(More) Swift Basics

Bas Broek

Force! Casting

```
var statusCode: Int?

print(statusCode!) // crash

if statusCode != nil {
  print(statusCode!) // Please don't do this.
}

statusCode = 200

if let statusCode = statusCode  {
  print(statusCode) // Int, not Optional<Int>
}
```

??

```
var statusCode: Int?

let result1 = statusCode // Optional<Int>

let result2 = statusCode ?? 200 // Int

// Which is the same as

let result3 = (statusCode == nil) ? 200 : statusCode! // or
let result4 = (statusCode != nil) ? statusCode! : 200
```

Enums

```
enum IntNumber: Int {
  case zero // = 0
  case one // = 1
}

enum StringNumber: String {
  case zero // = "zero"
  case one // = "one"
}

// Here, you get an initializer for free:

IntNumber(rawValue: 1) // .one
StringNumber(rawValue: "zero") // .zero
```

Con(struct)or

Constructor

Initializer

Initializer

```
struct Book {
  let title: String
  let author: String
}

Book(title: "De IJzeren Wil", author: "Bas Haring")
```

More Enums

```
func track(number: IntNumber) {
  print(number.rawValue)
}

// You can call this as

track(number: IntNumber.zero) // 0

// .. but also as

track(number: .zero) // 0
```

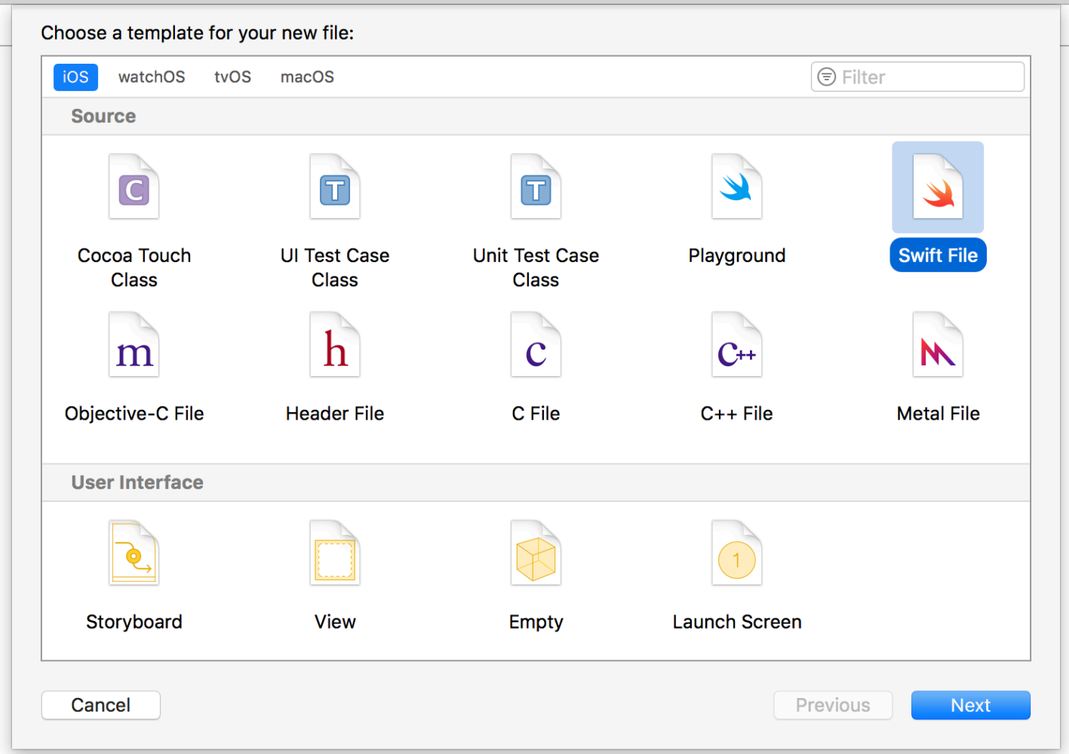
Extensions

```
extension String {
  func localized() -> String {
    return NSLocalizedString(self, comment: "")
  }
}

"hello".localized()

extension Int {
  func times(task: (Int) -> Void) {
    (1...self).forEach {
      task($0)
    }
  }
}

3.times { number in
  print(number) // 1, 2, 3
}
```

Typealias

https://github.com/BasThomas/Analysis

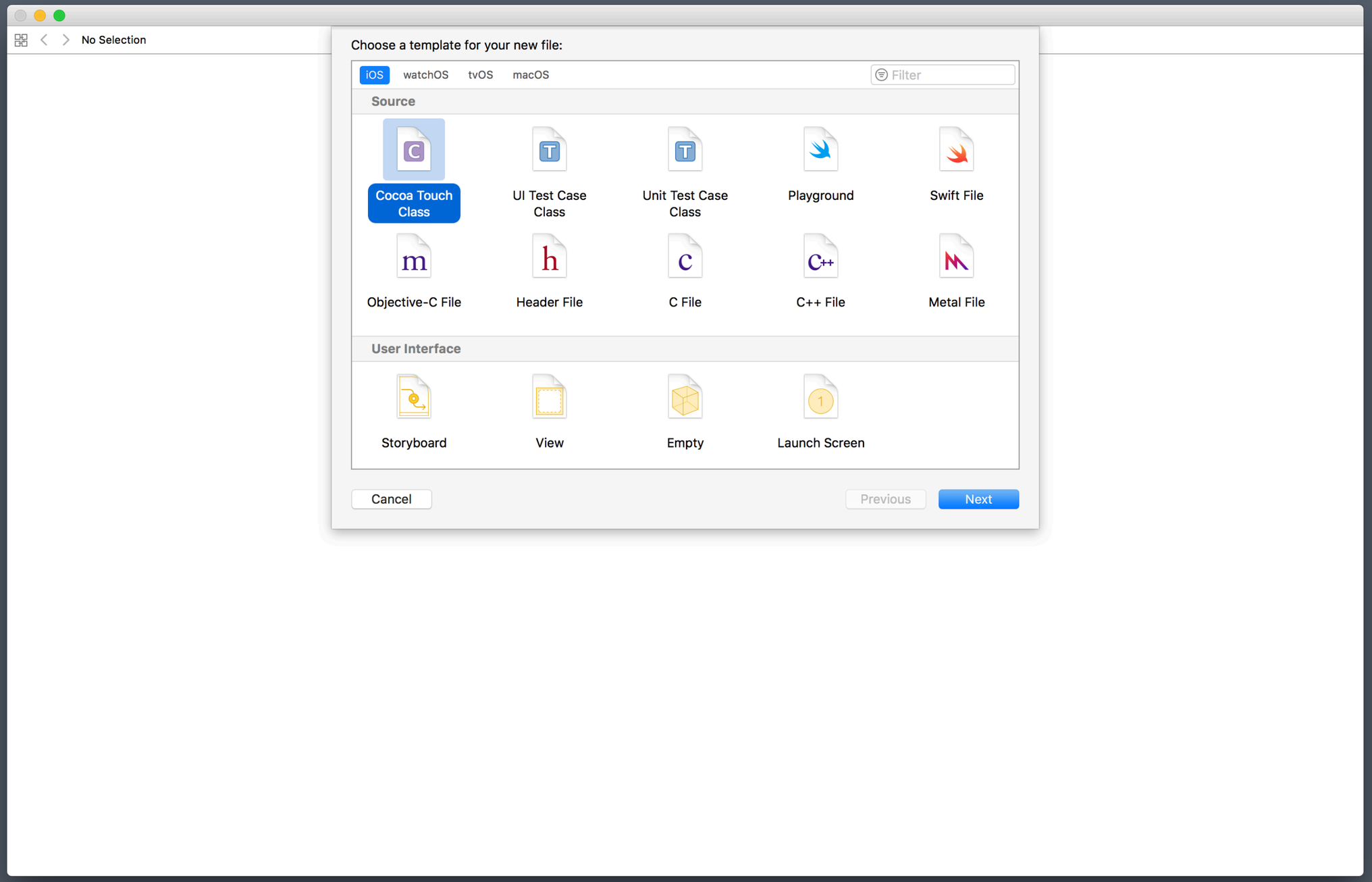
```swift
public struct Analysis {
  public typealias Percentage = Double

  /// Returns the frequency of the specified word.
  ///
  /// - Parameter caseSensitive: Indicating if words
  /// should be counted regardless of their case sensitivity.
  /// Defaults to `false`.
  ///
  /// - Returns: A percentage based on the `wordCount()`.
  public func frequency(of word: String, caseSensitive: Bool = false) -> Percentage {
    return Double(occurrences(of: word, caseSensitive: caseSensitive)) / Double(wordCount()) * 100.0
  }
}

let analysis = "What is this?".analysed()
analysis.frequency(of: "what") // 33.33
```

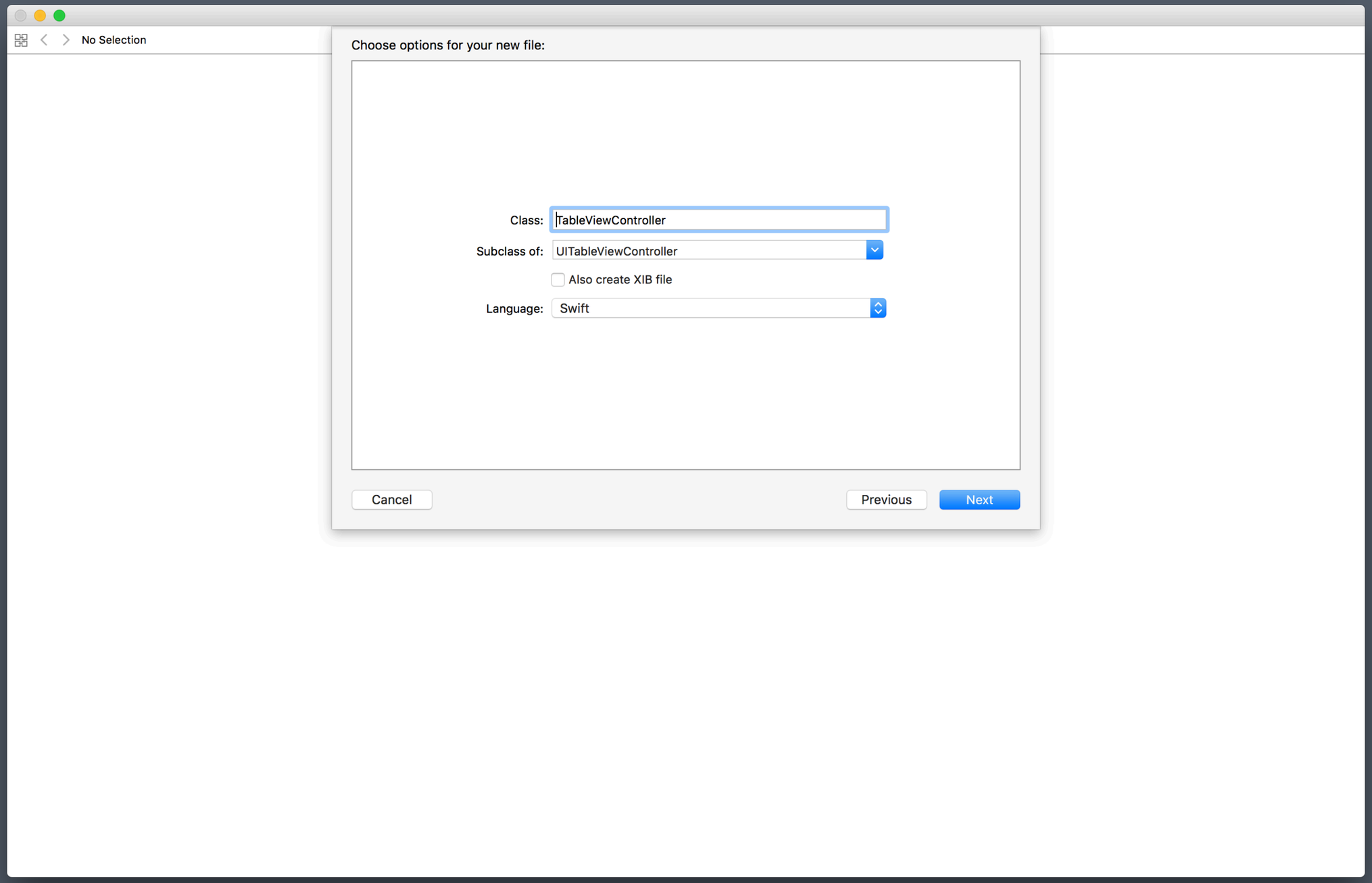


File creation

File.swift 〉 No Selection

```
1  //
2  //  File.swift
3  //
4  //
5  //  Created by Bas Broek on 07/12/2016.
6  //
7  //
8
9  import Foundation
10
```


No Selection
Choose a template for your new file:
iOS
watchOS
tvOS
macOS
Filter
Source
Cocoa Touch Class
UI Test Case Class
Unit Test Case Class
Playground
Swift File
Objective-C File
Header File
C File
C++ File
Metal File
User Interface
Storyboard
View
Empty
Launch Screen
Cancel
Previous
Next


No Selection
Choose options for your new file:
Class:
TableViewController
Subclass of:
UITableViewController
Also create XIB file
Language:
Swift
Cancel
Previous
Next

```swift
//
//  TableViewController.swift
//
//
//  Created by Bas Broek on 07/12/2016.
//
//

import UIKit

class TableViewController: UITableViewController {

    override func viewDidLoad() {
        super.viewDidLoad()

        // Uncomment the following line to preserve selection between presentations
        // self.clearsSelectionOnViewWillAppear = false

        // Uncomment the following line to display an Edit button in the navigation bar for this view controller.
        // self.navigationItem.rightBarButtonItem = self.editButtonItem()
    }

    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
        // Dispose of any resources that can be recreated.
    }

    // MARK: - Table view data source

    override func numberOfSections(in tableView: UITableView) -> Int {
        // #warning Incomplete implementation, return the number of sections
        return 0
    }

    override func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {
        // #warning Incomplete implementation, return the number of rows
        return 0
    }

    /*
    override func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {
        let cell = tableView.dequeueReusableCell(withIdentifier: "reuseIdentifier", for: indexPath)

        // Configure the cell...

        return cell
    }
    */

    /*
    // Override to support conditional editing of the table view.
    override func tableView(_ tableView: UITableView, canEditRowAt indexPath: IndexPath) -> Bool {
        // Return false if you do not want the specified item to be editable.
        return true
    }
    */

    /*
    // Override to support editing the table view.
    override func tableView(_ tableView: UITableView, commit editingStyle: UITableViewCellEditingStyle, forRowAt indexPath: IndexPath) {
        if editingStyle == .delete {
            // Delete the row from the data source
            tableView.deleteRows(at: [indexPath], with: .fade)
        } else if editingStyle == .insert {
            // Create a new instance of the appropriate class, insert it into the array, and add a new row to the table view
        }
    }
    */

    /*
    // Override to support rearranging the table view.
    override func tableView(_ tableView: UITableView, moveRowAt fromIndexPath: IndexPath, to: IndexPath) {

    }
    */
```