

1. Dart គឺជា **Programming Language** មួយដែលបង្កើតដោយ Google សម្រាប់បង្កើត

- Mobile apps ( Android & iOS )
- Web apps
- Desktop apps

Dart ប្រើជាចម្បងជាមួយ **Flutter Framework**។

**ប្រយោជន៍របស់ Dart**

- មានសមត្ថភាព **Fast Development** ( Hot Reload )
- **Cross-platform** កូដមួយដើរបានលើ Android, iOS, Web, Desktop
- Syntax ងាយស្រួល ដូច Java/C#
- Performance លឿន ( ជាទូទៅ compile ទៅ native code )
- Support OOP ( Class, Object, Inheritance, Polymorphism )

**ទម្រង់ទូទៅ ( Structure ) របស់ Dart Program**

```
void main( ) {  
  
    print( "Hello Dart" );  
  
}
```

2. ហេតុអ្វីបានជាកម្មវិធីមួយចំនួនប្រើ Framework និង Pattern ?

**ហេតុអ្វីប្រើ Framework**

- Framework ជួយ ងាយស្រួលសរសេរកម្មវិធី
- មាន **structure ត្រៀមរួច** → Developer សរសេរកូដតិច
- មាន Function និង Library ផ្តល់ជាស្រេច
- បង្កើនល្បឿនអភិវឌ្ឍន៍

- ក្នុងមាន **standard** និងគួរឱ្យទុកចិត្ត
- ការថែទាំ (maintenance) ងាយ

ឧទាហរណ៍៖ Flutter, Laravel, React, Spring Boot

### ហេតុអ្វីប្រើ Design Pattern

- ដោះស្រាយបញ្ហាដែលប្រហែលជា repeat
- ធ្វើឱ្យ code **Reusable**
- ក្នុងមាន **clean, organized, scalable**
- ការងាយក្នុងការបង្កើត team work
- មិនបង្កើតបញ្ហា architecture នៅពេល project ធំ

ឧទាហរណ៍ Pattern: MVC, MVVM, Singleton, Factory

### 3. ហេតុអ្វីត្រូវការ UX/UI ក្នុងការបង្កើត Application? ហើយខុសគ្នាយ៉ាងដូចម្តេច?

#### UX (User Experience)

- គឺការចនាបទពិសោធន៍អ្នកប្រើ
- ផ្ដោតលើភាពងាយប្រើ, សាមញ្ញ, និងលំហូរដំណើរការ (Flow)
- ត្រូវគិតពី Logic, Navigation, Interaction

#### UI (User Interface)

- គឺរចនាមុខមាត់ Application
- ផ្ដោតលើការរចនារូបរាង (Colors, Icons, Layout, Fonts)

#### UX/UI ត្រូវការដោយសារ៖

- បង្កើត App ដែលអ្នកប្រើ **ងាយស្រួល និងរីករាយ**
- កាត់បន្ថយ Bug នៅខាងប្រើប្រាស់
- បង្កើនល្បឿនពេល User ធ្វើ task
- ធ្វើឱ្យ App professional និងគួរឱ្យទុកចិត្ត

#### 4. Class សម្រាប់ព័ត៌មាននិស្សិត និងពិន្ទុ ( Dart OOP )

##### a. Class សម្រាប់ព័ត៌មានផ្ទាល់ខ្លួន

```
class StudentInfo {  
    String name;  
    String gender;  
    String year;  
    String classroom;  
  
    StudentInfo( this.name, this.gender, this.year, this.classroom );  
}
```

##### b. Class សម្រាប់ពិន្ទុតាមមុខវិជ្ជា + ពិន្ទុសរុប

```
class StudentScore {  
    double se;  
    double mobileApp;  
    double ooad;  
    double linux;  
  
    StudentScore( this.se, this.mobileApp, this.ooad, this.linux );  
  
    double totalScore( ) {  
        return se + mobileApp + ooad + linux;  
    }  
}
```

##### c. Class សម្រាប់រួមទាំងព័ត៌មានផ្ទាល់ខ្លួន និងពិន្ទុ

```
class Student {
```

```
StudentInfo info;
```

```
StudentScore score;
```

```
Student( this.info, this.score );
```

```
void display( ) {
```

```
    print( "Name: ${info.name}" );
```

```
    print( "Gender: ${info.gender}" );
```

```
    print( "Year: ${info.year}" );
```

```
    print( "Class: ${info.classroom}" );
```

```
    print( "Total Score: ${score.totalScore( )}" );
```

```
}
```

```
}
```