

# Dart extension之扩展方法



向日葵开 关注

1 2020.08.16 09:43:33 字数 391 阅读 7,385

今早无意间看到了 Dart 居然也支持 Api 的扩展，感觉就好像发现了新大陆，之前写的好多转换方法都可以换一个更简洁，更优雅的写法。

这么好的点当然要记录一下，同时也分享给各位亲。

Dart 扩展需要关键词:extension。这个关键字只有在 Dart 版本 2.7 及其以上才支持。所以请确保你工程项目中 pubspec.yaml 文件中:

```
1 | environment:
2 |   sdk: ">=2.7.0 <3.0.0"
```

如果你之前的是老项目，修改了 pubspec.yaml，执行 pub get 之后，重启 IDE 之后设置才会生效。看一下 extension 的用法。超级简单。

形如：

```
1 | extension <extension name> on <type> {
2 |   (<member definition>)*
3 | }
4 |
```

示例：

```
1 | extension StringExtension1 on String {
2 |   //字符转换成Color对象
3 |   toColor() {
4 |     var hexColor = this.replaceAll("#", "");
5 |     if (hexColor.length == 6) {
6 |       hexColor = "FF" + hexColor;
7 |     }
8 |     if (hexColor.length == 8) {
9 |       return Color(int.parse("0x$hexColor"));
10 |    }
11 |  }
12 |
13 |   //字符转int
14 |   parseInt() {
15 |     return int.parse(this);
16 |   }
17 | }
18 |
19 | extension DateTimeExtension on DateTime {
20 |   toFormatString() {
21 |     DateFormat dateFormat = new DateFormat("yyyy-MM-dd HH:mm:ss");
22 |     return dateFormat.format(this);
23 |   }
24 | }
25 |
```

扩展不仅可以定义方法，还可以定义,setter,getter,operator。如果要使用这些扩展方法，只要引用导入对应的 dart 文件即可。

```
1 | import 'util/extensions.dart';//导入
2 |
3 | void main() {
4 |   print("#90F7EC".toColor());
5 |   print("23".parseInt());
6 | }
7 |
```



这种写法是不是超级简单。。。。

这种扩展对于申明为动态类型的变量不可使用。也就是：

```
1 | dynamic h = "78";
2 | print(h.parseInt()); //Runtime exception: NoSuchMethodError
```

运行报错。

如果你定义了很多扩展，扩展方法之间有同名冲突，可以用 hide,show, as 限制，选择你想用扩展方法。

```
1 | import 'util/extensions.dart';
2 | import 'util/extensions1.dart' hide StringExtension1;
3 | import 'util/extensions.dart' as ex1;
4 |
5 | print("#90F7EC".toColor());
6 | print("23".parseInt());
7 | print(ex1.DateTimeExtension(DateTime.now()).toFormatString());
8 |
```

扩展也可以作用于泛型：

```
1 | //扩展list的方法
2 | extension HclList<T> on List<T> {
3 |   //扩展属性
4 |   int get doubleLength => length * 2;
5 |   //扩展操作符
6 |   List<T> operator -( ) => reversed.toList();
7 |   //一个链表分割成两个
8 |   List<List<T>> split(int at) => <List<T>>[sublist(0, at), sublist(at)];
9 | }
```

## 扩展在 Widget 控件中的应用

我们会经常有类似控件：

```
1 | Column(
2 |   mainAxisAlignment: MainAxisAlignment.center,
3 |   children: <Widget>[
4 |     Container(
5 |       margin: EdgeInsets.all(10),
6 |       alignment: Alignment.center,
7 |       child: Text("ZL"),
8 |     ),
9 |     Container(
10 |      margin: EdgeInsets.all(10),
11 |      alignment: Alignment.center,
12 |      child: Text("ZL"),
13 |    ),
14 |     Container(
15 |      margin: EdgeInsets.all(10),
16 |      alignment: Alignment.center,
17 |      child: Text("ZL"),
18 |    ),
19 |   ],
20 | )
```

好多重复的 Container()是不是？我们可以扩展 一下：

```
1 | extension HclWidget on Widget {
2 |   Widget marginAll(double margin) {
3 |     return Container(
4 |       alignment: Alignment.center,
5 |       margin: EdgeInsets.all(margin),
6 |       child: this,
7 |     );
8 |   }
9 | }
```



```
9 |   }
   | }
```

之后我们就可以改成这样：

```
1 |         Column(
2 |             mainAxisAlignment: MainAxisAlignment.center,
3 |             children: <Widget>[
4 |                 Text("ZL").marginAll(10),
5 |                 Text("ZL").marginAll(10),
6 |                 Text("ZL").marginAll(10),
7 |             ],
8 |         )
9 |
```

扩展类的所有代码：

```
1 | import 'dart:ui';
2 | import 'package:flutter/material.dart';
3 | import 'package:intl/intl.dart';
4 |
5 | extension StringExtension on String {
6 |     //字符转换成Color对象
7 |     toColor() {
8 |         var hexColor = this.replaceAll("#", "");
9 |         if (hexColor.length == 6) {
10 |             hexColor = "FF" + hexColor;
11 |         }
12 |         if (hexColor.length == 8) {
13 |             return Color(int.parse("0x$hexColor"));
14 |         }
15 |     }
16 |
17 |     //字符转int
18 |     parseInt() {
19 |         return int.parse(this);
20 |     }
21 | }
22 |
23 | extension DateTimeExtension on DateTime {
24 |     toFormatString() {
25 |         DateFormat dateFormat = new DateFormat("yyyy-MM-dd HH:mm:ss");
26 |         return dateFormat.format(this);
27 |     }
28 | }
29 |
30 | //扩展list的方法
31 | extension HclList<T> on List<T> {
32 |     //扩展属性
33 |     int get doubleLength => length * 2;
34 |
35 |     //扩展操作符
36 |     List<T> operator -( ) => reversed.toList();
37 |
38 |     //一个链表分割成两个
39 |     List<List<T>> split(int at) => <List<T>>[sublist(0, at), sublist(at)];
40 | }
41 |
42 | extension HclWidget on Widget {
43 |     Widget marginAll(double margin) {
44 |         return Container(
45 |             alignment: Alignment.center,
46 |             margin: EdgeInsets.all(margin),
47 |             child: this,
48 |         );
49 |     }
50 | }
51 |
```



13人点赞 >



Flutter学习



更多精彩内容，就在简书APP




"小礼物走一走，来简书关注我"

赞赏支持

还没有人赞赏，支持一下



**向日葵开**  我愿追逐你的阳光。。。  
总资产109 共写了3.2W字 获得216个赞 共90个粉丝

关注

