

Forum:[Hack Forums](#)

Thread: [My first python project \(Calculator\)](#)

TISA - January 18, 2016, 10:35 am

Is there anything I have done wrong? Is there anything that I could do better? What can i do next as a follow up project?

(This is my first project and I've only just started learning python)

pastebin.com/A1fEBmPt

(It was too many characters)

Thanks in advance

Ariesfangz - December 1, 2017, 4:12 pm

(Jan 18, 2016 11:35 AM)TISA Wrote: [\[CITE\]](#) Is there anything I have done wrong? Is there anything that I could do better? What can i do next as a follow up project?

(This is my first project and I've only just started learning python)

pastebin.com/A1fEBmPt

(It was too many characters)

Thanks in advance

you have made this far more difficult than it really needs to be.

ok what type of maths are you trying to do and I will see what I can do

scrubjay - December 2, 2017, 7:03 am

Seriously, you can see that it's your first project.

- Use constants, then you don't need to type everything twice.
- Use methods as you did with CalcTop() etc.
- You don't need to convert strings to a string :D
- Wtf is this?

Really.. I don't want to say anything about the logic of this. I hope you know that this is a absolutely retarded way to write a calculator.

But what ever. Keep going on! I'm sure you will get better :)

Good luck and have fun!

oh and project suggestion: Code something you want to code. You will get frustrated and at this point you'll learn. "Exercise projects" doesn't make sense I think.

michalmonday - December 2, 2017, 8:01 am

hi, I like the graphical representation of the calculator xD

I'd try to avoid long if/elif sequences used to adjust the amount of empty spaces depending on the length of some number, you could check the lenght of the number and use multiplication to get desired amount of spaces using something like this:

Code result = 10000

```
print ("SPACE" * len(str(result))) # just replace "SPACE" with " "
```

Also to make the code more readable while having many options for the user to pick you could use list with functions instead of long if/elif sequences like this:

```
Code def Func_Addition(x, y):  
    return x+y
```

```
def Func_Substraction(x,y):  
    return x-y
```

```
functions = [  
    Func_Addition,  
    Func_Substraction  
]
```

```
while True:  
    option = int(input("\nPick an option: 1-Addition, 2-Substraction\n> "))  
    print(functions[option - 1](5,10))
```

Ultimately you could avoid using multiple functions dedicated for the calculations and replace them with simple eval() function which takes a string parameter and treats it as if it was a python instruction so it does the calculation for you as long as you put the right mathematical operation inside the string like eval("2 + 2") which will return 4. You could create another list with mathematical operators and use it like this:

```
Code first_number = 5  
second_number = 10
```

```
math_operators = [  
    "+",  
    "-",  
]
```

```
while True:  
    option = int(input("\nPick an option: 1-Add, 2-Sub\n> "))  
    print(eval(str(first_number) + math_operators[option - 1] + str(second_number)))
```

I'd recommend this video to you: <https://www.youtube.com/watch?v=OSGv2VnC0go>

I wouldn't bother about the "I hope you know it's retarded" useless comments because the code efficiency will come with experience as long as you learn+practice, especially with a language like python which allows to automate complex tasks in a very concise way

Gliese - December 2, 2017, 8:03 am

This thread was created a year ago