Swift Style Guide

This guide helps to improve the developer’s ability to compile code without warnings. Swift style recommends using camelCase instead of snake case with clarity. Using non-complex terms is preferred so that even beginners can understand it. Swift discourages using abbreviation. While using prose being unambiguous is preferred. While creating delegates the first unnamed parameter is delegate source. Generic parameter type should be descriptive. To match apple’s API we should use US English. While organizing the code comments are preferred so that everyone using the source code know what specification that code brings also if you’re using protocols, creating extensions are recommended. If there is some code unused either it should be deleted or should be mentioned properly and commented out. Minimum data is recommended to import so that the program looks clean and builds fast. Proper spacing and comments are very important while writing code in swift. Be careful about using classes and structures. Use of self is only referred if necessary. Marking the classes or members using “final” is not recommended since it might be confusing more than helpful. Be extra careful while using function declarations. Use trailing closure syntax if there’s a single closure expression parameter at the end of the argument list. Use of swift’s native types and expressions are always preferred. We use ‘let’ to define a constant whereas ‘var’ is used for variables. It the value of precise then ‘static let’ is preferred more. If a nil value is acceptable then ‘?’ is recommended to use for optional parameter whereas ‘!’ is used to force unwrap. Lazy initialization is recommended while writing control over objects. Type annotations are recommended for empty array and dictionaries. Prefer shortcut version of type annotation. Using of swift specific control flow makes it less time consuming. Use of access control type is recommended only if it is considered really useful. Ternary operator ‘?:’ is used only when it proves to increase the neatness of our code. If you want to use guard statement, try not to use it in a bulk of code.

Parenthesis are not required around conditionals. Emojis cannot be used while writing codes. Copyright statement should be mentioned at the top of every file.