Async

Function in Dart can either produce no value, or a value synchronously, or a value asynchronously. An asynchronous function is a function that goes off and does work and comes back later producing a value, or even void. The “async” keyword in Dart is a way to mark up a function as an asynchronous function, meaning that the function doesn’t completely immediately upon being invoked.

Further reading:

- Asynchronous programming: futures, async, await - dart.dev
- dart:async library - dart.dev
- Asynchronous programming: Streams - dart.dev
- Exploring Async Programming In Dart & Flutter - medium.com
- Dart's async/await in Flutter - Educative.io

Examples

```dart
// this function is marked as "async" meaning that inside
// it we are allowed to use the "await" keyword in order to
// wait on the result of other functions marked with "async"
Future<int> fetchStatusCode(String urlStr) async {
  final url = Uri.parse(urlStr);
  // for instance the "getUrl" function is an async function
  // that we can "await" on since our function itself is
  // marked as "async"
  final getResult = await HttpClient().getUrl(url);
  // smae for the close() function, it's async so we can
  // await on it since we are an async function too
  final result = await getResult.close();
  return result.statusCode;
}
```