# Flutter Help

## Talha Saqib

1. I am Rich App

import 'package:flutter/material.dart';  
  
void main() {  
 runApp(MaterialApp(  
 home: Scaffold(  
 appBar: AppBar(  
 title: Text('I am rich'),  
 backgroundColor: Colors.*blueGrey*[900],  
 ),  
 backgroundColor: Colors.*blueGrey*,  
 body: Center(  
 child: Image(image: AssetImage('images/diamond.png')),  
 )),  
 ));  
}

1. Shortcuts

* stless – Stateless Widget (Alt + Enter to convert it into statefull)
* stful – Statefull Widget
* Alt + Enter (Wrap Center, Padding, etc)
* Ctrl + Shift + I to check source code in same screen
* Ctrl + Alt + L to fix indentation
* Shift + F6 to reformat code
* Ctrl + Alt + O to remove unused imports
* Ctrl + Q to check widget properties

1. Margins/Padding
   * EdgeInsets.symmetric(vertical:, horizontal)
   * EdgeInsets.only(left:)
   * EdgeInsets.fromLTRB(0,0,0,0)
2. Columns
   * mainAxisSize: MainAxisSize.min // change col height
   * verticalDirection: VerticalDirection.down // change starting direction
   * mainAxisAlignment: MainAxisAlignment.spaceEvenly // change alignment
   * crossAxisAlignment: …end/stretch // align over horizontal axis for row/col
   * SizedBox(height: 10) // use this between containers in columns as spacers
3. Expanded
   * It acts like container-fluid
   * Flex: 2 // It will give a widget twice the width of Flex:1
4. Dart Specific
   * ‘\_’ means private e.g. \_var
   * ‘extends’ for inheritance
   * Method({int namedParam1, int namedParam2} // don’t require order
   * ‘…’ spread operation, to extract items from a list (or others) to outer container.
   * @required before a attr in constructor to make them required
5. Theme

return MaterialApp(  
 theme: ThemeData(  
 primarySwatch: Colors.*purple*,  
 accentColor: Colors.*amber*,  
 fontFamily: 'Quicksand',  
 textTheme: ThemeData.light().textTheme.copyWith(  
 headline6: TextStyle(  
 fontFamily: 'Quicksand',  
 fontSize: 18,  
 fontWeight: FontWeight.*bold*,  
 ),  
 ),  
 appBarTheme: AppBarTheme(  
 textTheme: ThemeData.light().textTheme.copyWith(  
 headline6: TextStyle(  
 fontFamily: 'OpenSans',  
 fontSize: 20,  
 fontWeight: FontWeight.*bold*,  
 ),  
 ),  
 ),  
 ),  
);

1. Where(), ToList(), Datetime.now().subtract()

List<Transaction> get \_recentTransacions {  
 return \_userTransactions.where((tx) {  
 return tx.date.isAfter(  
 DateTime.now().subtract(  
 Duration(days: 7),  
 ),  
 );  
 }).toList();

1. Modal/Pop-up

void \_startAddNewTransaction(BuildContext ctx) {  
 showModalBottomSheet(  
 context: ctx,  
 builder: (\_) {  
 return GestureDetector(  
 child: NewTransaction(\_addNewTransaction), //Widget  
 onTap: () {},  
 behavior: HitTestBehavior.opaque, //opaque means entire size of gesture detector is hit region  
 );  
 },  
 );  
}

// to close navigatior

Navigator.*of*(context).pop();

1. AppBar IconButton, Scroll, Float Button

return Scaffold(  
 appBar: AppBar(  
 title: Text(  
 'My Expenses',  
 ),  
 actions: <Widget>[  
 IconButton(  
 icon: Icon(Icons.*add*),  
 onPressed: () => \_startAddNewTransaction(context),  
 )  
 ],  
 ),  
 body: SingleChildScrollView(  
 child: Column(  
 children: <Widget>[  
 Chart(\_recentTransacions),  
 TransactionList(\_userTransactions),  
 ],  
 ),  
 ),  
 floatingActionButtonLocation: FloatingActionButtonLocation.*endFloat*,  
 floatingActionButton: FloatingActionButton(  
 child: Icon(Icons.*add*),  
 onPressed: () => \_startAddNewTransaction(context),  
 ),  
);

1. ListView, Date Format, List

ListView.builder(

itemCount:   
 itemBuilder: (ctx, index) {

// index can be used for a list e.g

DateFormat.yMMMd().format(transactions[index].date),

//another example of dateformat

'day': DateFormat.E().format(weekDay).substring(0, 1),

}

List can be generated via List.generate(3, (int index)=>index+index;)

//.fold(intital, (prev, item)) iteratively combines all list values into one

double get totalSpending {  
 return groupedTransactionValues.fold(0.0, (sum, item) {  
 return sum + item['amount'];  
 });  
}

//.map()

To convert a list/map of items into other type i.e. widgets/list etc

.map() returns a lazy iterator, therefore .toList() needs to be used

1. Flexible, Expanded

Expanded() is nothing more than Flexible() with

Flexible (fit: FlexFit.tight) = Expanded()

but, Flexible uses fit :FlexFit.loose by default.

**FlexFit.tight** = Wants to fit tight into parent taking as much space as possible.

**FlexFit.loose** = Wants to fit loose into parent taking as little space as possible for itself.

1. Widgets

* FrictionallySizedBox() – sizes its child to a fraction of total available space.