Scoping

Swift's Scoping ("Access Control") Model

Module:

- A single unit of code distribution
- A framework or application that is built and shipped as a single unit and that can be imported by another module using Swift's import keyword

Source file:

- A single Swift source code file within a module (app or framework)
- A single source file *can* contain definitions for multiple types, functions, etc., although it's typical to define individual types in separate source files

Access Levels – from least to most restrictive

`public`

- entities can be used within any source file from their defining module, as well as in a source file from another module that imports the defining module.
- Typically used to specify the public interface to a framework.

`internal`

- Entities can be used within any source file from their defining module, but NOT in any source file outside that module. This is the default.
- Typically used to define an app or a framework's internal structure.

Access Levels

`fileprivate`

- Entities can only be used within in their defining source file.
- Typically used to hide implementation details at the file level.

`Private`

- Entities can only be used in their enclosing declaration.
- Typically used to hide implementation details at the declaration level.

Examples of Syntax

```
public class SomePublicClass {}
internal class SomeInternalClass {}
private class SomePrivateClass {}

public var somePublicVariable = 0
internal let someInternalConstant = 0
fileprivate func someFilePrivateFunction() {}
```