العطاء الرقمي
Attaa Digital

تطوير التطبيقات باستخدام Flutter

Eng Mouaz M. Al-Shahmeh
BSc / ITC - Software Engineer & Developer

أهداف اللقاء

- مقدمة في Flutter
- تهيئة بيئة التطوير
- تشغيل أول تطبيق Flutter
- Declarative UI
- التصميم المتجاوب

أهداف اللقاء

- Null Safety
- State Management
- المفاتيح
- التنقل
- خيارات قواعد البيانات

أهداف اللقاء

- اختبارات الأداء
- أنماط بناء التطبيقات
- ملخص ومراجع

مقدمة في Flutter

What is Flutter?

Flutter is Google's portable UI toolkit for crafting beautiful, natively compiled applications for mobile, web, and desktop from a single codebase. Flutter works with existing code, is used by developers and organizations around the world, and is free and open source.

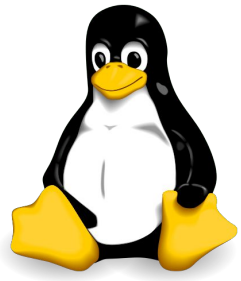
مقدمة في Flutter

- Hot Reload - Hot Restart
- One Code Base (Dart)

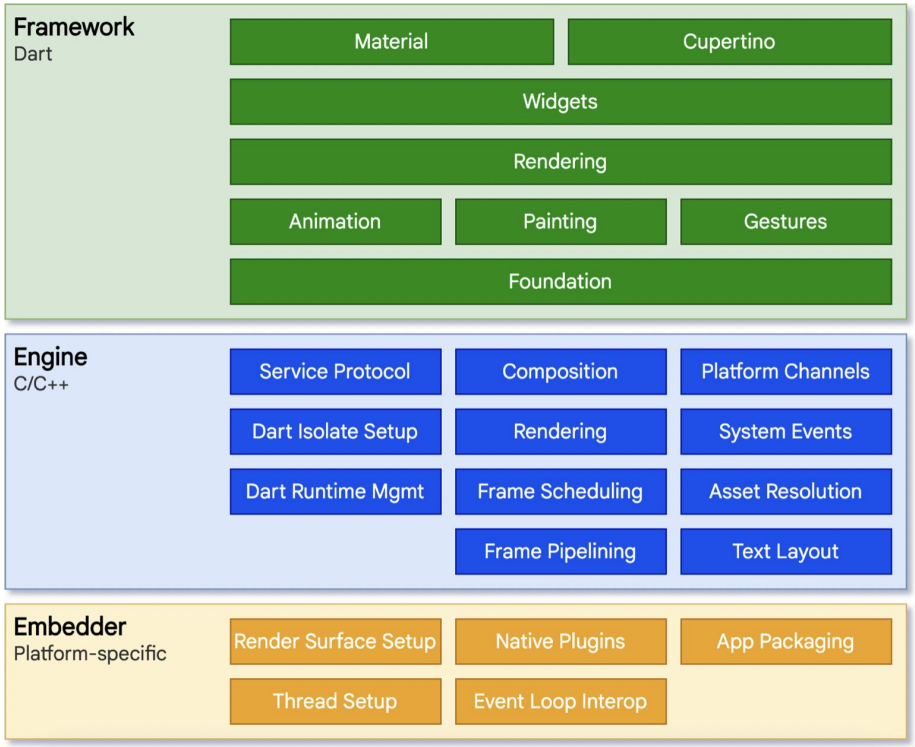
- بيئات تطوير منافسة لتقنية Cross-Platform
  Kotlin Multi Mobile - Xamarin - React Native - ionic
- تطوير التطبيقات الاصلي: اندرويد (java - kotlin) - ابل (objective-c - swift)
- تطبيقات ثلاثية الابعاد (Unity)

مقدمة في Flutter

- تطبيقات الجوال (اندرويد - ابل)
- تطبيقات سطح المكتب (ويندوز - ماك - لينكس)
- تطبيقات الويب

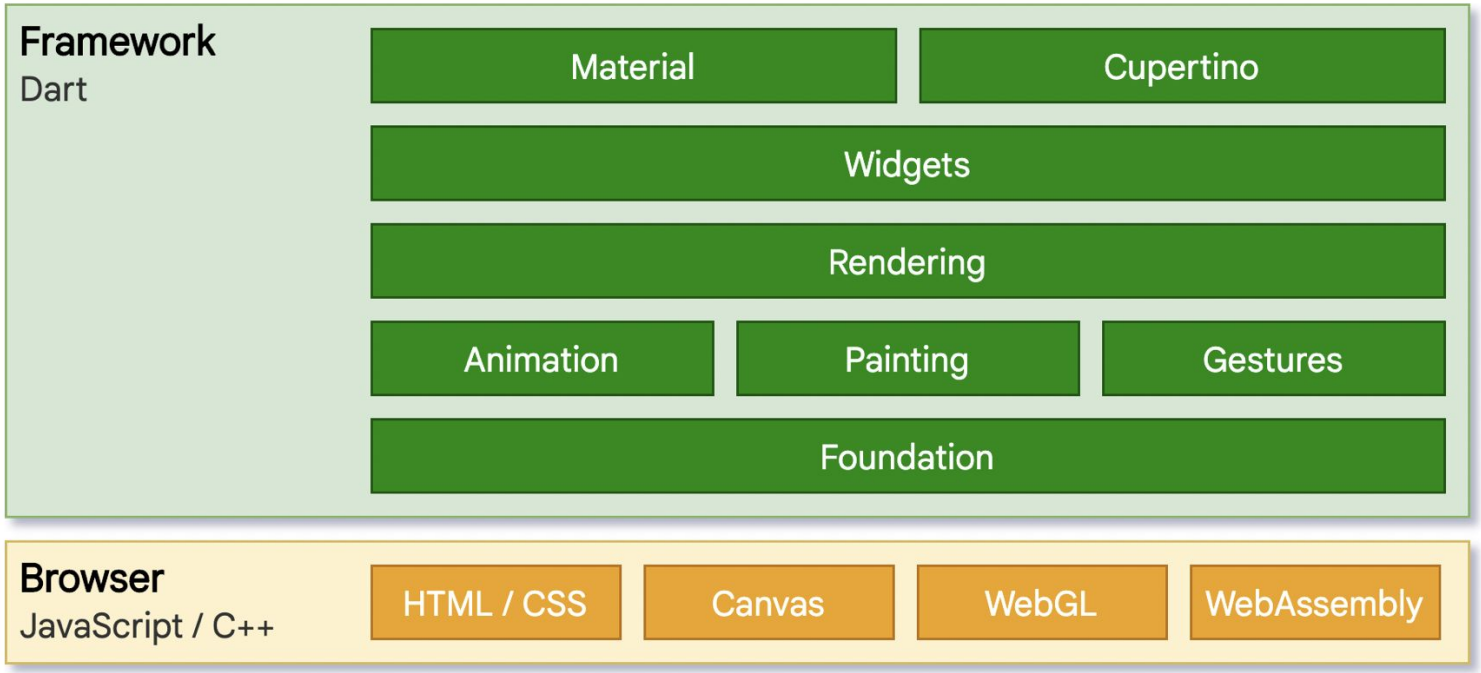


مقدمة في Flutter

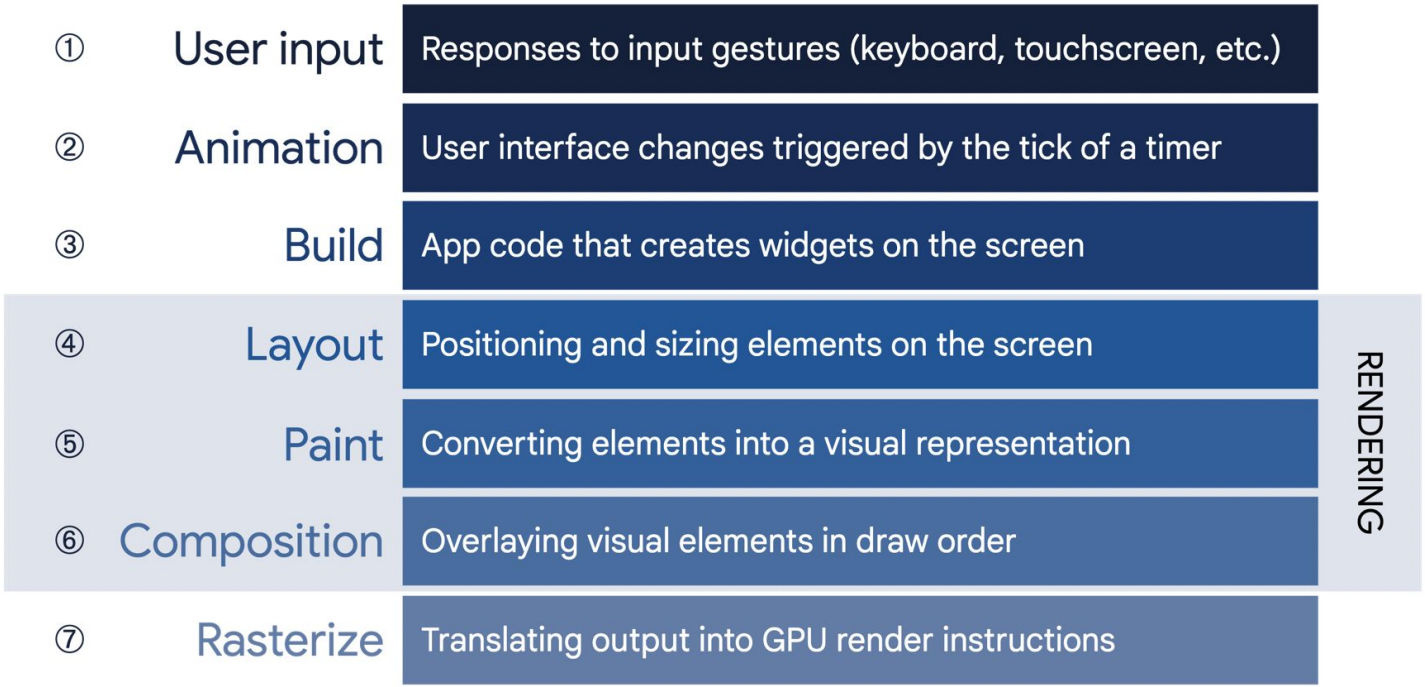


مقدمة في Flutter

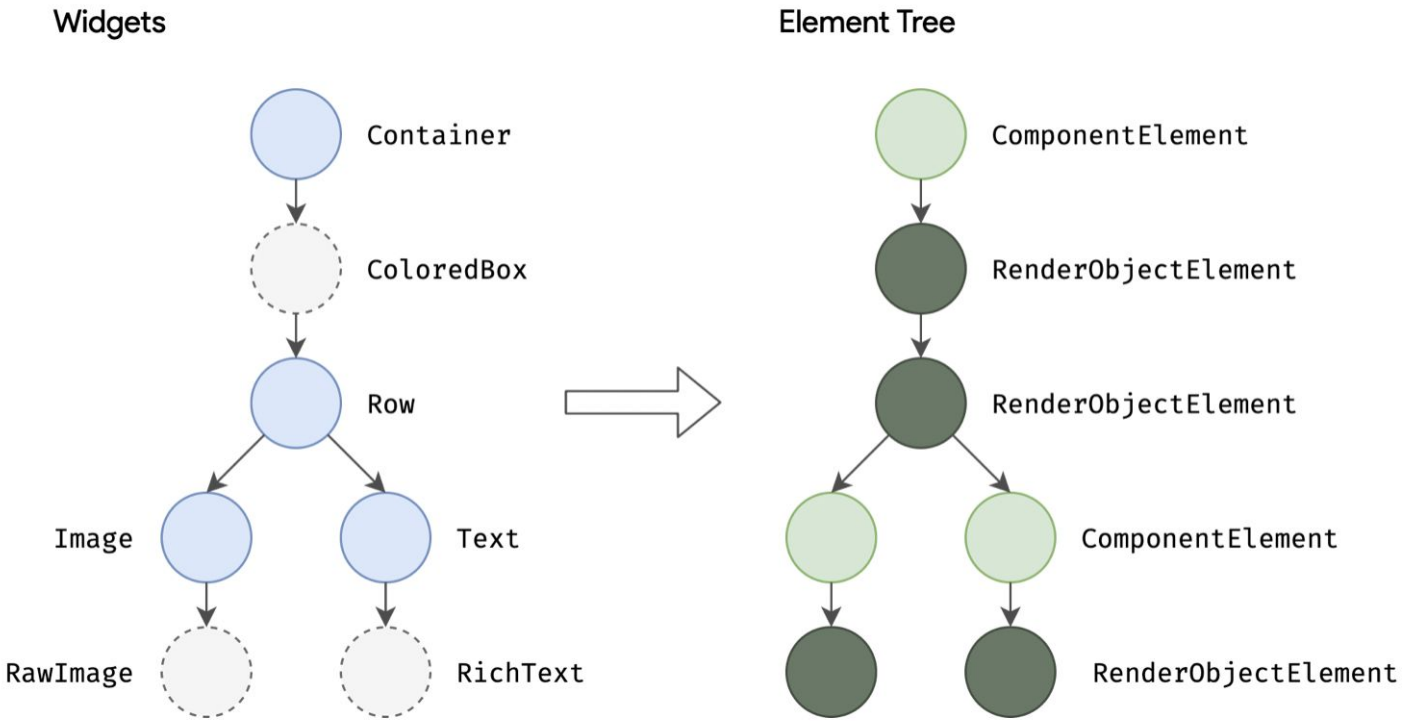
The web version of the architectural layer diagram is as follows:



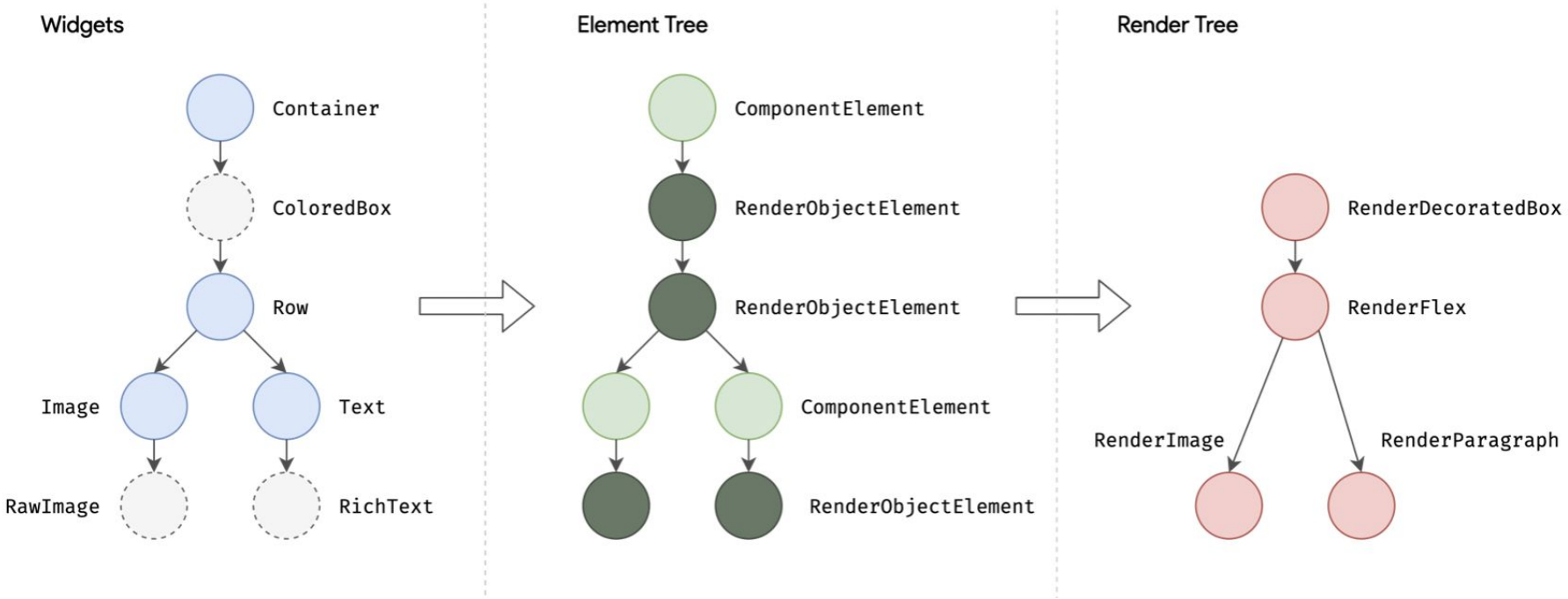
مقدمة في Flutter



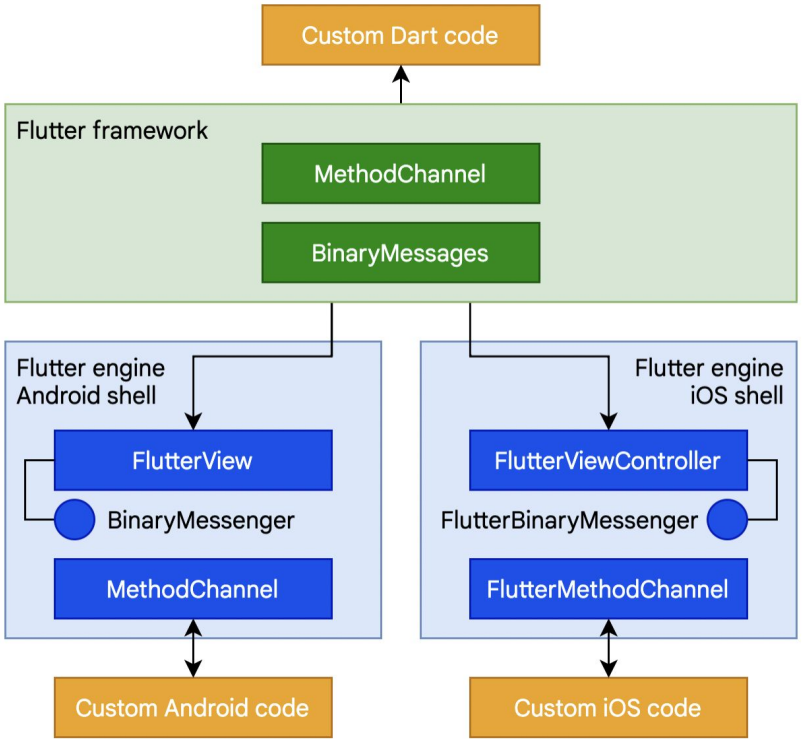
مقدمة في Flutter



مقدمة في Flutter



مقدمة في Flutter

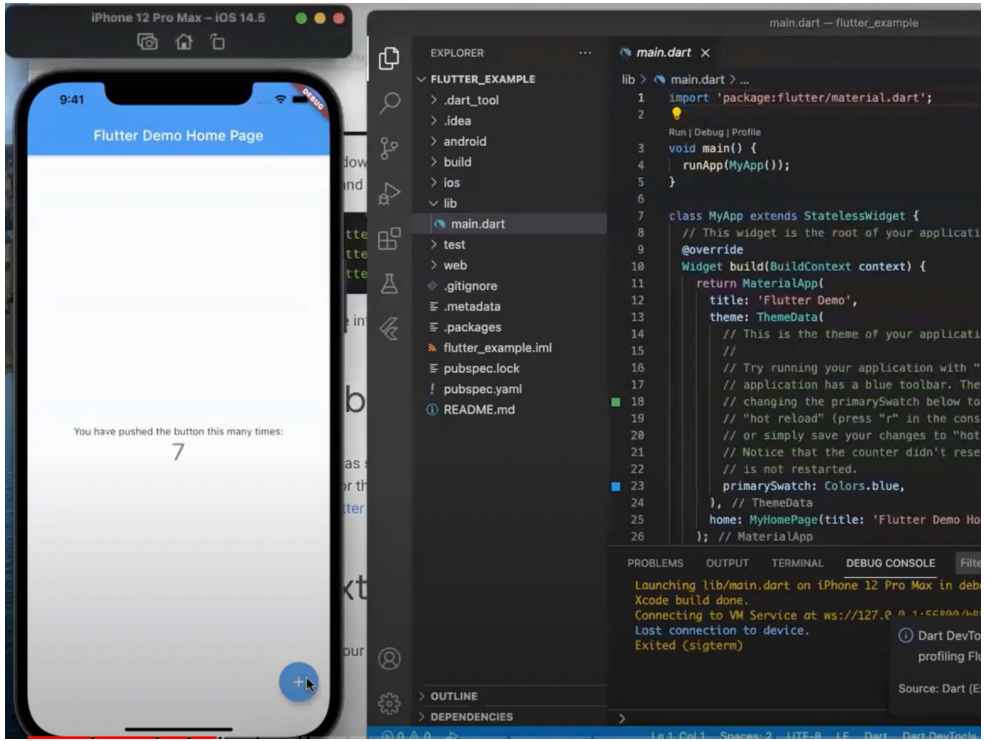


تهيئة بيئة التطوير

- مواصفات جهاز جيدة
- تحميل Flutter SDK
- تحميل برنامج أكواد
- ابل (cocoapods) - اندرويد (JDK java)

تهيئة بيئة التطوير

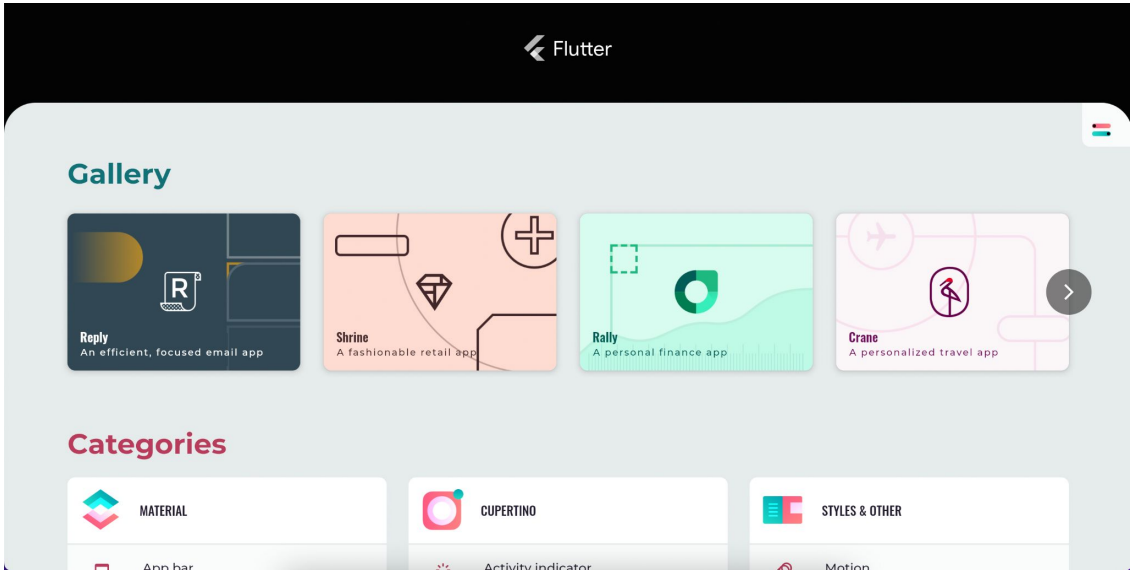
تشغيل أول تطبيق



Declarative UI

التصميم المتجاوب

- MediaQuery.of(context)
- Cupertino (Apple)
- Material (Android)
- gallery.flutter.dev

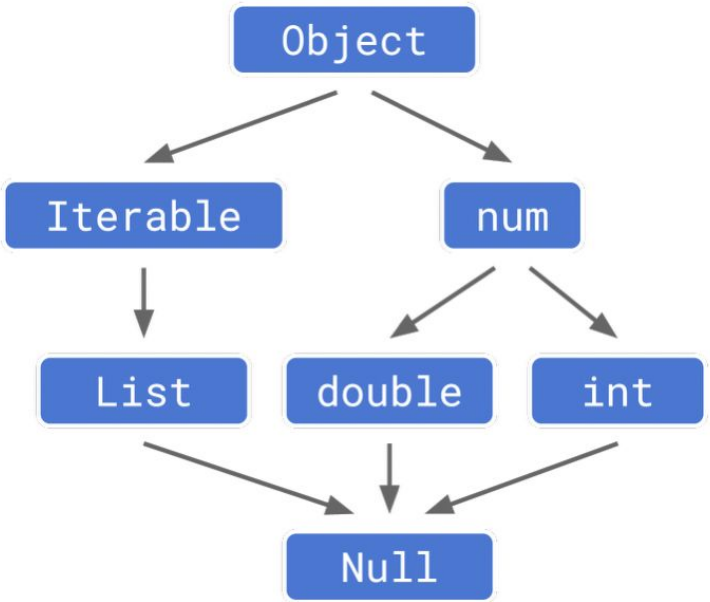


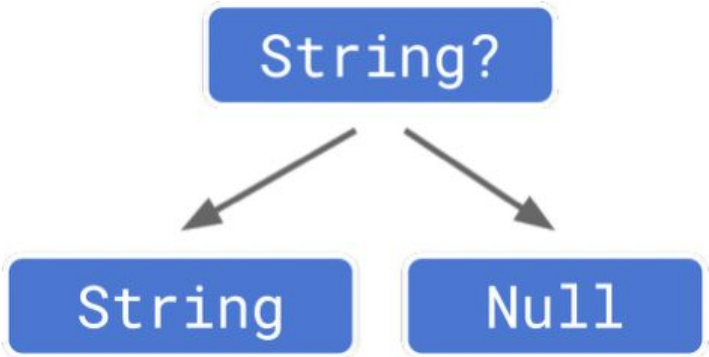
Null Safety

Sound null safety is available in Dart 2.12 and Flutter 2.

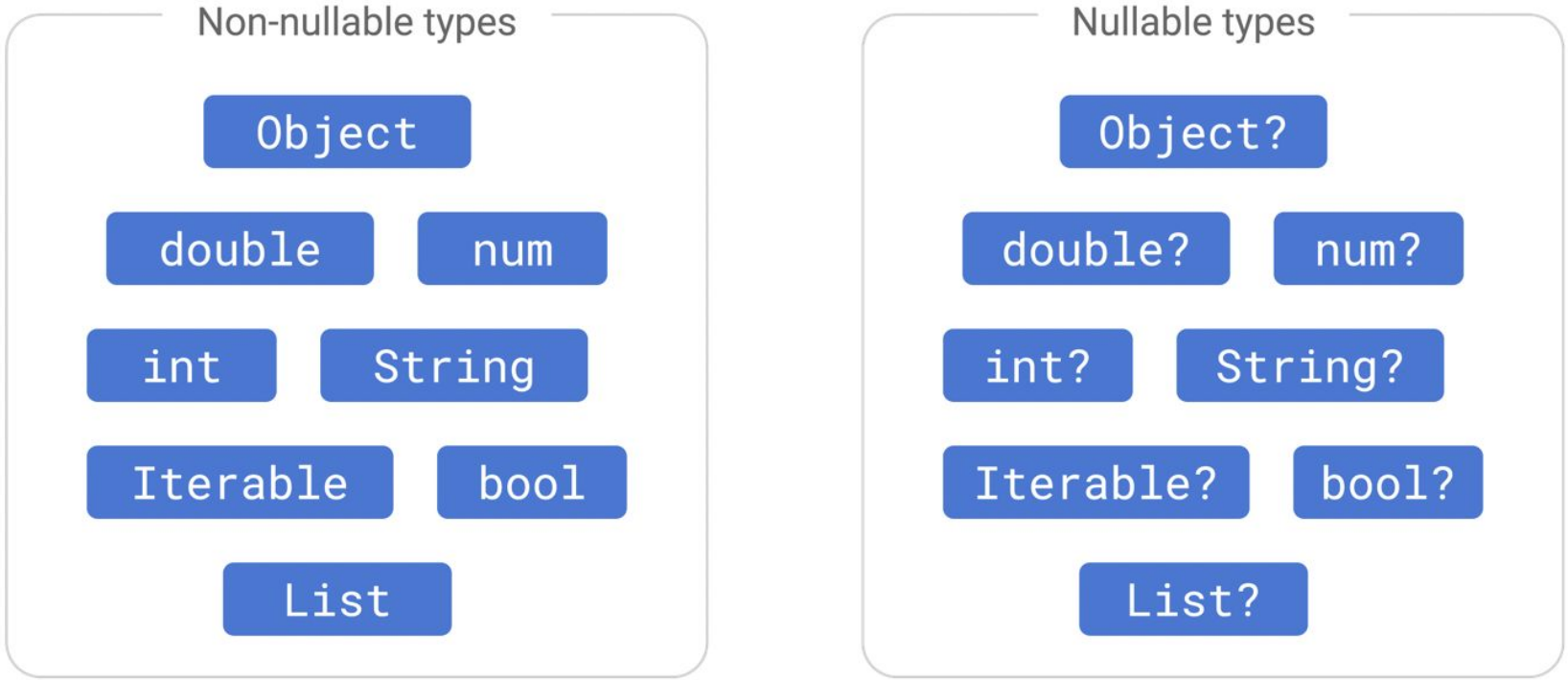
Null Safety

In type theory lingo, the `Null` type was treated as a subtype of all types:

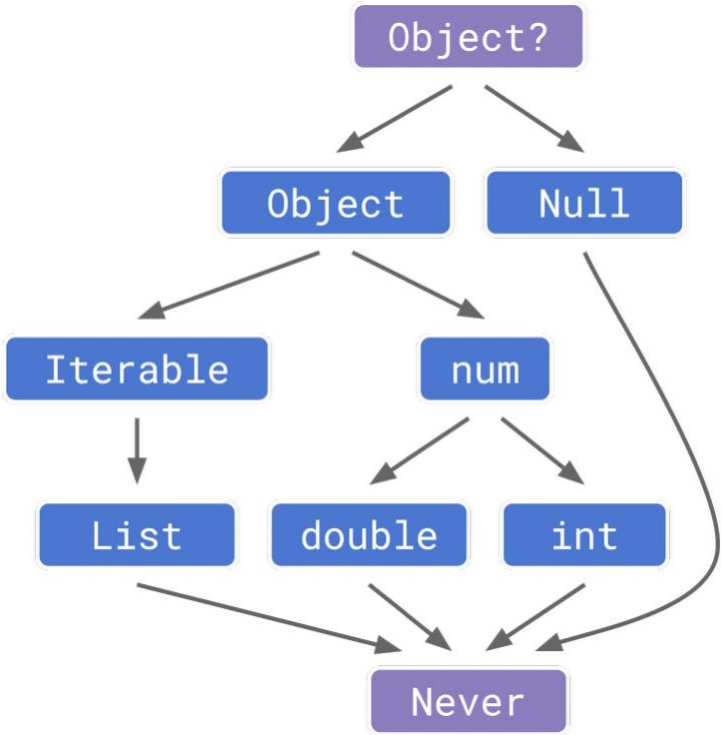




Null Safety


Non-nullable types
Object
double
num
int
String
Iterable
bool
List
Nullable types
Object?
double?
num?
int?
String?
Iterable?
bool?
List?

Null Safety



Null Safety

- Types are non-nullable by default and made nullable by adding `?`.
- Optional parameters must be nullable or have a default value. You can use `required` to make named parameters non-optional. Non-nullable top-level variables and static fields must have initializers. Non-nullable instance fields must be initialized before the constructor body begins.
- Method chains after null-aware operators short circuit if the receiver is `null`. There are new null-aware cascade (`?..`) and index (`?[]`) operators. The postfix null assertion “bang” operator (`!`) casts its nullable operand to the underlying non-nullable type.
- Flow analysis lets you safely turn nullable local variables and parameters into usable non-nullable ones. The new flow analysis also has smarter rules for type promotion, missing returns, unreachable code, and variable initialization.
- The `late` modifier lets you use non-nullable types and `final` in places you otherwise might not be able to, at the expense of runtime checking. It also gives you lazy-initialized fields.
- The `List` class is changed to prevent uninitialized elements.

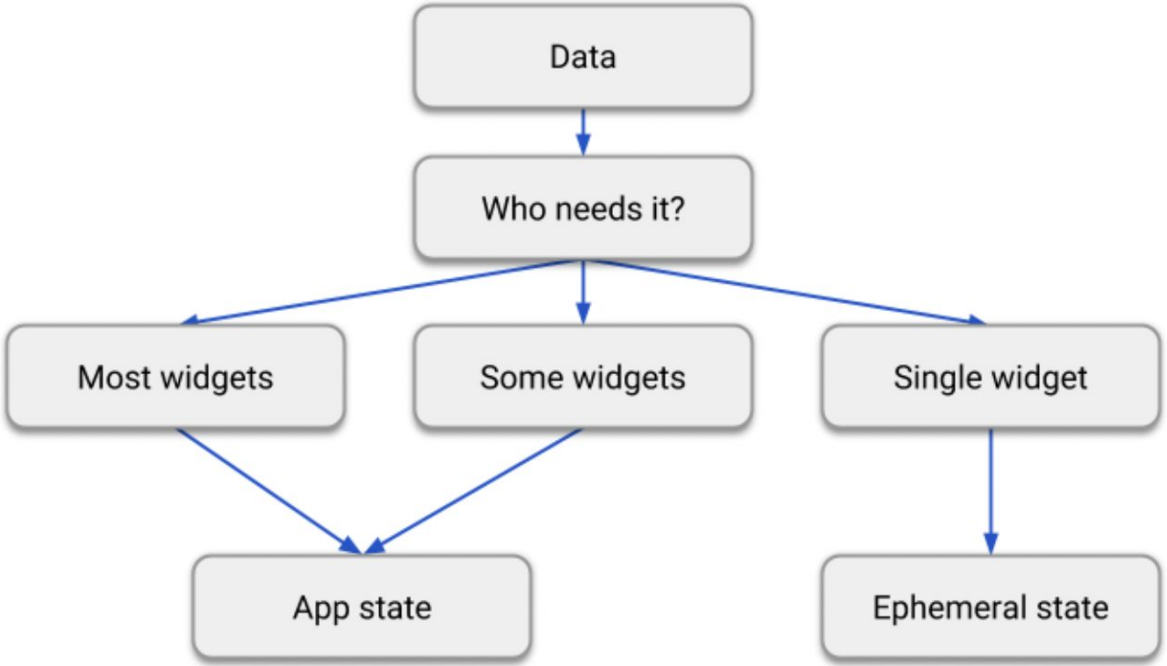
State Management

- Provider
- Riverpod
- setState (StatefulWidget - StatelessWidget - InheritedWidget)
- GetX
- Redux

State Management

- Fish-Redux
- BLoC / Rx
- GetIt
- MobX
- Binder

State Management



Ephemeral state

Ephemeral state (sometimes called *UI state* or *local state*) is the state you can neatly contain in a single widget.

This is, intentionally, a vague definition, so here are a few examples.

- current page in a `PageView`
- current progress of a complex animation
- current selected tab in a `BottomNavigationBar`

App state

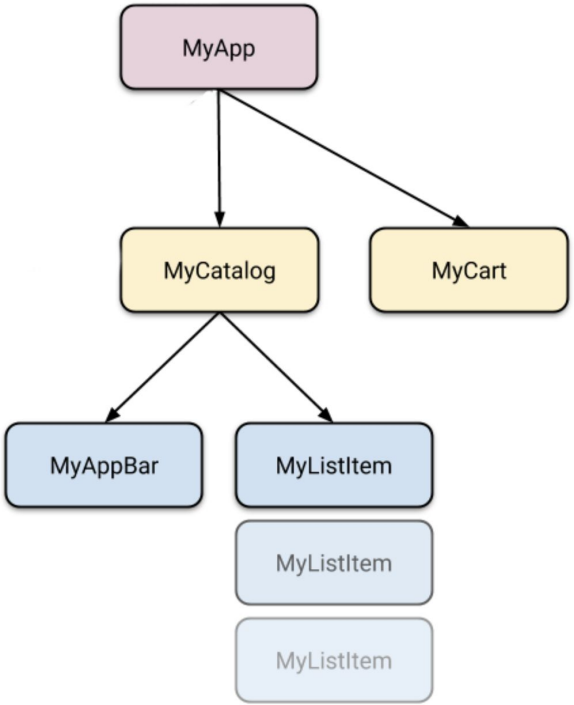
State that is not ephemeral, that you want to share across many parts of your app, and that you want to keep between user sessions, is what we call application state (sometimes also called shared state).

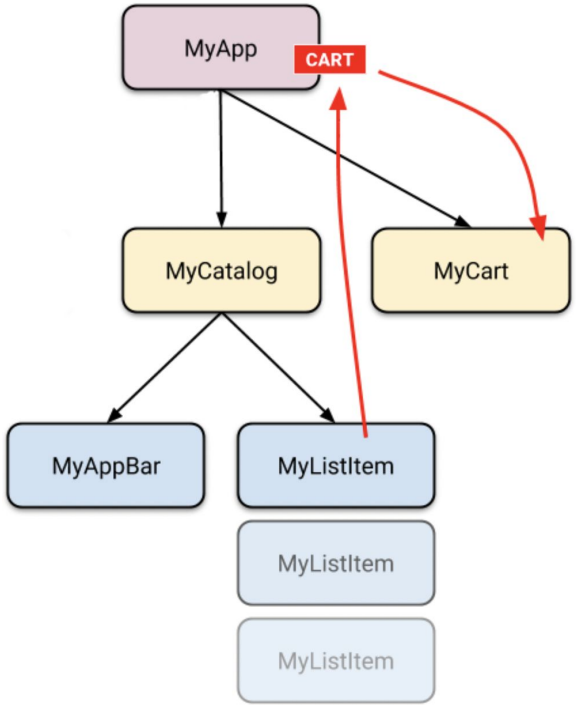
Examples of application state:

- User preferences
- Login info
- Notifications in a social networking app
- The shopping cart in an e-commerce app
- Read/unread state of articles in a news app

State Management

- Provider





State Management

- Provider

```dart
class CartModel extends ChangeNotifier {
  /// Internal, private state of the cart.
  final List<Item> _items = [];

  /// An unmodifiable view of the items in the cart.
  UnmodifiableListView<Item> get items => UnmodifiableListView(_items);

  /// The current total price of all items (assuming all items cost $42).
  int get totalPrice => _items.length * 42;

  /// Adds [item] to cart. This and [removeAll] are the only ways to modify the
  /// cart from the outside.
  void add(Item item) {
    _items.add(item);
    // This call tells the widgets that are listening to this model to rebuild.
    notifyListeners();
  }

  /// Removes all items from the cart.
  void removeAll() {
    _items.clear();
    // This call tells the widgets that are listening to this model to rebuild.
    notifyListeners();
  }
}
```

State Management

- Provider

```
void main() {
  runApp(
    ChangeNotifierProvider(
      create: (context) => CartModel(),
      child: const MyApp(),
    ),
  );
}
```

```
void main() {
  runApp(
    MultiProvider(
      providers: [
        ChangeNotifierProvider(create: (context) => CartModel()),
        Provider(create: (context) => SomeOtherClass()),
      ],
      child: const MyApp(),
    ),
  );
}
```

State Management

- Provider

```
// DON'T DO THIS
return Consumer<CartModel>(
  builder: (context, cart, child) {
    return HumongousWidget(
      // ...
      child: AnotherMonstrousWidget(
        // ...
        child: Text('Total price: ${cart.totalPrice}'),
      ),
    );
  },
);
```

```
// DO THIS
return HumongousWidget(
  // ...
  child: AnotherMonstrousWidget(
    // ...
    child: Consumer<CartModel>(
      builder: (context, cart, child) {
        return Text('Total price: ${cart.totalPrice}');
      },
    ),
  ),
);
```

المفاتيح

When? Adding - Removing - Reordering (collection of widgets of the same type of some state).

- Global Key
- Local Key (
  ValueKey('Emily Furtuna') -
  ObjectKey(MutableRectangle(1,2,3,4)) -
  UniqueKey() -
  PageStorageKey(scrollLocation)
)

Example: Scaffold and Form.

التنقل

- GoRouter (push - go) - (Declarative Navigation - no context)
- Navigator 2 (push - pop - pushReplacement - pushNamed - popUntill)

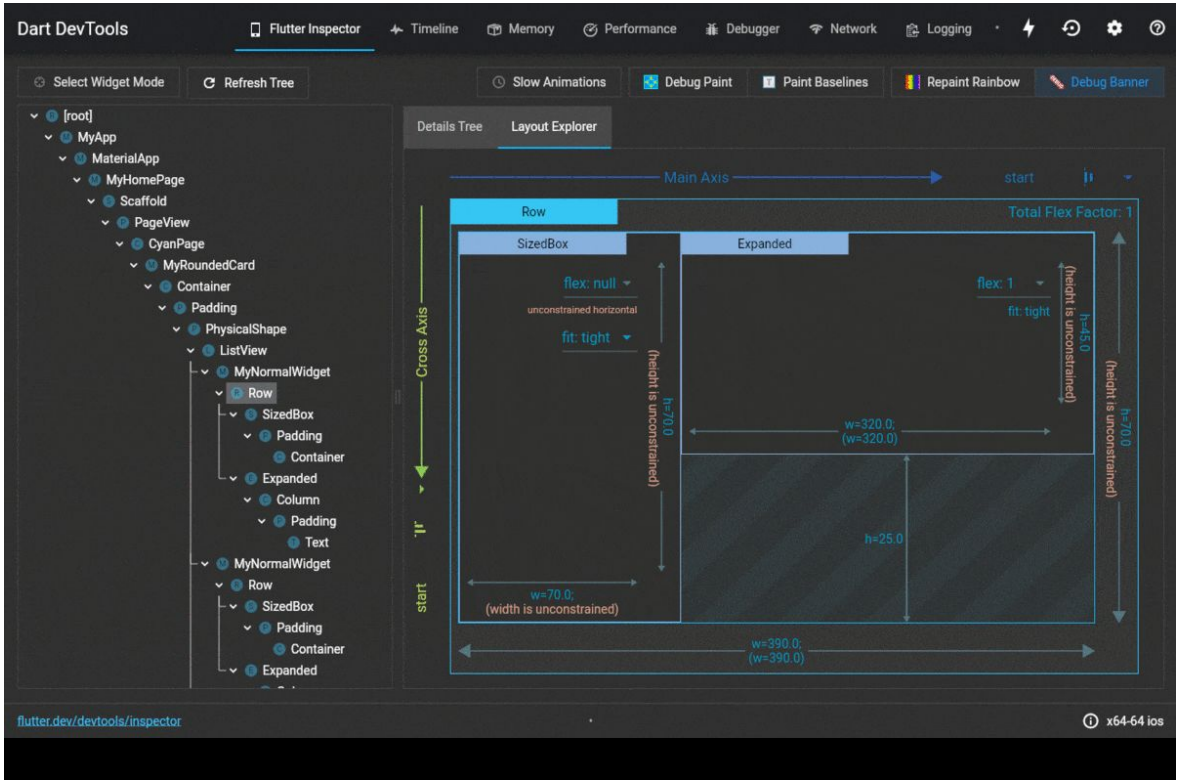
خيارات قواعد البيانات

- Firebase (FlutterFire) - BAAS (Backend As A Service)
- Sqflite - SqfEntity (local)
- Supabase (PostgreSQL)
- API http (json)
- Shared Preferences (path)

اختبارات الأداء

- Dev Tools
- flutter doctor
- flutter debug mode (code trace)

اختبارات الأداء



أنماط بناء التطبيقات

- Debug (print - debugShowCheckedModeBanner - run - hot reload & restart - debug mode)
- Profile (run - real performance - real device - hot reload & restart)
- Release (archive in xcode - apk or bundle in android - app signing - unique id - play store - app store)

ملخص ومراجع

- flutter.dev
- The Boring Flutter Development Show (Youtube)
- Widget of the Week (Youtube playlist)
- pub.dev
- fuchsia.dev (zircon is dart package)