



Linting & Formatting

Goals

- Better understand "the why" and the differences between linting and formatting
- How to lint and format & what tools to use
- Different workflows
 - command-line
 - Xcode integration
 - CI/CD

What is linting and formatting?

Why lint/format your code?

Linting vs Formatting

Linting

- Deeper analysis of your code (e.g., cyclomatic complexity, file length)
- Can be automatically run (Xcode & in CI/CD systems)
- Fixes typically require human intervention
- Most popular tool: SwiftLint

Formatting

- Mainly focused on code style and format (e.g., spaces, wrapping brackets, etc.)
- Can be automatically run (Xcode & in CI/CD systems)
- Applying fixes can be done automatically (recommended)
- Most popular tool: SwiftFormat

Let's play with some tools (overview)

```
let tools = [  
    [SwiftLint](https://github.com/realm/SwiftLint),  
    [SwiftFormat](https://github.com/apple/swift-format),  
]  
  
tools.forEach {  
    /*  
    - install $0  
    - Try $0 out via the (e.g., command-line)  
    - Config $0, aka customizing the behavior (yaml files)  
    - How to integrate $0 into your Xcode project & local workflow  
    - How to automate & add $0 to a CI/CD system  
    - if $0 == "SwiftLint" {  
        Stretch goal, look at setting up [Danger](https://danger.systems)  
        in a continuous integration (CI)  
    }  
    */  
}
```