# Introduction to Combine

Combine is “a [declarative](https://tylermcginnis.com/imperative-vs-declarative-programming/) Swift API for processing values over time”. It is a functional reactive programming library.

Streams of elements. Transform data flowing through streams.

## Functional Reactive Programming

Functional programming: declarative programming paradigm done with declarations instead of statements.

Functional reactive programming: functional programming applied to streams of elements.

Reactive is an API for asynchronous programming with observable streams.

“The Observer pattern done right” - <http://reactivex.io/>

## Publisher and Subscriber Overview

These are defined by two Swift protocols: [Publisher](https://developer.apple.com/documentation/combine/publisher) and [Subscriber](https://developer.apple.com/documentation/combine/subscriber). A publisher is an observable event stream. Subscribers observe events from the publisher. The Combine framework allows us to process numerous values over time *and* handle failure.

Publisher provides data when available and upon request. A publisher without at least one subscription does not provide any data. Publishers declare two associated types: Output and Failure. Publishers emit Output values until completion (finished or failure). Completion is not guaranteed; however, upon completion, no further data is sent through the stream.

A subscriber receives a stream of elements from a publisher. It also receives lifecycle events from the publisher. Subscribers have associated types of Input and Failure which must match the Output and Failure of its corresponding publisher. Without a subscriber, a publisher remains idle.

The connection between a publisher and subscriber is a subscription. Rules of subscription are:

* A subscriber can only have one subscription
* Zero or more values can be published
* At most one completion will be called (may never complete)

<Diagram>

Operators are an additional concept. Operators adopt both the Subscriber and Publisher protocol and allow for additional processing between a publisher and subscriber.

## When to use Combine

Reacting to various inputs, for example, observing a textfield for changes.

Performing asynchronous operations, especially as part of a sequence of operations. Example, a series of network requests and decoding and/or merging the results.

### Difficulties using Combine

Paradigm shift for Developers. May slow development initially. Onboarding new Developers can be challenging.

Difficult to debug. <expand> <talk about debugging tips>

Difficult to find help and examples of more specific use cases. <examples?>

# Combine Concepts

## Publisher

Publishers as a concept.

Convenience publishers.

Publishers in Foundation and SwiftUI.

## Subscriber

Cancellable

## Subjects

Publisher & Subscriber <mindblown>. Ability to send values through the stream, but also subscribe to the stream.

## Operators

Talk about what operators do. Give a few simple examples.