



Piscine Mobile - 0

Introduction to Mobile Development

Summary: This document contains the subject matter for Module00 of the Piscine Mobile.

Version: 2.3

Contents

I	Instructions	2
II	Introduction	3
II.1	What is a mobile application?	3
II.2	What is Flutter?	3
III	Exercise 00: A basic display	4
IV	Exercise 01: Say Hello to the World	6
V	Exercise 02: More Buttons	8
VI	Exercise 03: It's Alive!	10
VII	Submission and peer-evaluation	12

Chapter I

Instructions

- Only this page will serve as reference. Do not trust rumors.
- Read attentively the whole document before beginning.
- Your exercises will be corrected by your piscine colleagues.
- The document can be relied upon, do not blindly trust the demos or pictures example which can contain not required additions.
- Got a question? Ask your peer on the right. Otherwise, try your peer on the left.
- By Odin, by Thor ! Use your brain !!!



Intra indicates the date and the hour of closing for your repositories. This date and hour also corresponds to the beginning of the peer-evaluation period for the corresponding piscine day. This peer-evaluation period lasts exactly 24h. After 24h passed, your missing peer grades will be completed with a 0.

Chapter II

Introduction

II.1 What is a mobile application?

A mobile application is a software program designed to run on mobile devices, such as personal digital assistants, enterprise digital assistants, or mobile phones.

These applications can either be pre-installed on phones during manufacturing or delivered as web applications using server-side or client-side processing.

Mobile applications often differ from desktop applications, which run on desktop computers, and web applications, which run in mobile web browsers rather than directly on the mobile device.


II.2 What is Flutter?

Flutter is Google's mobile UI framework for crafting high-quality native interfaces on iOS and Android in record time.

Flutter works with existing code, is used by developers and organizations around the world, and is free and open source.

Chapter III

Exercise 00: A basic display

	Exercise :
A basic display	
Turn-in directory : <code>mobileModule00</code>	
Files to turn in : <code>ex00</code> and all necessary files	
Forbidden functions : None	



As explained in the main project, we are using Flutter for these projects, so we will sometimes use terms specific to this framework. It is up to you to adapt and find the equivalent in the framework you choose to use.

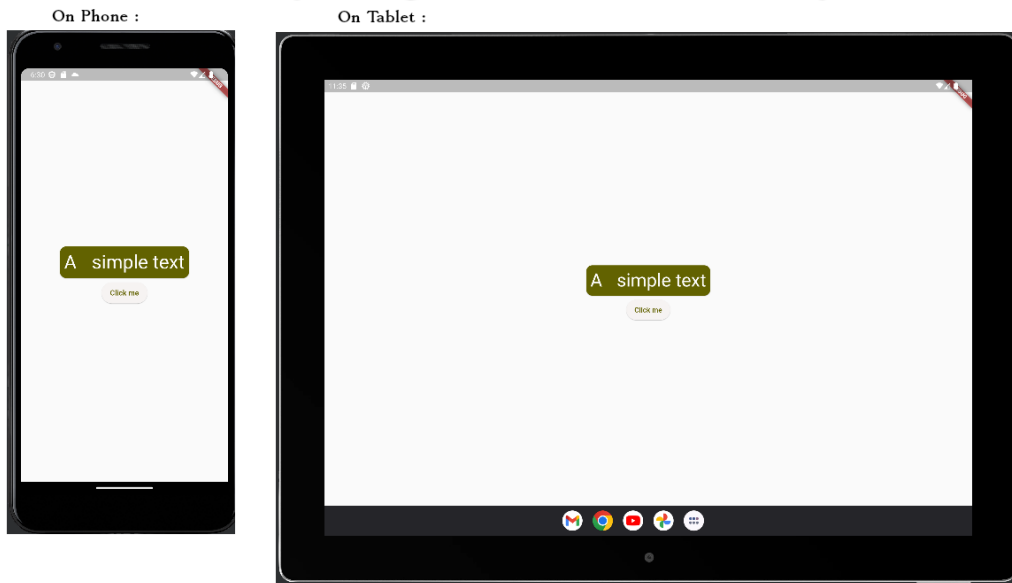
For your first exercise, you will need to create a new `ex00` project using the tools provided by the framework of your choice.

If you are using Flutter, it is important to understand the structure of a Flutter project and, for this exercise, what **widgets** are and their different **states**.

For now, your project must contain a single page with some widgets:

- A text widget with a button below it, both centered horizontally and vertically.
- When the button is clicked, you must display “Button pressed” in the debug console.
- Your application must be responsive.

It should look something like this:




On the debug
console:

```
I/flutter ( 5024): Button pressed
```

Chapter IV

Exercise 01: Say Hello to the World

	Exercise :
Say Hello to the World	
Turn-in directory : <code>mobileModule00</code>	
Files to turn in : <code>ex01</code> and all necessary files	
Forbidden functions : None	

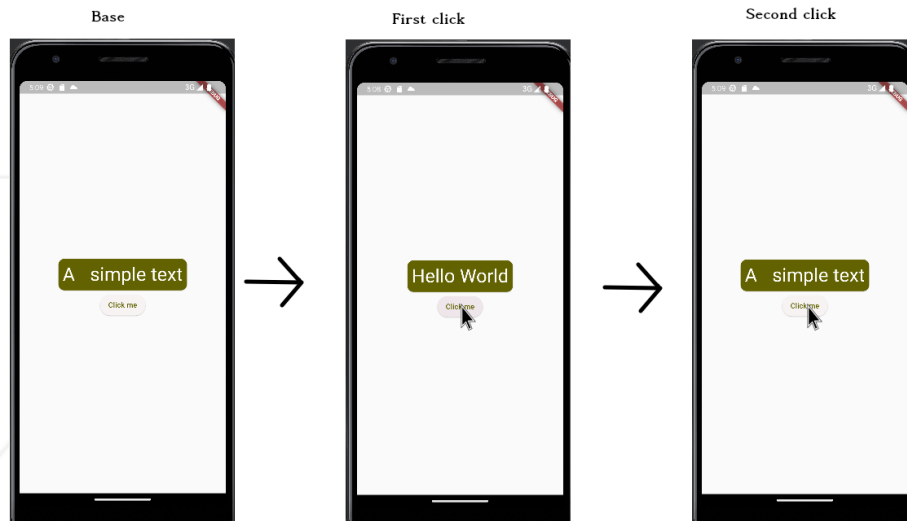


For this exercise, you will need to retrieve the code from your previous exercise and create a new `ex01` project.

Now you will need to ensure that the text displayed in the application changes when you click the button.


It should display “Hello World!” instead of the initial text.

Each time you click the button, the text should toggle between the initial text and “Hello World!”.



Chapter V

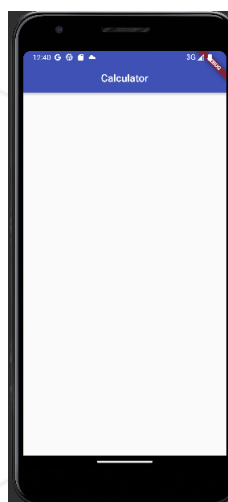
Exercise 02: More Buttons

	Exercise :
	More Buttons
	Turn-in directory : <code>mobileModule00</code>
	Files to turn in : <code>ex02</code> and all necessary files
	Forbidden functions : None

Now that you understand the basics of displaying text and buttons, create a new project called `ex02`.

In this new project, display an `AppBar` at the top of your screen with the title “Calculator”.

Like this:



You also need to add two `TextFields` (one to display the expression and one to display the result) and several buttons. For now, just display “0” inside both `TextFields`; you will handle this in the next exercise.

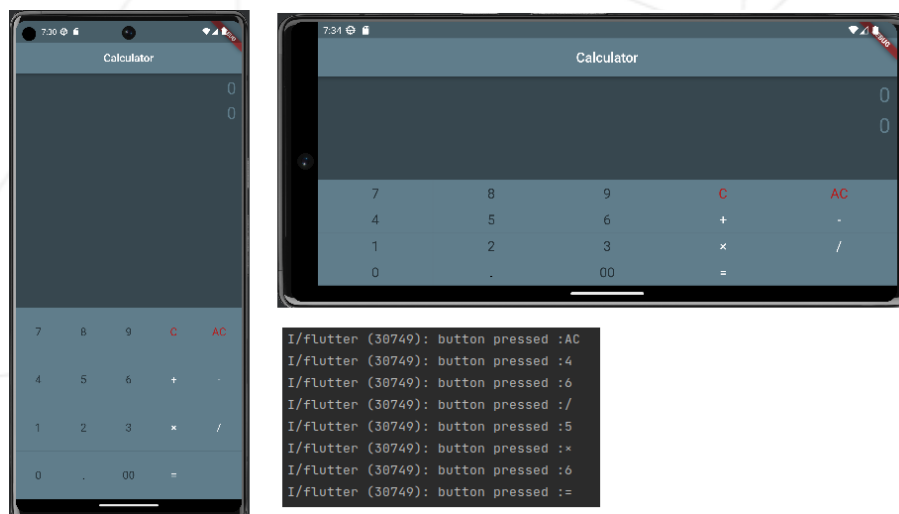
Buttons:

- Numbers from 0 to 9.
- “.” for decimal numbers.
- “AC” to reset the expression and result.
- “C” to delete the last character of the expression.
- “=” to display the result of the expression.
- Operators: “+”, “-”, “*”, “/”.

Add a debug feature. For each button you press, display the text of the button in the debug console.


Once all the buttons are properly set, make sure the display is responsive for all devices (phone, tablet, etc.).

You should have something like this:



Chapter VI

Exercise 03: It's Alive!

	Exercise :
	It's Alive!
	Turn-in directory : <code>mobileModule00</code>
	Files to turn in : <i>calculator_app</i> and all necessary files
	Forbidden functions : None



For this exercise, you will need to retrieve the code from your previous exercise and create the *calculator_app* project. You can use the `math_expressions` library or an equivalent library.

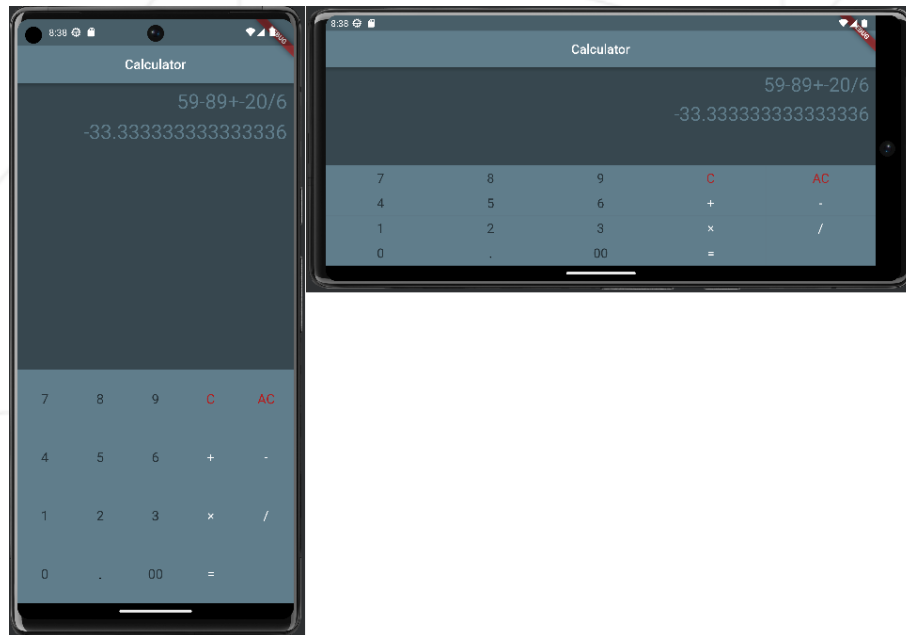
Now it's time to make your calculator work!
You need to add the logic behind it.

The TextFields created in the previous exercise must now display the expression and the result of the expression.

- You must be able to perform the following operations:
 - Addition
 - Subtraction
 - Multiplication
 - Division
- You can perform multiple operations in one expression (e.g., $1 + 2 * 3 - 5 / 2$).
- You must be able to enter a negative number (by pressing the “-” button before the number).
- You must be able to enter decimal numbers.

- You must be able to delete the last character of the expression.
- You must be able to clear the whole expression and result.

Your result could look something like this:



Be careful—if you don't thoroughly test your code, you might encounter issues. For example, an incorrect expression, division by 0, or very large numbers can cause problems.



Your application must NEVER crash!

Chapter VII

Submission and peer-evaluation

Turn in your assignment in your `Git` repository as usual. Only the work inside your repository will be evaluated during the defense. Don't hesitate to double check the names of your folders and files to ensure they are correct.



The evaluation process will happen on the computer of the evaluated group.