## Source control ⚀

**Definition / example**

A system that keeps a history of changes so you can roll back to earlier versions and recover work if something goes wrong.

## Repository (repo) ⚁

**Definition / example**

A project “home” that stores your files and their change history in one place.

## Local repository ⚂

**Definition / example**

The project history that lives on your own computer while you work day-to-day.

## Remote repository (remote) ⚃

**Definition / example**

An off-computer copy of the project history stored on another machine (often a cloud service) for backup and sharing.

## git ⚄

**Definition / example**

The tool that records versions and helps coordinate changes across computers or collaborators.

## Commit ⚅

**Definition / example**

Creating a named checkpoint in the project’s history that captures your changes at a moment in time.

## Push ⚀

**Definition / example**

Uploading your new checkpoints so the off-computer copy includes your latest work.

## Pull ⚁

**Definition / example**

Downloading updates so your computer has the newest changes from the off-computer copy.

## Preview (Canvas) ⚂

**Definition / example**

Shows just the contents of the single view/file you are working on so you can rapidly iterate on its interface or logic.

## Simulator ⚃

**Definition / example**

Runs the entire application in a virtual device so you can test navigation, multiple screens, and full app behaviour.

## App entry point (@main) ⚄

**Definition / example**

The “starting place” Swift uses to launch the app; this is where the first screen shown to the user is created.

## VStack / HStack / ZStack ⚅

**Definition / example**

Layout containers that arrange views top-to-bottom, left-to-right, or in overlapping layers.

### Push-out vs pull-in views ⚀

**Definition / example**

Some views expand to fill available space, while others stay only as large as their content; mixing them affects how layouts “stretch” or “hug.”

### Spacer ⚁

**Definition / example**

An invisible element that expands to take up leftover room, forcing other content into position.

### View modifier (.containerRelativeFrame) ⚂

**Definition / example**

A chained method you apply to a view to change layout; this one lets you divide available space into equal parts and claim a chosen number of those parts.

## @State ⚃

**Definition / example**

Marks stored data that can change; when it changes, SwiftUI automatically refreshes what’s on screen to match.

## Optional (e.g., String?) ⚄

**Definition / example**

A variable that may contain a real value or represent “no value provided yet.”

## Optional binding (if let) ⚅

**Definition / example**

A safe pattern that attempts to extract a real value; if successful you get a non-optional constant inside the first branch, otherwise the alternate branch runs.