```swift
import UIKit
import XCPlayground

//: # NSOperationQueue
//: NSOperationQueue is responsible for scheduling and running a set of
    operations, somewhere in the background.

//: To prevent the playground from killing background tasks when the
    main thread has completed, need to specify indefinite execution
XCPlaygroundPage.currentPage.needsIndefiniteExecution = true


//: ## Creating a queue
//: Operations can be added to queues directly as closures
let printerQueue = NSOperationQueue()

printerQueue.maxConcurrentOperationCount = 2

startClock()
printerQueue.addOperationWithBlock({ sleep(5); print("Hello") })
printerQueue.addOperationWithBlock({ sleep(5); print("my") })
printerQueue.addOperationWithBlock({ sleep(5); print("name") })
printerQueue.addOperationWithBlock({ sleep(5); print("is") })
printerQueue.addOperationWithBlock({ sleep(5); print("Vincenzo") })
stopClock()

startClock()
printerQueue.waitUntilAllOperationsAreFinished()
stopClock()





//: ## Adding NSOperations to queues
let images = ["city", "dark_road", "train_day", "train_dusk",
    "train_night"].map { UIImage(named: "\($0).jpg") }
var filteredImages = [UIImage]()

//: Create the queue with the default constructor
let filterQueue = NSOperationQueue()

//: Create a filter operations for each of the iamges, adding a
    completionBlock
for image in images {

    let filterOp = TiltShiftOperation()
    filterOp.inputImage = image
    filterOp.completionBlock = {
        guard let output = filterOp.outputImage else { return }
        filteredImages.append(output)
    }

    filterQueue.addOperation(filterOp )
}

//: Need to wait for the queue to finish before checking the results
filterQueue.waitUntilAllOperationsAreFinished()

//: Inspect the filtered images
filteredImages
```