Dart extension之扩展方法



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

```
dynamic h = "78";
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。

扩展也可以作用于泛型:

扩展在 Widget 控件中的应用

我们会经常有类似控件:

```
1
                mainAxisAlignment: MainAxisAlignment.center,
                children: <Widget>[
 4
                  Container(
                    margin: EdgeInsets.all(10),
 5
                    alignment: Alignment.center,
                    child: Text("ZL"),
 7
 8
                  Container(
10
                    margin: EdgeInsets.all(10),
                    alignment: Alignment.center,
11
12
                    child: Text("ZL"),
13
14
                   Container(
                    margin: EdgeInsets.all(10),
15
                    alignment: Alignment.center,
16
                    child: Text("ZL"),
17
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 | }
```

之后我们就可以改成这样:

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

扩展类的所有代码:

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



13人点赞>





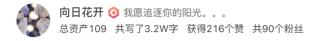




"小礼物走一走,来简书关注我"

赞赏支持

还没有人赞赏, 支持一下



关注