

Question-1 ----- What is combine ? -----

The Combine framework provides a declarative Swift API for processing values over time. These values can represent many kinds of asynchronous events. Combine declares publishers to expose values that can change over time, and subscribers to receive those values from the publishers.

Question-2 -----What are publisher and subscribers? Explain creating examples.-----

PUBLISHERS-->

Publisher is a protocol which has two associated type first one is Output which is the kind of value reproduced by publisher and second one is Failure which is kind of error produces by publisher.

Publisher has one key function which is called subscribe. This function needs subscriber input as parameter to match with publishers output.

Combine provides a number of additional convenience publishers:

- i) Just
- ii) Future
- iii) Deferred
- iv) Empty
- v) Sequence
- vi) Fail
- vii) Record
- ix) Share
- x) Multicast
- xi) ObservableObject
- xii) @Published

SUBSCRIBERS-->

1)A subscriber is responsible for requesting data and accepting the data (and possible failures) provided by a publisher.

2)This receives values and a completion, this is of reference types which means classes.

3)A subscriber is described with two associated types, one for Input and one for Failure.

The subscriber initiates the request for data, and controls the amount of data it receives.

It can be thought of as “driving the action” within Combine, as without a subscriber, the other components stay idle.

4)Publishers and operators are pointless unless something is listening to the published events. That something is the Subscriber.

5)A subscriber receives a stream of value, completion or failure events from a publisher.

6)There are two built-in subscribers in Combine i.e. sink and assign. There is a subscriber built in to SwiftUI: onReceive.

7)Subscribers can support cancellation, which terminates a subscription and shuts down all the stream processing prior to any Completion sent by the publisher. Both Assign and Sink conform to the cancellable protocol.

NOTE--- EXAMPLES OF " PUBLISHERS AND SUBSCRIBERS" ARE ATTACHED IN THIS PROJECT

QUESTION3 -----What are subjects and operators? Explain creating examples.

SUBJECTS-->

1)Subject is also a type of publisher on which we can subscribe but also dynamically send events to them.

Subject exposes a method for outside callers to publish elements.

2)A subject can be used to "inject" values into a stream, by calling its send( :) method . This can be useful for adapting existing imperative code to the Combine model.

3)A subject can also broadcast values to multiple subscribers.

If multiple subscribers are connected to a subject, it will fan out values to the multiple subscribers when send() is invoked

4)There are two types of built-in subjects with Combine: PassthroughSubject, CurrentValueSubject

OPERATORS-->

1)Operators are a convenient name for a number of pre-built functions that are included under Publisher.

2)This describes when and where particular event is delivered.

3)This is supported by runloop and dispatch queue.

NOTE--- EXAMPLES OF "subjects and operators" ARE ATTACHED IN THIS PROJECT

Question -4 -----Implement URLSessionTask using combine.

implementation done in this app