

Forum:[Hack Forums](#)

Thread: [Miles per gallon calculator: GUI interface for Python](#)

billybluiceez - December 8, 2016, 2:57 pm

Here's a homework assignment that I've been tinkering with and it is mostly finished, the only thing needs to be done is code the error stuff.

when you hit the calculate button it has to calculate miles per gallon, that works properly as this far but I need it to send back errors boxes if any of the 2 entries are "0", "negative", or "non-numeric", if anybody can figure it out, it would be appreciated. I will attach the file.

[code]

```
class MPG:
```

```
#create name of initializer function
```

```
    def __init__(self):
```

```
# main window/frames
```

```
    self.main_window = tkinter.Tk()
```

```
    uframe = tkinter.Frame(self.main_window)
```

```
    lframe = tkinter.Frame(self.main_window)
```

```
    button_frame = tkinter.Frame(self.main_window)
```

```
    gas_lbl = tkinter.Label(uframe, text="Enter # of gallons used",  
                           font=("Calibri",18,"bold"))
```

```
    self.gal_gas = tkinter.Entry(uframe, width = 10, font=("Calibri",\n                                   20,"bold"))
```

```
    miles_lbl = tkinter.Label(lframe, text="Enter miles travelled", \n                              font=("Calibri",18,"bold"))
```

```
    self.miles_travel = tkinter.Entry(lframe, width = 10, font=("Calibri",\n                                                                20,"bold"))
```

```
    btn_calculate = tkinter.Button(button_frame, text = "Calculate!", font=("Consolas",\n                                                                              15,"bold"),
```

```
                                   command = self.btn_calculate_event_handler)
```

```
    btn_exit = tkinter.Button(button_frame, text = "Exit", font=("Consolas",\n                                                                15,"bold"),command = self.cmd_btn_event_handler)
```

```
    gas_lbl.pack(side="left")
```

```
    self.gal_gas.pack(side="top")
```

```
    miles_lbl.pack(side="left")
```

```
    self.miles_travel.pack(side="bottom")
```

```
    btn_calculate.pack(side="left")
```

```
    btn_exit.pack(side="right")
```

```
    button_frame.pack(side="bottom")
```

```
    uframe.pack()
```

```
    lframe.pack()
```

```
    tkinter.mainloop()
```

```
# calculate button
```

```

def btn_calculate_event_handler(self):
    self.gal = float(self.gal_gas.get())
    if self.gal <= 0:
        tkinter.messagebox.showerror("Must be greater than 0")
    if float(self.gal) == int:
        print("use numbers")
    else:
        print("Use numbers only")
    self.miles = float(self.miles_travel.get())
    if self.miles <=0:
        tkinter.messagebox.showerror("Must be greater than 0")
    if float(self.miles) == int:
        print("why")
    else:
        print("Use numbers only") #this prints up in background 2 times
    self.miles_per_gallon = self.miles / self.gal
    self.miles_per_gallon = (str(round(self.miles_per_gallon, 1)))
    result_msg = "MPG = " + self.miles_per_gallon
    tkinter.messagebox.showinfo("", result_msg)
# exit button
def cmd_btn_event_handler(self):
    self.main_window.destroy()

def main():
    main_window = MPG()

main()

```

[code]

i've been told Python 64 bit will not work for GUI, but I've been using 64 bit on my system with no problems, just a heads up if you have any problems.

Ensky - December 8, 2016, 3:04 pm

Hey man, with the first [code] tag, remove the slash and it should work!

Goodies - December 8, 2016, 5:06 pm

1) Not sure what your original code meant by this:

PHP Code

```
<?php
```

```
if float(self.gal) == int:
```

```
# ...
```

a) that will never be true because an instance of the float class will never be equal to the int class itself, and b) you shouldn't use == sign when comparing types. Get in the habit of using "is" for type comparison.

2) The values in your btn_calculate_event_handler() method should not be instance variables. They should be local only to the function. There is absolutely no reason to use self() at that point.

3) Some design issues: any calculations that don't directly depend on the state of the object should preferably be done in a static method. I didn't change the design at all, but you should have a calculate method that is static and is only called by the btn_calculate_event_handler() method. This is where you should perform your checks and return any values or exceptions that may be raised.

4) I added a "safe cast" method that will attempt to convert one value into the provided type. Upon

failure, it will return whatever default value you provide (in my case, None). I then check if the values are None (meaning the conversion failed, which indicates an invalid number) and if they're 0.

Final code: [\(Click to View\)](#)

billybluiceez - *December 8, 2016, 7:13 pm*

Goodies! perfect work, I looked over your project and made some adjustments here and there so i can write it out the way i would code and i saw the areas you pointed out and you made alot of sense so thank you very much for your help bro