

A. SwiftLint

workshop guide (1/3)

<https://github.com/realm/SwiftLint>

Part 1 - Setup

1. Clone workshop repo
 - `git clone https://github.com/jasonzurita/swift-linting-formatting-workshop``
 - Get added to repo (simplifies testing CI)
2. Install SwiftLint — `brew install swiftlint``
 - Requires Homebrew to be installed: <https://brew.sh>
3. Text editor is up to you (Xcode, Vim, TextEdit, etc.)

Part 2 - Exploration (SwiftLint)

Note: “Folders” below are located within the “linting” parent folder

Step	Description	Command(s) (not including file edits)	Folder
1	What can we do?	<ul style="list-style-type: none">• <code>swiftlint help</code>• <code>swiftlint help <cmd></code>	n/a
2	Available lint rules	<code>swiftlint rules</code>	n/a
3	Make “hello.swift”, edit, and lint	<ul style="list-style-type: none">• <code>touch hello.swift</code>• <code>swiftlint lint .</code>	1
4	Make some “triggering” changes and lint	<code>swiftlint lint .</code>	2
5	Let’s copy in some examples from the rules and lint	<ul style="list-style-type: none">• <code>swiftlint docs</code>• <code>swiftlint lint .</code>	3
6	Try and lint a Swift file or folder on your machine!	<code>swiftlint lint .</code> OR <code>swiftlint lint <file.swift></code>	n/a
7	Disabling rules in source code files	<code>// swiftlint:disable <rule></code> <code>// swiftlint:enable <rule></code>	4
8	Configuration	<code>touch .swiftlint.yml</code>	5

Part 3 - Integration (SwiftLint)

Step	Description	Notes	Folder
1	Create new Xcode project named "quotes"	n/a	6
2	Add functionality for random quote request then lint from command-line	<ul style="list-style-type: none">• Quote API<ul style="list-style-type: none">• Note: see below json• Decodable model gen• Http request	7
3	Integrate SwiftLint with Xcode	Setup	8
4	Make some changes, build, watch lint warning/errors show up, and disable rules!	n/a	n/a
5	Demo of other integrations (e.g., Vim)	n/a	n/a
6	Continuous Integration / Continuous Delivery (CI/CD) integration — failing build to improve code quality	Manually in GitHub Actions (can also use Fastlane)	9

—

QUOTE MODEL JSON

Below makes use of: <https://github.com/lukePeavey/quotable>
Specifically: <https://api.quotable.io/random?tags=technology,famous-quotes>

```
{
  "length" : 76,
  "_id" : "VDDodhULiss2",
  "content" : "It is through science that we prove, but through intuition that we discover.",
  "tags" : [
    "famous-quotes",
    "science",
    "technology"
  ],
  "authorSlug" : "henri-poincare",
  "author" : "Henri Poincaré"
}
```

B. SwiftFormat

workshop guide (2/3)

<https://github.com/nicklockwood/SwiftFormat>

Part 1 - Setup

1. Install SwiftFormat — `brew install swiftformat`

Part 2 - Exploration (SwiftFormat)

Note: “Folders” below are located within the “formatting” parent folder

Step	Description	Command(s) (not including file edits)	Folder
1	What can we do?	swiftformat --help	n/a
2	Available format rules	<ul style="list-style-type: none">• swiftformat --rules• swiftformat --ruleinfo• swiftformat --options	n/a
3	Make “hello.swift”, edit, and format	<ul style="list-style-type: none">• touch hello.swift• swiftformat .	1
4	Make some changes to “hello.swift” and format	swiftformat .	2
5	Try and format a Swift file or folder on your machine	Warning: this will replace the source file, so version control is your friend here	n/a
6	Biggest format diff challenge!	git diff --stat	n/a
7	Disabling rules in source code files	// swiftformat:disable <rule> // swiftformat:enable <rule>	3
8	Configuration	touch .swiftformat	4

Part 3 - Integration (SwiftFormat)

Step	Description	Notes	Folder
1	Integrate SwiftFormat with Xcode	<ul style="list-style-type: none">• Using the workshop repo version• Setup	5
2	Make some changes, test, watch format get updated, and look at diff!	<ul style="list-style-type: none">• git status• git diff	n/a
3	Continuous Integration / Continuous Delivery (CI/CD) integration — failing build to improve code quality	Manually in GitHub Actions	6

C. CI/CD Setup

workshop guide (3/3)

A possible way to integrate and automate use of SwiftLint & SwiftFormat

Part 1 - Setup

1. Using the same repo starting point
2. Create a new branch

Part 2 - Integration

Step	Description	Notes	Folder
1	A quick look at Danger	Danger site	n/a
2	Add Danger on a separate branch	<ul style="list-style-type: none">• Dangerfile• Gemfile• GH Actions workflow	Root folder "C - CI"
3	Test it out!		