**Calculator GUI with Python**

**In this article, I will introduce you to how to create a graphical user interface of the calculator with the Python programming language. Here I am from the package Kivy I will use Python to create this user interface .**

**Kivy is a free and open source Python library that enables fast and easy development of cross-platform and interactive applications. Execution speed Kivy It is the same compared to other mobile development alternatives like Java for Android and Objective C for iOS .**

**In addition, Kivy It has a great advantage, which is that it can run on multiple platforms. Just like HTML5. In this case, Kivy It performs best because it doesn't rely on a heavy browser and most of its components are in C using libraries Cython are implemented so that most of the graphics processing can be performed directly on the GPU .**

**Kivy It creates a good balance between performance and portability in hardware and software environments .**

**How is the graphical user interface of the calculator coded with Python?**

**We do not need to use a file called file.kv We don't have to make a calculator, because the calculator is a very simple program. Let's see how we can build a simple user interface for a calculator in a free Python project :**

**from kivy \_ app Import App**

**from kivy \_ uix . button Import Button**

**from kivy \_ uix . boxlayout import BoxLayout**

**from kivy \_ uix . gridlayout import GridLayout**

**from kivy \_ uix . label import Label**

**class myApp ( App ):**

**def build( self ):**

**root\_widget = BoxLayout ( orientation = 'vertical' )**

**output\_label = Label ( size\_hint\_y = 0.75 , font\_size = 50 )**

**button\_symbols = ( '1' , '2' , '3' , '+' ,**

**'4' , '5' , '6' , '-' ,**

**'7' , '8' , '9' , '.' ,**

**'0' , '\*' , '/ ' '=' )**

**button\_grid = GridLayout ( cols = 4 , size\_hint\_y = 2 )**

**for symbol in button\_symbols :**

**button\_grid . add\_widget ( Button ( text = symbol ))**

**clear\_button = Button ( text = 'Clear' , size\_hint\_y = None , height = 100 )**

**def print\_button\_text ( instance ):**

**output\_label . text += instance . text**

**for button in button\_grid . children [ 1 ::]**

**button . bind ( on\_press = print\_button\_text )**

**def resize\_label\_text ( label , new\_height ):**

**label . fontsize = 0.5 \* label . height**

**output\_label . bind ( height = resize\_label\_text )**

**def evaluate\_result ( instance ):**

**try :**

**output\_label . text = str ( eval ( output\_label . text ))**

**except SyntaxError :**

**output\_label . text = 'Python syntax error!'**

**button\_grid . children [ 0 ]. bind ( on\_press = evaluate\_result )**

**def clear\_label ( instance ):**

**output\_label . text = ""**

**clear\_button . bind ( on\_press = clear\_label )**

**root\_widget . add\_widget ( output\_label )**

**root\_widget . add\_widget ( button\_grid )**

**root\_widget . add\_widget ( clear\_button )**

**return root\_widget**

**myApp (). run ()**

Amirshayan Jalili

[Shayan138190@gmail.com](mailto:Shayan138190@gmail.com)

<https://github.com/Amirshayan2002>