

Asynchronous Programming in Dart In Arabic

المحتوي :

- 1.introduction to Asynchronous Programming in Dart
2. Threads
3. Concurrency and parallelism
- 3.Dart isolates
- 4.How to write async code
5. How to get the value with callbacks and async-await

متطلبات:

1. اساسيات Dart

2. فكرة عن OOP علشان لما استخدم كلمة class فانا متوقع انك عارف معناها

المحتوى ده مجاني وليه جزئين وده الجزء الاول اتمنى لو عجبك واستفادت [تدعمني من هنا](#) او تدعمني بلاك
ونقد على لينكد إن

تخيل إنك في يوم قررت تنزل تشتري كيلو سمك وتشويه في المطعم وتشتري كيلو فاكهة رحت الشوايه وقالك السمك هيجوز في خلال ربع ساعة فأنت دلوقت قررت أنك تروح تجيب الفاكهة لما السمك يجهز هو ده **Asynchronous Programming**, حالات كثير مش هتبقى مضطر تخلي التطبيق بتاعك يقف علشان تعمل task معينة

وهنا يجي دور **Asynchronous Programming** "Asynchronous operations let your program **complete work while waiting** for another operation to finish."

Synchronous vs. asynchronous code

Synchronous	Asynchronous code
<div><div><div><div></div><div></div><div></div></div><div>Synchronous.dart</div><div><pre>fun main(){ print("الوضوء"); print("الصلاة"); }</pre></div><div>● Output:</div><div>الوضوء الصلاة</div></div></div>	<div>> buy fish</div> <div>>buy fruits</div> <div>#The fish is not ready yet.</div> <div>Output:</div> <div>>buy fruits</div> <div>> buy fish</div>
هنا مينفعش tasks تتأجل يعني لازم وضوء علشان اعرف اصلي	هنا الموضوع عادي انا ممكن اجل السمك شويه لحد لما يجهز وروح اعمل حاجه ثانيه اني اشترى الفاكهة وبالتالي انا نجزت نفسي ووفرت وقت

ال tasks اللي هتاخد وقت نوعين

1- I/O(input/output) tasks

2-Computationally intensive tasks

ال I/O tasks دي مهام ملهاش علاقة بـ CPU بتاعك زي مثلا انك عاوز تعمل حاجه من النت و Computationally tasks دي حجات او مهام اللي بيقيم بيها CPU زي مثلا انك تعمل عملية حسابية

Threads:

"A thread is basically a single process that a program can use to complete tasks. Each thread can only do a single task at once"

فيه لغات برمجة بتدعم multiple-threaded ولغات تانية single-threaded

يعني ايه بقى الكلام ده ؟

تخيل أن فيه مطعم شغال فيه شخصين بس الطباخ والمساعد

المساعد هياخد الطلبات ويوديها للطباخ بعدين يرجع يقدمها تاني ويغسل الاطباق وينظف المكان ويوصل الدليفري.

المساعد ده ميعرفش اكثر من task في نفس الوقت وبالتالي هيبقي ابطأ

ده كده single-threaded أو خلينا نقول مصطلح تاني **Concurrency**

Concurrency: multiple tasks take turns running on a **single** CPU core.



إنما multiple-threads ان كل شغلانه هيبقي ليها واحد متخصص بيها واحد لغسيل الاطباق وشخص لتوصيل الطلبات وشخص لتنظيف المكان فكدده tasks كلها بتتعمل بشكل متوازي وبالتالي هيبقى اسرع

وخلينا نقول علي ده مصطلح **Parallelism**

Parallelism: multiple tasks run at the same time on **multiple** processors or CPU cores.

واحد يقولي يا دوك كده الكلام بان ووضح أن خلاص Parallelism اجمد واحسن

الحقيقة أن Dart اختارت انها تبقي single-threaded !

ايه ؟؟ يعني ايه ؟! يعني انا بعت اللي ورايا واللي قدامي علشان اجيب CPU فيه 16 core وكل core فيه اثنين

thread علشان تقولي في الاخر dart is single-threaded

يا خساره بجد, يا بجد خساره

"To benefit from parallelism, the total amount of processor-intensive work in a program must be large enough to dwarf the overheads of parallelization, and a large fraction of that work must be decomposable to run in parallel."

في المثال اللي انا قلته بتاع المطعم, Parallelism هيبقي احسن فعلا لو كان مهام كثير والمهام دي نعرف نقسمها كويس من غير محد يضايق حد أو يعمل حاجه مش بتاعته فالتاني يضايق أو يعمل شغله غلط مثلا لو اللي بيطبخ راح هو اخذ الطلبات من الزبون من غير ما اللي بياخد الطلبات يعرف (هنسميه محمود) بعدين محمود هيروح ياخذ الطلبات فالزبون يقوله ان في حد اخذ الطلبات بالفعل محمود يقوله " لا أنا عارف الحركات دي كويس ده انا من حامول السعيد يا بيه, عاوز تقعد من غير متطلب حاجه ليه بيه؟"

بعدين محمود يروح يسأل الطباخ فيقوله فعلا أنا اخذت الطلبات فمحمود يضايق ويضرب الطباخ تعالي ناخذ المثال ده شويه لحتة ابعد

دلوقت اخذ الطلبات من الزبون ده بيخزن جوا ميموري يا true or false مفروض لما يجي زبون جديد محمود يتوقع دايما أنه false يعني لسه ما مطلبش حاجه فلو راح ولقاه true هيحصل مشكله bug

"The problem isn't so much with parallelism itself, but rather with multiple threads having access to the same state in memory."

المشكله اللي حصلت أن الطباخ ومحمود كان ليهم access لنفس memory فكان لازم علشان يبغي فيه parallelism أننا نضمن يبقى فيه نظام محكم يتأكد ان محدش يدخل في شغل التاني.

بس خلينا لحد نتفق على كام نقطة

1- concurrency بطنى بس مش اوي (هنعرف ليه بعدين)

2- Dart is based around concurrency on a single thread.

Dart isolates:

"All dart code runs in as isolate"

Isolate = Event loop + Memory

Event loop : thread execute events.



كل isolate عبارة عن event loop و memory

الكود بتاعك executed in isolate

من اسمها isolate يعني عازل يعني كل isolate يقدر يتحكم بس في الميموري و event loop بتاعه بس

Isolate is single-threaded

ممکن التطبيق بتاعك يبقی فیہ اکثر من isolate وبالتالي بقي معاك multi-threaded app

مش معني كده انك لازم تستخدم اكثر من isolate لأن معظم الحالات احنا بنستخدم واحد بس

طب لما هم معزولين عن بعض هيقدرُوا يتواصلوا ازاي ؟

عارف اعصابك ؟ عليها myelin sheath عازلة كل عصب عن الثاني مينفعش يلمسوا بعض

طب والأعصاب دي بتتواصل ازاي أو بيكلموا بعض ازاي ؟

الاعصاب بتتواصل عن طريق مواد كيميائية معينة اسمها neurotransmitters بتحمل رسالة من عصب للثاني

علشان توصل **رسالة** معينة زي يا تنشط يا توقف العصب اللي بعده

سيبك بقا من حصة الاحياء دي وقلبي يا عم لما isolate is isolate ازاي ممكن يكلموا بعض او يتواصلوا ؟

بنفس الفكرة عن طريق نقل رسائل معينة بين ports

خلي بالك هم منفصلين في الميموري بس كل isolate يقدر يرسل او يستقبل داتا

ااااا مش فاهم

بص لما تبقي امتحان وعاوز اجابة سؤال من صاحبك وانت عارف ان المعلومات متخزنه في دماغه بس مش

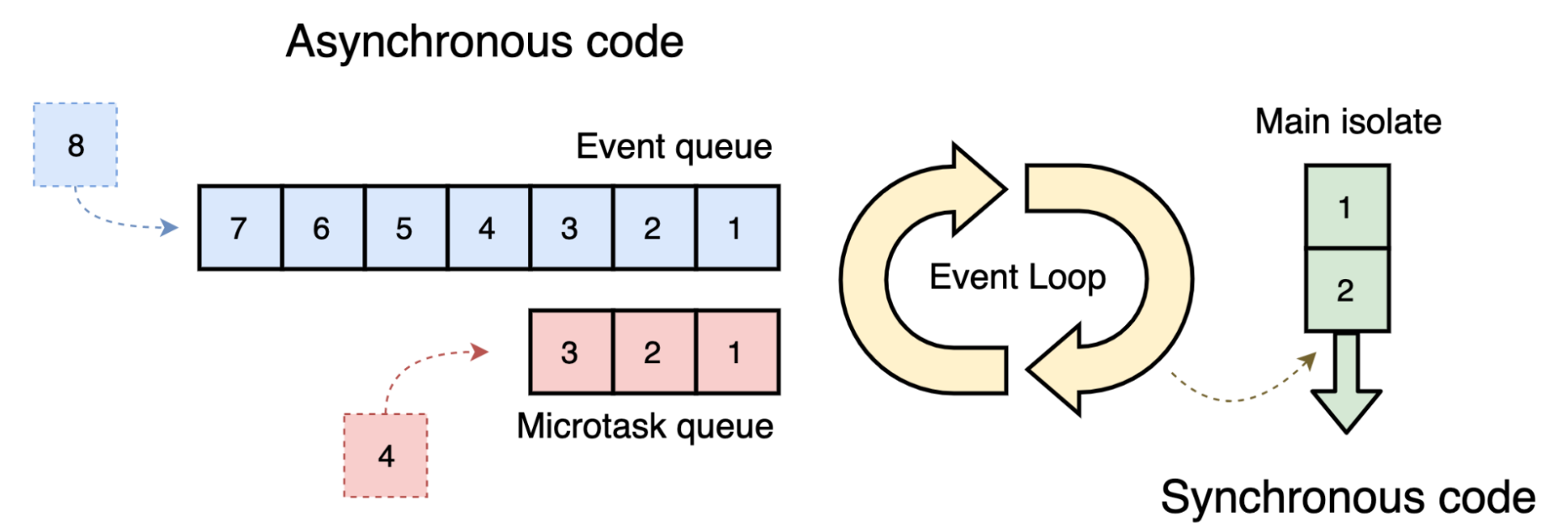
هتقدر تحط ايدك في مخه وتأخذ المعلومه لان دماغه معزولة فيه عظم وحاجات بس تعرف تسأله بكل ادب

وذوق ايه غششني يسطا السؤال ده فيديك الاجابة

حصل دلوقت حاجتين صاحبك بعثلك الداتا **send port**

انت استقبلت الداتا وحطتها في الامتحان **receive port**

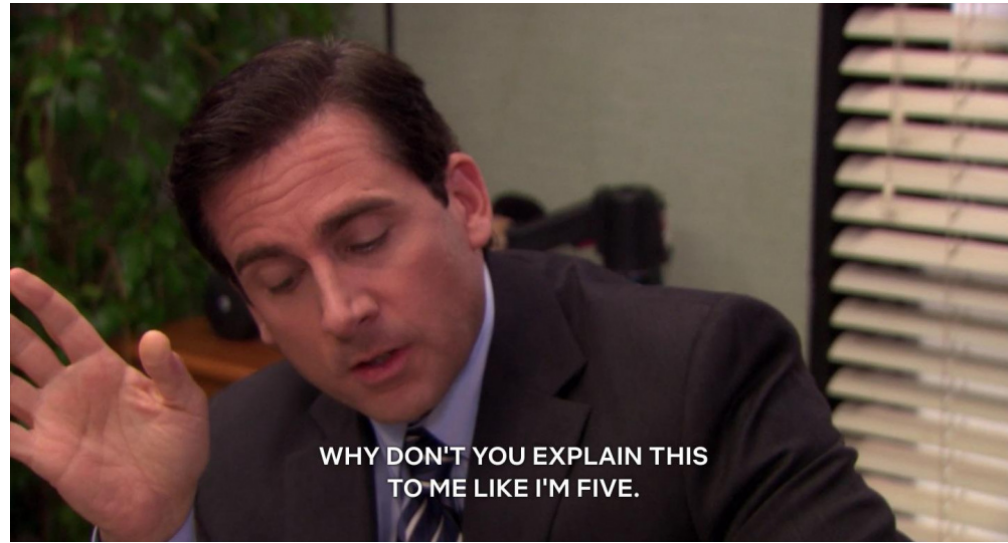
خلينا نروح لسؤال وهو لما اغلب التطبيقات اللي هنعملها هتبقى isolate واحد ازاي لما يبغي فيه تطبيق المستخدم بيضغط علي زر او بيكتب بالكيورد... ازاي isolate واحد هيقدر ينفذ كل ده ؟
<الإجابة في الرسمة دي



ال tasks اللي بتبقى main isolate بتننفذ علطول ومينفعش نأجلها Synchronous tasks
بعد كده ممكن اقسم ال tasks التانيه لنوعين

Event queue	Microtask queue
for events like a user entering a keystroke or touching the screen, or data coming from a database, file, or remote server.	mostly used internally by Dart
لو dart وجد long-running tasks ينفع تتأجل بيحطها هنا في Event queue	
Low priority	High priority

طيب بعد منفذت tasks اللي في main isolate ال event loop هشوف لو فيه tasks في Microtask queue هحطها في main isolate وينفذها لحد لما تخلص وفي الاخر يروح يشوف Event queue لو فيه حاجه ينفذها
Synchronous tasks in the main isolate thread are always run immediately. You can't interrupt them.
حط القاعدة دي في دماغك طالما حاجه دخلت main isolate لازم بتننفذ يعني tasks اللي في event queue لما
يروح في main isolate بتننفذ Synchronous tasks



يلا بينا ناخذ أمثلة :

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.

النتيجة اللي هتيجيلي من Future ده بحرف كبتل يعني class بنعبر عنها باسم future حرف small

```
Asynchronous.dart

void main(List<String> args) {
  final myFuture =
    Future<int>.delayed(
      Duration(seconds: 5),
      () => 20);
}

● Output:
  Instance of 'Future<int>'
```

تعالوا بقى نشوف ايه اللي حصل في الكود ده

عرفت متغير اسمه myFuture

المتغير ده من نوع Future

الحقيقة أن Future is generic علشان كده انا عرفتها من النوع int

استخدمت constructor هنا اللي هوا Future.delayed وفي الكود ده واخذ اتنين argument

اول واحد Duration والثاني anonymous function وهو القيمة اللي myFuture هياخذها

دورات بعد مرور duration دي هيحط anonymous function في anonymous function

جرب تكتب نفس الكود اللي فوق وبعدين اعمل print لي myFuture

```
print(myFuture);
```

Output: Instance of 'Future<int>'

المتغير myFuture مش 20 لاهو future اللي هو Instance of Future

طب ايه future بحرف small ده ؟ ده يعتبر وعد انك هيرجعلك يا int يا error في [المستقبل](#)

طب ازاى دلوقت اطبع القيمة اللي انا مديها للمتغير ؟!

قدامك طريقين يا async-await syntax أو callback

Getting the result with [callbacks](#)

"A callback is an anonymous function that will run after some event has completed."

في حالة future فيه تلت حالات لي callbacks اما catchError , then أو whenComplete

```
Run | Debug
void main(List<String> args) {
  print("before");
  final myFuture = Future<int>.delayed(Duration(seconds: 5), () => 20)
    .then(
      (value) => print("$value"),
    )
    .catchError((onError) => print(onError))
    .whenComplete(() => print("Done"));
  print("after");
}
```

بص كده علي الكود ده نفس الكود اللي فات بس استخدمت التلت حالات بتوع callback

لاحظ أن أنا استخدمت ";" في اخر سطر بس عند whenComplete

طيب then بتعمل ايه ؟ بترجع value

و catchError بتتعامل مع الاخطاء اللي ممكن تحصل

و whenComplete سواء رجعت value or error عندك فرصه انك تكتب كود برده هنا ودي اخر حاجة بتم

طيب تتوقع output هيبقي ايه ؟

```
PROBLEMS
before
after
```

في الاول

```
before
after
20
Done
Exited
```

بعد خمس ثواني

دورات أو event loop شافت synchronous code حطتهم في main isolate ولاقت asynchronous code حطته في event queue الاتنين print اتنفذوا لما main isolate فضت راح بص علي microstatic queue لاقاها فاضيه فراح لي الكود اللي في event queue وحطه في main isolate ونفذه

Working with futures: async and await:

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

بناء على القاعدتين دول تعالي نحول main function من synchronous الي asynchronous

1. هنضيف async قبل function body

```
void main() async { ... }
```

2. لو فانكشن ليها Type هنضيف Future قدام type

#Future<T> T is the type that a function return

طب لو مفيش ؟ زي حالة main هتبقى كالاتي

```
Future<void> main() async { ... }
```

دلوقت نفذها رقم واحد وبقي معانا async function نقدر نستخدم await

كلمة await بتستني future انها تنفذ

```
print(await createOrderMessage());
```

تعالي نلقي نظرة علي المثال الاتي

```
String createOrderMessage() {
  var order = fetchUserOrder();
  return 'Your order is: $order';
}

Future<String> fetchUserOrder() =>
  // Imagine that this function is
  // more complex and slow.
  Future.delayed(
    const Duration(seconds: 2),
    () => 'Large Latte',
  );

void main() {
  print('Fetching user order...');
  print(createOrderMessage());
}
```

اللي انا طالبه منك دلوقت انك تاخد الكود ده VS Code او اي IDE وتحوله لي asynchronous اشطاط ؟

خلينا قبل منحوله نشوف output بتاع الكود ده

```
Fetching user order...
```

```
Your order is: Instance of '_Future<String>'
```

تمام متوقع جدا

طب نعالى نحوله

طبقا للقاعدتين اللي هفكرك بيهم تاني

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

```
Future<String> createOrderMessage() async {
  var order = await fetchUserOrder();
  return 'Your order is: $order';
}

Future<String> fetchUserOrder() async =>
  // Imagine that this function is
  // more complex and slow.
  Future.delayed(
    const Duration(seconds: 2),
    () => 'Large Latte',
  ); // Future.delayed

Run | Debug
Future<void> main() async {
  print('Fetching user order...');
  print(await createOrderMessage());
}
```

Output:

```
Fetching user order...
Your order is: Large Latte
Exited
```

ايه اللي حصل ؟ اول حاجه خرجنا "Fetching user order" وخلصنا منها لانه synchronous

اي حاجه بعد await بتروح في event queue وهوريك ازاى في الجزء الثاني

بعد كده لف ودوران event loop معتش قدامه غير event queue علشان ينفذها صح ؟ شرحنا دي كثير

مفروض دلوقت يطبع creatOrderMessage() اللي بترجع قيمة order اللي هتتاخر ثانيتين وبعدين ترجع قيمة Large latte

"ياريت لو المحتوى ده عاجبك تدعمني بليك أو نقد أو لو شفته مفيد [ممکن تتبرع من هنا](#) واستني الجزء الثاني "