

Actions Flow

Shortly about homework

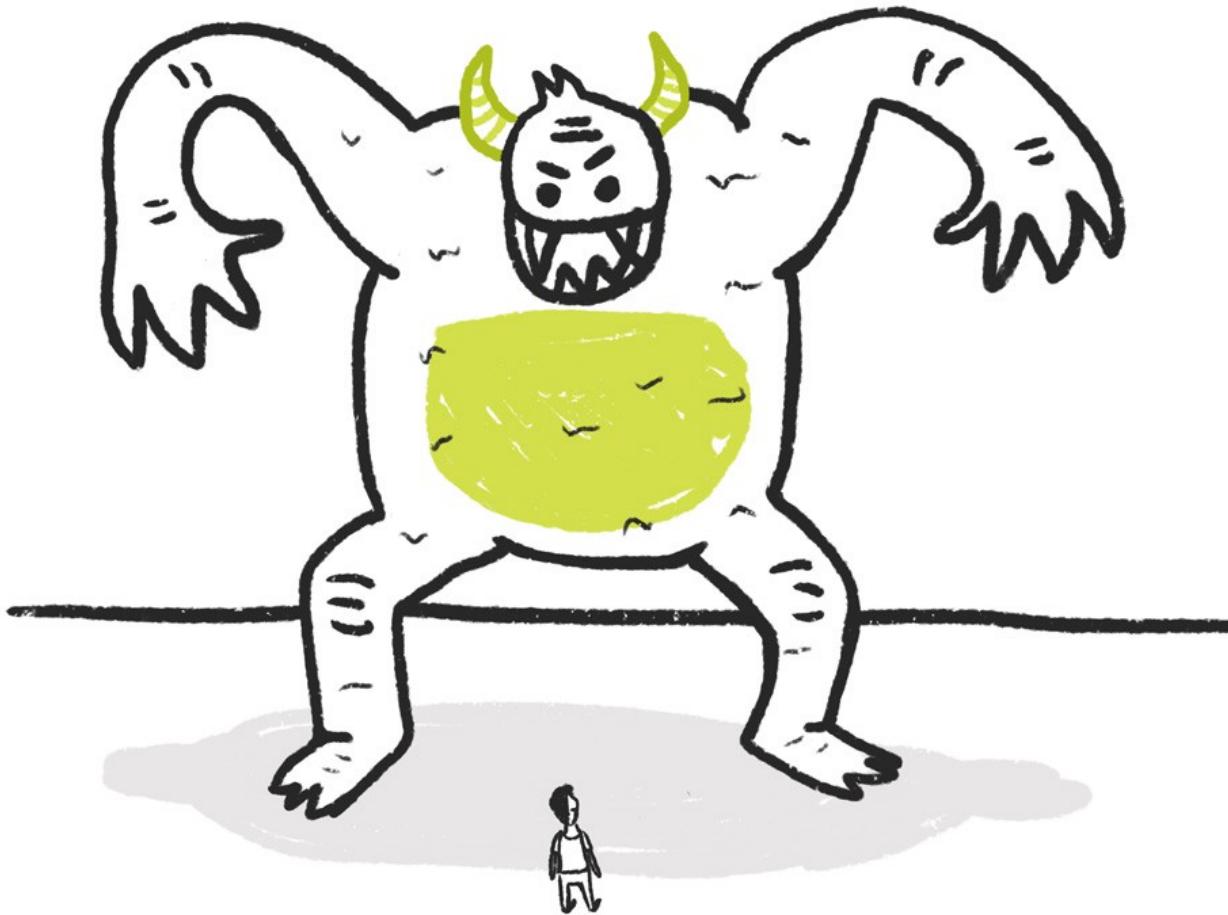
LESS IS MORE



Common mistakes

1. No linter
2. No **final** variables
3. No clear for text when tap cancel in alert
4. No private variables
5. Code for actions in build method
6. Global methods and variables
7. BIG widgets

LEGACY CODE ✖ DEVELOPER



Code smell



Legends about clean code

Principles

YAGNI

- You Ain't
Gonna Need It

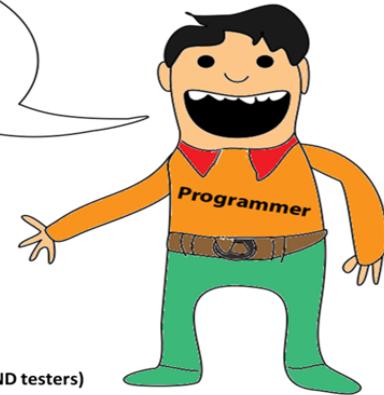


Keep. It. Simple. Stupid.

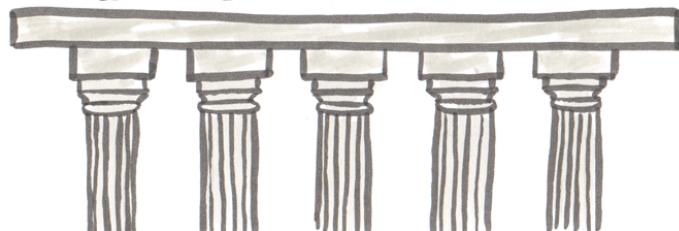
D.R.Y.

Don't
Repeat
Yourself

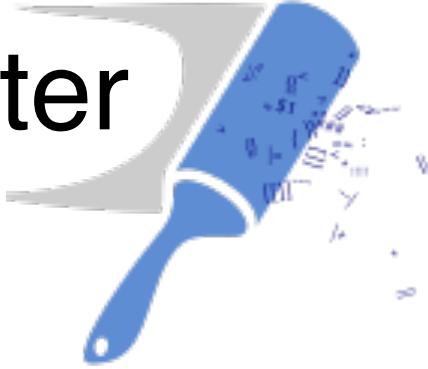
(Good advice for programmers AND testers)



S O L I D



Linter



Programming 101

Naming Conventions

1. Variables tend to start with a letter.
2. Spaces are not allowed in variables
3. Special characters such as “!,-,@” are not allowed in the name
4. Digits are allowed but keep Rule number 1 in mind.

Folder's structure

- ▼  lib
 - ▶  blocprovs
 - ▶  blocs
 - ▶  components
 - ▶  models
 - ▶  screens
 - ▶  services
 - ▶  theme
 -  main.dart

Create app from scratch



Log In



Sign Up



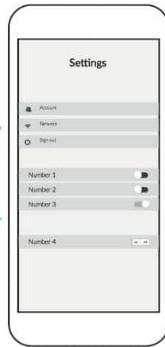
Home



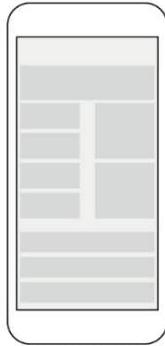
Menu



Settings



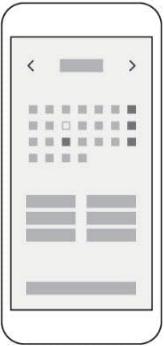
Dashboard



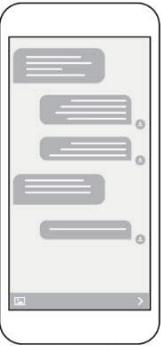
News



Calendar



Message



Profile

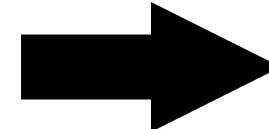


2:44
◀ Tele2 Work Prod



DEBUG

Main Page



2:44
◀ Tele2 Work Prod



DEBUG

Details

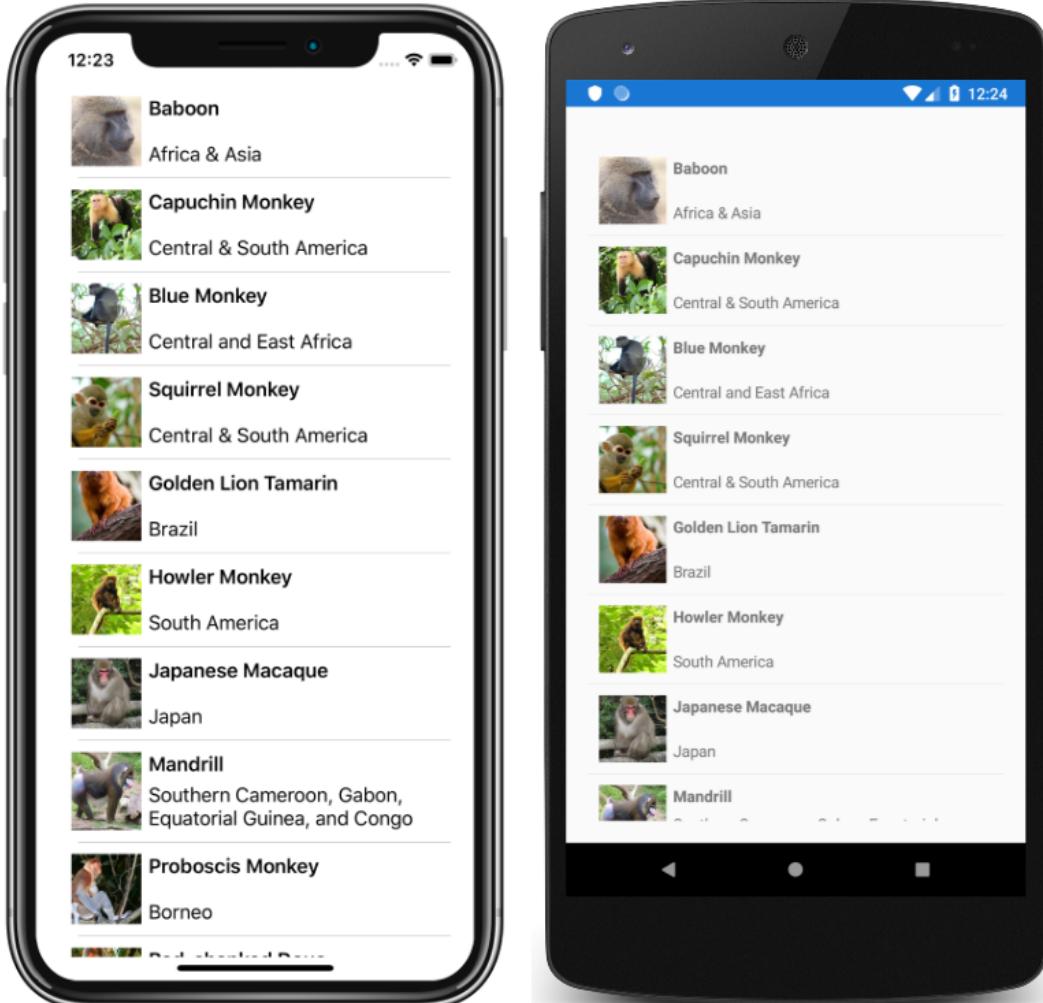


Main Page

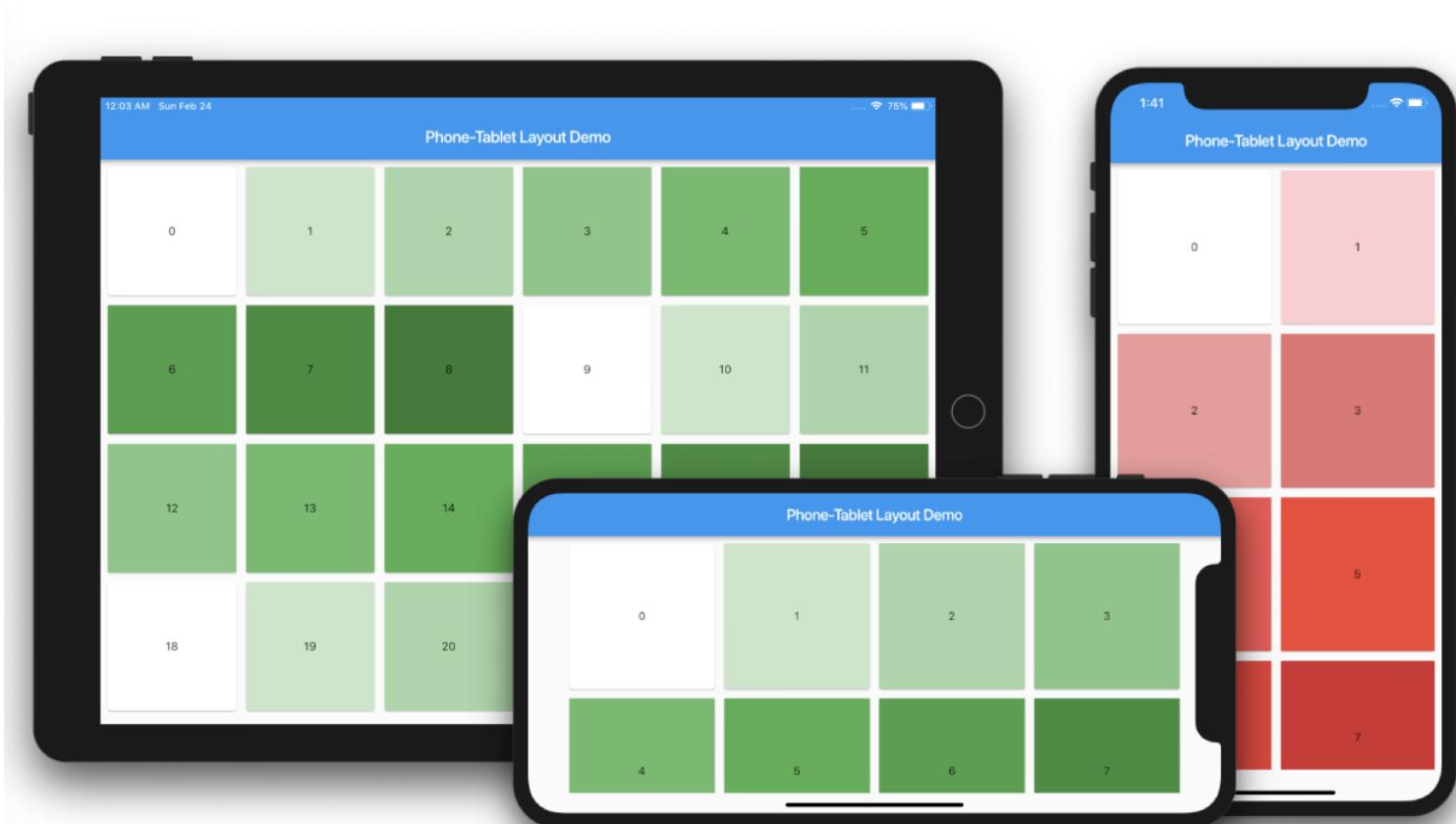


ListView

```
Widget _buildList() {  
  return ListView.builder(  
    itemCount: items.length,  
    itemBuilder: (BuildContext ctxt, int index) {  
      return ListTile(title: Text(items[index]));  
    }  
  );  
}
```



GridView



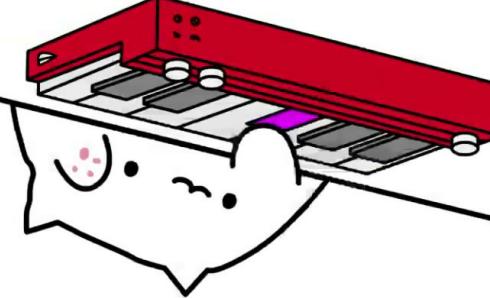
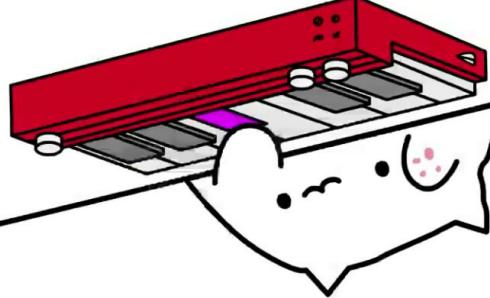
```
GridView.count(  
    crossAxisCount: 2,  
    padding: EdgeInsets.all(5),  
    mainAxisSpacing: 5,  
    crossAxisSpacing: 5,  
    children: <Widget>[  
        // Widgets  
    ],  
)
```

Image

```
Image.asset(  
    'resources/cats/1.jpg'  
    fit: BoxFit.cover)
```



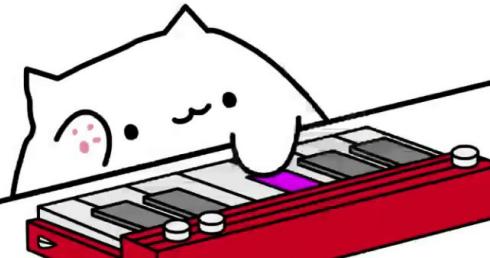
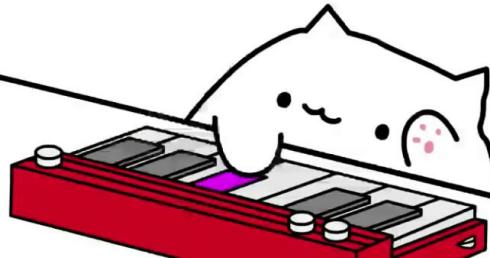
pubspec.yaml

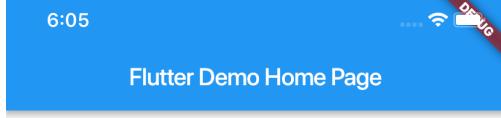


flutter:

assets:

- resources/cats/[1.jpg](#)





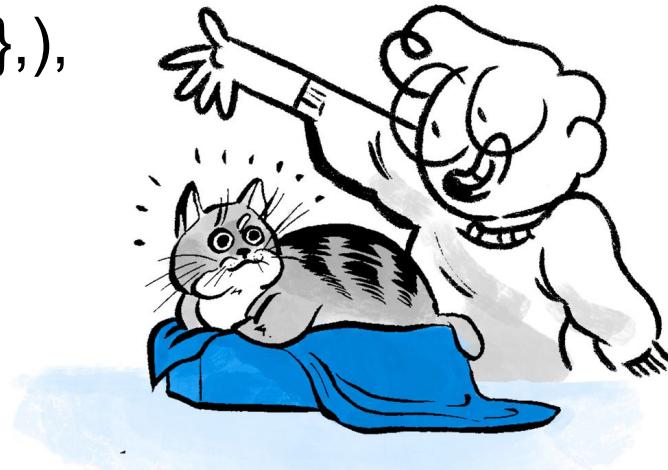
User interaction

InkWell & InkResponse

```
InkWell(  
  onTap: () {},  
  child: Image.asset(  
    'resources/cats/1.jpg'  
    fit: BoxFit.cover)  
)
```

Ink.Image

```
Ink.image(  
    image: AssetImage('resources/cats/2.jpeg'),  
    fit: BoxFit.cover,  
    child: InkWell(onTap: () {},),  
)
```



Basic navigation

Navigator class

A widget that manages a set of child widgets with a stack discipline.

Many apps have a navigator near the top of their widget hierarchy in order to display their logical history using an [Overlay](#) with the most recently visited pages visually on top of the older pages. Using this pattern lets the navigator visually transition from one page to another by moving the widgets around in the overlay. Similarly, the navigator can be used to show a dialog by positioning the dialog widget above the current page.

Using the Navigator

Mobile apps typically reveal their contents via full-screen elements called "screens" or "pages". In Flutter these elements are called routes and they're managed by a [Navigator](#) widget. The navigator manages a stack of [Route](#) objects and provides methods for managing the stack, like [Navigator.push](#) and [Navigator.pop](#).

When your user interface fits this paradigm of a stack, where the user should be able to *navigate* back to an earlier element in the stack, the use of routes and the Navigator is appropriate. On certain platforms, such as Android, the system UI will provide a back button (outside the bounds of your application) that will allow the user to navigate back to earlier routes in your application's stack. On platforms that don't have this build-in navigation mechanism, the use of an [AppBar](#) (typically used in the [Scaffold.appBar](#) property) can automatically add a back button for user navigation.

MaterialPageRoute

For Android, the entrance transition for the page slides the page upwards and fades it in. The exit transition is the same, but in reverse.

The transition is adaptive to the platform and on iOS, the page slides in from the right and exits in reverse. The page also shifts to the left in parallax when another page enters to cover it. (These directions are flipped in environments with a right-to-left reading direction.)

MaterialPageRoute

```
Navigator.push(  
    context,  
    MaterialPageRoute(  
        builder: (context) => NewPage()  
    )  
);
```



Let's code!

Homework

1. During the lesson, we use static GridView for showing content. But from the previous lesson, we know about builder. Your task is show "Add" button on the list screen and add a new image by a tap on the button.
2. Show an image from the internet by URL.
3. Write a title for app bar with a custom font.