

Generators

Generators in Dart are used to lazily create a sequence of values. A lazy sequence is an iterable whereas its counterpart is a list that is not created lazily. A generator is a function that allows you to return an iterable list of values, such as fibonacci numbers, without you having to calculate the entire sequence at once. Instead you can calculate the values and return those values with `yield` as the caller consumes them.

Further reading:

- [Generators - dart.dev](#)
- [Dart Generators - techLog](#)
- [Generators in Dart - GeeksforGeeks](#)
- [Dart Generators - W3Schools | W3Adda](#)
- [Dart Generators & Callable Class In Flutter - Medium](#)



Examples

```
1 // a generator is a special function in Dart that is
2 // marked with an iterable/stream as its return value
3 // plus a sync* or async* as its decorator, denoting
4 // synchronous or asynchronous generators in that order
5 Iterable<int> fromZero({required int upTo}) sync* {
6   // this function simply creates a loop that goes from
7   // 0 and up to, but not including, the value passed
8   // to it as "upTo"
9   for (var i = 0; i < upTo; i++) {
10    // and for every value in the loop it uses the
11    // "yield" keyword to return its value in the result
12    yield i;
13  }
14 }
```