

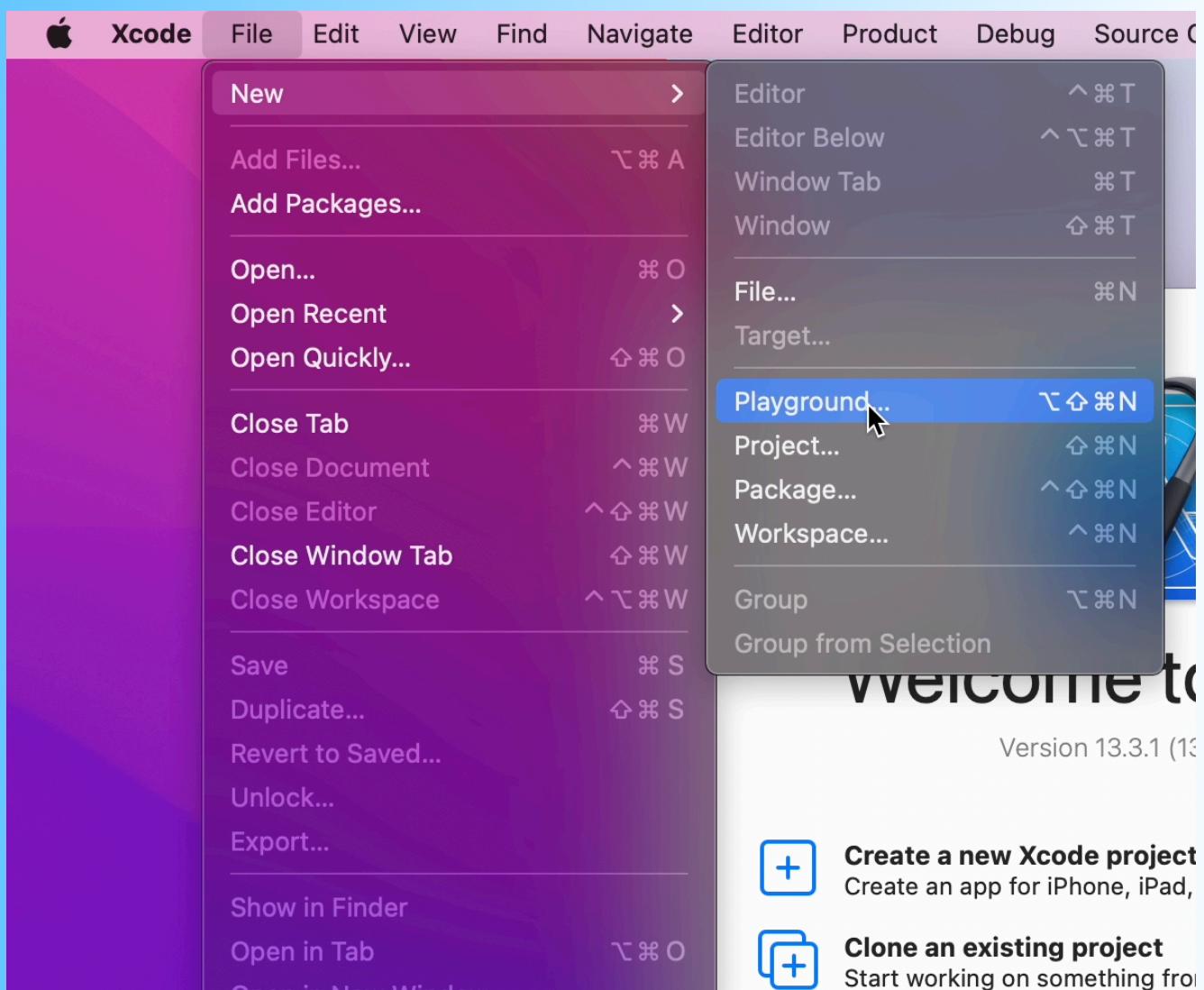
Introduction to XCode and Swift Playgrounds

XCODE

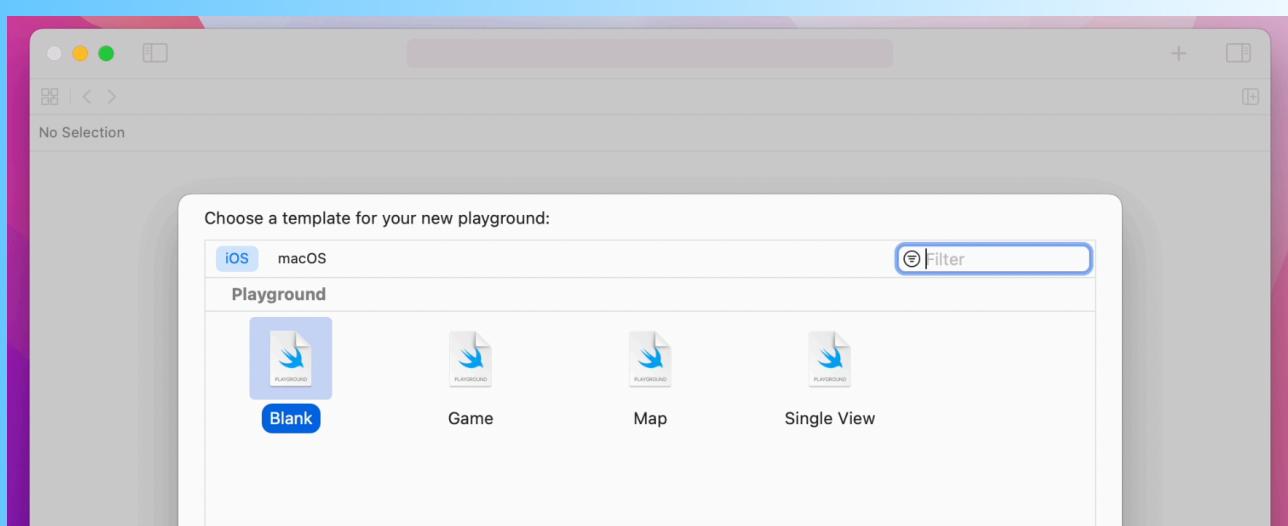
1. First Download Xcode from App Store, then Open it.



2. Then, click **File -> New -> Playground**

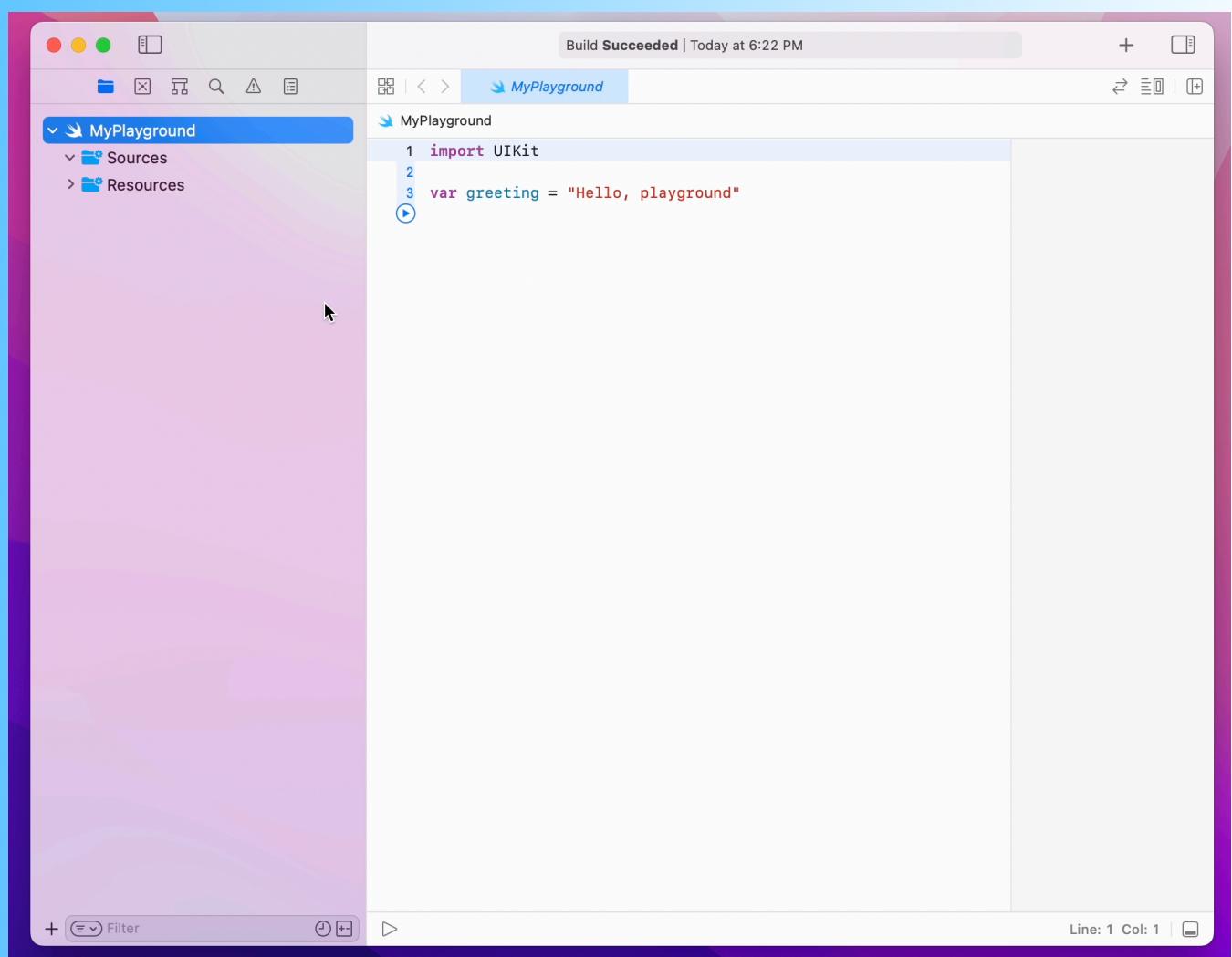


3. Click on **Blank** file and then **Next**.



4. Now give name to playground file and **Save** accordingly. Eg. *main.playground*

5. “On a technical level, a playground is a file wrapper around a *main.swift* file. Every time you edit the code, the playground runs the results. You can include additional files and resources, which you’ll do in a future activity.



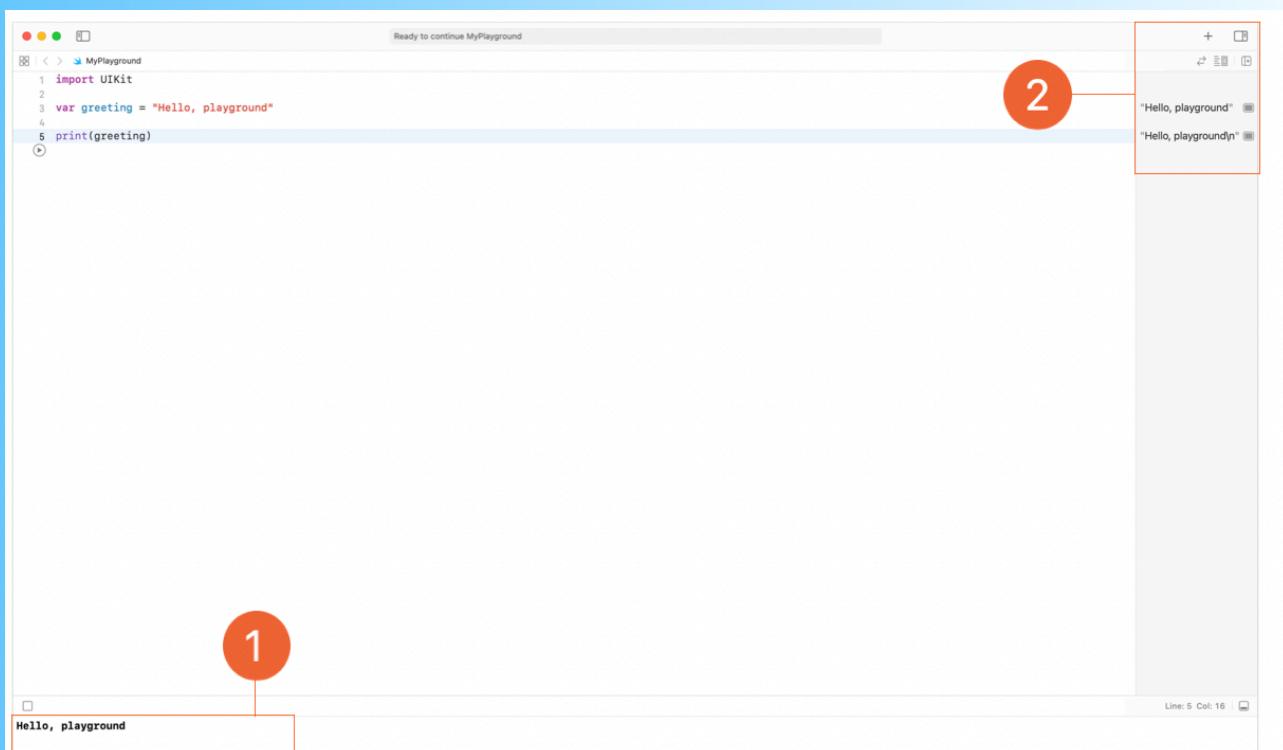
The screenshot shows the Xcode interface with a playground project open. The left sidebar displays the project structure with 'MyPlayground' selected. The main editor area shows the following Swift code:

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

The status bar at the bottom right indicates 'Line: 1 Col: 1'.

6. To view printed results from your Swift code, you can open a debug console area in a playground. [1]

Remember, every time you edit the Swift code, the code is run from top to bottom. For every expression, results are shown on the right side.[2]



Did you notice that the default playground comes with an import statement and the variable var greeting = “Hello, playground”? Open the results sidebar (if it’s not already displayed) and note that the string value is printed in the sidebar.

The import UIKit statement allows you to use anything from the UIKit framework. This will become more important as you work through the labs in this course. You should typically leave the import statement in all of your playgrounds.

Hello World

When you start with a new programming language, it's a time-honored tradition to build a "Hello, world!" app, one of the simplest programs you can write in any language. All this program does is print "Hello, world!" to the screen.

Swift code is written in plain text files with a `.swift` file extension. Each line in the file represents a statement, and a program is made up of one or more statements. These are the instructions you wish your app to run. Generally, code is executed starting at the top of the file and works its way to the bottom of the file.

As you'll learn, you can control whether specific sections of code are executed using control flow statements (`if`, `else`), how many times they're executed using loops (`for-in`, `while`), and how to use data that can be passed to different statements.

Some programs are made up of millions of statements spread across thousands and thousands of files. The compiler makes the code executable by combining all the files into a program.

But for now, let's keep it simple.

In Swift, the default file is called `main.swift`. Any Swift code included in the `main.swift` file will be executed from top to bottom.

So how would you write a "Hello, world!" app in Swift?

```
print("Hello, world!")
```