

ChatGPT Python

⚡ Default (GPT-3.5)

IN

Skriv et python script der fungerer som en normal terning. Det skal have en GUI med en knap der kaster "terningen" og kastets resultat nedenunder.



Selvfølgelig! Her er et enkelt eksempel på et Python-script ved hjælp af biblioteket Tkinter for at oprette en GUI, der fungerer som en normal terning:



python

Copy code

```
import tkinter as tk
import random

class DiceRollerApp:
    def __init__(self, root):
        self.root = root
        self.root.title("Dice Roller")

        self.result_label = tk.Label(root, text="", font=("Helvetica", 24))
        self.result_label.pack(pady=20)

        self.roll_button = tk.Button(root, text="Roll Dice", command=self.roll_dice)
        self.roll_button.pack()

    def roll_dice(self):
        result = random.randint(1, 6)
        self.result_label.config(text=f"Result: {result}")

if __name__ == "__main__":
    root = tk.Tk()
    app = DiceRollerApp(root)
```

Regenerate

```
self.root = root
self.root.title("Dice Roller")

self.result_label = tk.Label(root, text="", font=("Helvetica", 24))
self.result_label.pack(pady=20)

self.roll_button = tk.Button(root, text="Roll Dice", command=self.roll_dice)
self.roll_button.pack()

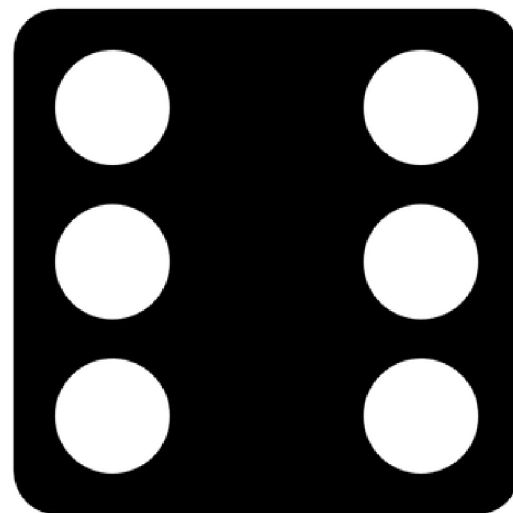
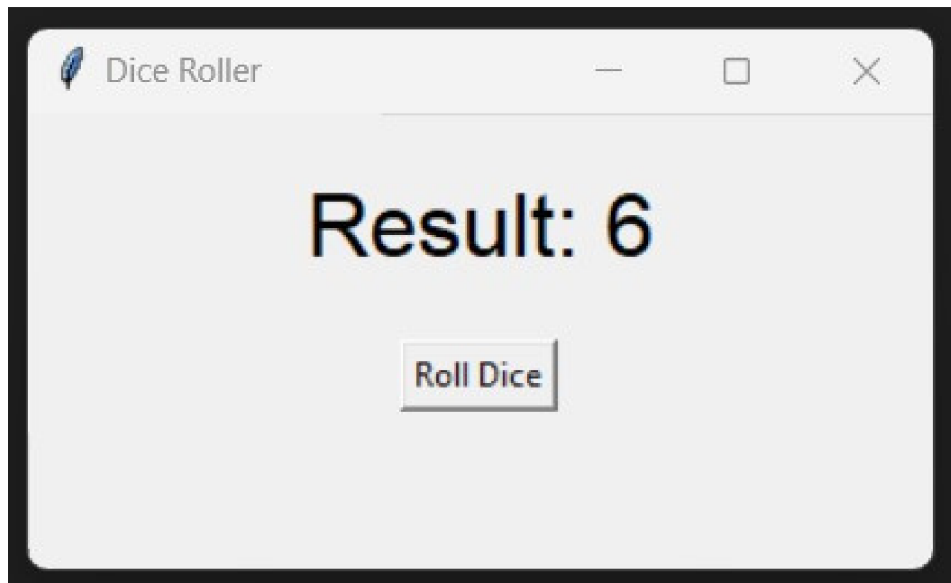
def roll_dice(self):
    result = random.randint(1, 6)
    self.result_label.config(text=f"Result: {result}")

if __name__ == "__main__":
    root = tk.Tk()
    app = DiceRollerApp(root)
    root.mainloop()
```

Dette script opretter et simpelt GUI-vindue med en knap og en etiket. Når knappen "Roll Dice" klikkes, genereres et tilfældigt tal mellem 1 og 6, som er det forventede resultat af et terningkast. Resultatet vises i etiketten nedenfor knappen.

Du skal have Tkinter installeret for at køre dette script. Du kan installere det ved at køre `pip install tk` i din terminal, hvis det ikke allerede er installeret.

Husk, dette er et grundlæggende eksempel. Du kan tilpasse det yderligere ved at tilføje flere funktioner eller designelementer efter behov.



Modified_ChatGPT.py

```
1
2 import tkinter as tk
3 from tkinter import ttk
4 import random
5
6 class DiceRollerApp:
7     def __init__(self, root):
8         self.root = root
9         self.root.title("Dice Roller")
10
11         self.result_label = tk.Label(root, text="", font=("Helvetica", 24))
12         self.result_label.pack(pady=20)
13
14         self.sides_label = tk.Label(root, text="Number of Sides:")
15         self.sides_label.pack()
16
17         self.sides_var = tk.IntVar()
18         self.sides_var.set(6) # Default value
19         self.sides_dropdown = ttk.Combobox(root, textvariable=self.sides_var, values=list(range(2, 21)))
20         self.sides_dropdown.pack()
21
22         # Adding the number of dice dropdown
23         self.num_dice_label = tk.Label(root, text="Number of Dice:")
24         self.num_dice_label.pack()
25
26         self.num_dice_var = tk.IntVar()
27         self.num_dice_var.set(1) # Default value
28         self.num_dice_dropdown = ttk.Combobox(root, textvariable=self.num_dice_var, values=list(range(1, 101)))
29         self.num_dice_dropdown.pack()
30
31         self.roll_button = tk.Button(root, text="Roll Dice", command=self.roll_dice)
32         self.roll_button.pack()
33
34     def roll_dice(self):
35         num_sides = self.sides_var.get()
36         num_dice = self.num_dice_var.get()
37
38         results = sorted([random.randint(1, num_sides) for _ in range(num_dice)], reverse=True)
39         results_str = ', '.join(map(str, results))
40         self.result_label.config(text=results_str)
41
42 if __name__ == "__main__":
43     root = tk.Tk()
44     app = DiceRollerApp(root)
45     root.mainloop()
46
```

```
Responsive_ChatGPT.py
```

```
1
2 import tkinter as tk
3 from tkinter import ttk
4 import random
5
6 class DiceRollerApp:
7     def __init__(self, root):
8         self.root = root
9         self.root.title("Dice Roller")
10
11         # Using a Text widget for displaying results with a scrollbar
12         self.result_text = tk.Text(root, height=10, width=40, font=("Helvetica", 16), wrap=tk.WORD)
13         self.result_text.pack(pady=20)
14         self.scroll = tk.Scrollbar(root, command=self.result_text.yview)
15         self.scroll.pack(side=tk.RIGHT, fill=tk.Y)
16         self.result_text.configure(yscrollcommand=self.scroll.set)
17
18         self.sides_label = tk.Label(root, text="Number of Sides:")
19         self.sides_label.pack()
20
21         self.sides_var = tk.IntVar()
22         self.sides_var.set(6) # Default value
23         self.sides_dropdown = ttk.Combobox(root, textvariable=self.sides_var, values=list(range(2, 21)))
24         self.sides_dropdown.pack()
25
26         # Number of dice dropdown
27         self.num_dice_label = tk.Label(root, text="Number of Dice:")
28         self.num_dice_label.pack()
29
30         self.num_dice_var = tk.IntVar()
31         self.num_dice_var.set(1) # Default value
32         self.num_dice_dropdown = ttk.Combobox(root, textvariable=self.num_dice_var, values=list(range(1, 101)))
33         self.num_dice_dropdown.pack()
34
35         self.roll_button = tk.Button(root, text="Roll Dice", command=self.roll_dice)
36         self.roll_button.pack()
37
38     def roll_dice(self):
39         num_sides = self.sides_var.get()
40         num_dice = self.num_dice_var.get()
41
42         results = sorted([random.randint(1, num_sides) for _ in range(num_dice)], reverse=True)
43         results_str = ', '.join(map(str, results))
44
45         # Clear the previous results and insert the new results
46         self.result_text.delete("1.0", tk.END)
47         self.result_text.insert(tk.END, results_str)
48
49 if __name__ == "__main__":
50     root = tk.Tk()
51     app = DiceRollerApp(root)
52     root.mainloop()
53
```