声明变量

var a = 1; js的声明方法

int b = 1; ts的声明方法

String s = “1”;

dynamic c = 1; == var方法

dart会自动推算var与dynamic声明的数据类型，声明变量不赋值默认为null;

声明常量

final a = 1; final可以通过变量赋值

const v = 1; 声明的时候就要定值

数据类型

numbers 包括浮点与整形

strings

booleans

lists(或者是arrays)

maps 字典

runes（UTF-32字符集的字符）

symbols 运算符或标识符

函数

int add(int a) {return a}声明返回值的类型 可以不声明

add() {}

add() => a =>代表return

add(String a, a: String){} 参数声明

add(String a, [String b]){} []中的参数可以不传，但是要放在最后

add(int a = 1){}为参数设置默认值

可以在双引号中打印变量值“hello $word” 跟php很像

应用入口函数

main(List<String>) {

return void

}

运算符

一般的都包括

a is b 判断运算符

a is！b

a ~/ b 下取整

a as b 强制类型转换

？. 能够打点才打点，不能反悔null

.. 类似于jq的链式调用，对一个对象使用new a() ..b()..c();

逻辑循环与js一致

多一个 try{} on IntegerDivisionByZeorException {} finally{} 虽然也有try{}catch(){}

类 类与js一样

独特之处

a(int b, int c) {

this.b = b;

this.c = c;

} == class a{

int a;

int b;

a(this.a, this.b);

}

命名构造

class Human {

String name;

Human.fromJson(Map data) {

print("Human's fromJson constructor");

}

}

继承extends

class Man extends Human {

Man.fromJson(Map data) : super.fromJson(data) {继承命名方法要用 ：super.

print("Man's fromJson constructor");

}

}

类的成员方法可以定义 get,set方法

class a {

b, c;

a(this.b, this.c);

get d => b + c;

set d(value) => left = value - b;

}

抽象类和抽象方法

使用abstract修饰一个类，则这个类是抽象类，抽象类中可以有抽象方法和非抽象方法，抽象方法没有方法体，需要子类去实现，如下代码：

abstract class Doer {

// 抽象方法，没有方法体，需要子类去实现

void doSomething();

// 普通的方法

void greet() {

print("hello world!");

}

}

class EffectiveDoer extends Doer {

// 实现了父类的抽象方法

void doSomething() {

print("I'm doing something...");

}

}

运算符重载

Dart中有类似于C++中的运算符重载语法，比如下面的代码定义了一个向量类，重载了向量的+ -运算：

class Vector {

num x, y;

Vector(this.x, this.y);

Vector operator +(Vector v) => new Vector(x + v.x, y + v.y);

Vector operator -(Vector v) => new Vector(x - v.x, y - v.y);

printVec() {

print("x: $x, y: $y");

}

}

main() {

Vector v1 = new Vector(1, 2);

Vector v2 = new Vector(3, 4);

(v1 - v2).printVec(); // -2, -2

(v1 + v2).printVec(); // 4, 6

}

类混合

// 使用with关键字，表示类C是由类A和类B混合而构成

class C = A with B;

类的静态方法

用static声明 static a = 1; static b() {};

库，文件导入

import xxx;/import abbc as ac; 跟py的库导入很像

import a show b; import b hide c;show与hide只导入b，导入b但不导入b中的c

import a deferred as a; deferred as 懒加载模式，调用才加载此库；

异步 async与await去学ES7拉；