关于点击事件:

在两个页面点击最右边的选择学期出来的东西是不一样的。可以使用tabbar controller监听index，然后返回选择的学期数据，传给下面的界面(拼接成字符串然后筛选)

1.TabBarView重新加载问题

在 TabBarView 组件中切换页面时，子页面每次均会重新 initState 一次，导致每次都切换页面均会重绘，

如果需要只在第一次进页面 initState 一次，后面再进入页面不再 initState ，需要在子页面加上以下内容

- 首先在继承的类后面加上 with AutomaticKeepAliveClientMixin

```
with  AutomaticKeepAliveClientMixin
```

- 然后在类中加入

```
@override
bool get wantKeepAlive => true; ///seeAutomaticKeepAliveClientMixin
```

最后在build中加入

```
super.build(context);/// see AutomaticKeepAliveClientMixin
```

2.监听父widget的变化

```
@override
  void didUpdateWidget(covariant ScoreSubject oldWidget){

  }
```

只要在父widget中调用setState，子widget的didUpdateWidget就一定会被调用，不管父widget传递给子widget构造方法的参数有没有改变。

只要didUpdateWidget被调用，接来下build方法就一定会被调用。

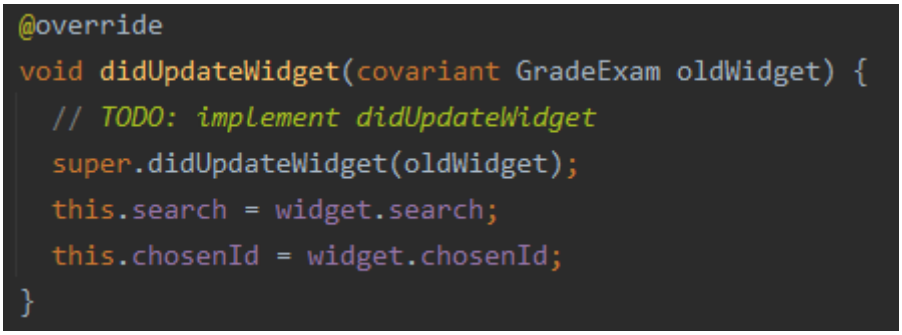
- 子widget首次被加载时的生命周期：

```
initState -> build
1
```

- 子widget首次被加载后，如果在父widget中调用setState，子widget的生命周期：

```
didUpdateWidget -> build
```

在项目中我通过重写该方法监听父widget变化后的参数变化

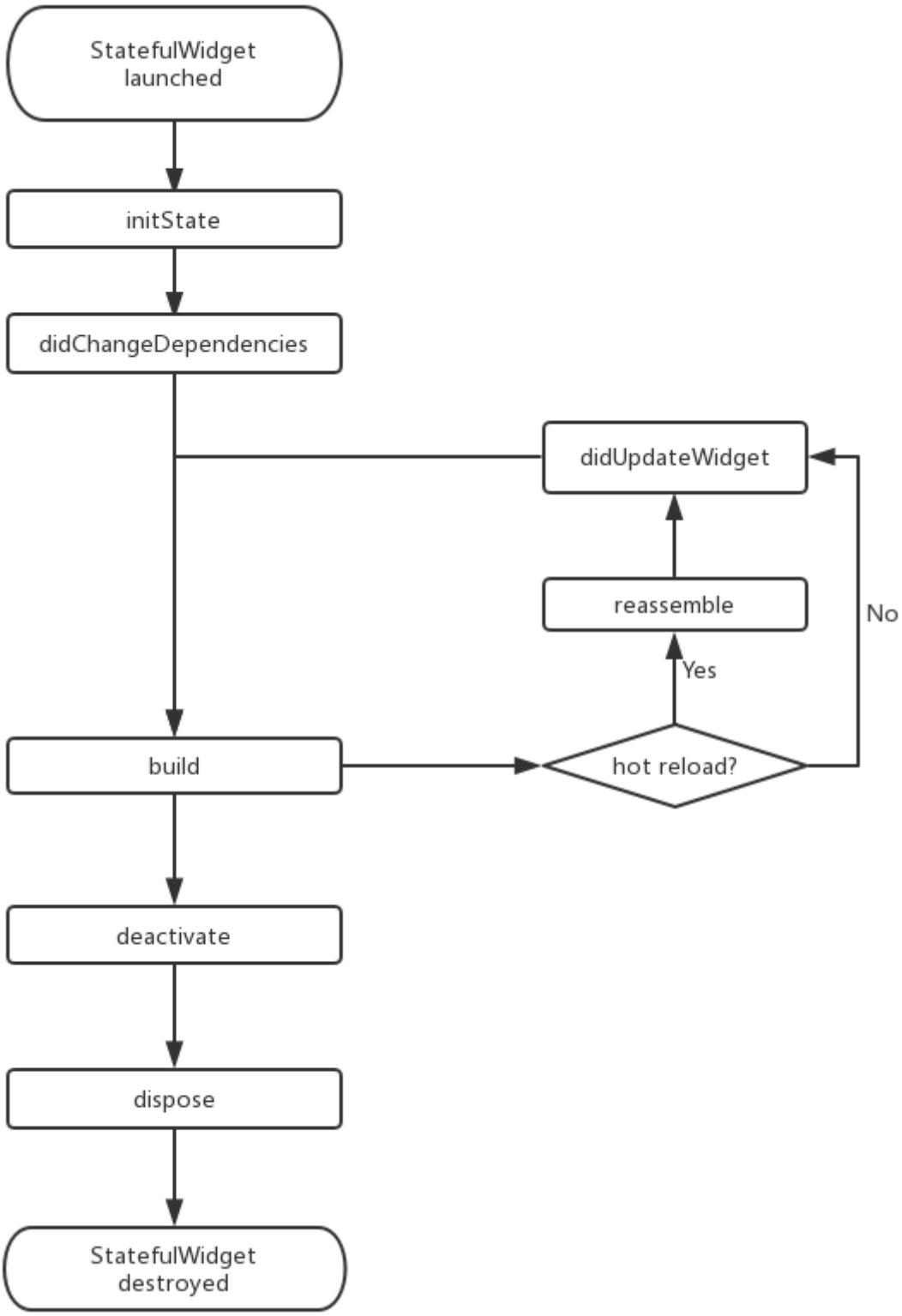


这里的`widget.search`实则是获取的statefulWidget的值：

```dart
class GradeExam extends StatefulWidget {
  final bool search;
  final int chosenId;
  const GradeExam({Key key,this.search=false,this.chosenId}) : super(key: key);

  @override
  _GradeExamState createState() => _GradeExamState(search:this.search,chosenId: this.chosenId);
}

class _GradeExamState extends State<GradeExam> with AutomaticKeepAliveClientMixin{

  bool search;//是否选择学期
  bool empty = true;//是否数据为空
  var dataList = List();//获取数据集
  int chosenId = 0;
  StuLoadingState _loadingState;
  bool fisrtInit  = true;
  String studentNumber;
```

> 因为父布局触发了setState()方法，所以重新调用子布局构造方法，但是这时候子布局不会再次调用createState，而是调用didUpdateWidget->build,所以说只能获取到stateful的值更新

StatefulWidget生命周期



3.集合对象的赋值

最好不要直接赋值，否则容易出错；例如列表使用:add/addAll等方法赋值

4.flutter网络请求的数据获取

- 要么在请求里面对数据解析然后赋值给全局变量，然后setState刷新
- 要么通过async/await组合来获取，然后回调函数

5.Boxshadow

BoxDecoration属性使用

```
    //阴影效果
    const BoxShadow({
    Color color = const Color(0xFF000000),//阴影默认颜色,不能与父容器同时设置color
    Offset offset = Offset.zero,//延伸的阴影，向右下偏移的距离
    double blurRadius = 0.0,//延伸距离,会有模糊效果
    this.spreadRadius = 0.0 //延伸距离,不会有模糊效果
    })
```

```
body: ListView(
          children: <Widget>[
            Container(
              width: 100.0,
              height: 100.0,
              margin: EdgeInsets.all(20.0),
//            color: Color(0xfff1f1f1),
              decoration: BoxDecoration(
                  color: Color(0xffffffff),
                  boxShadow: [
                    BoxShadow(
                      color: Color(0xffff0000),
                      blurRadius: 5.0,
                    ),
                  ]
              ),
            ),
            Container(
              width: 100.0,
              height: 100.0,
              margin: EdgeInsets.all(20.0),
              decoration: BoxDecoration(
                  color: Color(0xffffffff),
                  boxShadow: [
                    BoxShadow(
                      color: Color(0xffff0000),
                      spreadRadius: 5.0,
                    ),
                  ]
              ),
            ),
            Container(
              width: 100.0,
              height: 100.0,
              margin: EdgeInsets.all(20.0),
              decoration: BoxDecoration(
                  color: Color(0xffffffff),
                  boxShadow: [
                    BoxShadow(
                      color: Color(0xffff0000),
                      spreadRadius: 5.0,
                      offset: Offset(3.0, 3.0),
                    ),
                  ],
```

```
            ),
          ),
        ],
      ),
```

6.offstage组件

> 一个可以视觉上隐藏子组件的widget

1. 当offstage为true，控件隐藏,当前控件不会被绘制在屏幕上，不会响应点击事件，也不会占用空间；当Offstage不可见的时候，如果child有动画等，需要手动停掉，Offstage并不会停掉动画等操作。
2. 当offstage为false，当前控件哥平常一样渲染绘制

```
Offstage({Key? key, bool offstage, Widget? child})
```

2021/7/19

1.tabbar的indicator等相关属性

```
indicator:BoxDecoration()//去掉下划线
```

2.AzListView制作滑动的通讯录时:

每一个item不要设置宽度，否则第一条记录会有偏移

2021/7/26

1.Stepper步进器

2.LinearProgressIndicator/CircularProgressIndicator

2021/8/4

1.ListView嵌套Column和Row

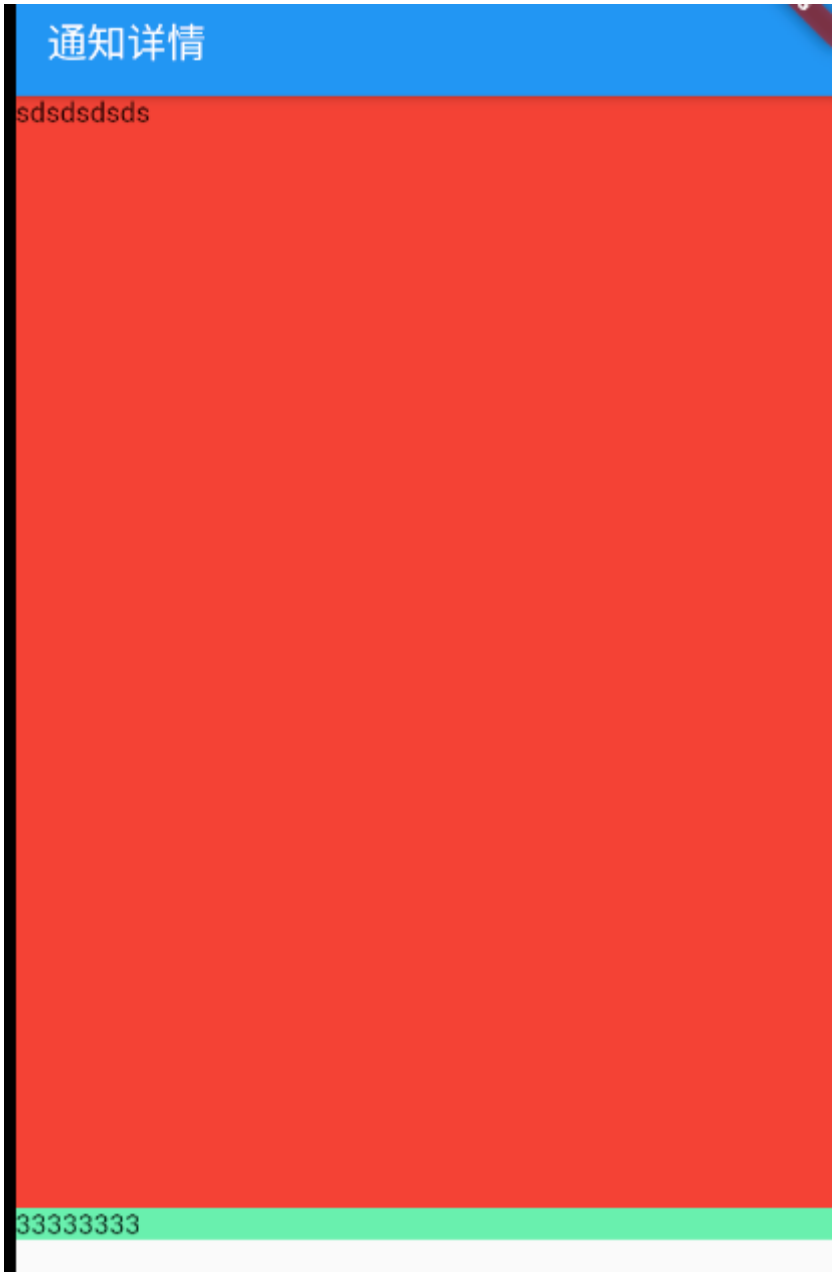
在Flutter的ListView的子View中，你可以在Row中使用Expanded填充水平方向的剩余空间，而无法在Column中使用Expanded填充垂直方向的剩余空间。

> 原因是ListView垂直方向的计算是包裹子View的，**也就是说子View必须有一个明确的高度，或者尽可能小的高度，而不能是无限高。**
> Row是横向排列，在Row中使用Expanded是填充水平方向的剩余空间，这和ListView的这一特性没有冲突，可以使用。
>
> 而Column是竖直排列，在Column中使用Expanded是填充竖直方向的剩余空间，这将和ListView的这一特性发生冲突，因为ListView将无法计算自己的子View的高度。

解决办法：

> IntrinsicHeight可以根据子控件的高度，智能调整自身高度
>
> 把这个控件加在itemVIew的最外层，把Row给包括起来。
>
> IntrinsicHeight可以根据子控件（左侧文案和右侧图片），智能选择一个合适的高度调整自身，使得ListView的子View的高度是确切的，这样ListView便不会因为子View想要无限高，而自己又要包括子View而报错了。

```
ListView(
       children: [
         IntrinsicHeight(
           child: Container(
             width: double.infinity,
             color: Colors.deepPurple,
             child: Column(
               children: [
                 Expanded(
                     flex: 35,
                     child: Container(
                       width: double.infinity,
                       color: Colors.red,
                       child: Text('sdsdsdsds'),
                     )
                 ),
                 Expanded(
                     flex: 1,
                     child: Container(
                       width: double.infinity,
                       color: Colors.greenAccent,
                       child: Text('33333333'),
                     )
                 )
               ],
             ),
           ),
         )
       ],
     )
```



2021/8/13

1.时间选择器：DataTimePicker

2.StreamBuilder

3.Notification

2021/8/16

1.DropdownButton

2.RawChip

3.BoxDecoration的gradient属性实现渐变

2021/8/18

1.格式转换：format

```
///千位数加逗号分割
  String getFormatStepCount(num) {
    if(num<1000&&num>-1000){
      return '$num';
    }
    var format = NumberFormat('0,000');
    if(num<0){
      return '-${format.format(num.abs())}';
    }

    return format.format(num);
  }
```

2021/8/19

1.SmartRefresher

2021/8/20

1.FocusScope、FocusNode

2021/8/27--2021/9/3

1.renderbox

2.WidgetsFlutterBinding.ensureInitialized();

3.WillPopScope

https://www.cnblogs.com/john-hwd/p/10760372.html

4.SafeArea

DA:42:72:48:0D:E6:F7:2F:51:61:13:00:5C:1A:AC:0C:2C:7D:CA:89

com.example.flutterpractice

5.eventBus

6.superController

7.webview

8.list.cast`<String>`()和toString