



Découvrir CustomTkinter

Sommaire

1. Qu'est-ce que CustomTkinter ?
2. Pourquoi utiliser ?
3. Création de votre première application CustomTkinter
4. Démo d'une application complexe (avec auth, navigation, dashboard, ...)
5. Ressources pour en savoir plus

Qui suis-je ?

-  Macktireh Abdi Soubaneh
-  Data Science > Développeur

Qu'est-ce que CustomTkinter ?

- Librairie Python développée par Tom Schimansky
- Basé sur librairie standard de Python Tkinter
- <https://github.com/tomschimansky/customtkinter> ★ 10.1K

```
pip install customtkinter
```

Pourquoi

CustomTkinter ?

- Apprentissage facile : Si vous êtes déjà familier avec Tkinter, la transition vers CustomTkinter sera transparente
- Modernité des Widgets : CustomTkinter offre des widgets modernes et entièrement personnalisables.
- Thèmes d'Apparence : Il permet de configurer facilement le mode d'apparence du GUI, avec des options comme ``système``, ``sombre`` et ``clair``.
- Compatibilité : Il peut être utilisé en combinaison avec les éléments Tkinter classiques, offrant ainsi une grande flexibilité.

Simple Application avec Tkinter

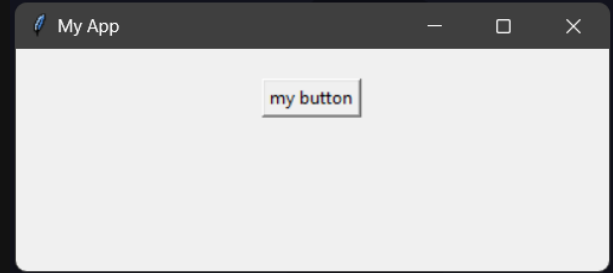
```
from tkinter import Tk, Button

class App(Tk):
    def __init__(self) -> None:
        super().__init__()
        self.title("My App")
        self.geometry("400x150")

        self.button = Button(self, text="my button", command=self.callback)
        self.button.pack(padx=20, pady=20)

    def callback(self) -> None:
        print("Hello World!")

if __name__ == "__main__":
    app = App()
    app.mainloop()
```



Avec CustomTkinter

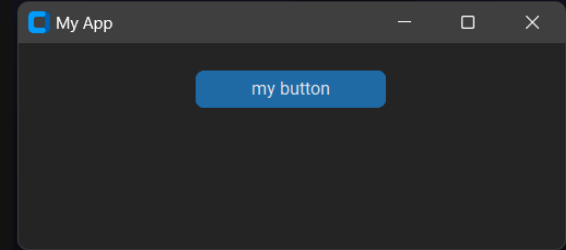
```
from customtkinter import CTk, CTkButton

class App(CTk):
    def __init__(self) → None:
        super().__init__()
        self.title("My App")
        self.geometry("400x150")

        self.button = CTkButton(self, text="my button", command=self.callback)
        self.button.pack(padx=20, pady=20)

    def callback(self) → None:
        print("Hello World!")

if __name__ == "__main__":
    app = App()
    app.mainloop()
```



Tkinter

```
from tkinter import Tk, Button

class App(Tk):
    def __init__(self) → None:
        super().__init__()
        self.title("My App")
        self.geometry("400x150")

        self.button = Button(
            self, text="my button", command=self.callback
        )
        self.button.pack(padx=20, pady=20)

    def callback(self) → None:
        print("Hello World!")

if __name__ == "__main__":
    app = App()
    app.mainloop()
```

CustomTkinter

```
from customtkinter import CTk, CTKButton

class App(CTk):
    def __init__(self) → None:
        super().__init__()
        self.title("My App")
        self.geometry("400x150")

        self.button = CTKButton(
            self, text="my button", command=self.callback
        )
        self.button.pack(padx=20, pady=20)

    def callback(self) → None:
        print("Hello World!")

if __name__ == "__main__":
    app = App()
    app.mainloop()
```


Live coding

Démo d'une application complexe

Pour en savoir plus

- Documentation officielle et tutoriel : <https://customtkinter.tomschimansky.com/>
- Tutoriel Youtube : https://www.youtube.com/playlist?list=PLfZw_tZWahjxJl81b1S-vYQwHs_9ZT77f

Merci !