



C1 - Python

C-COD-140

Python Day03

Bootstrap



FOREWORD

Welcome,

The purpose of this bootstrap is to make a calculator. During this bootstrap, we will start by creating the interface of our program. Then, we will perform the first calculation (an addition).



BOOTSTRAP

+ STEP 1

Restrictions: we cannot use `exec` and `eval` for this project.

The first step is to check that tkinter is installed correctly.

```
Terminal
~/C-COD-140> python3
>>>import tkinter
>>>tkinter._test()
```

If a window appears, tkinter is installed correctly.
Otherwise you can run this command:



```
sudo apt-get install python3-tk
```

We are now ready to create our interface. Here is how to generate a window with tkinter, in a Python file:

```
Terminal
from tkinter import *
window = Tk()
window.mainloop()
```



The window is empty because no elements have been added

Then we will add a first button. This is linked to the 'quit' function which will leave the interface when it is clicked.

```
Terminal
from tkinter import *
window = Tk()
Button(window, text="Close", command=window.quit).pack()
window.mainloop()
```



+ STEP 2

In this step, we will fill in the interface so that it looks like a calculator.
We will add the sign '+' and '=' as well as the numbers from 0 to 9.

```
Terminal
Button(window, text=" 0 ", command=lambda: input_key("0")).pack()
Button(window, text=" 1 ", command=lambda: input_key("1")).pack()
Button(window, text=" 2 ", command=lambda: input_key("2")).pack()
Button(window, text=" 3 ", command=lambda: input_key("3")).pack()
Button(window, text=" 4 ", command=lambda: input_key("4")).pack()
Button(window, text=" 5 ", command=lambda: input_key("5")).pack()
Button(window, text=" 6 ", command=lambda: input_key("6")).pack()
Button(window, text=" 7 ", command=lambda: input_key("7")).pack()
Button(window, text=" 8 ", command=lambda: input_key("8")).pack()
Button(window, text=" 9 ", command=lambda: input_key("9")).pack()
Button(window, text=" + ", command=lambda: input_key("+")).pack()
Button(window, text=" = ", command=lambda: equal()).pack()
```

Each button is assigned a value, just give the desired value for each element.



+ STEP 3

In the previous example, the numbers and the sign '+' are linked to a function 'input_key(value)' which will retrieve the input.

Once this input is retrieved, we can build a string containing the desired equation.

```
Terminal
def digit(value):
    global calc_input
    calc_input += value
    print(calc_input)
```



We will use a global variable to keep the value even after exiting the function.

+ STEP 4

We now have the full equation. All that remains is to calculate this one.



Use split to split the equation into several blocks of numbers.



Don't forget to display the result!

```
Terminal
def equal():
    global calc_input
    additions = calc_input.split("+")
    result = 0
    for value in additions:
        result += int(value)
    calc_input = ""
    print(result)
```



+ STEP 5

Let's improve our calculator.

All buttons are one below the other.

Let's rearrange the buttons in a grid pattern.

```
-----
|               | Close |
|-----|
|               |
|-----|
|-----|
| 7 | 8 | 9 | + |
|---+---+---+---|
| 4 | 5 | 6 |   |
|---+---+---+---|
| 1 | 2 | 3 | = |
|---+---+---+---|
| 0 |   |   |   |
|-----|
```

Instead of using the `pack()` method on the buttons, let's use the `grid(row, column)` method.

```
Terminal
Button(window, text="Close", command=window.quit).grid(row=0, column=3)
Button(window, text=" 0 ", command=lambda: input_key("0")).grid(row=6, column=0)
Button(window, text=" 1 ", command=lambda: input_key("1")).grid(row=5, column=0)
Button(window, text=" 2 ", command=lambda: input_key("2")).grid(row=5, column=1)
Button(window, text=" 3 ", command=lambda: input_key("3")).grid(row=5, column=2)
# ...
Button(window, text=" 7 ", command=lambda: input_key("7")).grid(row=3, column=0)
Button(window, text=" 8 ", command=lambda: input_key("8")).grid(row=3, column=1)
Button(window, text=" 9 ", command=lambda: input_key("9")).grid(row=3, column=2)
Button(window, text=" + ", command=lambda: input_key("+")).grid(row=3, column=3)
Button(window, text=" = ", command=lambda: equal()).grid(row=5, column=3)
```



+ STEP 6

Now let's try to display our calculus and its result in our graphical interface.
Let's add two 'Label' that will be used to display our calculus and result

```
Terminal
calc_input_text = StringVar()
Label(window, textvariable=calc_input_text).grid(row=1, column=0)
result_text = StringVar()
Label(window, textvariable=result_text).grid(row=2, column=0)
```



The `Label` object allows us to display a text (not directly modifiable by the user)

It is possible to bind a `StringVar` object to a `Label` by using a `textvariable` parameter in the `Label`.
When the `StringVar` object is modified via the `set()` method, the text in the `Label` will change automatically.

Each time the user clicks on a key, we update the `Label` corresponding to the calculus display.

```
Terminal
def input_key(value):
    # ...
    calc_input_text.set(calc_input)
```

When he clicks on the '=' button, we update the `Label` corresponding to the result display.

```
Terminal
def equal():
    # ...
    calc_input_text.set(calc_input)
    result_text.set(result)
```



+ CONCLUSION

Congratulations, if you have arrived here, you have designed a calculator that allows you to add in Python. You are free to add functionalities to it so that it corresponds to what is requested in the subject.

Here is the file you should get.

```
Terminal
~/C-COD-140> cat calcul.py
from tkinter import *

window = Tk()
calc_input = ""

def input_key(value):
    global calc_input
    calc_input += value
    calc_input_text.set(calc_input)
    print(calc_input)

def equal():
    global calc_input
    additions = calc_input.split("+")
    result = 0
    for value in additions:
        result += int(value)
    calc_input = ""
    calc_input_text.set(calc_input)
    result_text.set(result)
    print(result)

Button(window, text="Close", command=window.quit).grid(row=0, column=3)
Button(window, text=" 0 ", command=lambda: input_key("0")).grid(row=6, column=0)
Button(window, text=" 1 ", command=lambda: input_key("1")).grid(row=5, column=0)
Button(window, text=" 2 ", command=lambda: input_key("2")).grid(row=5, column=1)
Button(window, text=" 3 ", command=lambda: input_key("3")).grid(row=5, column=2)
# ...
Button(window, text=" 7 ", command=lambda: input_key("7")).grid(row=3, column=0)
Button(window, text=" 8 ", command=lambda: input_key("8")).grid(row=3, column=1)
Button(window, text=" 9 ", command=lambda: input_key("9")).grid(row=3, column=2)
Button(window, text=" + ", command=lambda: input_key("+")).grid(row=3, column=3)
Button(window, text=" = ", command=lambda: equal()).grid(row=5, column=3)
calc_input_text = StringVar()
Label(window, textvariable=calc_input_text).grid(row=1, column=0)
result_text = StringVar()
Label(window, textvariable=result_text).grid(row=2, column=0)
window.mainloop()
```


