Swift Coding Standard

Reference Link
https://google.github.io/swift/

General Formatting

Imports come first

Use “MARK” comment to separate sections of multiple code sections/functions
Example:

```
//MARK: -Load UI for main class
func loadUIViews(){
}
Func setColors(){
}
```

Braces follow Kernighan and Ritchie (K&R) style
The is no return after the function name
Example:

```
func requestAPI(){
}
```

Semi-colons are not used at all
Swift does not need any

One statement per line

Whitespace

- After reserved words
  - if (condition)
- Before and after { } when on the same line
  - var data = { 1, 2, 3 }
- Before and after Operators like =, +, -, ==, *, /
  - var distance = getDistance(d: x * y)
- After commas
  - 1, 2, 3, 4
- After the colon in type declarations
  - var num: Int = 0
- Two spaces before comments and one space after
  - var x = 0. // sample comment

Blank Lines

Permitted, but don't overuse

Parentheses

Are not necessary to wrap around the whole if statement

Example:

```
if (x == y) && x > 2 {
}
```

Formatting Specific Constructs

Add commas to every element when separating

```
let configurationKeys = [
    "bufferSize",
    "compression",
    "encoding",
]
```

Naming

Always use self when init creates a new instance

Example:

```
public init(name: String) {
    self.name = name
}
```

Never suffix the name of a variable or function with its type

Example:

```
var red: UIColor = UIColor.red
```

Not Allowed:

```
var redColor: UIColor = UIColor.red
```

Returning a computed value uses return, not get

Example:

```
var totalCost: Int {
    return items.cost + (items.cost * taxPercentage)
}
```

Force unwraps should be avoided at all costs, but are still allowed. Comments recommended after.
Example:

```
var x = getNum()!  // This unwrap is allowed because of some reason
```

Always use guard statements when applicable. (This is opposed to if checks)
Example:

```
guard let first = values.first else {
    return nil
}
```

Use “where” in for loops when applicable
Example:

```
for item in collection where item.hasProperty {
    //loop
}
```

Tuple assignment with variables
Example:

```
let (a, b) = (x: 4, y: 5)
```

Documenting functions and such uses Javadoc ///
Example:

```
/// General description: Returns the sum of two numbers
///
/// - Parameters:
///     - length: The z length of a given house
///     - width: The x dimension of a house
/// - Returns: The square footage representing the cornerstone of a house
func getCornerstoneSize(length: Int, width: Int) {
    return length * width
}
```
