第一章、Dart语法

一 、重要概念

1. 一切变量都是对象，包括数字，函数和null
2. 对象是累的实例，所有类都继承Object
3. 强类型支持类型推断
4. 不想指定具体类型，使用dynamic
5. 支持泛型，比如List<Int>和list<dynamic>
6. 支持顶层函数，类方法，实例方法，嵌套函数
7. 支持顶层变量，类变量，实例变量，局部变量
8. 以下划线(\_)开头变量是私有的，只能在库内部使用
9. 标识符只能以字母或下划线打头，后面跟任意多个字母，数字，下划线字符
10. 表达式在运行时可以求值，而语句不行
11. 代码问题分为warning和error两种，waring不会阻止程序运行，但error会

二 、关键字



三 、变量

1. 变量类型可以自动推断或者显示指定
2. 未初始化的变量都是null，包括数字
3. 如果某个变量的值在定义之后不会改变，就申明final或者const
4. final或者const变量只能赋值一次，但是const变量的值需要在编译的时候就能确定
5. final和const必须在定义的时候就完成初始化
6. const变量的值是编译时常量，包括数字，字符或者字符串常量，其他常量或算术运算结果
7. cosnt除了申明变量，也可用来申明值，也就是常量
8. 初始化cosnt变量时可以省去值前面的const，因为const变量的值肯定是一个常值
9. 常值用来防止内容变化，常值用来避免变量寻址，final变量用来提升代码安全性

注意：

1. 实例变量可以为 final 但是不能是 const 。
2. 被final，const 修饰的值是常值，不可被修改
3. 在生产模式 assert() 语句被忽略了。在检查模式 assert(*condition*) 会执行，如果条件不为 true 则会抛出一个异常

四 、内置类型

1. 数字（num）

1） num是int和double的基类

2） int是64位整数，double是64位整数

2. 字符串（String）

1） 字符串是utf-16码元（code unit）序列，每个unicode字符映射为1个或2个码元

2） 字符串字面量可以使用单引号或双引号

3） 使用${expression}来插入变量值，如果expression是个合法的标识符，则可以省去{}

4） 使用毗连字符串常量或+操作符来拼接字符

5） 使用三重引号来定义多行字符

6） 使用r前缀来声明原始字符串，其内不做字符转义

7） 只要内部插入的表达式为编译时常量，则字符串字面量为编译时常量

3. 布尔（bool）

1） 只有2个值，true和false，都是编译时常量

2） 在需要布尔类型的地方必须使用布尔值

4. 列表（list）

1） 列表就是其他语言里面的数组

2） 列表索引序号从0开始，直到list.length-1

3） 在列表字面量前使用const来声明常值

4） 使用[]操作符来操作列表元素

5. 哈希表（Map）

1） 哈希表用来关联键和值，键和值可以是任意类型的对象，但键必须唯一

2） 使用[]操作符来操作哈希表的值

3） 在哈希表字面量前使用const来声明常值

6. 符文（Rune）

1) Rune是由utf-32码点（code point）组成的字符串，一个Rune字符对应一个Unicode字符

2） 码点一般使用4个十六进制字符\uXXXX来表示，如果超过4个，则需要使用{}将十六进制字符括起来

3） 字符串和Rune之间可以互相转换

注意：

1. String->int，int.parse(‘xxx’)

String->double，double.parse(‘xxx’)

Int->string，xxx.toString

Double->string，xxx.toStringAsFixed(2),保留2位小数

1. << 是前面数字二进制左移 \*2 ，>>是前面数字右移 /2
2. 编译时常量，在编译时可以互相赋值
3. Xxx.runtimeType 返回变量的类型

五、 函数

1. 函数也是一种对象，类型为Function

2. 函数参数和返回值类型可以省略，以支持动态类型

3. 如果函数体只包含一个表达式，可使用箭头语法来定义

4. 可选参数：

可选命名参数使用{}来指定，并且可使用注解@required标注为必须

可选位置参数使用[]来指定

可选参数默认值使用=来指定，如未指定则默认值为null

5. 每个程序都应该有个位于顶层的main函数，它是程序的入口

6. 函数可作为函数参数值，也可赋值给变量

7. 可定义匿名函数，一般作为函数参数值或赋值给变量

8. 变量作用域静态确定，也就是同代码布局，每对大括号定义一个作用域，嵌套大括号定义嵌套作用域

9. 闭包是一个能访问其外层作用域变量的函数，即便该函数在其他地方被调用

10. 如果函数没有指定函数值，则默认返回null，如果确实不想返回任何值，则可指定返回类型为Void

六、 操作符

1. 操作符有优先级，从高到低为一元后缀，一元前缀，乘除，加减，移位，位运算，关系运算和类型测试，相等性运算，逻辑运算，null判断，三元表达式，级联调用和赋值

2. 在优先级判断比较模糊的地方，使用（）来提升可读性

3. 算术运算

+、-、\*、/、-expr、~/(整除)、%(求余)

++var 、var++、--var、var—

相等性和关系运算

==、！=、>、<、>=、<=

==判断值是否相等，如果要判断是否为同一个对象，使用identical()

4. 类型测试

As、is、is!

5. 赋值

=、？？=（null赋值）

-=、/=、%=、>>=、^=、+=、\*=、~/=、<<=、&=、|=

6. 逻辑运算

!expr、||、&&

7. 位运算

&、|、^、~expr(按位取反)、<<、>>

8. 条件表达式

三元表达式condition?expr1:expr2

Null判断expr1??expr2

9. 级联调用..，严格来说只是个语法糖

10. 函数调用（）

11. 小标访问[]

12. 成员访问.

13. 条件式成员访问?.

七、 控制流程语句

1. if else 条件必须为布尔值

2. for loop

1>循环内的闭包会捕获循环变量的当前值

2>迭代对象可使用for in 来迭代其内部元素

1. While , do while

While 先判断条件，满足后再执行循环体，do while 先执行再判断

1. Break ,continue

break终止循环，continue 跳过本次循环剩余代码，直接进入下一次循环

1. Switch case

1>使用== 操作符来比较整数，字符串或编译时常量，包括枚举类型

2>非空case语句正常情况下应该以break结束，也可使用continue,throw和return来结束

3>使用default语句来匹配其他情况

4>空case语句会落入下方case语句

5>可结合使用continue和lebel来跳转到其他case语句

6>一般用于解释器和扫描器，编写应用时，尽量少看

1. Assert

1>只再开发模式下有效，生产模式下呗忽略

2>条件判断识别时会抛出assertionError异常，可以通过第二个参数指定错误消息

八 异常

1. 异常表示有一些预料之外错误发生
2. 如果异常未被捕捉，则程序徐将终止运行
3. Dart内置了exception和error两种类型，Exception用于核心库异常，Error用于应用代码异常
4. 可使用throw抛出任何非null对象作为异常
5. Throw语句是一个表达式
6. 使用thy catch 语句来捕捉异常，以防止异常扩散，同时可使用on语句来处理特定类型的异常
7. 使用rethrow来再次抛出异常
8. 使用finally语句来执行，无论是否出现异常都要运行的代码
9. Dart语言进阶
10. 类
11. 使用类成员
    1. 对象的成员包括函数和数据，分别称为方法和实列变量
    2. 方法在某个对象上被调用，可以访问该对象的实列变量
12. 使用构造函数
    1. 构造函数可以是className或className.identifier,关键字new可选
    2. 如果类提供了常量构造函数，可使用const来创建编译时常量，创建2个完全一样的编译时常量将产生同一个实例
    3. 在一个常量上下文里，可以省去构造函数前面的const
    4. 在一个非常量上下文里，如果省去构造函数前面的const，则会创建一个非常量对象
13. 获取对象类型
    1. 运行时使用对象的runtimeType属性来获得对象类型，一个Type类型的对象
14. 实列变量
    1. 未初始化的实列变量值为null
    2. 每个实列变量都会自动生成一个隐式的getter方法，非final实列变量还会生成setter方法
    3. 在声明实列变量的同时进行的初始化，将先于构造函数和初始化器运行
15. 构造函数
    1. 引用实列变量一般不需要使用this,但如果出现变量名冲突，则可以使用this区分
    2. 在构造函数参数里使用this来简化赋值
    3. 如果不声明构造函数，则构造函数会使用默认构造函数，它会调用父类的无参数非命名构造函数
    4. 构造函数不会继承于父类
    5. 如果需要多个构造函数，可以使用命名构造函数
    6. 默认会调用父类的无参数非命名构造函数，如果没有则需显示调用某个构造函数
16. 泛型
17. 库
18. 异步
19. 生成器
20. 可调用类
21. 类型定义
22. 元数据
23. 注释
24. Flutter框架基础
25. App开发实战