Flutter - Dart - List的基本使用


小醉快跑 LV.3
2020年06月13日 18:43 · 阅读 6505

关注

前言： VSCode版本: 1.46.0 ,Dart版本v3.11.0,Flutter版本v3.11.0

List 定义

1.使用类型推导定义

ini 复制代码

```ini
var list1 = ['a','b','c','d']; //初始就赋值，限定了长度，限定了类型，只能是String
print('$list1, ${list1.runtimeType}'); //[a, b, c, d], List<String>

var list3 = ['x','y','z'];
var list4 = List(); //不限定长度，不限定类型，可添加任意类型的数据
List list5 = List(); //不限定长度，不限定类型，可添加任意类型的数据
List list6 = List(2); //限定长度为2  越界会报错，不限定类型，可添加任意类型的数据
List list7 = ['a',2,'b',false]; //初始化就赋值了，限定了长度，限定了类型，任意位置可用任意类型替换
```

2.明确指定类型

perl 复制代码

```perl
List<int> list2 = [1,2,3,4]; //初始就赋值，限定了长度，限定了类型，只能是int
print("$list2,${list2.runtimeType}"); //[1, 2, 3, 4],List<int>
```

List 常见属性

scss 复制代码

```scss
print(list1.length); //4  获取list1长度
print(list1.isEmpty); //false   字符串是否为空
print(list1.isNotEmpty);//true  字符串是否不为空
print(list1.reversed); //(d, c, b, a)   返回一个倒序排列的List，不影响原List;
print(list1.first);  //a  第一个元素
```

```
print(list1.last);  //d 最后一个元素
```

List 常用的一些方法

- add()添加元素
- addAll()添加List

scss 复制代码

```
list1.add('f');
list2.add(5);
print('List增加 $list1,$list2'); //List增加 [a, b, c, d, f],[1, 2, 3, 4, 5]
 //在这里会报错 list2 是一个int类型的，他无法加入到一个string的list里面
//list1.addAll(list2); //Error: The argument type 'List<int>' can't be assigned to the para

 list4.add(1);
 list4.add('aaa');
 list4.add(true);
 print(list4); //[1, aaa, true]


list1.addAll(list3);//点击addAll这个方法进入官方文档可以看到注解：1.将[iterable]的所有对象追加到该列
print('拼接后的List$list1');// +++++++[a, b, c, d, f, x, y, z]
```

- 指定索引位置 插入值，其余顺延

scss 复制代码

```
 list1.insert(4, 'g');
print(list1); //[a, b, c, d, g, f, x, y, z]
```

- insertAll() 指定位置插入list，其余顺延

scss 复制代码

```
list1.insertAll(5, list3);
print(list1); //[a, b, c, d, g, x, y, z, f, x, y, z]
```

- followedBy()将自身 和参数内的list 元素合并 返回的类型一定是 Iterable

ini 复制代码

```ini
List<String> list8 = ['aa','bb','cc','x','y','z'];
Iterable<String> list9 = list8.followedBy(['aa','zz']);
print('合并后的List $list9'); //合并后的List (aa, bb, cc, x, y, z, aa, zz)
```

- remove() 删除元素

dart 复制代码

```dart
list1.remove('a');  //list1 删除a 元素
print('List指定删除某个元素 $list1'); //List指定删除某个元素 [b, c, d, g, x, y, z, f, x, y, z]
```

- removeAt() 删除指定下标的元素

scss 复制代码

```scss
 //在这里可以看出 下标是从0 开始的
list1.removeAt(1); // list1 删除下标为1 的元素
print('List删除下标为1 的元素 $list1');//List删除下标为1 的元素 [b, d, g, x, y, z, f, x, y, z]
```

- removeLast() 删除最后一个元素

scss 复制代码

```scss
 list1.removeLast(); // list 删除最后一个元素
print('List删除最后一个元素 $list1');//List删除最后一个元素 [b, d, g, x, y, z, f, x, y]
```

- removeRange() //删除范围内的元素

scss 复制代码

```scss
 list1.removeRange(0, 3); //
 print('List 删除 从 0 - 3 的元素 后得到的 $list1');//List 删除 从 0 - 3 的元素 后得到的 [x, y, z,
```

- contains() 是否包含某个元素

```scss
list1.contains(1); //是否包含某个元素,
print('${list1.contains(1)}');//这里返回false,因为现有的list1 里面只有 x,y  所以返回flase
```

- removeWhere() 根据条件删除

```ini
list1.removeWhere((element) {return element == 'x';});

list1.removeWhere((element) => element == 'x'); //这里是函数简写
```

- List 中元素重新赋值

```scss
 list1[0] = '第一个元素替换';
print(list1); //[第一个元素替换, z, f, y]
```

- setRange() 范围内修改List

```go
//在range 范围内修改List 超出范围报错
list1.setRange(0, 4, ['a','b','c','d']);
print(list1);//[a, b, c, d]
```

- replaceRange() 范围内替换List

```scss
//在range 范围内替换List 超出范围报错 包含头不包含尾
List<int> list10 = List();
list10.add(0);
list10.add(1);
list10.add(2);
list10.add(3);
```

```
list10.add(4);
list10.replaceRange(1, 4, [7,8,9,10,11]);
print(list10); //[0, 7, 8, 9, 10, 11, 4]
```

- fillRange(start,end,value) 在start end 每个元素用value替换

scss 复制代码

```
List<int> list11 = List();
list11.add(0);
list11.add(1);
list11.add(2);
list11.add(3);
list11.add(4);
list11.add(5);
list11.fillRange(1, 5,9);
print(list11);//[0, 9, 9, 9, 9, 5]
```

- retainWhere(()=>(bool)) 根据条件筛选元素

scss 复制代码

```
List<int> list12 = List();
list12.add(0);
list12.add(1);
list12.add(2);
list12.add(3);
list12.add(4);
list12.retainWhere((element) => (element > 2)); //根据条件保留元素
print(list12);//[3, 4]
```

- setAll(index,list) 从index开始，使用list内的元素逐个替换本list中的元素

scss 复制代码

```
List<int> list13 = List();
list13.add(0);
list13.add(1);
list13.add(2);
list13.add(3);
list13.add(4);
list13.setAll(1, [5,6,7,8]); //从index开始替换，但是替换的List长度不能超过现有List的长度
```

```
print(list13); //[0, 5, 6, 7, 8]
```

- indexOf(element,[start]) 查找指定元素在list中的索引

scss 复制代码

```
List<int> list14 = List();
list14.add(0);
list14.add(1);
list14.add(2);
list14.add(3);
list14.add(4);
list14.add(4);
int index14 = list14.indexOf(4); //默认从索引0开始查找指定元素，返回指定元素的索引 如果存在多个相同的
int index141 = list14.indexOf(2,3); //从3 开始查找 2，如果存在返回指定元素索引，反则返回 -1
print(index14); //4
print(index141); //-1
```

- lastIndexOf(element,[start]) 从后往前查找指定元素在list中的索引

ini 复制代码

```
List<int> list15 = List();
list15.add(0);
list15.add(1);
list15.add(2);
list15.add(3);
list15.add(4);

int index15 = list15.lastIndexOf(2);
int index151 = list15.lastIndexOf(1,4); //从指定索引位置往后查找指定元素的索引位置，查到返回该元素的

print(index15); //2
print(index151); //1
```

- elementAt(index) 根据索引值获取元素

ini 复制代码

```
List<int> list16 = List();
list16.add(0);
list16.add(1);
```

```
list16.add(2);
list16.add(3);
list16.add(4);

var resultList16 = list16.elementAt(3);
print(resultList16); //3
```

- any((element) => (bool)) 判断List中是否任意一个元素符合条件赛选

```scss
List<int> list17 = List();
list17.add(1);
list17.add(2);
list17.add(3);
list17.add(4);
list17.add(5);
//every((element)=>(bool)) 判断List中是否有元素符合参数函数
bool resultList17 = list17.any((element) => (element >3));//判断List元素中是否有任意一个元素满足
print(resultList17); //true
```

- firstWhere((element) =>(bool)) 返回第一个满足条件的元素

```scss
List<int> list18 = List();
list18.add(1);
list18.add(2);
list18.add(3);
list18.add(4);
list18.add(5);
list18.add(1);
list18.add(2);

 var resultList18 = list18.firstWhere((element) => (element > 2)); //找不到则会报错 Bad state
 print(resultList18); //3
```

- int lastIndexWhere(bool test(E element), [int start]); 返回第一个满足条件的元素下标,start表示从第几个下标开始查找,不存在则返回-1

scss 复制代码

```
List<int> list19 = List();
list19.add(1);
list19.add(2);
list19.add(3);
list19.add(4);
list19.add(5);

var resultList19 = list19.indexWhere((element) => (element >3)); //不存在则返回-1
var resultList191 = list19.indexWhere((element) => (element > 4),2);

var resultList192 = list19.lastIndexWhere((element) => (element > 5));
var resultList193 = list19.lastIndexWhere((element) => (element > 2),1);

print(resultList19);//3
print(resultList191);//4
print(resultList192);//-1
print(resultList193);//-1
```

- lastWhere(bool test(E element), {E orElse()}) 从后往前查找，返回第一个满足条件的元素

ini 复制代码

```
List<int> list20 = List();
list20.add(1);
list20.add(1);
list20.add(3);
list20.add(4);
list20.add(5);
var resultList20 = list20.lastWhere((element) => element >2);

var resultList21 = list20.lastWhere((element) => element > 6,orElse:()=>(66));

print(resultList20); //5
print(resultList21); //66
```

- forEach 遍历List中的每个元素 进入文档内查看，forEach方法内部其实也是forIn方法进行遍历

scss 复制代码

```
List list21 = List();
list21.add(1);
list21.add(2);
list21.add(3);
```

```
list21.add(4);
list21.add(5);
list21.forEach((element) { //遍历时不可以add或remove，但是可以修改element内的值
  element += 1; //这里修改了element的值，但是遍历完，list21里面的值是不变的
  print(element); //2,3,4,5,6;
  list21[1] = 66;
  list21[2] = 77;
});
print(list21); //[1, 66, 77, 4, 5]

for (var list21Element in list21) {
   print(list21Element); // 1, 66, 77, 4, 5;
}
```

- fold(T initialValue, T combine(T previousValue, E element)) 根据现有的List和给定的initialValue,指定一个参数函数规则，对List每个元素做操作，并将结果返回

scss 复制代码

```
List<int> list22 = [1,2,3,4,5,6];
var resultList22 = list22.fold(2, (a, element) => (a * element)); //2 *(1*2*3*4*5*6)
var resultList221 = list22.fold(2, (a, element) => (a + element)); // 2+(1+2+3+4+5+6)
print(resultList22);//1440
print(resultList221);//23
```

- reduce(E combine(E value, E element)) 用指定的函数对元素做连续操作，将结果返回

dart 复制代码

```
List<int> list23 = [1,2,3,4,5,6];
var resultList23 = list23.reduce((a, b) => (a +b )); //1+2+3+4+5+6
var resultList231 = list23.reduce((a, b) => (a * b)); //1*2*3*4*5*6
print(resultList23);//21
print(resultList231);//720

//skip(count) 越过count个元素后，开始返回Iterable;
List<int> list24 = [1,2,3,4,5];
Iterable<int> resultList24 = list24.skip(2);
print('$resultList24,${list24.runtimeType}');//(3, 4, 5),List<dynamic>
```

- skipWhile(bool test(E value)) 根据参数函数，找到第一个不符合条件的元素，然后将其及以后的元素返回，如果都符合就返回一个空List，如果都不符合则全部返回

ini 复制代码

```ini
List list25 = [3,1,2,4,5,6];
Iterable resultList25 = list25.skipWhile((value) => (value > 1));
print(resultList25);//[]
```

- take(int count) 从0 - count 获取元素 并返回
- takeWhile(bool test(E value)) 从0 开始获取，直至第一个不符合函数的元素，将其前面的元素都返回

dart 复制代码

```dart
List list26 = [2,4,5,6,1];
var resultList26 = list26.take(2);
var resultList261 = list26.takeWhile((value) => value > 1.2);
print('$resultList26,${resultList26.runtimeType}'); //(2, 4),SubListIterable<dynamic>
print('$resultList261,${resultList261.runtimeType}'); //(2, 4, 5, 6),TakeWhileIterable<d
```

- where(bool test(E element)) => WhereIterable(this, test) 根据指定的函数筛选，符合条件的元素组成新的Iterable

ini 复制代码

```ini
List list27 = [3,1,0,5,7];
Iterable resultList27 = list27.where((element) => element >2);
print(resultList27); //(3, 5, 7)
```

- singleWhere(bool test(E element), {E orElse()}) 根据函数找出唯一的元素，如果不唯一，则报错：Too many elements

ini 复制代码

```ini
List list28 = [1,3,2,5,8];
var resultList28 = list28.singleWhere((element) => (element >7),orElse:()=>(7));
print(resultList28);//8
```

- whereType() 从无指定类型的List中，筛选出指定类型的数据

ini 复制代码

```
List list29 = [1,9,8,2,7,1.11,'abc'];
Iterable<int> resultList29 = list29.whereType();
Iterable<double> resultList291 = list29.whereType();
Iterable<String> resultList292 = list29.whereType();
print('$resultList29,$resultList291,$resultList292'); //(1, 9, 8, 2, 7),(1.11),(abc)
```

- cast() 将List的泛型提升到期祖父类

ini 复制代码

```
List<String> list30 = ['aa','bb','cc','dd'];
List<Object> list31 = list30.cast();
list31.add('ff');
print('$list31,${list31.runtimeType}'); //[aa, bb, cc, dd],CastList<String, Object>
```

- expand() 根据现有的List,指定一个规则，生成新的List

ini 复制代码

```
List list32 = [1,4,3,6];
var resultList32 = list32.expand((element) => ([element +1,element +2]));
print('$resultList32,${resultList32.runtimeType}'); //(2, 3, 5, 6, 4, 5, 7, 8)  Expandi
```

- toSet() 将List转化为Set 并去除后面重复的元素

ini 复制代码

```
List list33 = [1,3,2,7,6,1,2];
Set set33 = list33.toSet();
print('$set33,${set33.runtimeType}'); //{1, 3, 2, 7, 6},_CompactLinkedHashSet<dynamic>
```

- asMap() 将List转化为Map 索引为key，元素为value

ini 复制代码

```ini
List list34 = [1,2,3,6,7,9];
Map map34 = list34.asMap();
print('$map34,${map34.runtimeType}'); //{0: 1, 1: 2, 2: 3, 3: 6, 4: 7, 5: 9},ListMapView<
```

- shuffle() 将List内元素重新随机排列，并且该函数没有返回值，直接修改该List

ini 复制代码

```ini
List list35 = [9,1,2,5,3];
list35.shuffle();
print('$list35'); //[9, 2, 3, 1, 5]
```

--

- sort() List自身排序

css 复制代码

```css
List list36 = [1,5,6,7,8,8,9,3,8,9,8];
//  list36.sort();
//  print(list36); //[1, 3, 5, 6, 7, 8, 8, 8, 8, 9, 9]


list36.sort((a,b) =>(a>b ? 1:-1));
print(list36);//[1, 3, 5, 6, 7, 8, 8, 8, 8, 9, 9]
```

- sublist(int start, [int end]) 截取List 返回截取后的List

dart 复制代码

```dart
List list37 = [2,5,1,7,9,10];
var result37 =  list37.sublist(2);
print('$result37,${result37.runtimeType}'); //[1, 7, 9, 10],List<dynamic>
var result371 = list37.sublist(1,3);
print('$result371,${result371.runtimeType}'); //[5, 1],List<dynamic>
```

- getRange(int start, int end) 在List 中截取 start-end 之间

ini 复制代码

```ini
List list38 = [8,9,1,0,3,5];
var resultList38 = list38.getRange(1, 3);
print('$resultList38,${resultList38.runtimeType}'); //(9, 1),SubListIterable<dynamic>
```

- join()拼接字符，返回String类型

ini 复制代码

```ini
List list39 = ['a','cc','fc','xx'];
var resultList39 = list39.join('-');
print('$resultList39,${resultList39.runtimeType}'); //a-cc-fc-xx,String
```

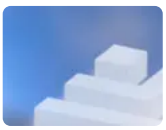
- 清除所有元素 clear();

scss 复制代码

```scss
//清除所有元素 clear()
 List list40 = ['c','b','z'];
 list40.clear(); //清除所有元素  如果是固定长度的则会报异常2
 print('$list40,${list40.runtimeType}'); //[],List<dynamic>
```

分类： 阅读 标签： Dart

文章被收录于专栏：



Flutter - Dart

Flutter - Dart 专栏

关注专栏