Dart

[Flutter]

・https://dart.dev/

・言語仕様

基本

Install

■ Windows

・http://www.gekorm.com/dart-windows/

■ Mac

・https://dart.dev/get-dart

```
$ brew tap dart-lang/dart
$ brew install dart
```

DartPad

・https://dartpad.dartlang.org/

API リファレンス

・https://api.dart.dev/stable/2.5.0/index.html

パッケージ

・Dart 標準は、dart:

```
import 'dart:html';
```

・ローカルは相対パス (dart: 省略)
・pub パッケージマネージャ

Hello

・hello.dart

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

・実行

```
>dart hello.dart
Hello Dart!
```

型

数値

■ num のサブクラス

・int
・double

真偽

・bool

文字列

■ String

・https://api.dart.dev/stable/2.5.0/dart-core/String-class.html

補完

・${}

```
print("Hello ${this.name}.");
```

■リテラル

・一重、二重引用符
・引用符 3 つ重ねで複数行

■ Runes

・UTF-32 型の文字
・絵文字などの表現

コレクション

■ List

・[]
・add
・insert
・removeAt
・indexOf
・sort
・shuffle
・forEach
・every
・map
・reduce

■ Map

・{}
・remove
・addAll
・forEach
・containsKey

■ Symbol

・Dart の構文での演算子や識別子

■ Functions

・関数

■ Enum

・列挙

文法

基本

・セミコロン必須

■エントリーポイント

・void main() もしくは、void main(List<String> args)

■コメント

・//
・/*...*/

変数

■ var

・型推論
・初期値 null
・型宣言

定数

■ final

・再代入不可

■ const

・コンパイル時点で評価

;; オブジェクトの変更禁止

```
final List<int> list1 = [1, 2, 3];
list1.add(4); // [1, 2, 3, 4]


List<int> list2 = const [1, 2, 3];
list2.add(4); // エラーが発生
```

関数

・第１級オブジェクト
・宣言

" 通常

```
int someFunc2(int a, int b) {
  return a + b;
}
```

■１行

```
bool isHoge(String str) => str == "hoge";
```

■引数の指定

```
someFunc3(a: 1, b: 2, c: 3);
```

■オプション引数

```
String conc(String a, String b, [String c])
```

■デフォルト引数

```
int someFunc4(int a, {int b = 2})
```

■無名関数 (ラムダ、クロージャ)

```
list.forEach((item) => print(item));
```

■ブロック

・変数のスコープ分離

■クロージャ

・変数のスコープから外れても利用できる

■カスケード表記

・.. でインスタンスの記述を省略できる
・メソッドを連続で呼び出すことができる

演算子

■型演算子

・as
・is
・is!

■条件付きメンバアクセス

・?.

構文

■ try catch

・例外チェックは行わない

```
try {
} on Exception catch (e) {
} catch (e, s) {
    // s はスタックトレース
} finally{}
```

ジェネリクス

```
void hoge(List<T> list){}
```

オブジェクト指向

・https://dart.dev/guides/language/language-tour#classes
・Dart はクラスとミックスインベースのインターフェースによるオブジェクト指向言語。
・すべてのオブジェクトはクラスのインスタンスで Object の子孫

■キーワード

・継承 extends
・継承元呼び出し super
・オーバーライド @override アノテーション

クラス

非同期処理

・async パッケージをインポート

```
import 'dart:async';
```

言語サンプル

・https://dart.dev/samples

File

・https://api.dart.dev/stable/2.5.0/dart-io/File-class.html

Directory

■ファイル一覧

```
var rootDir = new Directory("c:\\work");
rootDir.list(recursive: false, followLinks: false)
  .listen((FileSystemEntity entity){
    print(entity.path);
});
```