**GIỚI THIỆU NGÔN NGỮ DART**

**Dart** là một ngôn ngữ lập trình mã nguồn mở (open source) đa năng (general purpose). Nó được phát triển bởi GoogleI. **Dart** là một ngôn ngữ lập trình hướng đối tượng sử dụng cú pháp của C (C-style syntax). Nó hỗ trợ các khái niệm như interface, class, ... không giống như các ngôn ngữ lập tình khác, Dart không hỗ trợ mảng (array). Dart collections có thể sử dụng các cấu trúc dữ liệu (data structure) thay thế.

Đoạn code dưới đây minh hoạ một chương trình Dart cơ bản:

void main() {

print("Dart language is easy to learn");

}

Biến và Kiểu dữ liệu

**Biến***(Variable)* là tên đại diện cho nơi lữu trự dữ liệu còn **Kiểu dữ liệu** (*Data types)* đơn giản là loại và kích thước của dữ liệu liện kết với biến và hàm.

Dart sử dụng từ khoá *var* để khai báo biến.

var name = 'Dart';

Từ khoá *final* và *const* được sử dụng để khai báo hằng số (constants). Như ví dụ dưới đây

void main() {

final a = 12;

const pi = 3.14;

print(a);

print(pi);

}

**Dart** hỗ trợ các kiểu dữ liệu dưới đây, chúng ta không cần thiết phải khai báo kiểu dữ liệu cho biến

1. **Numbers** − Được sử dụng cho số – Integer và Double.
2. **Strings** − Được sử dụng cho chuỗi kí tự. Giá trị của String được đặt trong dấu nháy đơn hoặc nháy kép.
3. **Booleans** − Được sử dụng cho giá trị Boolean đúng và sai
4. **Lists and Maps** − Được sử dụng cho nhóm đối tượng. Ví dụ một Danh sách đơn giản có thể được mô tả như sau:

void main() {

var list = [1,2,3,4,5];

print(list);

}

Map có thể được mô tả như sau:

void main() {

var mapping = {'id': 1,'name':'Dart'};

print(mapping);

}

1. Dynamic − Trường hợp kiểu dữ liệu chưa được định nghĩa thì giá trị mặc đinh là  dynamic.

void main() {

dynamic name = "Dart";

print(name);

}

Điều khiển và vòng lặp

Một khối điều khiển (decision making block) đánh giá một điều kiện trước khi hướng dẫn thực thi. **Dart** hỗ trợ các khối lênh If, If..else và switch.

Vòng lặp được sử dụng để lặp lại một khối lệnh cho đến khi một điều kiện cụ thể được đáp ứng. **Dart** hỗ trợ các vòng lặp for..in, while và do.. while

Ví dụ đoạn code hiển thị các số từ 1 đến 10

void main() {

for( var i = 1 ; i &lt;= 10; i++ ) {

if(i%2==0) {

print(i);

}

}

}

Hàm (Functions)

Hàm là một nhóm các câu lệnh nhằm thực hiện một tác vụ nhất định. Chúng ta cùng xem một ví dụ về hàm trong **Dart** dưới đây:

void main() {

add(3,4);

}

void add(int a,int b) {

int c;

c = a+b;

print(c);

}

Hàm trên làm nhiệm vụ cộng hai tham số truyền vào, và in ra kết quả trên màn hình

Lập trình hướng đối tượng

**Dart** là một ngôn ngữ lập trình hướng đối tượng (object-oriented language). Nó hỗ trợ một số tính năng của lập trình hướng đối tượng như class, interface,...

Mỗi một class (lớp) định nghĩa cho một loại đối tượng. Một class bao gồm những nội dung sau đây:

* Các thuộc tính (Fields)
* Các hàm Getter và Setter
* Hàm khởi tạo (Constructor)
* Phương thức (Function)

Dưới đây là một minh hoạ các thành phần của một class

class Employee {

String name;

//getter method

String get emp\_name {

return name;

}

//setter method

void set emp\_name(String name) {

this.name = name;

}

//function definition

void result() {

print(name);

}

}

void main() {

//object creation

Employee emp = new Employee();

emp.name = "employee1";

emp.result(); //function call

}