

# Break and Continue

Both the `break` and the `continue` keywords allow you to control the flow inside a loop. The “`break`” keyword as its name implies allow you to break out of a loop should a certain condition be met. It’s useful when trying to find a specific value inside a loop. The “`continue`” keyword allows you skip the current iteration of the loop and go to the next iteration inside the same loop should a condition be met.

Further reading:

- [Break and continue - dart.dev](https://dart.dev)
- [Dart Programming - continue Statement - Tutorialspoint](#)
- [Dart - Loop Control Statements \(Break and Continue\)](#)
- [Dart Continue statement - W3Schools | W3Adda](#)
- [Dart continue Statement Tutorial With Examples - FlutterRDart](#)



## Examples

```
1  const names = ['Foo', 'Bar', 'Baz'];
2  String? foo;
3  for (final name in names) {
4    if (name.isEmpty) {
5      // although not necessary, we are using the continue
6      // statement here that jumps to the next iteration
7      // of this loop should the current name in the list
8      // be empty, meaning that it's of 0 length
9      continue;
10  }
11  if (name == 'Foo') {
12    foo = name;
13    // otherwise as soon as we find 'Foo', we break out
14    break;
15  }
16 }
```