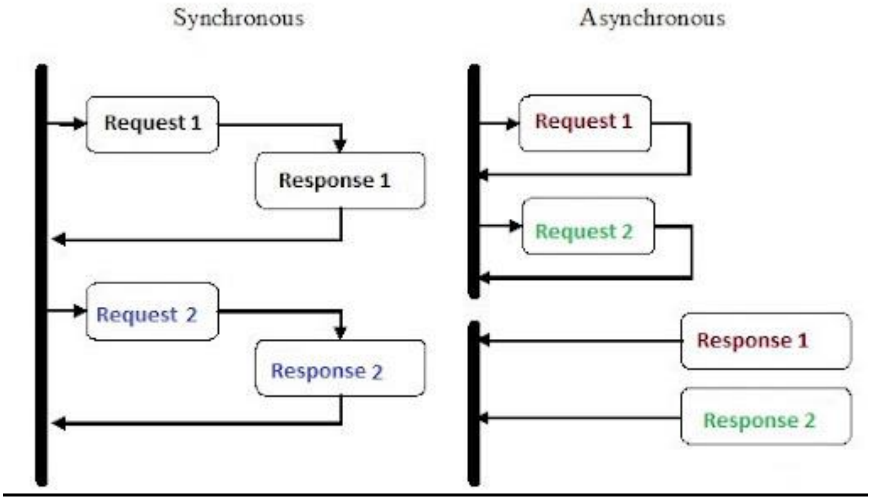
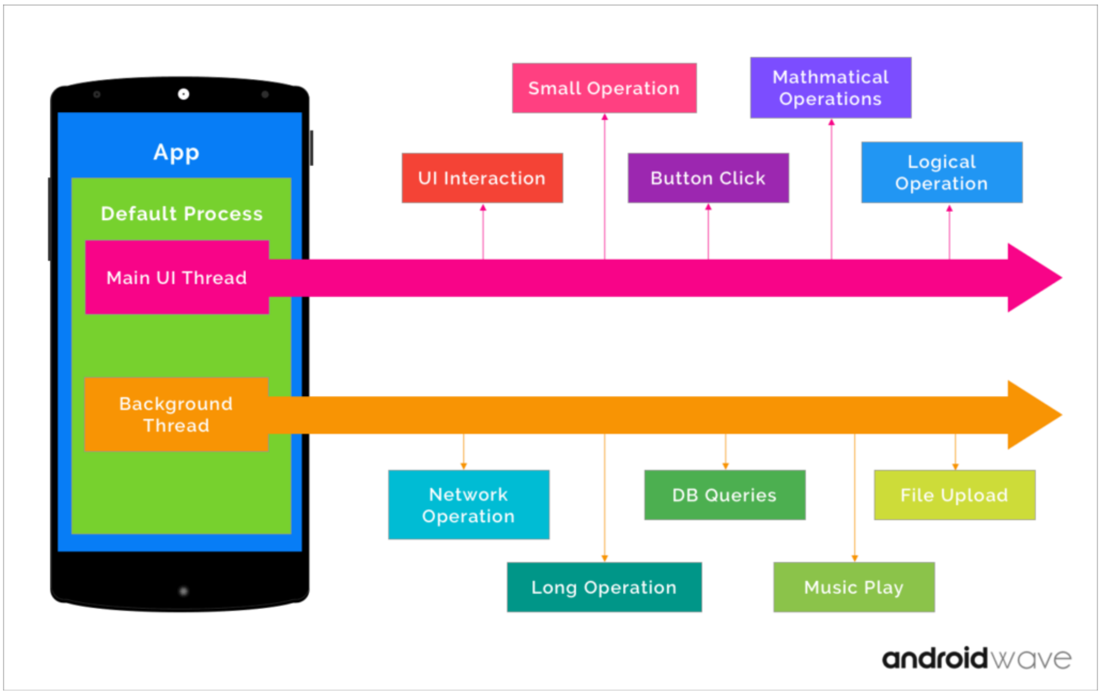
Asynchronous Programming- future, async and await keyword:

- It is a kind of programming in which operations or task are independent of the main thread. At the point when the works is finished, it tells the main thread.

- In other words, asynchronous programming is programming technique in which task or operation will Completed on certain time interval.



In Android Application (Native):



Main UI Thread- for small operation.
Background Thread- for heavy operation

But **Dart/Flutter is single threaded.**

- We can have only main thread in dart/flutter.
- We cannot have background thread.

So, all the heavy operation like network operation, file download, Image loading is done using Main UI thread.

But in flutter doing this long operation in main UI thread freeze your UI i.e. Your application hangs/ resulting poor UI experience/ crashing application.

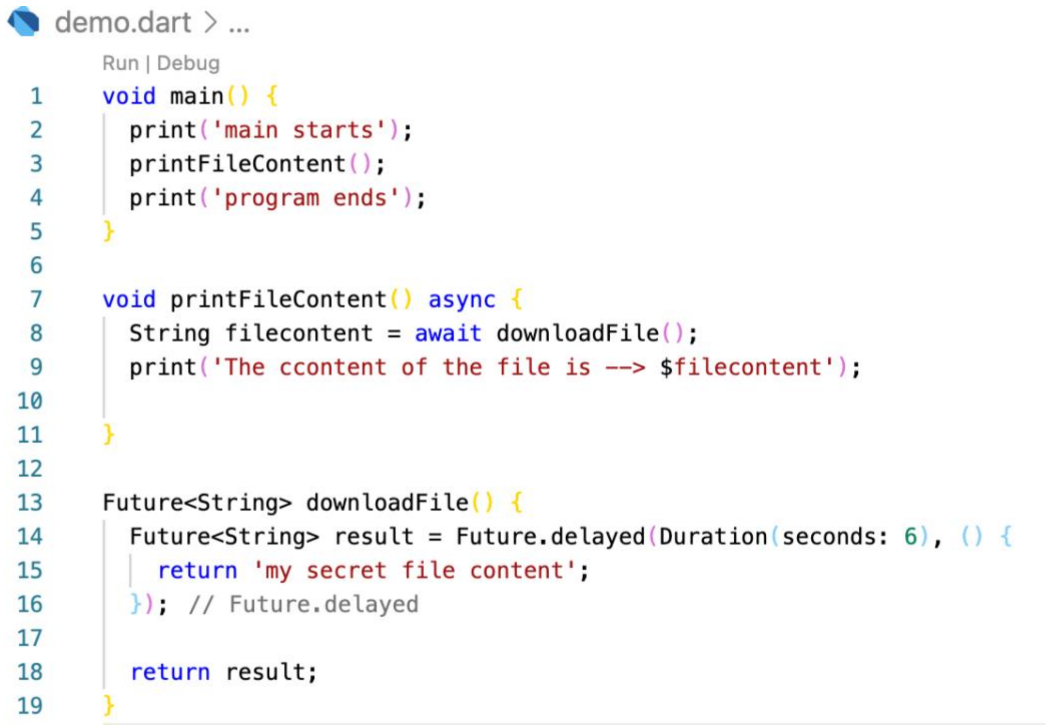
Now the Question arise? How to perform asynchronous programming in flutter?

➔ The solution for this is using **future, async and await**

Future class- *The result of an asynchronous computation.*

An asynchronous computation *cannot provide a result immediately when it is started, unlike a synchronous computation which does compute a result immediately by either returning a value or by throwing. An asynchronous computation may need to wait for something external to the program (reading a file, querying a database, fetching a web page) which takes time. Instead of blocking all computation until the result is available, the asynchronous computation immediately returns a Future which will* eventually *"complete" with the result.*

Async/await keyword - *To perform an asynchronous computation, you use an async function which always produces a future. Inside such an asynchronous function, you can use the await operation to delay execution until another asynchronous computation has a result. While execution of the awaiting function is delayed, the program is not blocked, and can continue doing other things.*

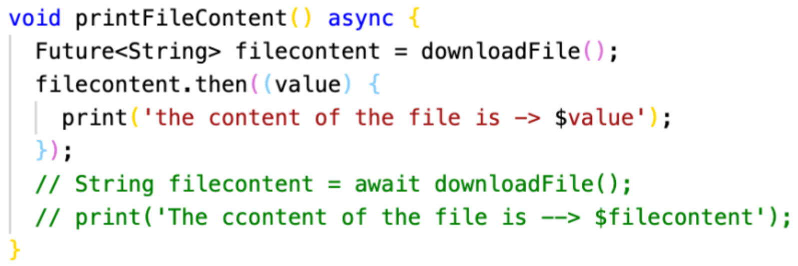


```
void main() {
  print('main starts');
  printFileContent();
  print('program ends');
}

void printFileContent() async {
  String filecontent = await downloadFile();
  print('The ccontent of the file is --> $filecontent');

}

Future<String> downloadFile() {
  Future<String> result = Future.delayed(Duration(seconds: 6), () {
    return 'my secret file content';
  }); // Future.delayed

  return result;
}
```

We can get the values from the future by using-

a. Async and await as above.

b. Using then () of future class.



```
void printFileContent() async {
  Future<String> filecontent = downloadFile();
  filecontent.then((value) {
    print('the content of the file is -> $value');
  });
  // String filecontent = await downloadFile();
  // print('The ccontent of the file is --> $filecontent');
}
```