

我的Flutter注意事项

1. Flutter 的跨平台兼容性目前最好，前期开发调试完全在 Android 端进行的情况下，第一次在 IOS 平台运行居然没有任何错误，并且还没出现UI兼容问题 这些特点其实这得益于Flutter Engine 和 Skia。

2. Dart 下只有 bool 型可以用于 if 等判断，不同于 JS 这种使用方式是不合法的

```
var g = "null"; if(g){}
```

3. 设置了decoration的color，就不能设置Container的color

4. MainAxisSize的取值有两种： max： 根据传入的布局约束条件，最大化主轴方向的可用空间； min： 与max相反，是最小化主轴方向的可用空间；

5. 使用Card，用于快速简单的实现圆角和阴影

6. 设置缓存图片个数与大小

```
缓存个数100: PaintingBinding.instance.imageCache.maximumSize = 100;  
缓存大小50M: PaintingBinding.instance.imageCache.maximumSizeBytes = 50 <<  
20;
```

7. flutter中的颜文字：\u{1f47f} //😏

Dart的基础语法

1. num、int和double都是Dart中的类，也就是说它是对象级别的，所以他们的默认值为null

2. Map为若干个键值对的容器，想用映射之名。其中一个Map对象中的键不能重复，值是可以重复的:

```
Map<int, String> map = {1: 'one', 2: 'two', 3: 'three'};
```

3. 对数组的所有元素做同一个操作，可用map方法:

```
input.map((e){return e-5;})
```

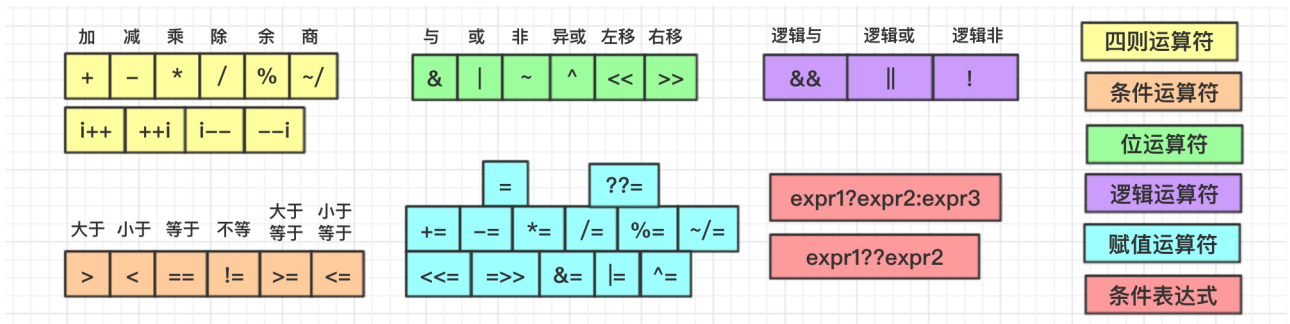
4. Runes是一个类，是一个可比遍历的int型元素组合体

5. var 关键字:如果只是用var声明变量,该变量之后是可以修改数据类型的(未赋值的时候，类型是dynamic:动态的) 如果声明的同时取赋值，那么该对象的类型就是固定的，不可修改

6. 如果一个变量你以后不打算修改, 可以使用 `final` 或者 `const` 进行修饰, 当你试图修改它的值, 就会报错。
`const`与`final`的区别: 一个`final`变量只能赋值一次: 它的值在可以在运行时获取 一个`const`变量是编译时常量: 代码还没有运行时我们就知道它声明变量的值

```
final f = DateTime.now(); // OK
const c = DateTime.now(); // ERROR Const variables must be initialized with
a constant value.
```

7. 运算符



对于我来说, 用的比较少需要注意的是(位运算符暂不考虑):

商“~/”: 取商 `print(10 ~/ 3); // 3` 商

“??=”: 当变量的值为`null`时, 执行赋值语句, 否则不赋值

```
---->[情况1: b值为null]----
var a = 20;
var b;
b ??= a;
print(b); // 20
```

```
---->[情况2: b值不为null]----
var a = 20;
var b = 2;
b ??= a;
print(b); // 2
```

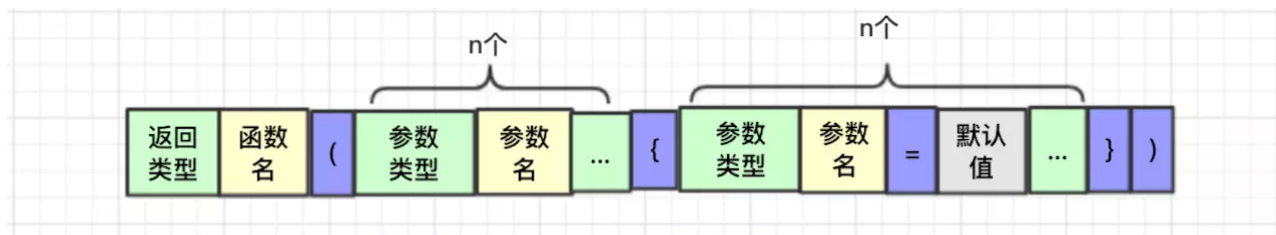
“??”: 前表达式值为`null`则取后者。否则, 执行前表达式, 不执行后表达式

```
---->[情况1: b值为null]----
var a = 20;
var b;
var c = b ?? a++;
print('a=$a,c=$c'); // a=21,c=20
```

```
---->[情况2: b值不为null]----
var a = 20;
var b = 2;
```

```
var c = b ?? a++;
print('a=$a,c=$c'); //a=20,c=2
```

8. Dart中的函数



可选参数加默认值“[]”

```
double add(double a,double b,discount=1.0)
{
    return (a+b)*discount;
}
调用： add(10,20,0.7);//21
```

属性型参数“{}”：将参数和key对应起来，方便清晰的传参模式

```
double add(double a,double b,{double discount=1.0,double c=0,double d =0})
{
    return (a+b+c+d)*discount;
}
调用： add(10, 20,discount: 0.7,c: 100);//91.0
```

函数参数：函数本身也是一个对象，可以作为参数传入

```
double add(double a,double b,Function deal)
{
    return deal(a)+deal(b);
}
调用：
var fun = (double i){
    return i*i;
};
函数简写为： var fun = (i)=>i*i;
print(add(1, 2, fun));
```

9. assert(条件)如果条件不满足，会中断程序的执行，否则继续执行流程。

```
assert(1>2); //程序中断;
```

10. r"ss" 字符串前面加一个r, 即使有空格或者\也会原样打出字符串, "ss"*2,字符串可以乘

11. List支持多类型

```
var list = [1,2,3,"a",false];
```

List的join方法: 在每一个元素之间插入参数, 然后返回字符串

```
var list = [1,2,3,4];  
print(list.join("&")); //1&2&3&4
```

List的sort方法: 把列表中的元素(需同一类型)按从大到小重新排列

```
var list = [3,2,1,4];  
print(list.sort()); //list = [1,2,3,4];
```

List的sublist(start,end)方法: 会返回列表中索引从start到end的子列表,为[start,end),且不会影响原List

```
var list = [3,2,1,4];  
print(list.sublist(1,3)); //newList = [2,1];
```

List的any方法,只要List中有满足any里的表达式的元素, 就返回true, 否则返回false

```
var any = numList.any((num) => num > 3);  
print(any); //只要有>3的任何元素,返回true
```

List的every方法,只有List中的所有元素满足表达式, 才返回true, 否则返回false

```
var every = numList.every((num) => num > 3);  
print(every); //所有元素大于3,返回true
```