# 第一部分Flutter基础篇

## 一 新建Flutter项目

略，可以参考我的博客

## 二 Dart语言基础

### 1 Dart数据类型

#### 1 Dart语言关键字

关键字如下



#### 2 变量（Variable）

变量赋值的例子

var name = 'Bob';

没有初始化的变量都会被赋予默认值 null.

即使是数字也是如此， 因为在Dart 中数字也是一个对象

可选类型

也可以在定义的时候指定变量的类型。

String name = 'Bob';

指定数据类型可以更好的辨明自己的使用意图，编译器和IDE 工具可以根据这些类型信息来做检查，更早的发现问题。

如前文所说，通过指定类型，也可以减少编译和运行时间。

#### 3 常量和固定值

如果定义的变量不会变化，可以使用final 或 const来指明。

也可以使用final 或 const来代替类型声明。

final name = 'Bob'; // Or: final String name = 'Bob';

// name = 'Alice'; // Uncommenting this causes an error

final的值只能被设定一次。

const 是一个编译时的常量。( Const variables are implicitly final.)

const name = "hello";

通过对const类型做四则运算将自动得到一个const类型的值。

const bar = 1000000; // Unit of pressure (dynes/cm2)

const atm = 1.01325 \* bar; // Standard atmosphere

可以通过const来创建常量的值

就是说const[] 本身是构造函数。

// Note: [] creates an empty list.

// const [] creates an empty, immutable list (EIA).

var foo = const []; // foo is currently an EIA.

final bar = const []; // bar will always be an EIA.

const baz = const []; // baz is a compile-time constant EIA.

// You can change the value of a non-final, non-const variable,

// even if it used to have a const value.

foo = [];

// You can't change the value of a final or const variable.

// bar = []; // Unhandled exception.

// baz = []; // Unhandled exception

### 2 面向对象

# 第二部分 Flutter基础控件

1 布局