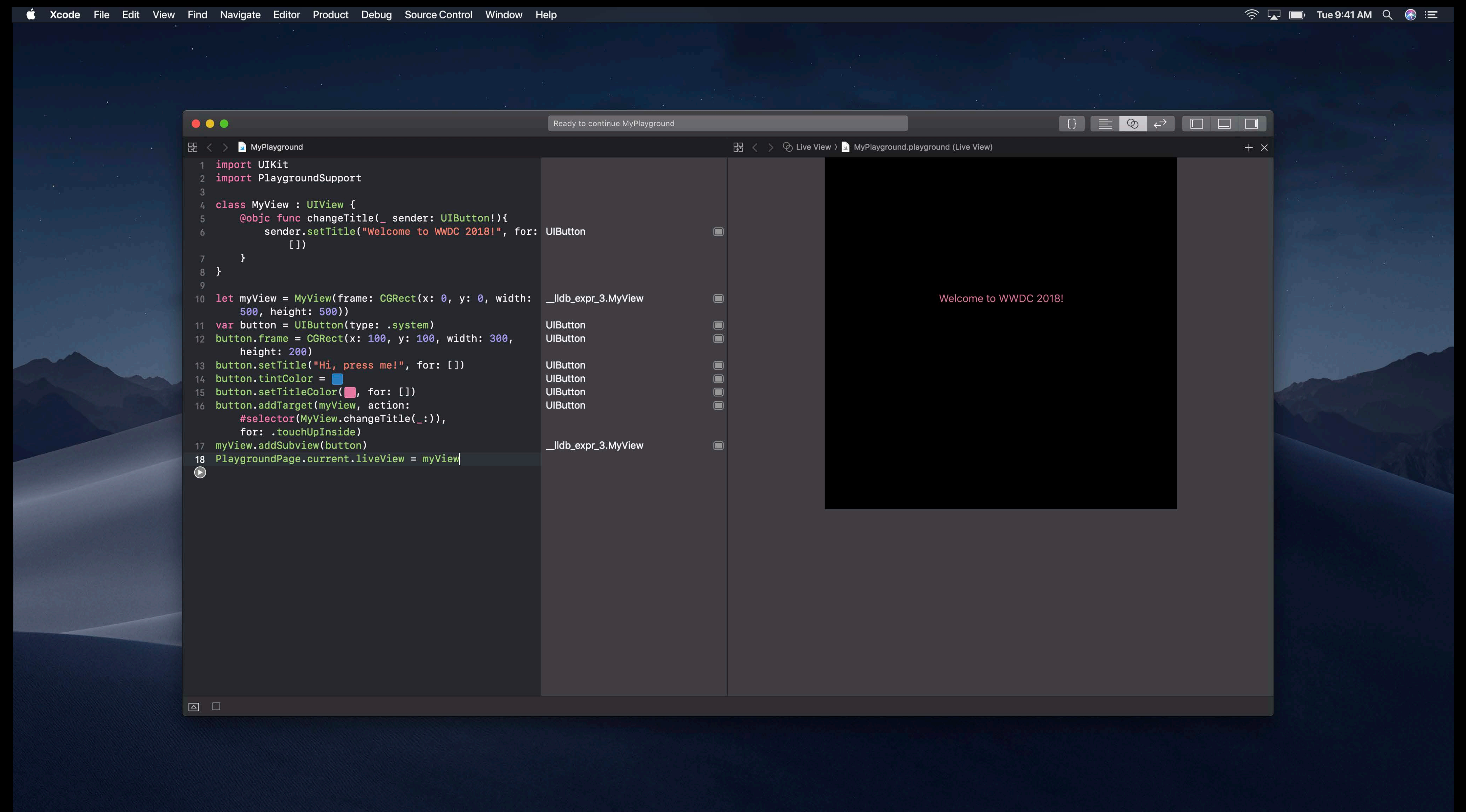
Play button

"Hello, playground"



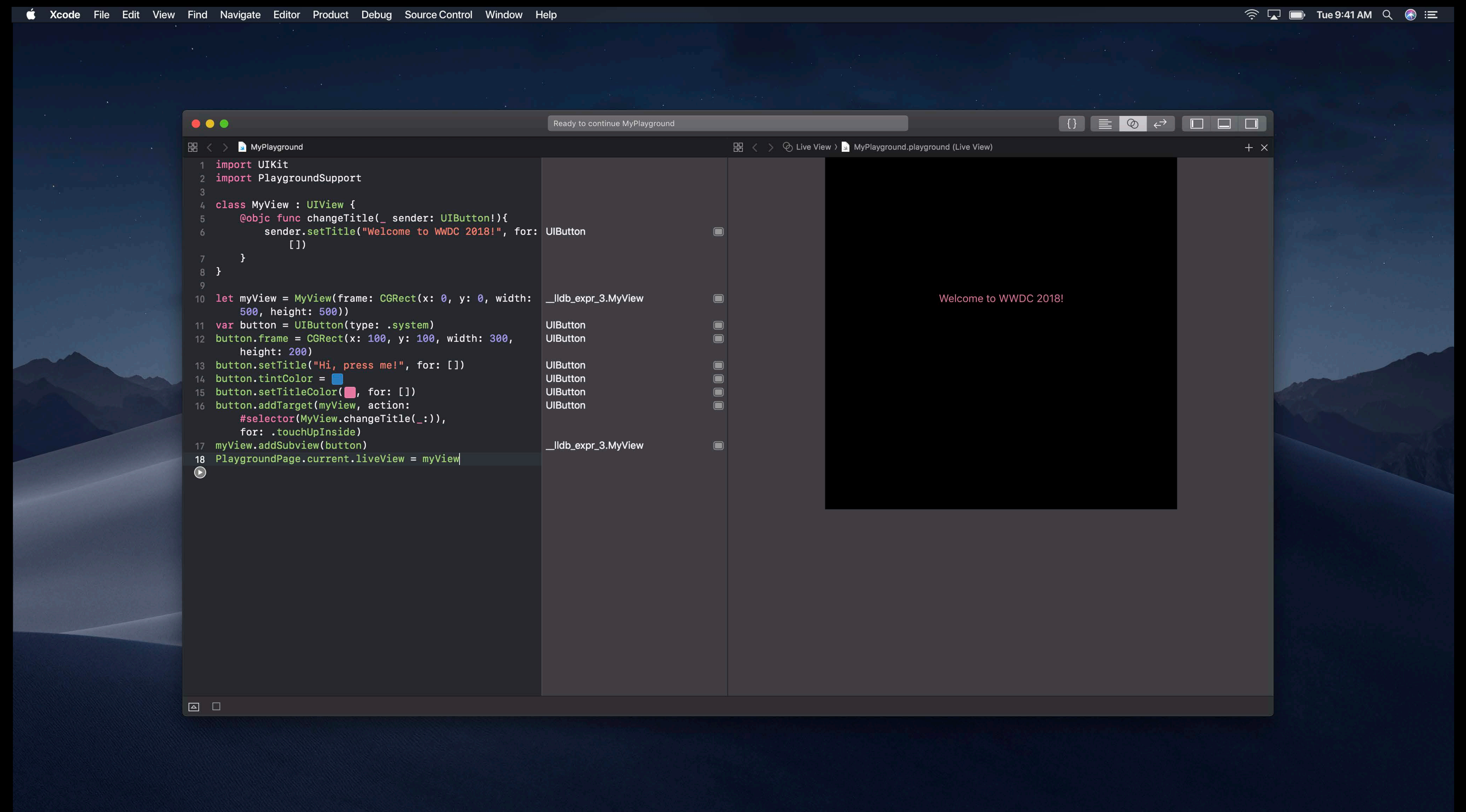


# Live View



# Live View

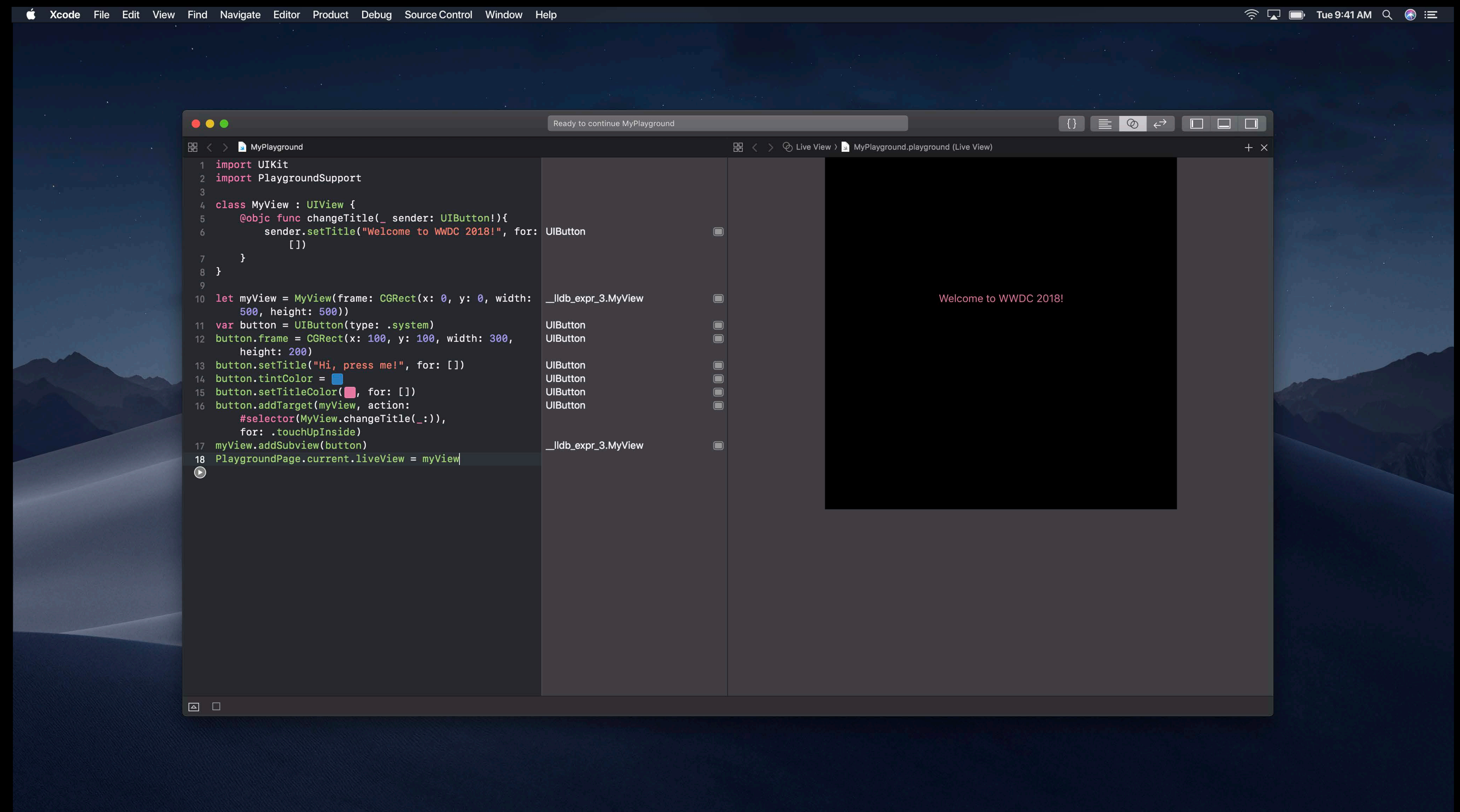
## UIView and UIViewController



# Live View

UIView and UIViewController

NSView and NSViewController



```
// Live View
```

```
import PlaygroundSupport
```

```
let viewController = /* ... */
```

```
PlaygroundPage.current.liveView = viewController
```

```
// Live View
```

```
import PlaygroundSupport
```

```
let viewController = /* ... */
```

```
PlaygroundPage.current.liveView = viewController
```

```
// Live View
```

```
import PlaygroundSupport
```

```
let viewController = /* ... */
```

```
PlaygroundPage.current.liveView = viewController
```

```
// Live View
```

```
import PlaygroundSupport
```

```
let viewController = /* ... */
```

```
PlaygroundPage.current.liveView = viewController
```

# Markup

Use markup to show formatted text



```
// Markup
```

```
//: Roses are red
```

```
//:
```

```
//: Violets are blue
```

```
//:
```

```
//: Writing your markup
```

```
//:
```

```
//: Is so fun to do!
```

```
// Markup
```

```
/*:
```

```
    Roses are red
```

```
    Violets are blue
```

```
    Writing your markup
```

```
    Is so fun to do!
```

```
*/
```

```
1 //: Roses are red
2 //:
3 //: Violets are blue
4 //:
5 //: Writing your markup
6 //:
7 //: Is so fun to do!
```

```
Ready to continue RosesAreRed  
RosesAreRed  
1 //: Roses are red  
2 //:  
3 //: Violets are blue  
4 //:  
5 //: Writing your markup  
6 //:  
7 //: Is so fun to do!
```

Ready to continue RosesAreRed

RosesAreRed

```
1 //: Roses are red
2 //:
3 //: Violets are blue
4 //:
5 //: Writing your markup
6 //:
7 //: Is so fun to do!
```

▶

Identity and Type

Name **RosesAreRed**

Location **Absolute**

Full Path **/Users/lscs/Documents/RosesAreRed.playground**

Playground Settings

Platform **iOS**

Render Documentation

Ready to continue RosesAreRed

RosesAreRed

```
1 //: Roses are red
2 //:
3 //: Violets are blue
4 //:
5 //: Writing your markup
6 //:
7 //: Is so fun to do!
```

▶

Identity and Type

Name **RosesAreRed**

Location **Absolute**

Full Path **/Users/lscs/Documents/RosesAreRed.playground**

Playground Settings

Platform **iOS**

Render Documentation

Ready to continue RosesAreRed

RosesAreRed

```
1 //: Roses are red
2 //:
3 //: Violets are blue
4 //:
5 //: Writing your markup
6 //:
7 //: Is so fun to do!
```

▶

Identity and Type

Name RosesAreRed

Location Absolute

Full Path /Users/lscs/Documents/RosesAreRed.playground

Playground Settings

Platform iOS

Render Documentation

Ready to continue RosesAreRed

RosesAreRed

Roses are red  
Violets are blue  
Writing your markup  
Is so fun to do!

Identity and Type

Name RosesAreRed

Location Absolute

Full Path /Users/lscs/Documents/RosesAreRed.playground

Playground Settings

Platform iOS

Render Documentation

The image shows a screenshot of an Xcode playground window. The main area displays a poem: "Roses are red", "Violets are blue", "Writing your markup", and "Is so fun to do!". Below the poem is a play button icon. The right sidebar contains the "Identity and Type" section with fields for Name (RosesAreRed), Location (Absolute), and Full Path (/Users/lscs/Documents/RosesAreRed.playground). Below that is the "Playground Settings" section with Platform (iOS) and a checked checkbox for "Render Documentation". The window title bar shows "Ready to continue RosesAreRed" and standard macOS window controls.



```
// Markup
```

```
// Headings
```

```
//: # Roses Are Red
```

```
//: ## An ode to markup
```

```
//: ### By: Tibet Rooney–Rabdau
```

# Roses Are Red

An ode to markup

By: Tibet Rooney-Rabdau

Roses are red

Violets are blue

Writing your markup

Is so fun to do!



```
// Markup
// Formatting Characters

//: Roses are *red*
//:
//: Violets are *blue*
//:
//: Writing your `markup`
//:
//: Is so **fun** to do!
```

```
// Markup
// Formatting Characters

//: Roses are *red*
//:
//: Violets are *blue*
//:
//: Writing your `markup`
//:
//: Is so **fun** to do!
```

```
// Markup
// Formatting Characters

//: Roses are *red*
//:
//: Violets are *blue*
//:
//: Writing your `markup`
//:
//: Is so **fun** to do!
```

```
// Markup
// Formatting Characters

//: Roses are *red*
//:
//: Violets are *blue*
//:
//: Writing your `markup`
//:
//: Is so **fun** to do!
```

Roses are *red*

Violets are *blue*

Writing your markup

Is so **fun** to do!



```
// Markup
```

```
// Numbered Lists
```

```
//: 1. Roses are red
```

```
//: 2. Violets are blue
```

```
//: 3. Writing your markup
```

```
//: 4. Is so fun to do!
```



1. Roses are red
2. Violets are blue
3. Writing your markup
4. Is so fun to do!



```
// Markup
```

```
// Bulleted Lists
```

```
//: * Roses are red
```

```
//: * Violets are blue
```

```
//: * Writing your markup
```

```
//: * Is so fun to do!
```

- Roses are red
- Violets are blue
- Writing your markup
- Is so fun to do!



```
// Markup
```

```
// Links
```

```
//: [Roses](https://en.wikipedia.org/wiki/Rose) are red
```

```
//:
```

```
//: [Violets](https://en.wikipedia.org/wiki/Viola_(plant)) are blue
```

```
// Markup
```

```
// Links
```

```
//: [Roses](https://en.wikipedia.org/wiki/Rose) are red
```

```
//:
```

```
//: [Violets](https://en.wikipedia.org/wiki/Viola_(plant)) are blue
```

```
//:
```

```
//: Writing your markup
```

```
//:
```

```
//: Is so [fun][1] to do!
```

```
//:
```

```
//: [1]: https://en.wikipedia.org/wiki/Fun
```

Roses are red

Violets are blue

Writing your markup

Is so fun to do!



- New
- Add Files to "MyPlayground"...
- Open...
- Open Recent
- Open Quickly...
- Close Window
- Close Tab
- Close "MyPlayground.playground"
- Close Playground
- Save
- Duplicate...
- Revert to Saved...
- Unlock...
- Export...
- Open in New Tab
- Open in New Window
- Show in Finder
- Open with External Editor
- Save As Workspace...
- Playground Settings...
- Page Setup...
- Print...

- Tab
- Window
- File...
- Playground...
- Target...
- Project...
- Workspace...
- Playground Page

Ready | Today at 11:41 AM

MyPlayground

```
1 import UIKit
2
3 var str = "Hello, playground"
```

Filter

```
// Markup
// Navigating Between Pages

//: [Previous](@previous)
//:
//: [Next](@next)
//:
//: [Formatting Characters](Formatting%20Characters)
```



```
// Markup
```

```
// Navigating Between Pages
```

```
//: [Previous](@previous)
```

```
//:
```

```
//: [Next](@next)
```

```
//:
```

```
//: [Formatting Characters](Formatting%20Characters)
```

```
// Markup
// Navigating Between Pages

//: [Previous](@previous)
//:
//: [Next](@next)
//:
//: [Formatting Characters](Formatting%20Characters)
```

```
// Markup
// Navigating Between Pages

//: [Previous](@previous)
//:
//: [Next](@next)
//:
//: [Formatting Characters](Formatting%20Characters)
```

# Sources

Swift source files

Multiple levels

- Playground wide
- Per page

Compiled as separate modules

- Automatically imported
- Use access control

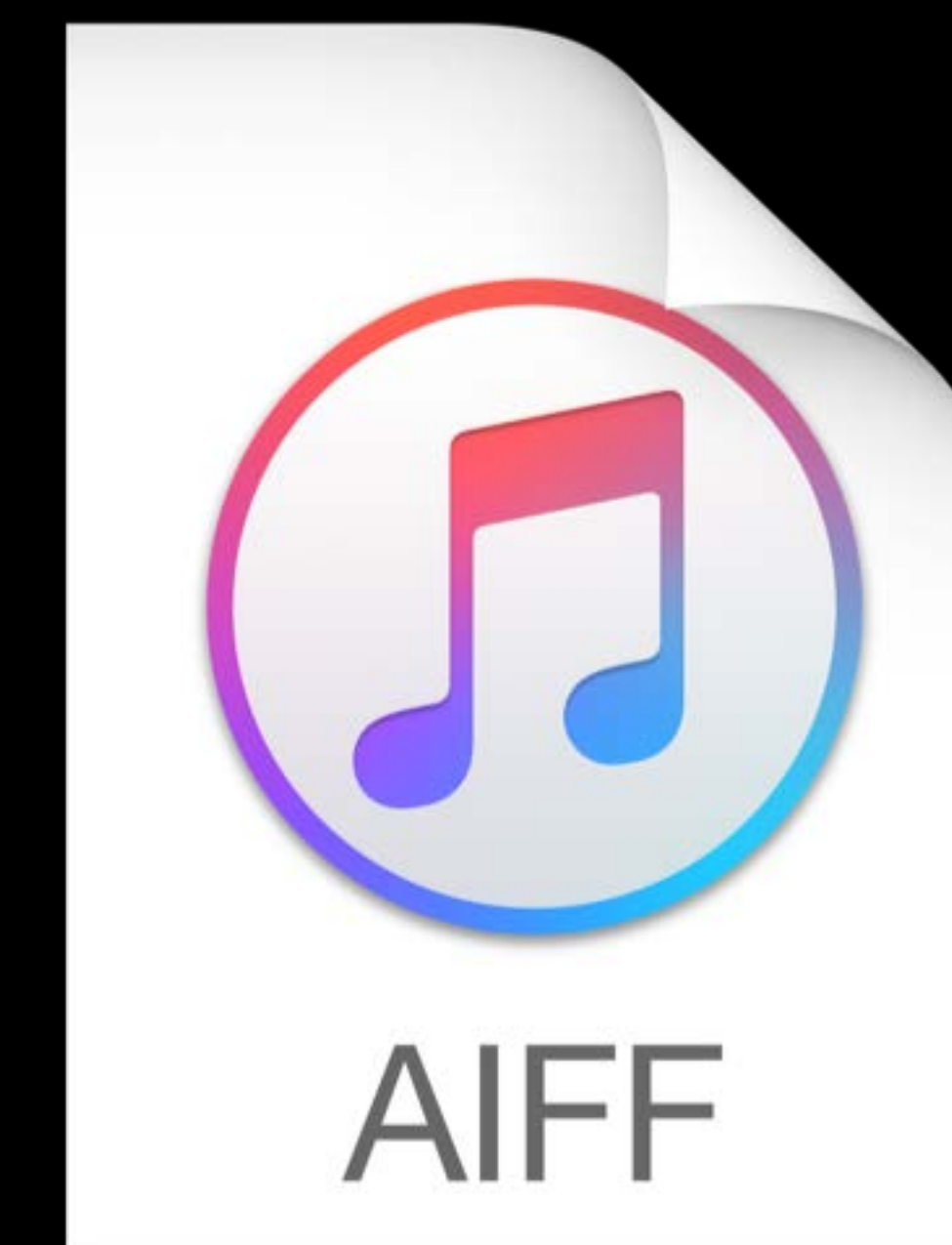


# Resources

Any file type

Multiple levels

- Playground wide
- Per page



```
// Using Resources in Playgrounds

//: You can embed an image in markup:
//: ![alternate text](MyPicture.jpg "hover title")

// You can also access the same image in code:
let image = UIImage(named: "MyPicture")

//: You can use videos in markup:
//: ![alternate text](MyVideo.mp4 poster="MyPoster.jpg" width="integer width"
//      height="integer height")

// You can find resources by using Bundle APIs:
let videoURL = Bundle.main.url(forResource: "MyVideo", withExtension: "mp4")
```

```
// Using Resources in Playgrounds
```

```
//: You can embed an image in markup:
```

```
//: ![alternate text](MyPicture.jpg "hover title")
```

```
// You can also access the same image in code:
```

```
let image = UIImage(named: "MyPicture")
```

```
//: You can use videos in markup:
```

```
//: ![alternate text](MyVideo.mp4 poster="MyPoster.jpg" width="integer width"  
height="integer height")
```

```
// You can find resources by using Bundle APIs:
```

```
let videoURL = Bundle.main.url(forResource: "MyVideo", withExtension: "mp4")
```

```
// Using Resources in Playgrounds
```

```
//: You can embed an image in markup:
```

```
//: ![alternate text](MyPicture.jpg "hover title")
```

```
// You can also access the same image in code:
```

```
let image = UIImage(named: "MyPicture")
```

```
//: You can use videos in markup:
```

```
//: ![alternate text](MyVideo.mp4 poster="MyPoster.jpg" width="integer width"  
height="integer height")
```

```
// You can find resources by using Bundle APIs:
```

```
let videoURL = Bundle.main.url(forResource: "MyVideo", withExtension: "mp4")
```



```
// Using Resources in Playgrounds
```

```
//: You can embed an image in markup:
```

```
//: ![alternate text](MyPicture.jpg "hover title")
```

```
// You can also access the same image in code:
```

```
let image = UIImage(named: "MyPicture")
```

```
//: You can use videos in markup:
```

```
//: ![alternate text](MyVideo.mp4 poster="MyPoster.jpg" width="integer width"  
height="integer height")
```

```
// You can find resources by using Bundle APIs:
```

```
let videoURL = Bundle.main.url(forResource: "MyVideo", withExtension: "mp4")
```

```
// Using Resources in Playgrounds
```

```
//: You can embed an image in markup:
```

```
//: ![alternate text](MyPicture.jpg "hover title")
```

```
// You can also access the same image in code:
```

```
let image = UIImage(named: "MyPicture")
```

```
//: You can use videos in markup:
```

```
//: ![alternate text](MyVideo.mp4 poster="MyPoster.jpg" width="integer width"  
height="integer height")
```

```
// You can find resources by using Bundle APIs:
```

```
let videoURL = Bundle.main.url(forResource: "MyVideo", withExtension: "mp4")
```

# More Information

Markup Formatting Reference  
Playground Support

[developer.apple.com](https://developer.apple.com)

NEW

# Running Step by Step

Alex Brown, Core OS Engineer

```
Ready | Today at 11:13 AM  
StepByStep  
1 "The Play button follows the pointer"  
2 "Blue means ready to execute"  
3 "Run to line..."  
4 do {  
5     "...to top-level statements only"  
6 }  
7 "⏪" // run to current line / cursor  
8 "Blue means..."  
▶ "Tap Stop (□) to reset"  
10
```

```
Ready | Today at 11:13 AM  
StepByStep  
1 "The Play button follows the pointer"  
2 "Blue means ready to execute"  
3 "Run to line..."  
4 do {  
5     "...to top-level statements only"  
6 }  
7 "⏪" // run to current line / cursor  
8 "Blue means..."  
9 "Tap Stop (⏹) to reset"  
10
```

```
Ready | Today at 11:13 AM  
StepByStep  
1 "The Play button follows the pointer"  
▶ "Blue means ready to execute"  
3 "Run to line..."  
4 do {  
5     "...to top-level statements only"  
6 }  
7 "⏪" // run to current line / cursor  
8 "Blue means..."  
9 "Tap Stop (⏹) to reset"  
10
```

```
Ready | Today at 11:13 AM  
StepByStep  
1 "The Play button follows the pointer"  
2 "Blue means ready to execute"  
3 "Run to line..."  
4 do {  
5     "...to top-level statements only"  
6 }  
7 "⏪" // run to current line / cursor  
8 "Blue means..."  
9 "Tap Stop (⏹) to reset"  
10
```



```
Ready to continue StepByStep
```

```
StepByStep
```

```
1 "The Play button follows the pointer"
2 "Blue means ready to execute"
3 "Run to line..."
4 do {
5     "...to top-level statements only"
6 }
7 "⏪" // run to current line / cursor
8 "Blue means..."
9 "Tap Stop (⏹) to reset"
10
```

```
"The Play button f...
"Blue means read...
"Run to line..."
```

```
Ready to continue StepByStep
```

```
StepByStep
1 "The Play button follows the pointer"
2 "Blue means ready to execute"
3 "Run to line..."
4 do {
5     "...to top-level statements only"
6 }
7 "⏪" // run to current line / cursor
8 "Blue means..."
9 "Tap Stop (⏹) to reset"
10
```

```
"The Play button f...
"Blue means read...
"Run to line..."
```

Ready to continue StepByStep

StepByStep

```
1 "The Play button follows the pointer"
2 "Blue means ready to execute"
3 "Run to line..."
4 do {
5     "...to top-level statements only"
6 }
7 "⏪" // run to current line / cursor
8 "Blue means..."
9 "Tap Stop (⏹) to reset"
10
```

"The Play button f...

"Blue means read...

"Run to line..."

Ready to continue StepByStep

StepByStep

```
1 "The Play button follows the pointer"
2 "Blue means ready to execute"
3 "Run to line..."
4 do {
5     "...to top-level statements only"
6 }
7 "⏪" // run to current line / cursor
8 "Blue means..."
9 "Tap Stop (⏹) to reset"
10
```

"The Play button f...

"Blue means read...

"Run to line..."

"...to top-level sta...

"⏪"

Ready to continue StepByStep

StepByStep

```
1 "The Play button follows the pointer"
2 "Blue means ready to execute"
3 "Run to line..."
4 do {
5     "...to top-level statements only"
6 }
7 "⏪" // run to current line / cursor
8 "Blue means edit without resetting"
9 "Tap Stop (⏹) to reset"
10
```

"The Play button f...

"Blue means read...

"Run to line..."

"...to top-level sta...

"⏪"

Ready to continue StepByStep

StepByStep

```
1 "The Play button follows the pointer"
2 "Blue means ready to execute"
3 "Run to line..."
4 do {
5     "...to top-level statements only"
6 }
7 "⏪" // run to current line / cursor
8 "Blue means edit without resetting"
9 "▶ Tap Stop (□) to reset"
10
```

"The Play button f...

"Blue means read...

"Run to line..."

"...to top-level sta...

"⏪"

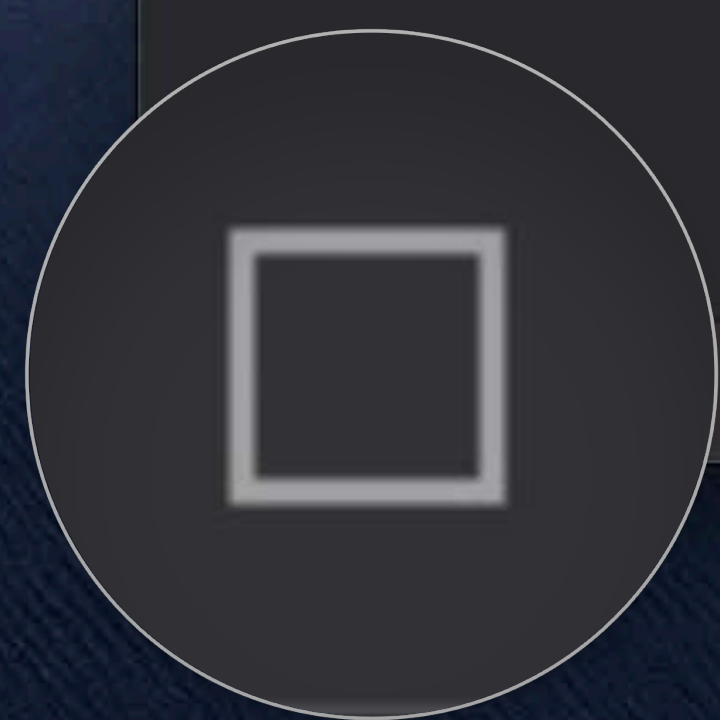
"Blue means edit ...

```
Ready to continue StepByStep
```

```
StepByStep
```

```
1 "The Play button follows the pointer"
2 "Blue means ready to execute"
3 "Run to line..."
4 do {
5     "...to top-level statements only"
6 }
7 "⏪" // run to current line / cursor
8 "Blue means edit without resetting"
9 "▶ Tap Stop (□) to reset"
10
```

```
"The Play button f... 
"Blue means read... 
"Run to line..." 
"...to top-level sta... 
"⏪" 
"Blue means edit ... 
```



Ready | Today at 11:15 AM

StepByStep

```
1 "The Play button follows the pointer"
2 "Blue means ready to execute"
3 "Run to line..."
4 do {
5     "...to top-level statements only"
6 }
7 "⏪" // run to current line / cursor
8 "Blue means edit without resetting"
9 "Tap Stop (⏹) to reset"
10
```

"The Play button f...

"Blue means read...

"Run to line..."

"...to top-level sta...

"⏪"

"Blue means edit ...



# Why Run Step by Step?

Execute one more line—fast

Respond to live data

Re-running can give you different values

Ready to continue RoShamBo

```
1 let roshambo = ["rock", "paper", "scissors"]
2 func check(_ left: String,
3           beats right: String) -> Bool {
4     return (3 + roshambo.firstIndex(of: left)! - roshambo.firstIndex(of:
5           right)!) % 3 == 1
6 }
7 let computerMove = roshambo.shuffled().first!
8
9 check("paper", beats: computerMove)
```

["rock", "paper", "...

rock

rock

***Demo***

# Running Step by Step

Explore your idea a line at a time

Have a conversation between code and data

Use Shift-Return to run up to current line

Interact with dynamic live views

# Step by Step Suggestions

Create a follow-along tutorial for your API

# Step by Step Suggestions

Create a follow-along tutorial for your API

Explore data from maps, public statistics, or a class project

# Step by Step Suggestions

Create a follow-along tutorial for your API

Explore data from maps, public statistics, or a class project

Create and play a game or animation on the fly

# Playgrounds Are Serious Fun

Explore code and data interactively

Learn and use Apple APIs

Import your Frameworks

Display focused results



# Advanced Techniques

TJ Usiyan, Xcode Engineer

# Advanced Techniques

Custom Playground Display Convertible

Using custom frameworks in playgrounds

Troubleshooting

# Advanced Techniques

Custom Playground Display Convertible



Using custom frameworks in playgrounds


Troubleshooting

Ready to continue MyPlayground

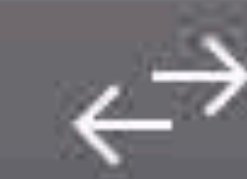
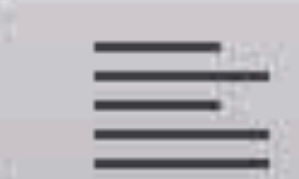
MyPlayground

```
1 import UIKit
2
3 var str = "Hello, playground"
```

"Hello, playgro...  



Ready to continue MyPlayground

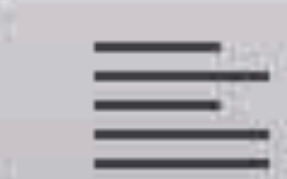


d"

"Hello, playgro...



Ready to continue MyPlayground

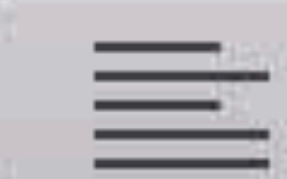


d"

"Hello, playgro...



Ready to continue MyPlayground

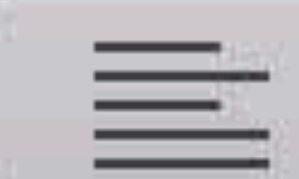


d"

"Hello, playgro.



Ready to continue MyPlayground



d"

"Hello, playgro...





```
20 let a4 = Pitch(.a, 4)
```

A4



"A4"



```
20 let a4 = Pitch(.a, 4)
```

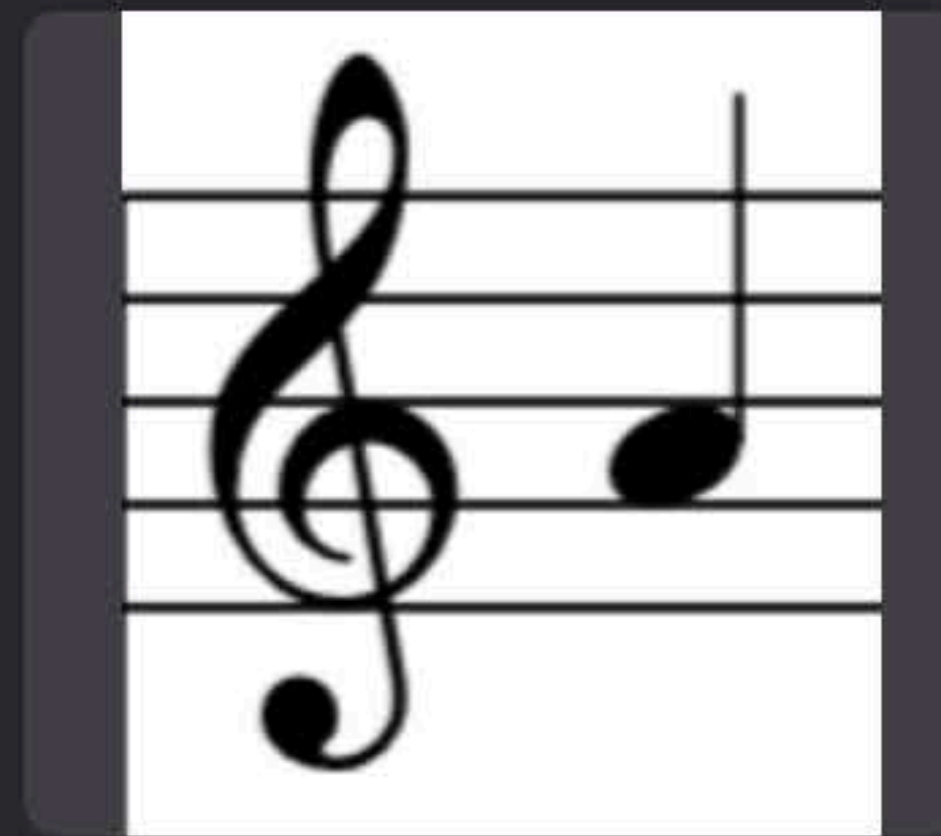
440.0



440



```
20 let a4 = Pitch(.a, 4)
```



w 140 h 

# Custom Playground Display Convertible



NEW

New `CustomPlaygroundDisplayConvertible` protocol

Introduced in Xcode 9.3 and Swift 4.1

Replaces `CustomPlaygroundQuickLookable` (Deprecated in Xcode 9.3)

# Custom Playground Display Convertible

```
extension MyType: CustomPlaygroundDisplayConvertible {  
    var playgroundDescription: Any { ... }  
}
```

# Custom Playground Display Convertible

Supported types in Xcode 9.3/Swift 4.1 and later

## Textual

String, NSString

Int, UInt (including the sized variants)

Float, Double

Bool

Date, NSDate

NSAttributedString

NSNumber

NSRange

## Graphical

URL, NSURL

CGPoint

CGSize

CGRect

NSColor, UIColor, CGColor, UIColor

UIImage, UIImage, CGImage, CIImage

NSBezierPath, UIBezierPath

NSView, UIView

# Advanced Techniques

Custom Playground Display Convertible

Using custom frameworks in playgrounds

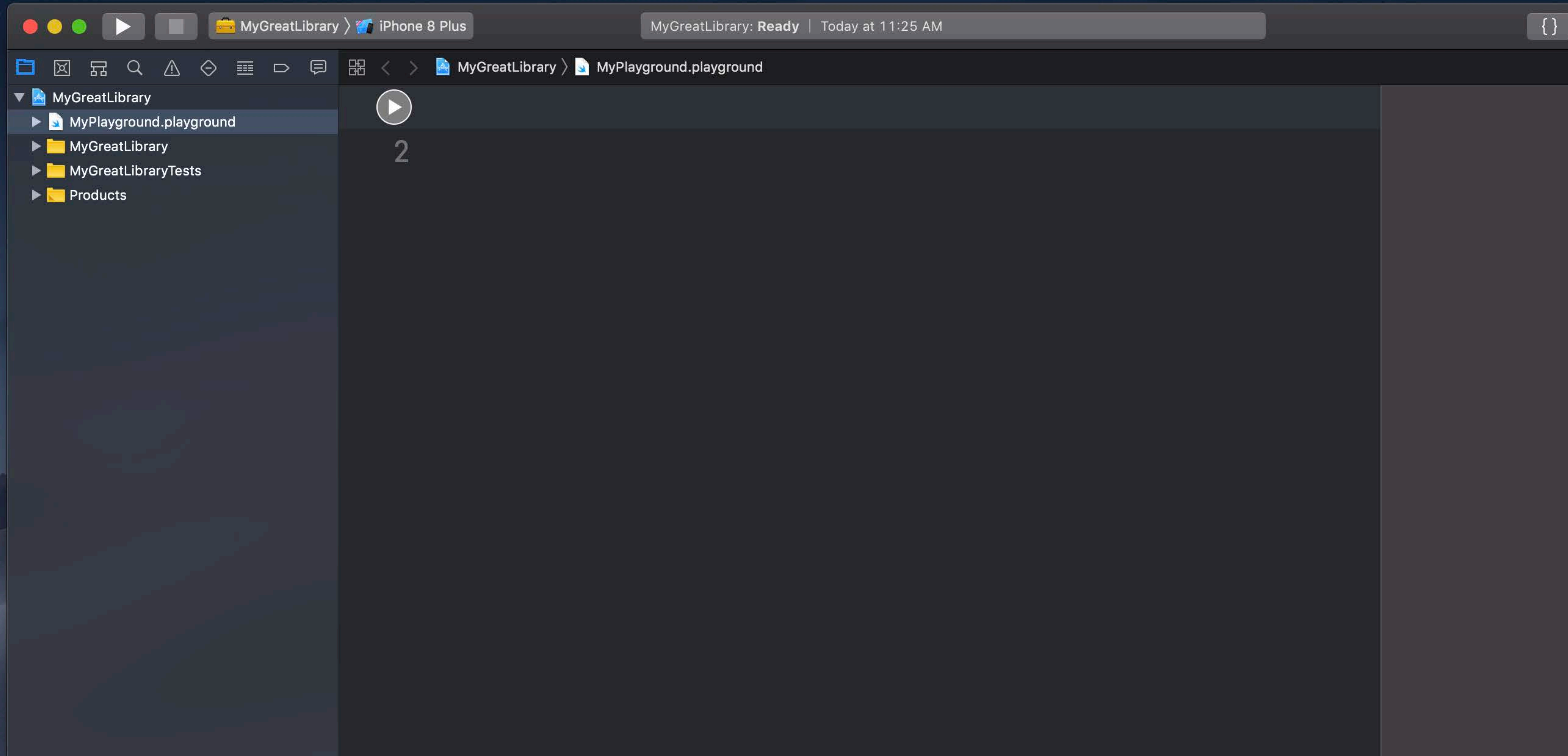
Troubleshooting



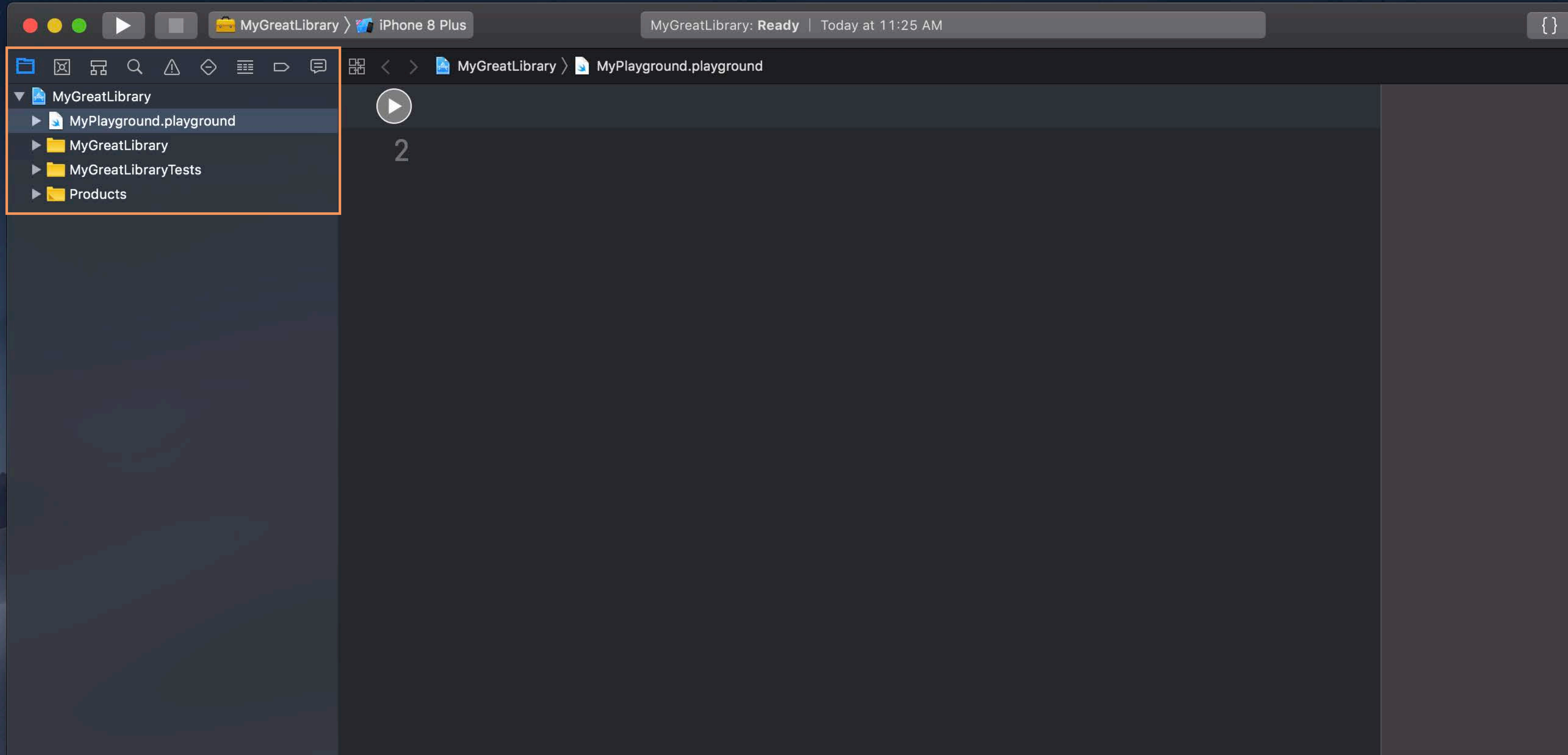




# Importing Your Code in Playgrounds



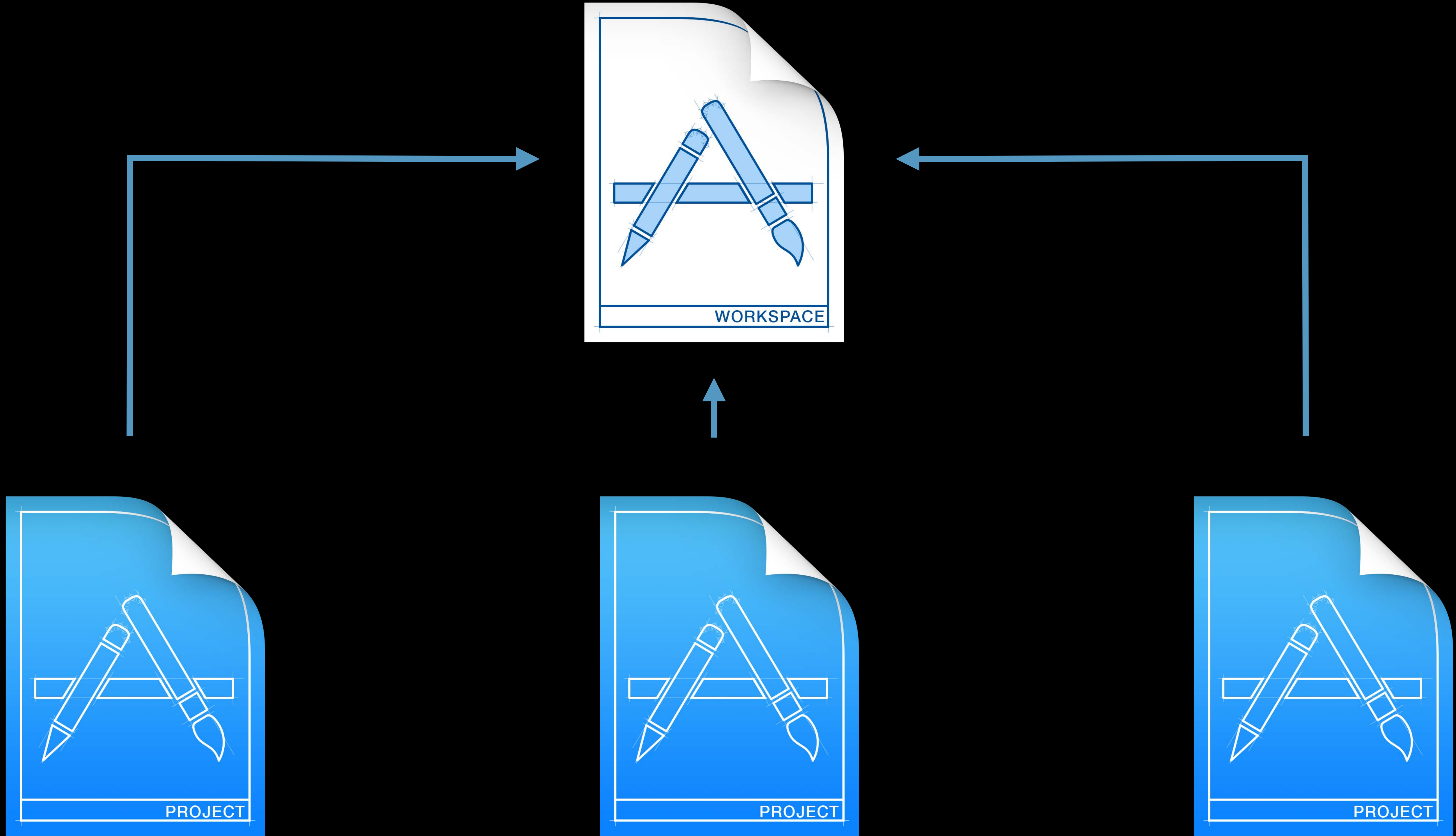
# Importing Your Code in Playgrounds

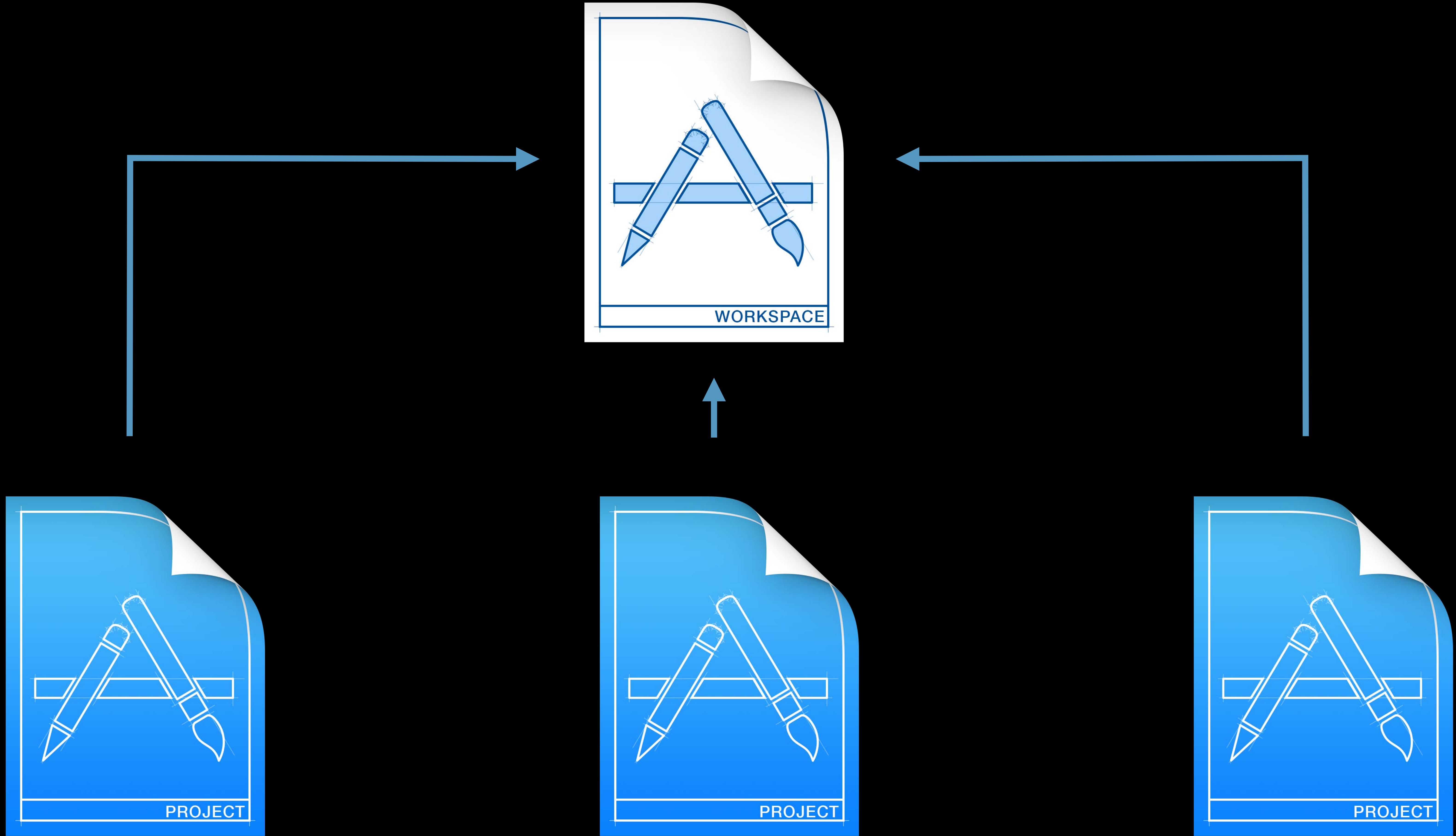


Remember to build before you run

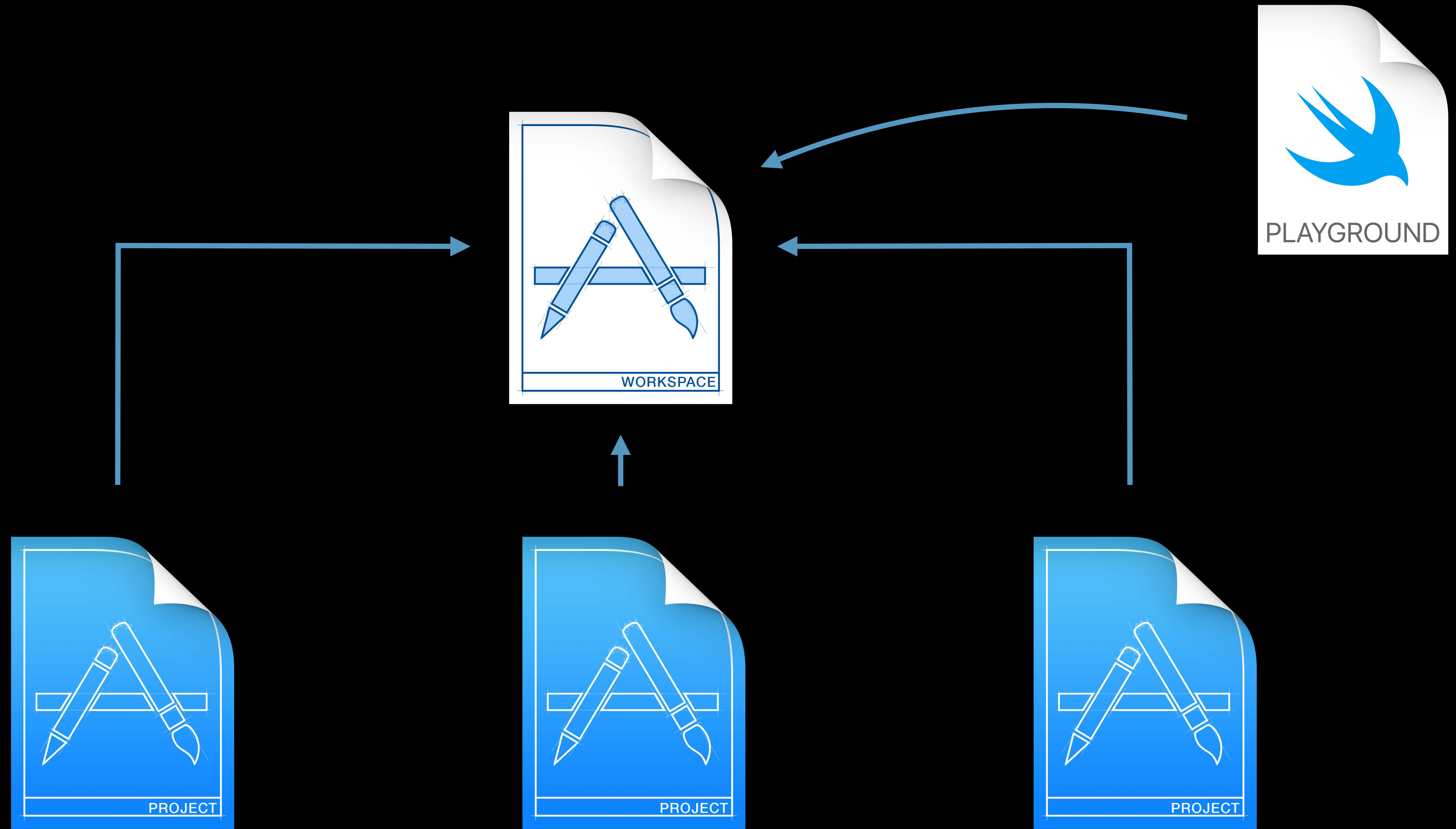




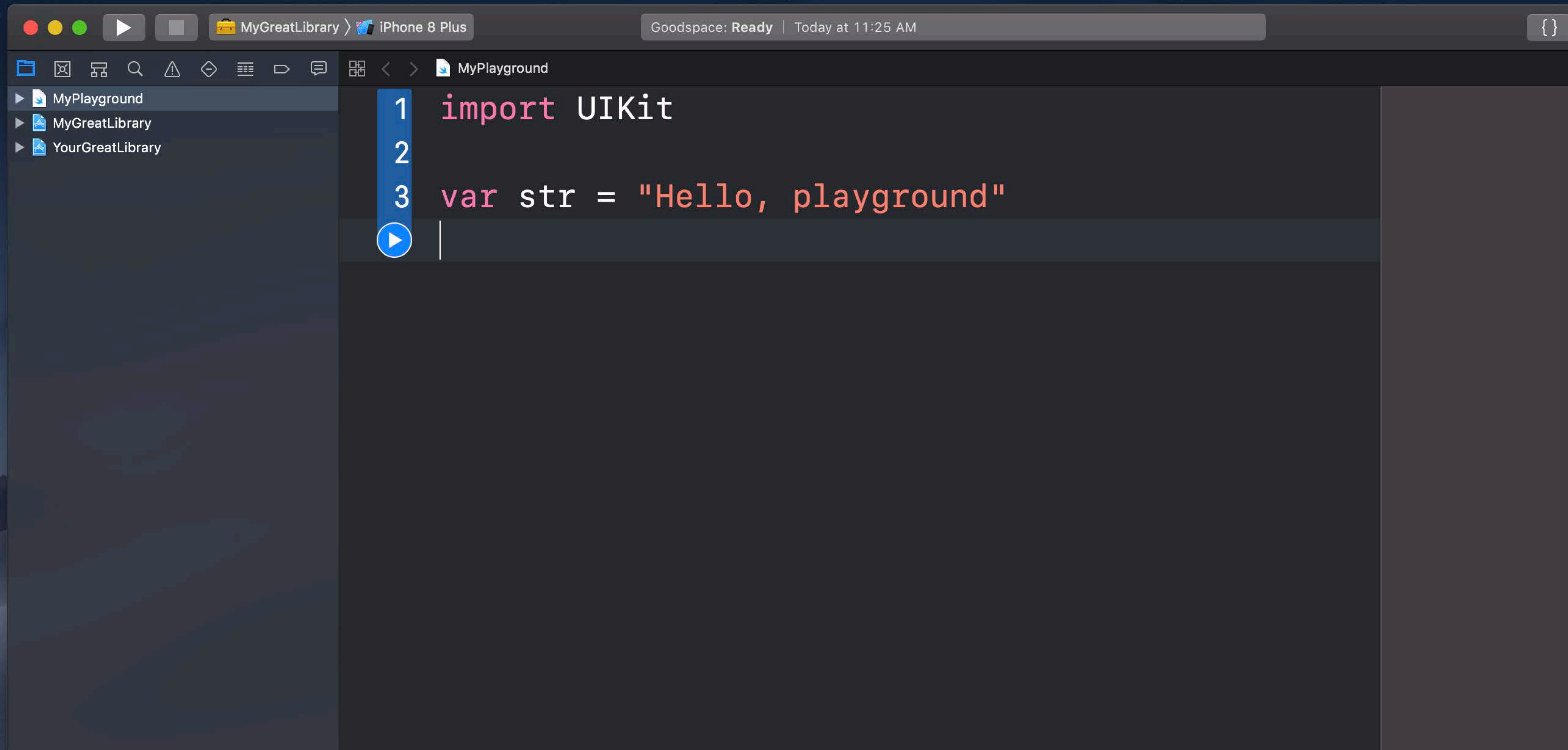




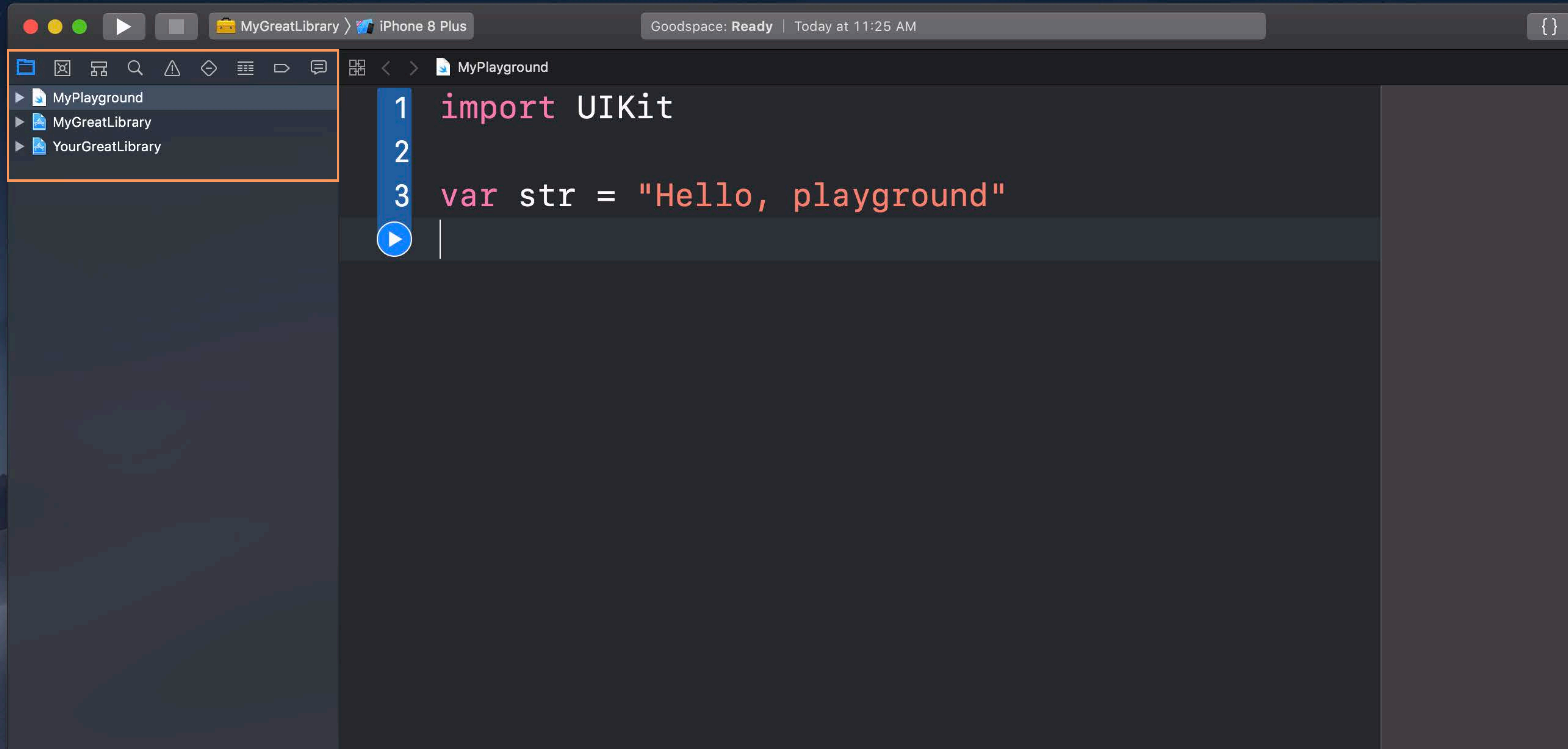




# Importing Your Code in Playgrounds



# Importing Your Code in Playgrounds



- New ▶
- Add Files to "MyPlayground.playground"... ⌘⌘A
- Open... ⌘O
- Open Recent ▶
- Open Quickly... ⌘⇧O
- Close Window ⌘⇧W
- Close Tab ⌘W
- Close "MyPlayground.playground" ⌘⇧W
- Close Project ⌘⇧⇧W
- Save ⌘S
- Duplicate... ⌘⇧S
- Revert to Saved...
- Unlock...
- Export...
- Open in New Tab
- Open in New Window
- Show in Finder
- Open with External Editor
- Save As Workspace...
- Project Settings...**
- Page Setup... ⌘⇧P
- Print... ⌘P

MyGreatLibrary: Ready | Today at 5:10 PM

MyGreatLibrary > MyPlayground.playground

1

Filter

MyGreatLibrary > iPhone 8 Plus | MyGreatLibrary: Ready | Today at 5:10 PM

MyGreatLibrary

- MyPlayground.playground
- MyGreatLibrary
- MyGreatLibraryTests
- Products

Shared Project Settings:

Build System: New Build System (Default)

Per-User Project Settings:

Build System: Use Shared Setting

Derived Data: Default Location

/Users/lscs/Library/...r/Xcode/DerivedData

Advanced...

Issues:

- Show live issues for source code
- Show issues for active scheme only
- Show all issues

Done

MyGreatLibrary > iPhone 8 Plus | MyGreatLibrary: Ready | Today at 5:11 PM

MyGreatLibrary

- MyPlayground.playground
- MyGreatLibrary
- MyGreatLibraryTests
- Products

**Build Location**

- Xcode Default**  
Use the location specified in Xcode preferences
- Unique**  
In a unique subfolder of Xcode's Derived Data location
- Shared Folder** \_\_\_\_\_  
In a shared subfolder of Xcode's Derived Data location
- Custom** **Relative to Derived Data** ▾  
Fully specify locations of build products and intermediates
  - Products \_\_\_\_\_
  - Intermediates \_\_\_\_\_
  - Index Datastore \_\_\_\_\_
- Legacy**  
Use target settings (used for legacy projects that cannot be relocated)
  - Products: ~/Library/Developer/Xcode/Derive...nzdopgmorqahhmb/Build/Products ↕
  - Intermediates: ~/Library/Developer/Xcode/Derive...hhmbe/Build/Intermediates.noindex ↕
  - Index Datastore: ~/Library/Developer/Xcode/Derive...dopgmorqahhmb/Index/DataStore ↕
- Full Index Datastore Visibility**  
Include index data information for files not referenced from the workspace

Done

Filter

MyGreatLibrary > iPhone 8 Plus | MyGreatLibrary: Ready | Today at 5:11 PM

MyGreatLibrary

- MyPlayground.playground
- MyGreatLibrary
- MyGreatLibraryTests
- Products

### Build Location

- Xcode Default  
Use the location specified in Xcode preferences
- Unique  
In a unique subfolder of Xcode's Derived Data location
- Shared Folder  
In a shared subfolder of Xcode's Derived Data location
- Custom Relative to Derived Data  
Fully specify locations of build products and intermediates
  - Products
  - Intermediates
  - Index Datastore
- Legacy  
Use target settings (used for legacy projects that cannot be relocated)

Products: ~/Library/Developer/Xcode/Derive...nzdopgmorqahhmb/Build/Products ↗

Intermediates: ~/Library/Developer/Xcode/Derive...nzdopgmorqahhmb/Build/Intermediates.noindex ↗

Index Datastore: ~/Library/Developer/Xcode/Derive...nzdopgmorqahhmb/Index/DataStore ↗

Full Index Datastore Visibility  
Include index data information for files not referenced from the workspace

Done

Products

Search

Favorites

- AirDrop
- Recents
- Applications
- Desktop
- Documents
- Downloads

Locations

Tags

- Red
- Orange
- Yellow
- Green
- Blue
- Purple
- Gray
- All Tags...

ModuleCache.noindex

MyGreatLibr...morqahhmbe

Planets-gvbz...atlygtmvtz

Build

- Index
- info.plist
- Logs
- scm.plist
- TextIndex

Intermediates.noindex

- Products

Debug-iphonesimulator



***Demo***

Fundamentals

Running step by step

Advanced techniques

A Playground in Every Project!

# More Information

<https://developer.apple.com/wwdc18/402>

 **WWDC18**