

Dart

数据类型 - 内置类型

- 数值型 - number
- - 整型 int
- - 浮点型 double
- 字符串型
- 布尔型
- 列表型
- 键值对 - Map
- Runes、Symbol
-

面向对象

类与对象，声明、创建及基本特性、计算属性

- 声明一个类：class
- 创建一个类：new、new可以省略
- 所有对象都继承Object类
- 属性默认会生成getter、setter方法
- 使用final声明的属性只有getter方法【只读不可写】
- 属性和方法通过 . 访问
- 方法不能被重载
- Dart中的可见性，以及library（库）位单位
- 默认情况下，每一个Dart文件就是一个库

构造方法及初始化列表

- 初始化列表会在构造方法体执行之前执行
- 使用逗号分隔初始化表达式
- 初始化列表常用于设置final变量的值

静态成员及对象操作符的使用

- 使用static关键字来实现类级别的变量和函数
- 静态成员不能访问非静态成员，非静态成员可以访问静态成员

- 类中的常量需要使用static、const来声明
- 对象操作符 - 条件成员访问: ?.
- 对象操作符 - 类型转换: as
- 对象操作符 - 是否指定类型: is, is!
- 对象操作符 - 级联操作: ..

面向对象扩展

三大特性: 封装、继承、多态、抽象

- 继承: extends关键字; 子类会继承父类可见的属性和方法, 不会继承构造方法; 子类能够覆写父类的方法, getter、setter; 单继承、多态性
- 继承中的构造方法: 子类的构造方法默认会调用父类的无名构造方法; 如果父类没有无名无参构造方法, 则需要显示调用父类构造方法; 在构造方法参数都使用: 显示调用父类构造方法。
- 构造方法执行顺序: 父类的构造方法在子类构造方法执行的位置调用; 如果有初始化列表, 初始化列表会在父类构造方法之前执行。

抽象类

- 抽象类使用 abstract, 不能被实例化。
- 抽象方法不用 abstract 修饰, 无实现。
- 抽象类可以没有抽象方法。
- 有抽象方法的类一定要声明成抽象类

接口

- 类和接口是统一的, 类就是接口
- 每个类都隐式的定义了一个包含所有实例成员的接口
- 如果是复用已有类的实现, 使用继承 (extends)
- 如果只是使用已有类的外在行为, 使用接口 (implements)

Mixins, 操作符的覆写

- Mixins类似于多继承, 是在多类继承中重用一個类代码的方式
- 作为Mixin的类不能有显式声明构造方法
- 作为Mixind的类只能继承来自Object
- 使用关键字with连接一个或多个mixin
- 覆写操作符需要在类中定义

```
返回类型 operator 操作符 (params1, params2, ...) {
    实现体...
    return 返回值
}
```

- 如果覆写 == , 还需要覆写对象的hashCode getter方法

one	two	three	four
<	+		[]
>	/	^	[]=
<=	~/	&	~
>=	*	<<	==
-	%	>>	

枚举

- 枚举是一种又穷序列集的数据类型
- 使用关键字 enum 定义一个枚举
- 常用于代替常量, 控制语句等
- index从0开始, 依次累加
- 不能指定原始值
- 不能添加方法

```
enum status {
    open,
    forbiden,
    close
}

void main() {
    print(status.close.index)
}
```

泛型

- Dart中类型是可选的, 可使用泛型限定类型
- 使用泛型能够有效的减少代码重复
- 类的泛型
- 方法的泛型

