

PythonBooklet2



TKinter Basics

Random Class

If Conditions

While Loops

Functions

Images in Buttons

Activate/Disable Buttons

TKinter Basics

```
import tkinter

def printName():
    nameGot = nameEntryField.get()
    print(nameGot)

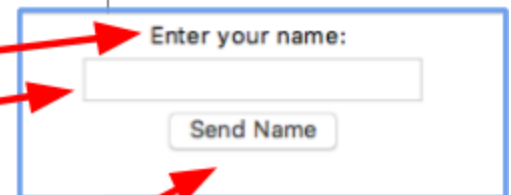
#WINDOW CALLED "window" - creates a new windows named "window"
window = tkinter.Tk()
window.title("My Window Title")
window.geometry("400x400")

#LABEL - like a text box
labelName = tkinter.Label(window, text="Enter your name: ")
labelName.pack()

#TEXT FIELD - user can enter text in these
nameEntryField = tkinter.Entry(window)
nameEntryField.pack()

#BUTTON - runs the "printName" method (top)
addButton = tkinter.Button(window, text="Send Name", command=printName)
addButton.pack()

#DRAW EVERYTHING IN A "window" - this always goes at the end
window.mainloop()
|
```



Random Class

```
import random

ranNum = random.randint(1, 10)
```

Generate a random integer (whole number) between 1 and 10, storing it under the "ranNum" variable name.

If Conditions

```
if userGuess < ranNum:  
    print("Higher")  
  
elif userGuess > ranNum:  
    print("Lower")  
  
elif userGuess == ranNum:  
    print("Correct! Great Job!")
```

1. If the “userGuess” variable variable is less than the “ranNum” variable, then print Higher to the terminal.
 2. ELSE IF it is greater, print Lower to the terminal
 3. ELISE IF it equals print Correct! Great Job to the terminal.
-

While Loops

```
while guessCounter < 5:  
    guessCounter = guessCounter + 1  
    userGuess = int(input("Guess a number: "))  
    if userGuess < ranNum:  
        print("Higher")  
    elif userGuess > ranNum:  
        print("Lower")  
    elif userGuess == ranNum:  
        print("Correct! Great Job!")  
        break
```

While “guessCounter” variable is less than 5, keep running the code below, adding one to the “guessCounter” variable each time.
break when it is not less than 5, also when the “userGuess” variable equals “ranNum” variable.

Functions

```
def numGameFunciton():  
    guessCounter = 0  
  
    while guessCounter < 5:  
        guessCounter = guessCounter + 1  
        userGuess = int(input("Guess a number: "))  
        if userGuess < ranNum:  
            print("Higher")  
        elif userGuess > ranNum:  
            print("Lower")  
        elif userGuess == ranNum:  
            print("Correct! Great Job!")  
            break  
  
numGameFunciton()
```

numGameFunction() runs all of the code indented under def numGameFunction():

Images in Buttons



Stores the load PhotoImage inside “loadTickPhoto” variable
Apply it to the button using the image condition (instead of text=’’) *Note: The Photo HAS to be in the same folder as the Python project.*

Activate/Disable Buttons

```
addButton.config(state="disabled")
```

The above line disables “addButton”

```
addButton.config(state="normal")
```

The above code reactivates “addButton”

Open/Closing Windows

```
import tkinter

#Open Window 3 Function
def openWindow3():
    openWindow2.label2.pack_forget()
    openWindow2.window2Button.pack_forget()

    openWindow3.label3 = tkinter.Label(openWindow1.window, text="Window 3")
    openWindow3.label3.pack()

#Open Window 2 Function
def openWindow2():
    openWindow1.label.pack_forget()
    openWindow1.windowButton.pack_forget()

    openWindow2.label2 = tkinter.Label(openWindow1.window, text="Window 2")
    openWindow2.label2.pack()

    openWindow2.window2Button = tkinter.Button(openWindow1.window, text="Go to W
    openWindow2.window2Button.pack()

#Open Window Function
def openWindow1():
    openWindow1.window = tkinter.Tk()
    openWindow1.window.title("Window 1")
    openWindow1.window.geometry("400x400")

    openWindow1.label = tkinter.Label(openWindow1.window, text="Window 1")
    openWindow1.label.pack()

    openWindow1.windowButton = tkinter.Button(openWindow1.window, text="Go to Wi
    openWindow1.windowButton.pack()

    openWindow1.window.mainloop()

openWindow1()
```

Always call variables with the function name they're in declared in before. e.g.

“funcitonName.VariableName

use “pack_forget()” to essential unpack the objects (textfiels/entrys, labels, buttons etc...)