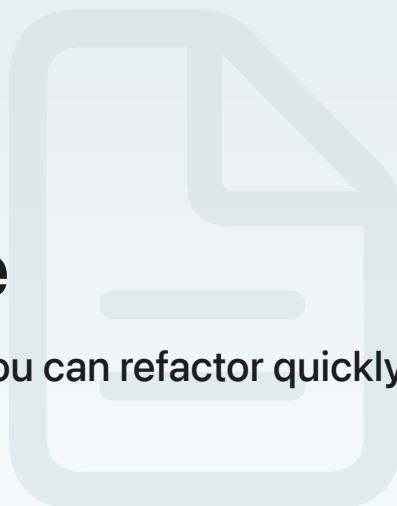


[Xcode](#) / [Source Editor](#) / Finding and refactoring code

Article

Finding and refactoring code

Search your code for text, patterns, and symbols that you can refactor quickly and easily.



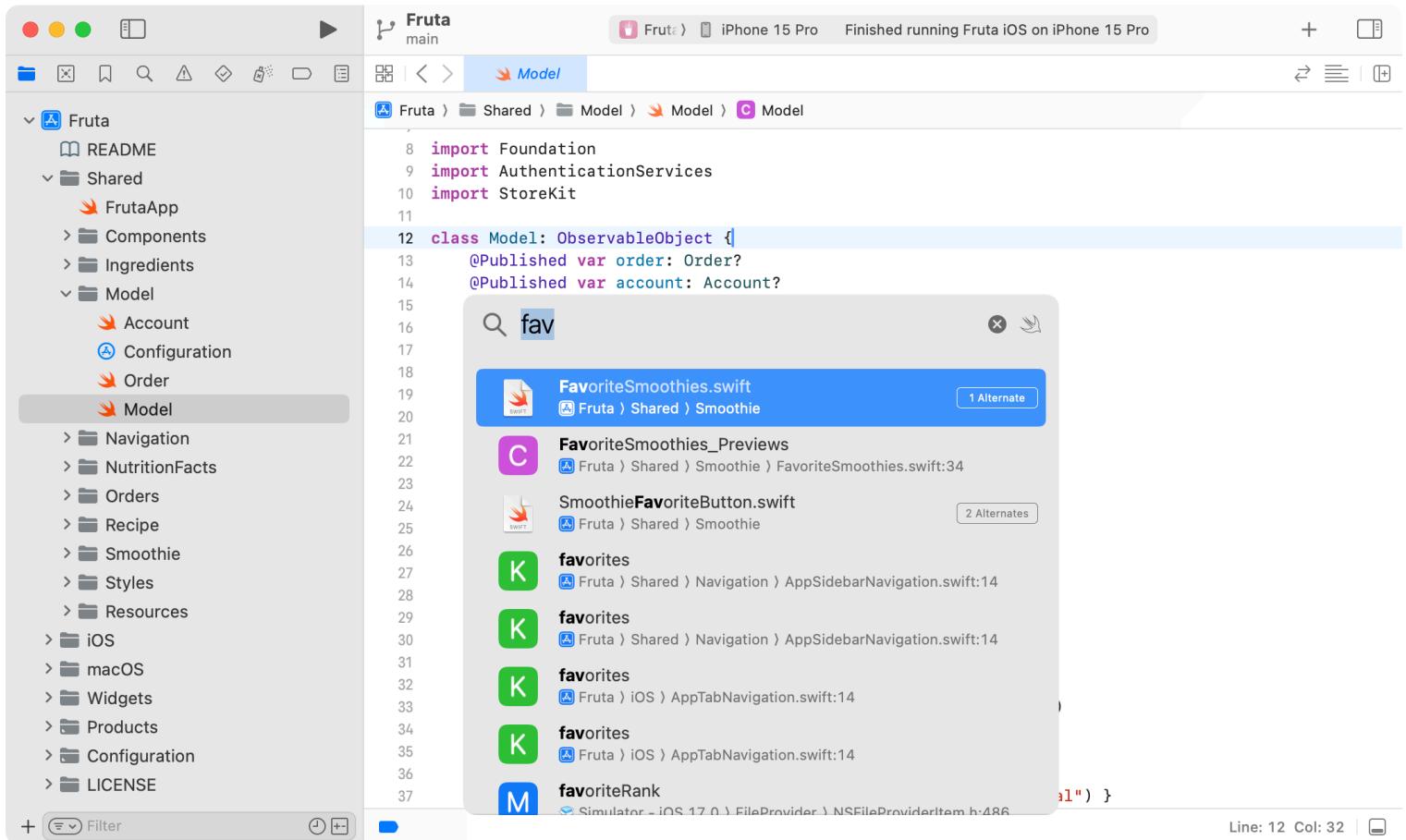
Overview

When writing and analyzing code, you frequently need to look up a symbol to learn how to use it and to determine how other code uses it. Use Xcode's search capabilities to find text and symbols in your code to analyze potential changes, determine current functionality, or debug code.

Rename files or symbols to clarify usage with a more descriptive and meaningful name. Refactor your code when you identify duplicate code or code that you can reuse. Transform existing code into more reusable functions, methods, or variables.

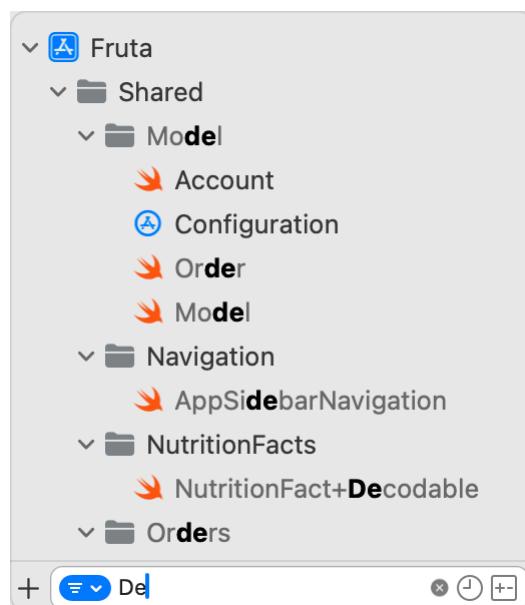
Find files by name or included symbols

Open a file quickly by pressing Shift-Command-O to bring up the Open Quickly dialog. Type in part of a filename or symbol name, and the dialog presents a list of potential matches.



Select an item from the search results to open it and navigate to the matched symbol or file. If the file is already open in a tab, Xcode switches to that tab, otherwise, Xcode opens the file in a new tab.

When there are a lot of files in your project, type part of the filename into the Filter field at the bottom of the Project navigator. Xcode displays matching group names and filenames with their parent groupings so you can easily see their locations in your project.

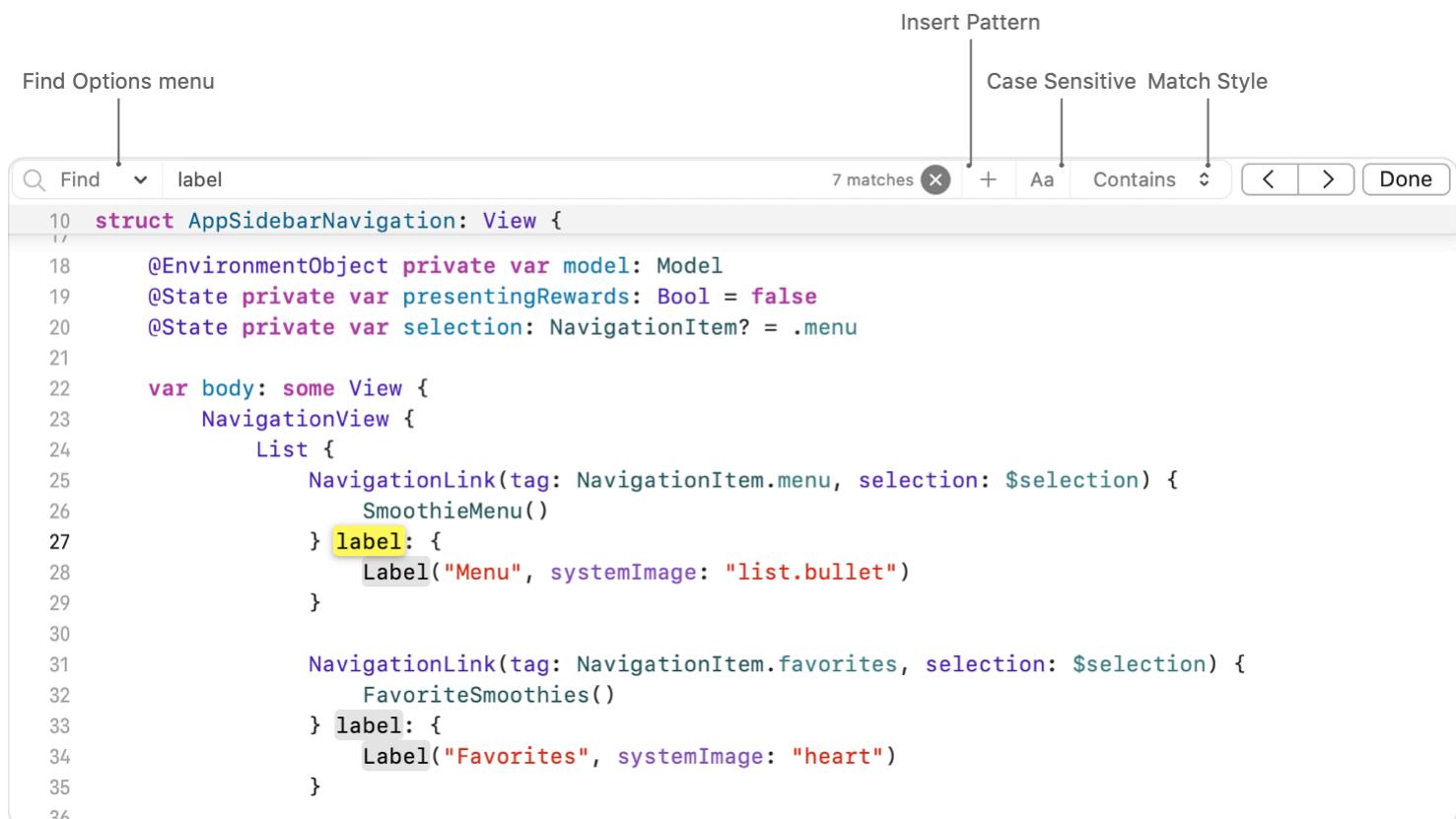


To find a symbol in your source code, open the Symbol navigator and enter part of a symbol name into the Filter field at the bottom.

Xcode matches the text you enter to the names of the class, structure, enumeration, function, method, and variable entities in your code, and displays those results in the navigator. Click a matching result to navigate directly to the definition of the symbol in your source code. Use the options in the Filter field to limit or expand your search by symbol type, to indicate whether the symbol's definition is in your source or in the SDKs, to choose whether to limit the results to containers (like classes or structures), or to show members (like variables and functions).

Find text or patterns in your source code

To find text in a file, open the file in the Xcode source editor and choose Find > Find from the menu bar. Xcode displays the Find bar and its search controls at the top of the file.



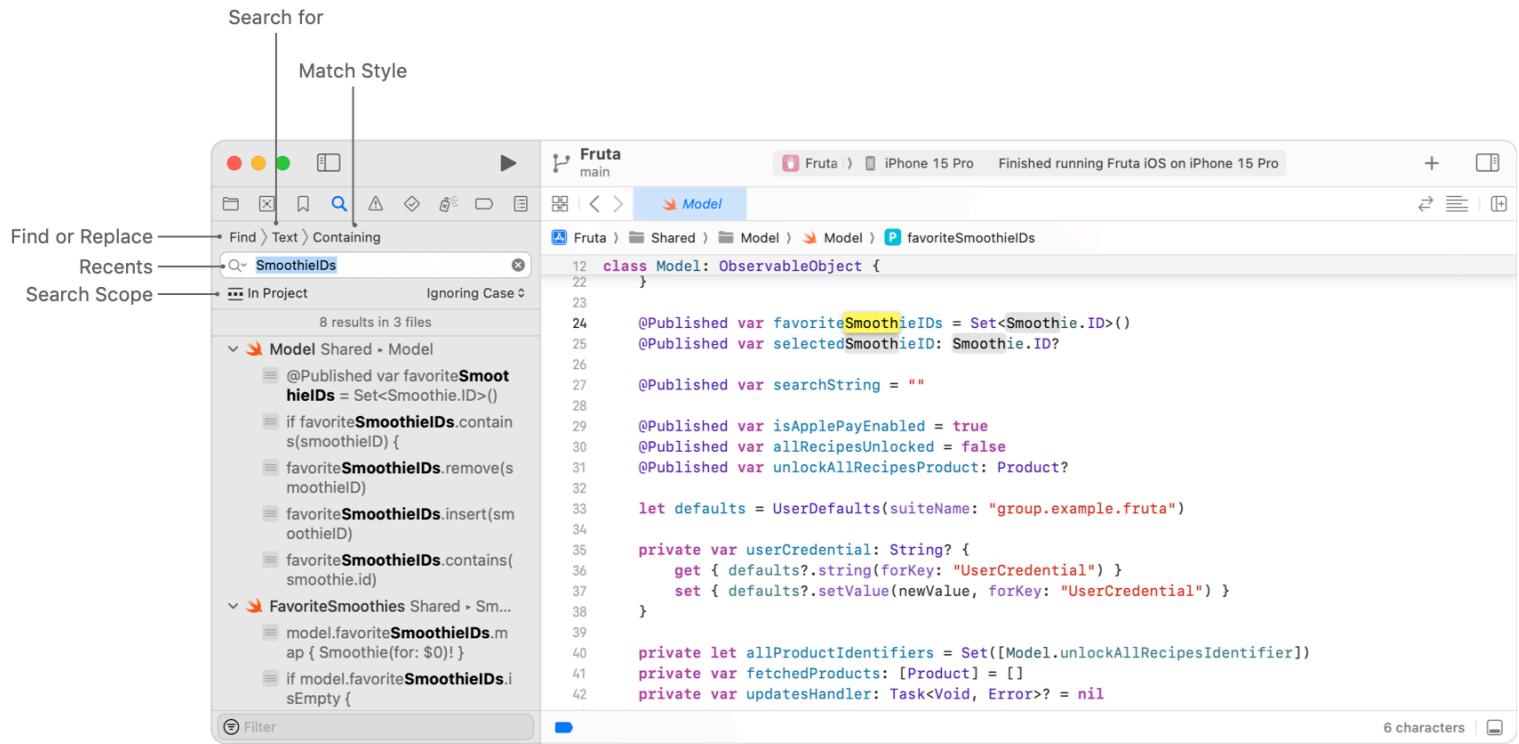
Enter a search term. Xcode searches the file, highlights matches, and notes how many it finds. Refine your search using the following options:

- Click the Find Options menu on the left to display options to replace text, run recent searches, or clear recent search history.
- Click the Insert Pattern button (+) to include tabs, line breaks, or special characters in your search term.
- Toggle the Case Sensitive button to indicate whether you want Xcode to match the case of your search term.
- Click the Match Style button to customize how Xcode looks for your search term in your source code.

- Navigate between matches using the Next and Previous buttons.
- Close the Find bar by clicking the Done button.

Find symbols in your source code

To find a symbol in your code, Control-click the variable or function name, and choose Find > Find Selected Symbol in Workspace.



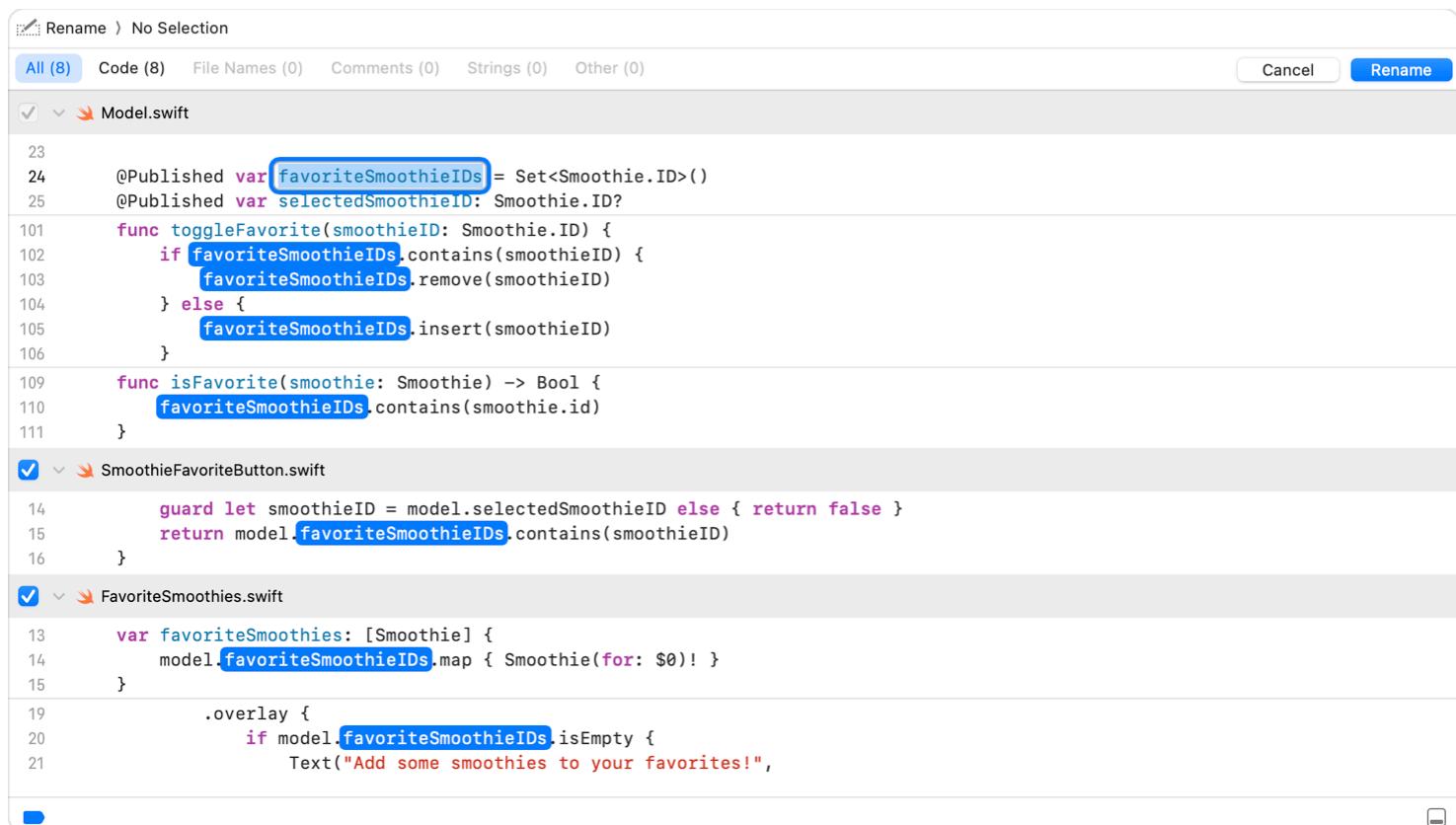
Xcode displays the declaration of the symbol in the Find navigator, along with any places where your code references the symbol. Refine your search using the following options:

- Choose Find or Replace.
- Choose what you want to search for: text, symbol references, symbol definitions, text that matches regular expressions, or call hierarchy.
- Select a match style to customize how Xcode looks for your search term.
- Click the magnifying glass icon to view recent searches.
- Customize the scope of your search: in your project, or in any groups/folders inside your project. If you're using a workspace, you can search across the workspace, or inside a project.
- Select Ignoring Case or Matching Case.

If your search returns a large number of matches, narrow the results with another term in the Filter field.

Rename symbols throughout your project

To rename a function, method, class, structure, or enumeration in your project, Control-click either the declaration of the symbol or a use of the symbol, and choose Refactor > Rename. Xcode highlights the symbol, searches your project for its name, and shows everywhere that the symbol appears.



The screenshot shows the Xcode 'Rename' dialog with the following details:

- Title Bar:** Shows 'Rename > No Selection'.
- Filter Bar:** Shows 'All (8)' (selected), 'Code (8)', 'File Names (0)', 'Comments (0)', 'Strings (0)', and 'Other (0)'. Buttons for 'Cancel' and 'Rename' are at the right.
- File List:** Shows three files with checked status:
 - Model.swift:** Contains code with several instances of 'favoriteSmoothieIDs' highlighted in blue.
 - SmoothieFavoriteButton.swift:** Contains code with one instance of 'favoriteSmoothieIDs' highlighted in blue.
 - FavoriteSmoothies.swift:** Contains code with one instance of 'favoriteSmoothieIDs' highlighted in blue.
- Code Preview:** Shows the Swift code from the selected files with the highlighted symbols.

Type a new name into the highlighted selection and Xcode previews all the changes. Click a proposed renaming instance to toggle whether Xcode renames it. Click Rename to complete the changes, or Cancel to not make the changes.

To rename a local variable or instance variable, Command-click the variable and choose Edit All in Scope. Xcode highlights instances of the variable in scope in the source editor. Type a new name and Xcode updates all instances to the same name.

Refactor code into functions

When you have code that repeats in a function or code that you can reuse, refactor it into a function. Select the lines of code you want to refactor, then Control-click and choose Refactor > Extract to Method. Xcode creates a new function, and highlights its name so you can rename it.

```
21  
22     fileprivate func extractedFunc(_ fruitItem: Fruit) {  
23         fruitDetailView.nameLabel.text = fruitItem.fruitName  
24         fruitDetailView.detailLabel.text = fruitItem.fruitDescription  
25     }  
26  
27     override func viewWillAppear(_ animated: Bool) {  
28         super.viewWillAppear(animated)  
29  
30         guard let fruitItem = fruit else {  
31             return  
32         }  
33  
34         extractedFunc(fruitItem)  
35     }  
36
```

If the lines of code reference parameters, Xcode includes those in the parameter list for the new function. To rename a parameter, Command-click it and choose Edit All in Scope, or Control-click it and choose Refactor > Rename.