


User's guide for U-value Calculator

Environment dependencies:

Code Runner



Code Runner

v0.12.0


Jun Han | 17,215,438 | ★★★★★ (242) | ❤️ 发起人

Run C, C++, Java, JS, PHP, Python, Perl, Ruby, Go, Lua, Groovy, PowerShell, CMD, BASH, F#, C#, VBScript, TypeScript, Coffe...

禁用 卸载 设置

此扩展已全局启用。

Jupyter



Jupyter

v2022.11.1003412109


Microsoft | 56,331,989 | ★★★★★ (258)

Jupyter notebook support, interactive programming and computing that supports Intellisense, debugging and more.

禁用 卸载 切换到预发布版本 设置

此扩展已全局启用。

Python



Python

v2022.20.2

Microsoft | 75,310,789 | ★★★★★ (525)

IntelliSense (Pylance), Linting, Debugging (multi-threaded, remote), Jupyter Notebooks, code formatting, refactoring, unit ...

禁用 卸载 切换到预发布版本 设置

此扩展已全局启用。

Tkinter Snippets



Tkinter Snippets

v2.0.2


Nikola Paunović | 90,458 | ★★★★★ (2)

Snippet pack for Tkinter module of Python.


禁用 卸载 设置

此扩展已全局启用。

Calculator.py

 Calculator.py

GUI main.py

 GUI main.py

Code Introduction :

In Calculator.py file , there are two main functions, namely 'read_and_calculate' and 'U_value_2'.

```
def read_and_calculate(K_values, ds, n):
    ##### This is variable for the GUI how many variable the user type
    ##### create two dicts for K and d
    dict_K = {} ##### the dict for K-value
    dict_d = {} ##### the dict for Material Thickness
    i = 0 ##### i the number in dic_K and dic_d
    x = 0 ##### x the number in K_values and ds
    while i < n: ##### This loop is to add the data to the dictionary
        dict_K[i] = K_values[x]
        dict_d[i] = ds[x]
        i = i + 1
        x = x + 1
    U_value_2(dict_K, dict_d, n)
    return U_value_2(dict_K, dict_d, n)
```

In 'read_and_calculate', there are two dictionaries, 'dict_k' and 'dict_d', storing the K-value and thickness of the material separately. And then we use loop to add the data to the dictionaries. After that, we return 'U_value_2(dict_K, dict_d, n)'.

```
def U_value_2(dict_K, dict_d, n):
    i = 0
    R_total = 0      ##### the original R-value is 0
    while i < n:
        R_total += dict_d[i]/dict_K[i]
        ##### Thermal Resistance =
        ##### Material Thickness / Thermal Conductivity
        ##### to add each layers up
        ##### is the total thermal resistances
        i = i + 1
    U_value_1 = 1/(R_total)
    ##### The U-value of a building element
    ##### is the inverse of the total thermal resistances
    ##### of the different layers making up the building element.
    return U_value_1
```

In 'U_value_2', we set the original R-value(R_{total}) is 0, then use 'Thermal Resistance = Material Thickness / Thermal Conductivity' equation and loop to calculate each layer's R-value. After add them up, we get the total R-value of the building element. Then, the inverse of the total thermal resistances is the total U-value. Finally, we return the 'U value 1'.

In GUI main.py file, we design the initial window first.

```
import tkinter as tk
# import the Tkinter library
from calculator import read_and_calculate
# introduce 'readread_and_calculate' in the 'calculator' module
import tkinter.messagebox

win=tk.Tk()    # creating the main window and storing the window object in 'win'
win.title('This is our own software!')    # setting title of the window
win.geometry('600x200')    # setting the size of the window
def start():    # def the function of the button
    win.destroy()
    win = tk.Tk()
    win.geometry("600x400")
    tkinter.messagebox.showinfo("Greetings","\n
    Welcome to U-value Calculator! \n
    Authors: Liyang YU and Zesheng YANG")

# setting the size of main window
lab = tk.Label(win, text = 'Please choose the number of layers of building material you want to analyze',
               font=('Arial',12,'bold'), width = 80, height = 13, bg = 'light blue')
lab.pack()
```

Initially, we import the Tkinter library, introduce the 'read_and_calculate' in the 'Calculator' module and import tkinter.messagebox to make the initial window, including the title, size. Then we define 'start' function of the button to realize the function of greeting and simple introduction.

When enter the main windows, there are four functions of the buttons, 'run_3', 'run_4', 'run_5' and 'run_6'. Since the functions of these buttons are similar, the 'run_3' will be taken as an example to introduce.

```
23     def run_3():
24         win.destroy()
25         # as soon as button 3 is pressed
26         # window will be destroyed
27         root=tk.Tk()
28     # setting the windows' name
29         root.title("U_value Calculator")
30     # setting the windows size
31         root.geometry("600x400")
```

Firstly, we define the 'run_3' function of the '3' button. We set the title and size of it and let the window disappear when it is clicked.

```
32
33     # declaring string variable
34     # for storing name and password
35         K_value_var=tk.StringVar()
36         K_value_1_var=tk.StringVar()
37         K_value_2_var=tk.StringVar()
38         d_var=tk.StringVar()
39         d_1_var=tk.StringVar()
40         d_2_var=tk.StringVar()
```

Secondly, we declare the string variable to store the name and password.

```
41
42     # defining a function that will
43     # get the d and K_value
44     # print them on the screen
45     def submit():
46         K_values=[float(K_value_var.get()), float(K_value_1_var.get()), float(K_value_2_var.get())]
47         ds=[float(d_var.get()), float(d_1_var.get()), float(d_2_var.get())]
48         n = 3
49         U_value = read_and_calculate(K_values, ds, n)
50
51         print(U_value)
52         tk.Label(root, text = U_value).grid(row = 4, column = 2)
53
```

Thirdly, we define another function 'submit' to get the d and K-value and print them on the screen.

```

54 # creating a label for
55 # name using widget Label
56 K_value_label = tk.Label(root, text = 'K_1', font=('calibre',10, 'bold'))
57 K_value_1_label = tk.Label(root, text = 'K_2', font=('calibre',10, 'bold'))
58 K_value_2_label = tk.Label(root, text = 'K_3', font=('calibre',10, 'bold'))
59 # creating a entry for input
60 # name using widget Entry
61 K_value_entry = tk.Entry(root,textvariable = K_value_var, font=('calibre',10,'normal'))
62 K_value_1_entry = tk.Entry(root,textvariable = K_value_1_var, font=('calibre',10,'normal'))
63 K_value_2_entry = tk.Entry(root,textvariable = K_value_2_var, font=('calibre',10,'normal'))
64 # creating a label for password
65 d_label = tk.Label(root, text = 'd_1', font = ('calibre',10,'bold'))
66 d_1_label = tk.Label(root, text = 'd_2', font = ('calibre',10,'bold'))
67 d_2_label = tk.Label(root, text = 'd_3', font = ('calibre',10,'bold'))
68 # creating a entry for password
69 d_entry=tk.Entry(root, textvariable = d_var, font = ('calibre',10,'normal'))
70 d_1_entry=tk.Entry(root, textvariable = d_1_var, font = ('calibre',10,'normal'))
71 d_2_entry=tk.Entry(root, textvariable = d_2_var, font = ('calibre',10,'normal'))
72 # creating the result Label
73 result_Label = tk.Label(root, text = 'result', font=('calibre',10, 'bold'))
74 # creating a button using the widget
75 # Button that will call the submit function
76 sub_btn=tk.Button(root,text = 'Submit', command = submit)
77

```

Fourthly, we create the labels and entries for K-value and d respectively. And there is a 'submit button' used to calculate the whole U-value.

```

78 # placing the label and entry in
79 # the required position using grid
80 # method
81 K_value_label.grid(row=0,column=0)
82 K_value_1_label.grid(row=1,column=0)
83 K_value_2_label.grid(row=2,column=0)
84
85 K_value_entry.grid(row=0,column=1)
86 K_value_1_entry.grid(row=1,column=1)
87 K_value_2_entry.grid(row=2,column=1)
88
89 d_label.grid(row=0,column=2)
90 d_1_label.grid(row=1,column=2)
91 d_2_label.grid(row=2,column=2)
92
93 d_entry.grid(row=0,column=3)
94 d_1_entry.grid(row=1,column=3)
95 d_2_entry.grid(row=2,column=3)
96
97 sub_btn.grid(row=5,column=3)
98 result_Label.grid(row=4,column=1)
99

```

Fifthly, there are the codes making the labels and entries in the required position.

```

100     def close_1():      # to quit the whole window
101         #this func is for unintended activation
102         res = tkinter.messagebox.askquestion('Exit the calculator',
103                                             'Are you sure to exit')
104         if res == 'yes':
105             root.destroy()
106
107         else :
108             None
109         Button = tk.Button(root, text = 'Quit', width=10, height=5, command = close_1)
110         Button.place(x=400,y=20)
111         root.mainloop()
112
113     # performing an infinite loop
114     # for the window to display
115     root.mainloop()
116     return

```

Finally, we define 'close_1' function to quit the whole window.

```

502     choose_but_1 = tk.Button(win, text = '3', width=10, height=5, command = run_3)
503     choose_but_2 = tk.Button(win, text = '4', width=10, height=5, command = run_4)
504     choose_but_3 = tk.Button(win, text = '5', width=10, height=5, command = run_5)
505     choose_but_4 = tk.Button(win, text = '6', width=10, height=5, command = run_6)
506     choose_but_1.place(relx = 0.1, rely = 0.7)
507     choose_but_2.place(relx = 0.3, rely = 0.7)
508     choose_but_3.place(relx = 0.5, rely = 0.7)
509     choose_but_4.place(relx = 0.7, rely = 0.7)
510     win.mainloop()
511
512     btn= tk.Button(win1,text="Click Me To Know More infomation", width=50, height=5, font=('Arial',10,'bold'), command=start,
513                   bg='yellow',activebackground='light yellow')
514     btn.place(x=95, y=50)
515     win1.mainloop()      # running the loop that works as a trigger
516

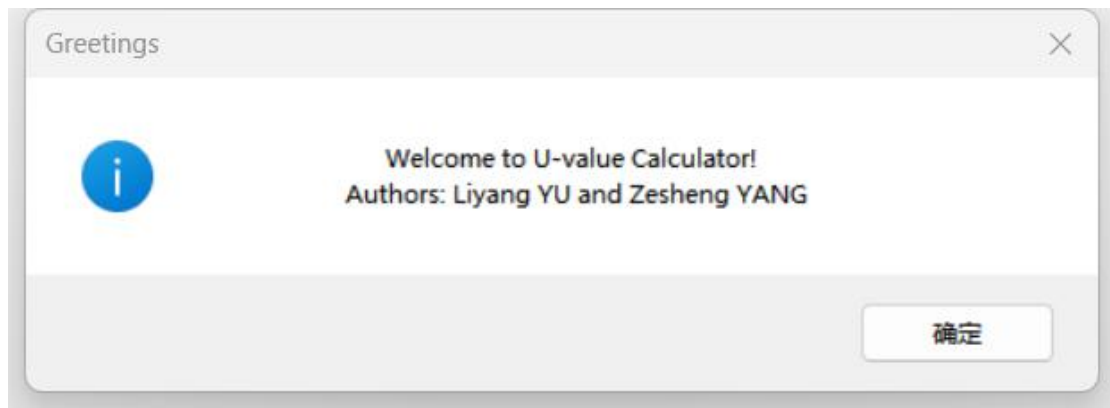
```

Here are all the parameters associated with the button.

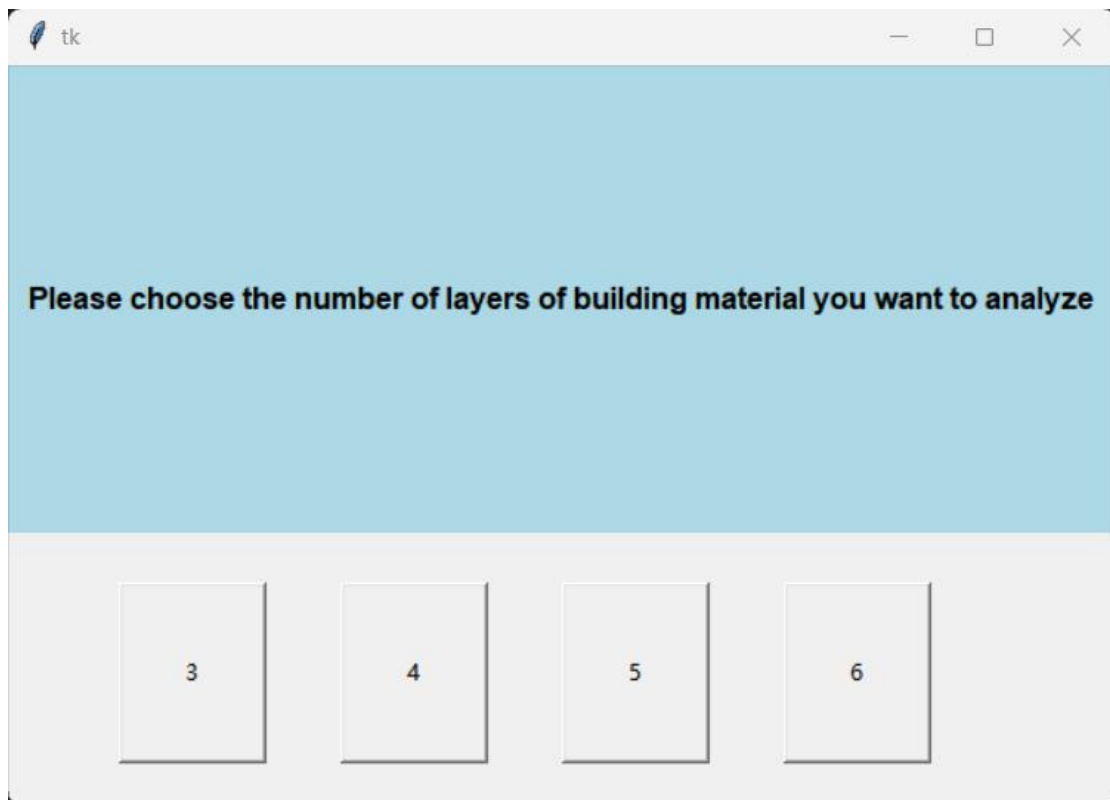
Operation demonstration :



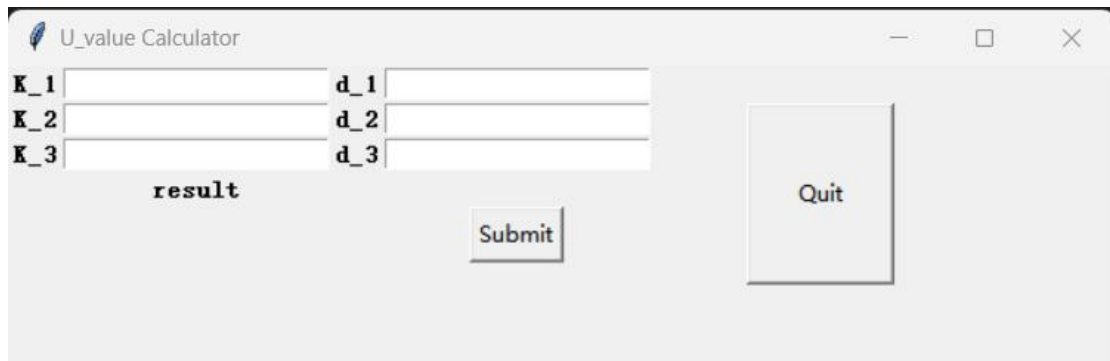
When you run the code, you will see the window like this, and there is a button 'Click Me To Know More information'. After you click it, another window will open.



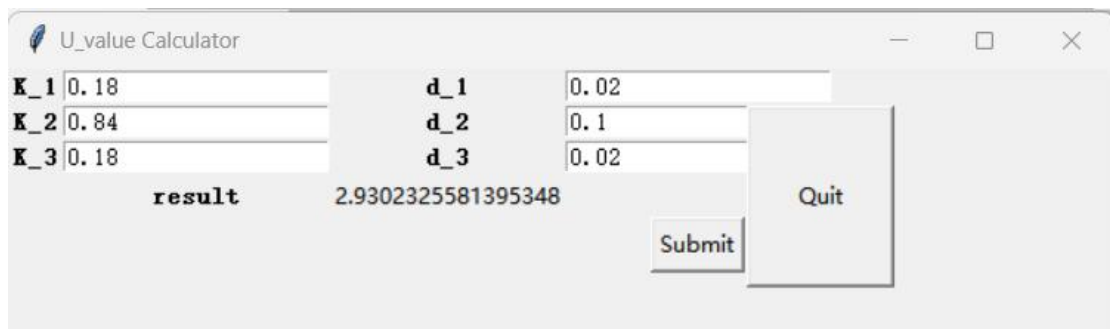
This is a 'Greetings' window, mainly responsible for introducing the name and authors of our software. And when you click the button ,this window will disappear and the main window will show up as below.



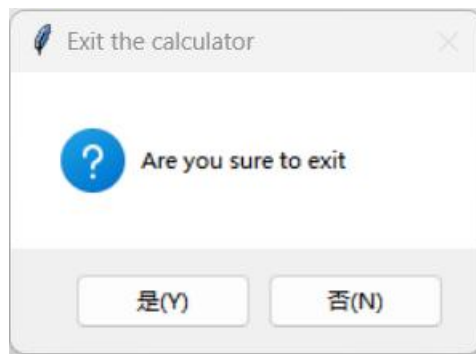
This is the main window of our U-value calculator. There is a label telling you 'please choose the number of layers of building material you want to analyze'. Then you can choose the corresponding button according to your own needs according to your needs. Here we take the '3' button as an example. When you click the button, it will jump automatically.



Then, you can type in the data and click the 'Submit' button. As shown in the picture below, it will show the result.



After you get the result, you can choose to continue using the software or quit. When you choose to quit, you can click the 'Quit' button.



Then, the software will confirm to you if you are sure to exit. If you want to leave, just click 'Y' is OK. This window will disappear automatically.