produce

```
public fun <E> produce(
        context: CoroutineContext = DefaultDispatcher,
        ...,
        block: suspend ProducerScope<E>.() -> Unit
): ReceiveChannel<E>
```

```
val squaresChannel: ReceiveChannel<Int> = produce {
    for (x in 0..5)
        send(x * x)
}
```

actor

```
public fun <E> actor(
        context: CoroutineContext = DefaultDispatcher,
        ... ,
        block: suspend ActorScope<E>.() -> Unit
): SendChannel<E>
```

```
val squaresChannel: SendChannel<TableUpdate> = actor(UI) {
    for (update in channel)
        redrawTable(update)
}
```