

# Yield

The `yield` keyword in Dart allows you to indicate to the Dart compiler that your generator function or your asynchronous stream function is producing a value. For instance if you have a synchronous function that produces an iterable of values from 0 to n, you can simply create a loop in the function and yield your values instead of having to construct a list inside the function. Uses also for sync generators!

Further reading:

- [Difference between yield and yield\\* in Dart - newbedev.com](https://newbedev.com/difference-between-yield-and-yield-in-dart/)
- [What is Yield Keyword In Flutter - flutteragency.com](https://flutteragency.com/what-is-yield-keyword-in-flutter/)
- [Asynchronous programming: Streams - dart.dev](https://dart.dev/asynchronous-programming-streams)
- [Dart Generators - Javatpoint](https://www.javatpoint.com/dart-generators)
- [Generators in Dart - GeeksforGeeks](https://www.geeksforgeeks.org/dart-generators/)



## Examples

```
1 // here is a synchronous generator function that produces
2 // three integers, 1, 2, 3, using the yield keyword
3 Iterable<int> threeNumbers() sync* {
4   yield 1;
5   yield 2;
6   yield 3;
7 }
8 // and here is the equivalent async function marked as
9 // async* that generates 3 integers, 1, 2, 3, inside the
10 // stream using yield
11 Stream<int> threeNumbersStream() async* {
12   yield 1;
13   yield 2;
14   yield 3;
15 }
```