

The following operators are part of the distinct `rxjava-async` module. They are used to convert synchronous methods into Observables.

- `start(_)` — create an Observable that emits the return value of a function
- `toAsync(_)` `or` `asyncAction(_)` `or` `asyncFunc(_)` — convert a function or Action into an Observable that executes the function and emits its return value
- `startFuture(_)` — convert a function that returns Future into an Observable that emits that Future's return value
- `deferFuture(_)` — convert a Future that returns an Observable into an Observable, but do not attempt to get the Observable that the Future returns until a Subscriber subscribes
- `forEachFuture(_)` — pass Subscriber methods to an Observable but also have it behave like a Future that blocks until it completes
- `fromAction(_)` — convert an Action into an Observable that invokes the action and emits its result when a Subscriber subscribes
- `fromCallable(_)` — convert a Callable into an Observable that invokes the callable and emits its result or exception when a Subscriber subscribes
- `fromRunnable(_)` — convert a Runnable into an Observable that invokes the runnable and emits its result when a Subscriber subscribes
- `runAsync(_)` — returns a `StoppableObservable` that emits multiple actions as generated by a specified Action on a Scheduler