Assignment 5 - Project

Date: 04-04-2022

Members:

- PHAN TAN MONG
- LA THI TUONG VY
- NGUYEN NGOC QUYNH NHU

Calculator

Build a program that can be used as a basic calculator. Your program should have a manu displayed for the user to choose from, where are listed basic operations: addition, subtraction, multiplication, division, second power, square root, exit.
The program should allow user to choose the desired operation over and over again until user chooses to quit using it

 # Import ipywidgets for GUI and math to use sqrt function

In [1]:
```
from ipywidgets import Button, GridBox, Layout, ButtonStyle, Text
from math import sqrt
from IPython.display import display, Markdown
```

 # Create display screens

In [2]:
```
# Create main display screen: shows the result
main_screen = Text(disabled=True,
                   layout=Layout(width='310px', height='50px', grid_area='disp'))
# Create sub display screen: shows the numbers with operation
sub_screen = Text(disabled=True,
                   layout=Layout(width='310px', height='50px', grid_area='sub'))
```

 # Create all buttons

In [3]:
```
# Create all buttons
button1  = Button(description='1',
                  layout=Layout(width='70px', height='50px', grid_area='b1'),
                  style=ButtonStyle(button_color='lightblue', font_weight = 'bold'))
button2  = Button(description='2',
                  layout=Layout(width='70px', height='50px', grid_area='b2'),
                  style=ButtonStyle(button_color='lightblue', font_weight = 'bold'))
button3  = Button(description='3',
                  layout=Layout(width='70px', height='50px', grid_area='b3'),
                  style=ButtonStyle(button_color='lightblue', font_weight = 'bold'))
button4  = Button(description='4',
                  layout=Layout(width='70px', height='50px', grid_area='b4'),
                  style=ButtonStyle(button_color='lightblue', font_weight = 'bold'))
button5  = Button(description='5',
                  layout=Layout(width='70px', height='50px', grid_area='b5'),
                  style=ButtonStyle(button_color='lightblue', font_weight = 'bold'))
button6  = Button(description='6',
                  layout=Layout(width='70px', height='50px', grid_area='b6'),
                  style=ButtonStyle(button_color='lightblue', font_weight = 'bold'))
```

```
button7  = Button(description='7',
                  layout=Layout(width='70px', height='50px', grid_area='b7'),
                  style=ButtonStyle(button_color='lightblue', font_weight = 'bold'))
button8  = Button(description='8',
                  layout=Layout(width='70px', height='50px', grid_area='b8'),
                  style=ButtonStyle(button_color='lightblue', font_weight = 'bold'))
button9  = Button(description='9',
                  layout=Layout(width='70px', height='50px', grid_area='b9'),
                  style=ButtonStyle(button_color='lightblue', font_weight = 'bold'))
button0  = Button(description='0',
                  layout=Layout(width='70px', height='50px', grid_area='b0'),
                  style=ButtonStyle(button_color='lightblue', font_weight = 'bold'))
button_cla  = Button(description='AC',
                  layout=Layout(width='70px', height='50px', grid_area='cla'),
                  style=ButtonStyle(button_color='gray', font_weight = 'bold'))
button_pw  = Button(description='x^2',
                  layout=Layout(width='70px', height='50px', grid_area='pw'),
                  style=ButtonStyle(button_color='gray', font_weight = 'bold'))
button_sqrt  = Button(description='Sqrt',
                  layout=Layout(width='70px', height='50px', grid_area='sqrt'),
                  style=ButtonStyle(button_color='gray', font_weight = 'bold'))
button_div  = Button(description='/',
                  layout=Layout(width='70px', height='50px', grid_area='div'),
                  style=ButtonStyle(button_color='DarkOrange', font_weight = 'bold'))
button_multi  = Button(description='x',
                  layout=Layout(width='70px', height='50px', grid_area='multi'),
                  style=ButtonStyle(button_color='DarkOrange', font_weight = 'bold'))
button_minus  = Button(description='-',
                  layout=Layout(width='70px', height='50px', grid_area='minus'),
                  style=ButtonStyle(button_color='DarkOrange', font_weight = 'bold'))
button_plus  = Button(description='+',
                  layout=Layout(width='70px', height='50px', grid_area='plus'),
                  style=ButtonStyle(button_color='DarkOrange', font_weight = 'bold'))
button_equal  = Button(description='=',
                  layout=Layout(width='150px', height='50px', grid_area='equal'),
                  style=ButtonStyle(button_color='MediumSlateBlue', font_weight = 'bold'))
button_dot  = Button(description='.',
                  layout=Layout(width='70px', height='50px', grid_area='dot'),
                  style=ButtonStyle(button_color='lightblue', font_weight = 'bold'))
```

 # Define on_click event for button '0'

 # It will be triggered as soon as button '0' is pressed and shows number on main screen

In [4]:

```
# Define on_click event for button 0
def on_clicked_b0(b):
    global dot_flag
    global result_flag
    if result_flag == True:
        main_screen.value = ""
        sub_screen.value = ""
        result_flag = False
    if main_screen.value == "" or float(main_screen.value) == 0 and dot_flag == False:
        main_screen.value = "0"
    else:
        main_screen.value = main_screen.value + "0"

button0.on_click(on_clicked_b0)
```

 # Define on_click event for button '1'

 # It will be triggered as soon as button '1' is pressed and shows number on main screen

In [5]:

```
# Define on_click event for button 1
```

```
def on_clicked_b1(b):
    global dot_flag
    global result_flag
    if result_flag == True:
        main_screen.value = ""
        sub_screen.value = ""
        result_flag = False
    if main_screen.value == "" or float(main_screen.value) == 0 and dot_flag == False:
        main_screen.value = "1"
    else:
        main_screen.value = main_screen.value + "1"

button1.on_click(on_clicked_b1)
```

 # Define on_click event for button '2'

 # It will be triggered as soon as button '2' is pressed and shows number on main screen

In [6]:

```
# Define on_click event for button 2
def on_clicked_b2(b):
    global dot_flag
    global result_flag
    if result_flag == True:
        main_screen.value = ""
        sub_screen.value = ""
        result_flag = False
    if main_screen.value == "" or float(main_screen.value) == 0 and dot_flag == False:
        main_screen.value = "2"
    else:
        main_screen.value = main_screen.value + "2"

button2.on_click(on_clicked_b2)
```

 # Define on_click event for button '3'

 # It will be triggered as soon as button '3' is pressed and shows number on main screen

In [7]:

```
# Define on_click event for button 3
def on_clicked_b3(b):
    global dot_flag
    global result_flag
    if result_flag == True:
        main_screen.value = ""
        sub_screen.value = ""
        result_flag = False
    if main_screen.value == "" or float(main_screen.value) == 0 and dot_flag == False:
        main_screen.value = "3"
    else:
        main_screen.value = main_screen.value + "3"

button3.on_click(on_clicked_b3)
```

 # Define on_click event for button '4'

 # It will be triggered as soon as button '4' is pressed and shows number on main screen

In [8]:

```
# Define on_click event for button 4
def on_clicked_b4(b):
    global dot_flag
    global result_flag
    if result_flag == True:
        main_screen.value = ""
        sub_screen.value = ""
        result_flag = False
```

```
    if main_screen.value == "" or float(main_screen.value) == 0 and dot_flag == False:
        main_screen.value = "4"
    else:
        main_screen.value = main_screen.value + "4"

button4.on_click(on_clicked_b4)
```

 # Define on_click event for button '5'

 # It will be triggered as soon as button '5' is pressed and shows number on main screen

In [9]:

```
# Define on_click event for button 5
def on_clicked_b5(b):
    global dot_flag
    global result_flag
    if result_flag == True:
        main_screen.value = ""
        sub_screen.value = ""
        result_flag = False
    if main_screen.value == "" or float(main_screen.value) == 0 and dot_flag == False:
        main_screen.value = "5"
    else:
        main_screen.value = main_screen.value + "5"

button5.on_click(on_clicked_b5)
```

 # Define on_click event for button '6'

 # It will be triggered as soon as button '6' is pressed and shows number on main screen

In [10]:

```
# Define on_click event for button 6
def on_clicked_b6(b):
    global dot_flag
    global result_flag
    if result_flag == True:
        main_screen.value = ""
        sub_screen.value = ""
        result_flag = False
    if main_screen.value == "" or float(main_screen.value) == 0 and dot_flag == False:
        main_screen.value = "6"
    else:
        main_screen.value = main_screen.value + "6"

button6.on_click(on_clicked_b6)
```

 # Define on_click event for button '7'

 # It will be triggered as soon as button '7' is pressed and shows number on main screen

In [11]:

```
# Define on_click event for button 7
def on_clicked_b7(b):
    global dot_flag
    global result_flag
    if result_flag == True:
        main_screen.value = ""
        sub_screen.value = ""
        result_flag = False
    if main_screen.value == "" or float(main_screen.value) == 0 and dot_flag == False:
        main_screen.value = "7"
    else:
        main_screen.value = main_screen.value + "7"

button7.on_click(on_clicked_b7)
```

 # Define on_click event for button '8'

 # It will be triggered as soon as button '8' is pressed and shows number on main screen

In [12]:
```
# Define on_click event for button 8
def on_clicked_b8(b):
    global dot_flag
    global result_flag
    if result_flag == True:
        main_screen.value = ""
        sub_screen.value = ""
        result_flag = False
    if main_screen.value == "" or float(main_screen.value) == 0 and dot_flag == False:
        main_screen.value = "8"
    else:
        main_screen.value = main_screen.value + "8"

button8.on_click(on_clicked_b8)
```

 # Define on_click event for button '9'

 # It will be triggered as soon as button '9' is pressed and shows number on main screen

In [13]:
```
# Define on_click event for button 9
def on_clicked_b9(b):
    global dot_flag
    global result_flag
    if result_flag == True:
        main_screen.value = ""
        sub_screen.value = ""
        result_flag = False
    if main_screen.value == "" or (float(main_screen.value) == 0 and dot_flag == False):
        main_screen.value = "9"
    else:
        main_screen.value = main_screen.value + "9"

button9.on_click(on_clicked_b9)
```

 # Define on_click event for button 'AC'

 # It will be triggered as soon as button 'AC' is pressed and clear all screens and variables

In [14]:
```
# Define on_click event for button Clr all
def on_clicked_cla(b):
    global dot_flag
    global first_num
    global second_num
    main_screen.value = ""
    sub_screen.value = ""
    first_num = 0
    second_num = 0
    dot_flag = False

button_cla.on_click(on_clicked_cla)
```

 # Define on_click event for button 'x^2'

 # It will be triggered as soon as button 'x^2' is pressed and

 # perform second power calculation of input number and shows on screens

In [15]:
```
# Define on_click event for button x^2
def on_clicked_pw(b):
    global dot_flag
```

```
    global first_num
    global oper
    global result_flag
    dot_flag = False
    first_num = float(main_screen.value)
    sub_screen.value = main_screen.value + " * " + main_screen.value
    main_screen.value = str(first_num ** 2)
    first_num = 0
    oper = ""
    result_flag = True

button_pw.on_click(on_clicked_pw)
```

 # Define on_click event for button 'Sqrt

 # It will be triggered as soon as button 'Sqrt' is pressed and

 # perform square root calculation of input number and shows on screens

In [16]:

```
# Define on_click event for button sqrt
def on_clicked_sqrt(b):
    global dot_flag
    global first_num
    global oper
    global result_flag
    dot_flag = False
    first_num = float(main_screen.value)
    sub_screen.value = "sqrt" + '(' + main_screen.value + ')'
    main_screen.value = str(sqrt(first_num))
    first_num = 0
    oper = ""
    result_flag = True

button_sqrt.on_click(on_clicked_sqrt)
```

 # Define on_click event for button '.'

 # It will be triggered as soon as button '.' is pressed to show decimal number

In [17]:

```
# Define on_click event for button dot
dot_flag = False
def on_clicked_dot(b):
    global dot_flag
    global result_flag
    if result_flag == True:
        main_screen.value = ""
        sub_screen.value = ""
        result_flag = False
    if main_screen.value == "":
        main_screen.value = "0."
        dot_flag = True
    elif main_screen.value != "" and dot_flag == False:
        main_screen.value = main_screen.value + "."
        dot_flag = True

button_dot.on_click(on_clicked_dot)
```

 # Define variables for final calculation

In [18]:

```
# Define global var for calculation
first_num = 0    # store complete first number when any operation buttons pressed
second_num = 0   # store complere second number when equal button pressed
oper = "" # used as a flag to know which operation is chosen: "", "div", "mul", "min", "p
```

 # Define on_click event for button '/'

 # It will be triggered as soon as button '/' is pressed

 # It then get complete number and assign to first_num variable

 # Shows this number together with / on sub screen

 # And mark operation as 'div'

In [19]:

```
# Define on_click event for button /
def on_clicked_div(b):
    global first_num
    global dot_flag
    global oper
    dot_flag = False
    if oper == "":
        if main_screen.value == "":
            first_num = 0
            sub_screen.value = "0 / "
        else:
            first_num = float(main_screen.value)
            sub_screen.value = main_screen.value + " / "
            main_screen.value = ""
    else:
        if first_num == 0:
            sub_screen.value = "0 / "
        else:
            sub_screen.value = sub_screen.value[:len(sub_screen.value)-3] + " / "
    oper = "div"

button_div.on_click(on_clicked_div)
```

 # Define on_click event for button 'x'

 # It will be triggered as soon as button 'x' is pressed

 # It then get complete number and assign to first_num variable

 # Shows this number together with x on sub screen

 # And mark operation as 'mul'

In [20]:

```
# Define on_click event for button x
def on_clicked_multi(b):
    global first_num
    global dot_flag
    global oper
    dot_flag = False
    if oper == "":
        if main_screen.value == "":
            first_num = 0
            sub_screen.value = "0 x "
        else:
            first_num = float(main_screen.value)
            sub_screen.value = main_screen.value + " x "
            main_screen.value = ""
    else:
        if first_num == 0:
            sub_screen.value = "0 x "
        else:
            sub_screen.value = sub_screen.value[:len(sub_screen.value)-3] + " x "
    oper = "mul"

button_multi.on_click(on_clicked_multi)
```

 # Define on_click event for button '-'

 # It will be triggered as soon as button '-' is pressed

 # It then get complete number and assign to first_num variable

 # Shows this number together with - on sub screen

 # And mark operation as 'min'

In [21]:

```
# Define on_click event for button -
def on_clicked_minus(b):
    global first_num
    global dot_flag
    global oper
    dot_flag = False
    if oper == "":
        if main_screen.value == "":
            first_num = 0
            sub_screen.value = "0 - "
        else:
            first_num = float(main_screen.value)
            sub_screen.value = main_screen.value + " - "
            main_screen.value = ""
    else:
        if first_num == 0:
            sub_screen.value = "0 - "
        else:
            sub_screen.value = sub_screen.value[:len(sub_screen.value)-3] + " - "
    oper = "min"

button_minus.on_click(on_clicked_minus)
```

 # Define on_click event for button '+'

 # It will be triggered as soon as button '+' is pressed

 # It then get complete number and assign to first_num variable

 # Shows this number together with + on sub screen

 # And mark operation as 'plus'

In [22]:

```
# Define on_click event for button +
def on_clicked_plus(b):
    global first_num
    global dot_flag
    global oper
    dot_flag = False
    if oper == "":
        if main_screen.value == "":
            first_num = 0
            sub_screen.value = "0 + "
        else:
            first_num = float(main_screen.value)
            sub_screen.value = main_screen.value + " + "
            main_screen.value = ""
    else:
        if first_num == 0:
            sub_screen.value = "0 + "
        else:
            sub_screen.value = sub_screen.value[:len(sub_screen.value)-3] + " + "
    oper = "plus"

button_plus.on_click(on_clicked_plus)
```

 # Define on_click event for button '='

 # It will be triggered as soon as button '=' is pressed

 # It then get complete number and assign to second_num variable

 # Shows 2 numbers together with operation on sub screen

 # Perform final calculation based on chosen operation

 # Show the result to main screen

In [23]:

```
# Define on_click event for button =
result_flag = False
def on_clicked_resl(b):
    global first_num
    global second_num
    global dot_flag
    global oper
    global result_flag
    dot_flag = False
    if oper != "":
        if main_screen.value == "":
            second_num = 0
            sub_screen.value = sub_screen.value + "0"
        else:
            second_num = float(main_screen.value)
            sub_screen.value = sub_screen.value + main_screen.value

        if oper == "div":
            if second_num == 0:
                main_screen.value = "Error"
            else:
                main_screen.value = str(first_num / second_num)
        elif oper == "mul":
            main_screen.value = str(first_num * second_num)
        elif oper == "min":
            main_screen.value = str(first_num - second_num)
        elif oper == "plus":
            main_screen.value = str(first_num + second_num)
        first_num = 0
        second_num = 0
        oper = ""
        result_flag = True

button_equal.on_click(on_clicked_resl)
```

#Create GridBox to make a layout for calculator

In [24]:

```
# Create grid box container to host all buttons
display(GridBox(children=[button1, button2, button3, button4, button5, button6, button7,
                  button8, button9, button0, button_cla, button_pw, button_sqrt,
                  button_div, button_multi, button_minus, button_plus, button_equal,
                  button_dot, main_screen, sub_screen],
        layout=Layout(
            grid_template_rows='18px 30px 50px 50px 50px 50px 50px',
            grid_template_columns='70px 70px 70px 70px',
            grid_gap='10px 10px',
            grid_template_areas='''
            "sub sub sub sub"
            "disp disp disp disp"
            "cla pw sqrt div"
            "b7 b8 b9 multi"
            "b4 b5 b6 minus"
            "b1 b2 b3 plus"
            "b0 dot equal equal"
            ''')
       ))
```