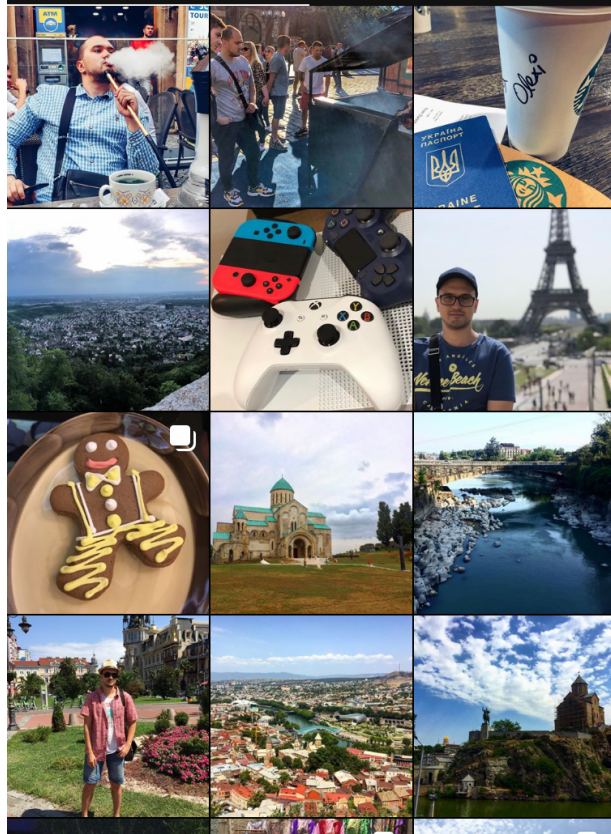


Asynchronous programming

Why?

shalex9154



Multithreading is everywhere!

Basic async operations

- Fetching data over a network
- Writing to a database
- Reading data from a file

Dart async basics

- Future class
- `async`
- `await`

What is a future?

- A future (lower case “f”) is an instance of the `Future` (capitalized “F”) class. A future represents the result of an asynchronous operation, and can have two states: *uncompleted* or *completed*.

Example

Async and Await

- To define an async function, add **async** before the function body
- The **await** keyword works only in **async** functions

Async Example

Key terms

- `async`: You can use the `async` keyword before a function's body to mark it as asynchronous.
- `async function`: An `async` function is a function labeled with the `async` keyword.
- `await`: You can use the `await` keyword to get the completed result of an asynchronous expression. The `await` keyword only works within an `async` function.

Code reuse!

pubspec.yaml

flutter:

dependencies:

http: ^0.12.0+2

```
import 'dart:async';
import 'dart:convert';
import 'package:http/http.dart' as http;
import 'package:lecture_4_weather/models/weather.dart';

class WeatherProvider {
  Future<Weather> getCurrentWeather() async {
    final response = await http.get(
      'https://api.openweathermap.org/data/2.5/weather?q=Kharkiv&units=metric&APPID=1ea55013049215603ece3fee22806975');
    if (response.statusCode == 200) {
      return Weather.fromJson(json.decode(response.body));
    } else {
      throw Exception('Failed to load weather data');
    }
  }
}
```

Let's code!

Homework

1. Add landscape orientation to current app.

