PYTHON

PROJECT REPORT

OF

INDUSTRIAL TRAINING

UNDER A SUPERVISION OF:

MS. MUSKAN

SUBMITED BY: SUBMITED TO:

MANISH KUMAR KRISHAN KUMAR.

DEPARTMENT OF GOVT. POLYTECNIC, SIRSA(HARYANA)

In these 45 days of training, we learn how to make a calculator.

C-110,INDUSTRIAL AREA,PHASE 7, MOHALI,PUNJAB-160055

PYTHON

python is a high-level, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation.

Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly procedural), object-oriented and functional programming. It is often described as a "batteries included" language due to its comprehensive standard library.

Guido van Rossum began working on Python in the late 1980s as a successor to the ABC programming language and first released it in 1991 as Python 0.9.0. Python 2.0 was released in 2000. Python 3.0, released in 2008, was a major revision not completely backward-compatible with earlier versions. Python 2.7.18, released in 2020, was the last release of Python ..Python consistently ranks as one of the most popular programming languages

INDENTATION

Python uses whitespace indentation, rather than curly brackets or keywords, to delimit blocks. An increase in indentation comes after certain statements; a decrease in indentation signifies the end of the current block. Thus, the program's visual structure accurately represents its semantic structure.[83] This feature is sometimes termed the off-side rule. Some other languages use indentation this way; but in most, indentation has no semantic meaning. The recommended indent size is four spaces

SYNTAX

Python is meant to be an easily readable language. Its formatting is visually uncluttered and often uses English keywords where other languages use punctuation. Unlike many other languages, it does not use curly brackets to delimit blocks, and semicolons after statements are allowed but rarely used. It has fewer syntactic exceptions and special cases than C or Pascal.

Calculator Program in Python – Python Calculator Project

Work on Python Calculator Project and get ready for a boost in your career

The calculator is one application that we all use in our day to day lives. If you are trying to get your hands dirty with programming in python, Calculator is a project which is easy and useful at the same time. Today, we are going to build a Python Calculator using Tkinter with easy to understand steps.

What is Tkinter?

Python offers various utilities to design the GUI wiz Graphical User Interface, and one such utility is Tkinter which is most commonly used. It is indeed one of the fastest and easiest ways to build GUI application. Moreover, Tkinter is cross-platform, hence the same code works on macOS, Windows, and Linux.

Step 1: Importing the necessary modules

To use the Tkinter we need to import the Tkinter module. We are also going to import the function factorial from math module.

**Code:**

*1. from tkinter import \**

*2. import parser*

*3.from math import factorial*

Step 2: Making a window for our calculator

Now we are going to draft the window for our calculator which will accommodate the buttons.

**Code:**

**1. root = Tk()**

**2. root.title('DataFlair - Calculator')**

**3. root.mainloop()**

**Explanation :The above code sets the title of python calculator window as ‘Data Flair – Calculator’. When you run the above code, you will get a window like this.**

Step 3: Designing the buttons

Now let’s quickly design the buttons for our calculator and put them on our application window.

**Code:**

1 #adding the input field

2 display = Entry(root)

3 display.grid(row=1,columnspan=6,sticky=N+E+W+S)

4 #Code to add buttons to the Calculator

5 Button(root,text="1",command = lambda :get\_variables(1)).grid(row=2,column=0, sticky=N+S+E+W)

6 Button(root,text=" 2",command = lambda :get\_variables(2)).grid(row=2,column=1, sticky=N+S+E+W)

7 Button(root,text=" 3",command = lambda :get\_variables(3)).grid(row=2,column=2, sticky=N+S+E+W)

8 Button(root,text="4",command = lambda :get\_variables(4)).grid(row=3,column=0, sticky=N+S+E+W)

9 Button(root,text=" 5",command = lambda :get\_variables(5)).grid(row=3,column=1, sticky=N+S+E+W)

10 Button(root,text=" 6",command = lambda :get\_variables(6)).grid(row=3,column=2, sticky=N+S+E+W)

11 Button(root,text="7",command = lambda :get\_variables(7)).grid(row=4,column=0, sticky=N+S+E+W)

12 Button(root,text=" 8",command = lambda :get\_variables(8)).grid(row=4,column=1, sticky=N+S+E+W)

13 Button(root,text=" 9",command = lambda :get\_variables(9)).grid(row=4,column=2, sticky=N+S+E+W)

14 #adding other buttons to the calculator

15 Button(root,text="AC",command=lambda :clear\_all()).grid(row=5,column=0, sticky=N+S+E+W)

16Button(root,text=" 0",command = lambda :get\_variables(0)).grid(row=5,column=1, sticky=N+S+E+W)

17 Button(root,text=" .",command=lambda :get\_variables(".")).grid(row=5, column=2, sticky=N+S+E+W)

18 Button(root,text="+",command= lambda :get\_operation("+")).grid(row=2,column=3, sticky=N+S+E+W)

19 Button(root,text="-",command= lambda :get\_operation("-")).grid(row=3,column=3, sticky=N+S+E+W)

20 Button(root,text="\*",command= lambda :get\_operation("\*")).grid(row=4,column=3, sticky=N+S+E+W)

21 Button(root,text="/",command= lambda :get\_operation("/")).grid(row=5,column=3, sticky=N+S+E+W)

22 # adding new operations

23 Button(root,text="pi",command= lambda :get\_operation("\*3.14")).grid(row=2,column=4, sticky=N+S+E+W)

24 Button(root,text="%",command= lambda :get\_operation("%")).grid(row=3,column=4, sticky=N+S+E+W)

25 Button(root,text="(",command= lambda :get\_operation("(")).grid(row=4,column=4, sticky=N+S+E+W)

26 Button(root,text="exp",command= lambda :get\_operation("\*\*")).grid(row=5,column=4, sticky=N+S+E+W)

27 Button(root,text="<-",command= lambda :undo()).grid(row=2,column=5, sticky=N+S+E+W)

28 Button(root,text="x!", command= lambda: fact()).grid(row=3,column=5, sticky=N+S+E+W)

29 Button(root,text=")",command= lambda :get\_operation(")")).grid(row=4,column=5, sticky=N+S+E+W)

30 Button(root,text="^2",command= lambda :get\_operation("\*\*2")).grid(row=5,column=5, sticky=N+S+E+W)

31 Button(root,text="^2",command= lambda :get\_operation("\*\*2")).grid(row=5,column=5, sticky=N+S+E+W)

32 Button(root,text="=",command= lambda :calculate()).grid(columnspan=6, sticky=N+S+E+W)

Explanation:

In this calculator program in python, the “Entry” function helps in making a text input field and we use .grid() method to define the positioning associated with the button or input field. We use the button method to display a button on our application window.

root – the name with which we refer to our window

text – text to be displayed on the button

row – row index of the grid

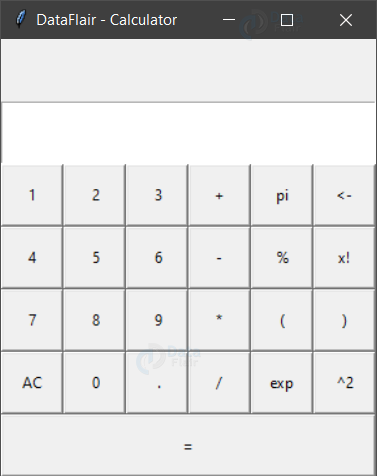
column – column index of the grid

columnspan – spans or combines the number of columns

sticky – If the resulting cell is larger than the widget then sticky defines how to expand the widget. The combination of constants used S, N, E, and W, or NW, NE, SW, and SE are analogous to the directions in compass. N+E+W+S means that the widget should be expanded in all directions

When you run the above code, you will get calculator output like this

OUTPUT:



NOTE:

If you have noticed an error such as no function get variable() found, then do not worry. We are just defining an action function associated with each button. Since we have just called and not declared them yet hence the error. Now, lets defined those functions.