iOS projects code formattings

Eduard Panasiuk

Tools

- ClangFormat
- SwiftFormat

ClangFormat

ClangFormat

https://clang.llvm.org/docs/ClangFormat.html

Instalation:

```
brew install clang-format
```

Usage:

```
clang-format -style=file ./MyClass.m
```

ClangFormat options

https://clang.llvm.org/docs/ClangFormatStyleOptions.html

```
---
# We'll use defaults from the LLVM style, but with 4 columns indentation.
BasedOnStyle: LLVM
IndentWidth: 4
---
Language: Cpp
# Force pointers to the type for C++.
DerivePointerAlignment: false
PointerAlignment: Left
---
Language: JavaScript
# Use 100 columns for JS.
ColumnLimit: 100
---
Language: Proto
# Don't format .proto files.
DisableFormat: true
...
```

ClangFormat predefined styles

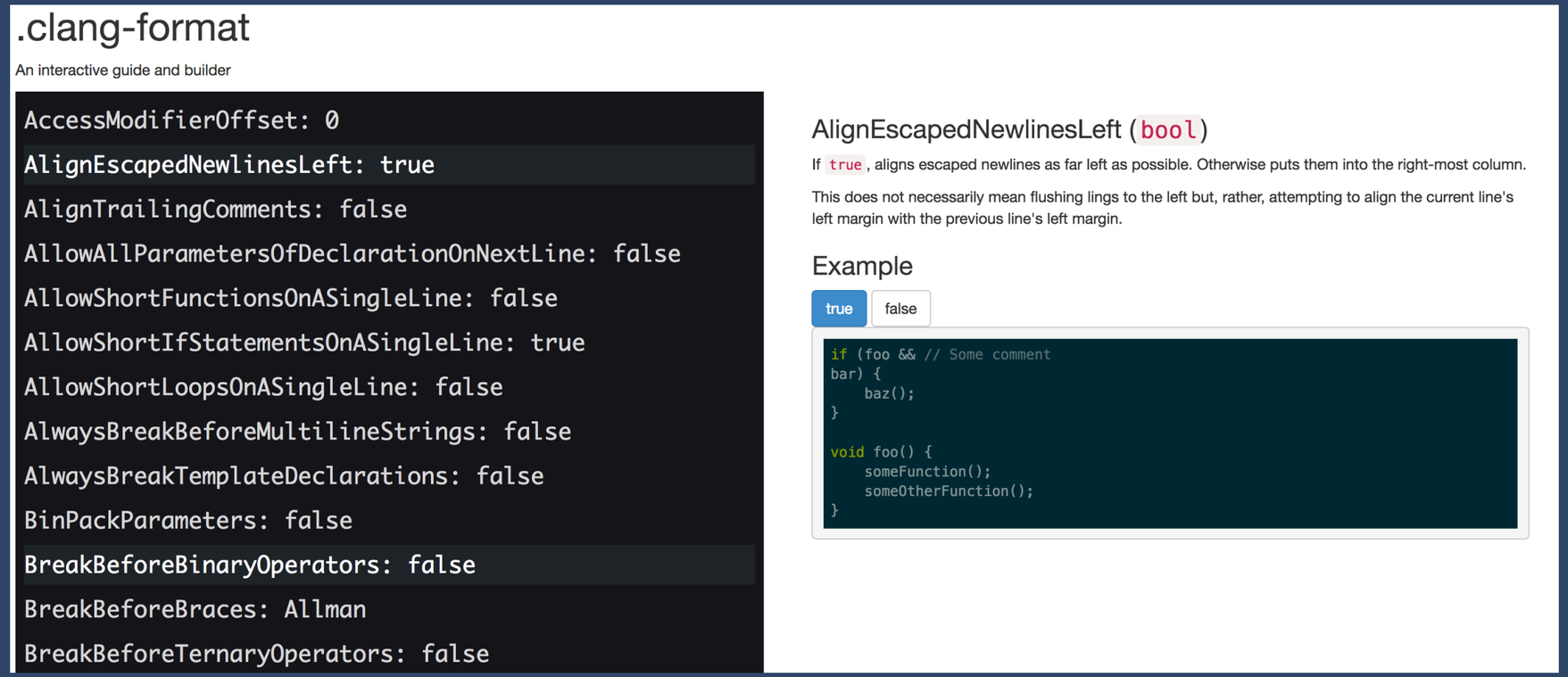
LLVM, **Google**, **Chromium**, **Mozilla**, **WebKit**

Dump predefined style:

```
clang-format -style=llvm -dump-config > .clang-format
```

ClangFormat options

https://clangformat.com/



ClangFormat usage

Git hook

Create `.git/hooks/pre-commit`

```
#!/bin/bash

CLANG_FORMAT=$(which clang-format)

git diff-index --cached --diff-filter=ACMR --name-only $against -- | while read file;
do
    "$CLANG_FORMAT" -i -style=file "$file"
    git add "$file"
done
```

SwiftFormat

SwiftFormat

https://github.com/nicklockwood/SwiftFormat

Instalation:

```
brew install swiftformat
```

SwiftFormat usage

Terminal

Run in project directory:

```
swiftformat .
```

SwiftFormat options

https://github.com/nicklockwood/SwiftFormat

```
--allman            use allman indentation style. "true" or "false" (default)
--binarygrouping    binary grouping,threshold or "none", "ignore". default: 4,8
--commas            commas in collection literals. "always" (default) or "inline"
--comments          indenting of comment bodies. "indent" (default) or "ignore"
--decimalgrouping   decimal grouping,threshold or "none", "ignore". default: 3,6
--elseposition      placement of else/catch. "same-line" (default) or "next-line"
--empty             how empty values are represented. "void" (default) or "tuple"
--experimental      experimental rules. "enabled" or "disabled" (default)
--exponentcase      case of 'e' in numbers. "lowercase" or "uppercase" (default)
--header            header comments. "strip", "ignore", or the text you wish use
--hexgrouping       hex grouping,threshold or "none", "ignore". default: 4,8
--hexliteralcase    casing for hex literals. "uppercase" (default) or "lowercase"
--ifdef             #if indenting. "indent" (default), "noindent" or "outdent"
--indent            number of spaces to indent, or "tab" to use tabs
--indentcase        indent cases inside a switch. "true" or "false" (default)
--linebreaks        linebreak character to use. "cr", "crlf" or "lf" (default)
--octalgrouping     octal grouping,threshold or "none", "ignore". default: 4,8
--operatorfunc      spacing for operator funcs. "spaced" (default) or "nospace"
--patternlet        let/var placement in patterns. "hoist" (default) or "inline"
--ranges            spacing for ranges. "spaced" (default) or "nospace"
--semicolons        allow semicolons. "never" or "inline" (default)
--self              use self for member variables. "remove" (default) or "insert"
--stripunusedargs   "closure-only", "unnamed-only" or "always" (default)
--trimwhitespace    trim trailing space. "always" (default) or "nonblank-lines"
--wrapparguments    wrap function args. "beforefirst", "afterfirst", "disabled"
--wrapelements      wrap array/dict. "beforefirst", "afterfirst", "disabled"
```

SwiftFormat options

Enable/Disable rules for concrete file

```
// swiftformat:disable <rule1> [<rule2> [rule<3> ...]]
// swiftformat:enable <rule1> [<rule2> [rule<3> ...]]
```

SwiftFormat usage

Xcode build phase

Install pod:

```
pod 'SwiftFormat/CLI'
```

Add build phase:

```
"${PODS_ROOT}/SwiftFormat/CommandLineTool/swiftformat"
```

SwiftFormat usage

Git hook

Create `.git/hooks/pre-commit`

```
#!/bin/bash
git diff --staged --name-only | grep -e '\(.*\).swift$' | while read file; do
   swiftformat ${file};
   git add $file;
done
```

iOS projects code formattings

Eduard Panasiuk