อัพเดทครั้งที่ 2

สูตรลัด Dart 3

เรียนรู้ภาษา Dart ก่อนไปเขียนแอปด้วย Flutter

อนุชิต ชโลธร

สำนักพิมพ์ก๊อปวาง

สูตรลัด Dart 3

เรียนรู้ภาษา Dart ก่อนไปเขียนแอปด้วย Flutter

พิมพ์ครั้งที่ 2

อนุชิต ชโลธร

สำนักพิมพ์ก๊อปวาง

สารบัญ

บทนำ

Dart เป็นภาษาที่ออกแบบมาให้ใช้งานง่าย และยังสามารถใช้งานได้หลายแพลทฟอร์ม ทำให้คุณสามารถคอมไพล์แอปให้ใช้งานได้หลายแพลทฟอร์ม ไม่ว่าจะเป็น Windows, Mac, Linux, Android, iOS, Embeded System หรือแม้กระทั่ง Web ก็สามารถใช้ภาษา Dart ได้ด้วย ความยืดหยุ่นของตัวภาษา ทำให้ Dart ถูกเลือกใช้งานคู่กับ Flutter และผนวกรวมเข้ากับ Flutter เพื่อให้การพัฒนาแอปแบบ GUI บนแพลตฟอร์มต่างๆ ได้ง่าย สะดวก รวดเร็วมากยิ่งขึ้น

หนังสือเล่มนี้จะพาคุณไปเรียนรู้ภาษา Dart การเขียนภาษา Dart ตลอดจนการคอมไพล์ ผ่านเครื่องมือ dart tool เพื่อให้เรียกใช้งานแอปบนแพลตฟอร์มต่างๆ ผ่านตัวอย่างในหนังสือ ได้แก่

- Command line App
- Command line App เชื่อมต่อกับ API
- Dart HTTP Server
- Web App แบบ Server Side Rendering โดยใช้ Angel Framework
- Web App แบบ Client Side Rendering โดยใช้ Flutter for Web

แนะนำภาษา Dart

Dart เป็นภาษาที่ออกแบบมาให้พัฒนาแอปบนแพลทฟอร์มต่างๆ ให้ง่ายและรวดเร็ว ช่วยใช้เขียนโปรแกรมในหลายๆ แพลทฟอร์มได้ง่ายและสะดวกยิ่งขึ้น ภาษา Dart มีความยืดหยุ่นทำให้สามารถ compile เป็นแอปสำหรับ run ได้ในหลายๆ แพลทฟอร์ม

ภาษา Dart เหมาะสำหรับการพัฒนาแอปในฝั่ง Client โดยมีคุณสมบัติอย่างเช่น Hot reload เพื่อช่วยให้นักพัฒนาเห็นความเปลี่ยนแปลงทันทีเมื่อแก้ไขโค้ดและยังรองรับการ compile ไปยังแพลทฟอร์มต่างๆ ทั้ง mobile, desktop และ web

ภาษา Dart ยังเป็นพื้นฐานของการพัฒนาแอปโดยใช้ Flutter อีกด้วย นอกจากคุณสมบัติสำหรับนักพัฒนา เช่น การจัดรูปแบบ การวิเคราะห์โค้ด และรองรับการทดสอบโค้ด (Testing) อีกด้วย

ภาษา Dart

Dart เป็นภาษาแบบ type safe คือ เป็น static type ตัวแปรทุกตัวต้องมี type ตัวภาษาจะคอยตรวจสอบ static type อยู่ตลอดเวลา ด้วยความยืดหยุ่นของภาษา Dart มี type แบบ dynamic เพื่อรองรับกรณีที่โค้ดของต้องการใช้ type ในลักษณะนี้

Dart ยังรองรับ sound null safety โดยบังคับให้ตัวแปร ไม่สามารถมีค่าเป็น null ได้ ยกเว้นคุณกำหนดให้ ตัวแปรนั้นเป็น null ได้ ซึ่งหากเทียบกับภาษาอื่น อาจจะแปลกสักหน่อยที่ตัวภาษากลับบังคับว่า ตัวแปรจะมีค่าเป็น

null ไม่ได้โดยปริยาย (non-nullable) หากตัวแปรนั้นจะมีค่าเป็น null นักพัฒนาจะต้องกำหนดให้ตัวแปรนั้นสามารถเป็น null ได้ (nullable)

ไลบรารี (Library)

Dart มีไลบรารีหลายตัวและเพิ่มขึ้นเรื่อยๆ เพื่อรองรับงานเขียนโปรแกรมในด้านต่างๆ เช่น

- build-in types, collections และ core function ต่างๆ ในภาษา Dart อยู่ในไลบรารี dart:core
- Collection type เช่น queues, linked list, hashmap, binary tree เป็นต้น อยู่ในไลบรารี dart:collection
- การเข้ารหัส ถอดรหัส ข้อมูลเพื่อใช้แสดงผลข้อมูล เช่น JSON, UTF-8 เป็นต้น อยู่ในไลบรารี dart:convert
- ฟังก์ชั่นเกี่ยวกับคณิตศาสตร์ สุ่มข้อมูล จะอยู่ในไลบรารี dart:math
- การจัดการ File, Socket, Http และ I/O สำหรับแอปที่ไม่ใช่ web อยู่ในไลบรารี dart:io
- Asynchronous programming และ Future class และ Stream อยู่ในไลบรารี dart:async
- List ที่มีขนาดเท่าๆ กัน เช่น unsigned 8 byte integer อยู่ในไลบรารี dart:type_data
- การเชื่อมต่อกับโค้ดภาษาอื่นๆ (Foreign function interface for interoperability : ffi) เช่น การเรียกใช้ไลยรารีจาก C, Python จะใช้ไลบรารี dart:ffi
- การเขียนโปรแกรมแบบ concurrent จะใช้ isolates สร้าง worker แยกออกมาเพื่อทำงานนั้นๆ คล้ายกับ thread แต่ไม่ share memory

ร่วมกัน และสื่อสารกันผ่าน message เท่านั้น อยู่ในไลบรารี dart:isolate

- HTML และ web application ที่ต้องสื่อสารกับ browser และการใช้ Dom จะใช้ไลบรารี dart:html

คอร์ไลบรารีข้างต้น ยังมี API อีกหลายอย่างที่อยู่ในรูปแบบ package ซึ่งทีมพัฒนาภาษา Dart ยังออก package เหล่านี้ออกมาเรื่อยๆ เพื่อเสริมความสามารถของภาษาให้ดียิ่งขึ้น เช่น

- characters
- intl
- http
- crypto
- markdown

นอกจาก package จากทีมพัฒนาภาษา Dart แล้ว ยังมี package จะนักพัฒนาอื่นๆ (3rd party publisher) อีกนับพัน package เพื่อรองรับ feature เพิ่มเติม เช่น

- XML
- Windows integration
- SQLite
- compression

คุณสามารถดูแพจเกจต่างๆ ได้จากเว็บไซต์ pub.dev

รองรับหลายแพลตฟอร์ม

Dart มีเทคโนโลยีของ compiler ที่ทันสมัย ช่วยให้คุณสามารถ run ได้หลายรูปแบบ

- Native platform สำหรับ mobile app, desktop device โดย Dart มี Dart VM ที่มาพร้อมกับการ compile แบบ just-in-time และการ compile แบบ ahead-of-time เพื่อใช้ในการ compile โค้ดภาษา Dart ให้เป็น binary ในแต่ละแพลตฟอร์มได้
- Web platform สำหรับ web app โดย Dart สามารถ compile จากโค้ดภาษา Dart ให้เป็น Javascript และ Web Assembly ได้

Flutter

Flutter เป็น Multi platform UI Toolkits ที่ได้รับความนิยม รองรับการแสดงผล UI ได้ในหลายแพลตฟอร์ม โดยใช้ความสามารถของภาษา Dart เข้ามาช่วย Flutter มีไลบรารีด้าน UI รองรับทั้ง iOS, Android, macOS, Windows, Linux และ Web

Dart Native (Machine code JIT และ AOT)

Dart เป็นภาษาที่รองรับ Hot-reload ทำให้ระหว่างการพัฒนา Dart VM จะใช้ just-in-time compiler (JIT) ใช้ในการ recompile โค้ดที่มีการแก้ไขเพื่อให้รองรับ Hot-reload และรองรับ DevTools เพื่อใช้ในการทำ debug โค้ดที่กำลัง run อยู่ได้

เมื่อพัฒนาแอปเสร็จเรียบร้อยแล้ว และต้องการ deploy ไปยัง production ไม่ว่าจะเป็นการ deploy app ไปยัง AppStore, PlayStore หรือการ

deploy แบบ backend application จะต้อง compile แบบ ahead-of-time เพื่อให้ได้ native machine code ใช้ run บน ARM หรือ x64 การ compile แบบ AOT คุณจะได้แอปที่มีขนาดเล็กลง และทำงานได้เร็วขึ้น

Dart Web (JavaScript dev & production)

Dart Web รองรับการ run คำสั่งภาษา Dart บน Web platform โดยใช้โค้ดภาษา Javascript ทำงานบน browser เหมือนกับ V8 ใน Google Chrome

Dart Web มีโหมดการ compile 2 แบบ

1. Incremental development ใช้สำหรับช่วงการพัฒนา
2. Optimizing production คือการ compile สำหรับงาน production ให้ได้โค้ดที่มีขนาดเล็กลง ทำงานได้เร็วขึ้น และที่สำคัญคือได้ผลลัพท์ออกมาเป็น Javascript ที่สามารถเอาไปใช้งานได้

เริ่มเรียนภาษา Dart

การเรียนภาษา Dart มีหลายช่องทาง คุณสามารถศึกษา / ทดลองเขียนภาษา Dart ได้จาก

- Dart Pad เป็น web IDE คุณสามารถเขียนโค้ดและ run บน browser เพื่อดูผลลัพท์ได้
- เรียนรู้จาก Tutorial จากเว็บไซต์ dart.dev
- อ่าน API Documentation เพื่อทำความเข้าใจ core library
- อ่านหนังสือเล่มนี้ ในหัวข้อต่างๆ ที่นำเสนอในบทถัดๆ ไป

Hello World

ภาษา Dart เริ่มทำงานที่ฟังก์ชั่น main หรือ main() โดยฟังก์ชั้นนี้ไม่มีอะไร return ออกมา

```
void main() {
  print('Hello, World');
}
```

ลองใช้ DartPad ใส่โค้ดข้างต้น แล้วลองกดปุ่ม Run เพื่อดูผลลัพท์