

# SwiftLint のオレオレカスタムルール一覧

## はじめに

こんにちは、ウホーイ (@the\_uhooi) です。  
SwiftLint で私が自作しているカスタムルールを紹介します。

## 環境

本稿の環境（2024/06/20 現在）です。多少環境が異なっても同様の動作になると思いますが、もし動作が異なっていたら X（旧 Twitter）などでご連絡くださると嬉しいです。

- OS: macOS Sonoma 14.5 (23F79)
- Xcode: 15.4 (15F31d)
- Swift: 5.10
- SwiftLint: 0.55.1

## オレオレカスタムルール一覧

私が自作しているカスタムルールの一覧です。

### enum\_lower\_case

enum の case 名の先頭が大文字より小文字を好むルールです。

```
enum_lower_case:
  name: 'Enum Lower Case'
  regex: '\bcase\s+([A-Z][a-zA-Z0-9]*)\b'
  capture_group: 1
  message: 'The enum case should be lower camel case'
  severity: warning

// Non Triggering Examples
enum Foo {
  case bar
}

// Triggering Examples
enum Foo {
  case ↓ Bar
}
```

### final\_class

class に final を必ず付けるルールです。

```
final_class:
  name: 'Final Class'
  regex: '\b(?<!final)\s+(class)\b'
```

```
capture_group: 1
message: 'Always add `final` to classes that do not inherit '
severity: warning

// Non Triggering Examples
final class Foo {}

// Triggering Examples
↓ class Foo {}
```

## date\_now

Date() より.now を好むルールです。  
.now は Date() と同等ですが、.now のほうがわかりやすいので好みます。

```
# iOS 15.0+
date_now:
  name: 'Date Now'
  regex: '[=:(]\s*(Date\(\))'
  capture_group: 1
  message: 'Prefer using `.now` over `Date()`'
  severity: warning

// Non Triggering Examples
let weekday = Calendar.current.component(.weekday, from: .now)

// Triggering Examples
let weekday = Calendar.current.component(.weekday, from: ↓ Date())
```

## enumerated

enumerated() より swift-algorithms の indexed() を好むルールです。

```
enumerated:
  name: 'Enumerated'
  regex: '.*\.(enumerated\[([~])*\))'
  capture_group: 1
  message: 'Prefer using `indexed()` over `enumerated()`'
  severity: warning

// Non Triggering Examples
import Algorithms
ForEach(monsters.indexed(), id: \.element.id) { index, monster in
  // Do something
}

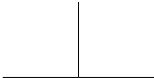
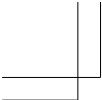
// Triggering Examples
ForEach(Array(monsters.↓enumerated()), id: \.element.id) { index, monster in
  // Do something
}
```

## aspect\_ratio\_fill

.aspectRatio(contentMode: .fill) より.scaledToFill() を好むルールです。  
.scaledToFill() は.aspectRatio(contentMode: .fill) と同等ですが、短いほうが好みます。

```
# iOS 13.0+
aspect_ratio_fill:
  name: 'Aspect Ratio Fill'
  regex: '.*\.(aspectRatio\(contentMode:\s*\.\fill\s*\))'
  capture_group: 1
  message: 'Prefer using `scaledToFill()` over `aspectRatio(contentMode: .fill)`'
  severity: warning

// Non Triggering Examples
Image(systemName: "swift")
.resizable()
.scaledToFill()
```



```
// Triggering Examples
Image(systemName: "swift")
  .resizable()
  .↓ aspectRatio(contentMode: .fill)
```

## aspect\_ratio\_fit

.aspectRatio(contentMode: .fit) より.scaledToFit() を好むルールです。  
aspect\_ratio\_fill と同じ理由でルール化しています。

```
# iOS 13.0+
aspect_ratio_fit:
  name: 'Aspect Ratio Fit'
  regex: '.*\.(aspectRatio\(contentMode:\s*\.\fit\s*\))\s*'
  capture_group: 1
  message: 'Prefer using `scaledToFit()` over `aspectRatio(contentMode: .fit)`'
  severity: warning
```

```
// Non Triggering Examples
Image(systemName: "swift")
  .resizable()
  .scaledToFit()

// Triggering Examples
Image(systemName: "swift")
  .resizable()
  .↓ aspectRatio(contentMode: .fit)
```

## corner\_radius

cornerRadius() は非推奨になったため、clipShape() または fill() を好むルールです。  
clipShape(RoundedRectangle(cornerRadius:)) に置き換えることが多いです。

```
# iOS 13.0+
corner_radius:
  name: 'Corner Radius'
  regex: '.*\.(cornerRadius\([^\)]*\))\s*'
  capture_group: 1
  message: 'Prefer using `clipShape(RoundedRectangle(cornerRadius:))` or `fill()` over `cornerRadius()` because it was deprecated'
  severity: warning
```

```
// Non Triggering Examples
Text("Foo")
  .clipShape(RoundedRectangle(cornerRadius: 8))

// Triggering Examples
Text("Foo")
  .↓ cornerRadius(8)
```

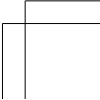
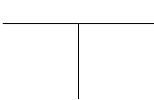
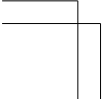
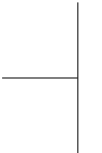
## foreground\_color

foregroundColor(\_) は非推奨になったため、foregroundStyle(\_) を好むルールです。

```
# iOS 15.0+
foreground_color:
  name: 'Foreground Color'
  regex: '.*\.(foregroundColor\([^\)]*\))\s*'
  capture_group: 1
  message: 'Prefer using `foregroundStyle()` over `foregroundColor()` because it was deprecated'
  severity: warning
```

```
// Non Triggering Examples
Text("Foo")
  .foregroundStyle(.blue)

// Triggering Examples
Text("Foo")
  .↓ foregroundColor(.blue)
```



## navigation\_bar\_leading

.navigationBarLeading は非推奨になったため、.topBarLeading を好むルールです。

```
# iOS 14.0+
navigation_bar_leading:
  name: 'Navigation Bar Leading'
  regex: '\bToolBarItem\(placement:\s*(.navigationBarLeading)\s*\)'
  capture_group: 1
  message: 'Prefer using `.topBarLeading` over `.navigationBarLeading` because it was deprecated'
  severity: warning

// Non Triggering Examples
Text("Foo")
.toolbar {
  ToolBarItem(placement: .topBarLeading) {
    // Do something
  }
}

// Triggering Examples
Text("Foo")
.toolbar {
  ToolBarItem(placement: ↓.navigationBarLeading) {
    // Do something
  }
}
```

## navigation\_bar\_trailing

.navigationBarTrailing は非推奨になったため、.topBarTrailing を好むルールです。

```
# iOS 14.0+
navigation_bar_trailing:
  name: 'Navigation Bar Trailing'
  regex: '\bToolBarItem\(placement:\s*(.navigationBarTrailing)\s*\)'
  capture_group: 1
  message: 'Prefer using `.topBarTrailing` over `.navigationBarTrailing` because it was deprecated'
  severity: warning

// Non Triggering Examples
Text("Foo")
.toolbar {
  ToolBarItem(placement: .topBarTrailing) {
    // Do something
  }
}

// Triggering Examples
Text("Foo")
.toolbar {
  ToolBarItem(placement: ↓.navigationBarTrailing) {
    // Do something
  }
}
```

## おわりに

SwiftLint のカスタムルールは正規表現で書くので、慣れていないと難しいです。私も試行錯誤しながら書いたので、誤りなどあればご連絡くださると嬉しいです。

## 参考資料

- <https://github.com/uhooi/iOSDC-Japan-2024-Manuscript>
- SwiftLint のオレオレカスタムルール一覧 - Qiita