Python - używanie modułu tkinter

> Brak wiary w siebie jest chorobą. Jeśli stracisz panowanie nad tym, wątpliwości staną się twoją rzeczywistością.
>
> ~John Flanagan - książka Zwiadowcy. Płonący Most

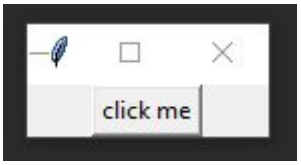
W poprzednich notatkach wykorzystywaliśmy moduł **turtle** do tworzenia prostej grafiki. Problem z naszym "żółwiem" jest taki, że jak to żółwie mają w zwyczaju jest bardzo wolny. Nawet gdy żółw osiągnie maksymalną prędkość to nie jest to bardzo szybko. Dla żółwi to nie jest problem, ale dla grafiki komputerowej już tak.

Tworzenie klikalnego przycisku

W pierwszym przykładzie z wykorzystania modułu **tkinter** stworzymy prostą aplikację z przyciskiem. Tak wygląda przykładowy kod:

```
from tkinter import *
tk = Tk()
btn = Button(tk, text="click me")
btn.pack()
tk.mainloop()
```

W pierwszym wierszu importujemy zawartość modułu tkinter. Używanie wyrażenia **from** *nazwa_modułu* **import *** pozwala nam używać zawartości modułu bez używania jego nazwy.
W kolejnym wierszu w naszym przykładzie tworzymy zmienną zawierającą obiekt klasy **Tk**. (Tak samo tworzyliśmy wcześniej "długopis" w module turtle. Wykorzystywaliśmy do tego klasę **Pen**.) Obiekt tk tworzy podstawowe okno, do którego możemy następnie dodawać inne rzeczy, takie jak przyciski, pola wprowadzania, płótno do rysowania itd. Jest to główna klasa zapewniona przed moduł **tkinter** - bez tworzenia obiektu klasy **Tk** nie będzie można wykonywać żadnych grafik.
W trzecim wierszu tworzymy przycisk. Używamy do tego klasy **Button**. Tym razem przekazujemy dwa parametry do konstruktora klasy. Pierwszym z nich jest zmienna tk, a drugim napis "click me" przekazany jako tekst, który ma zostać wyświetlony.
Pomimo stworzenia przycisku, nie pojawi się on w naszym oknie, dopóki nie dodamy: btn.pack() (w kolejnej linii), która informuje przycisk, że ma się pojawić oraz ustawia wszystko poprawnie na ekranie. Wynik tego kodu powinien wyglądać następująco (na następnej stronie):



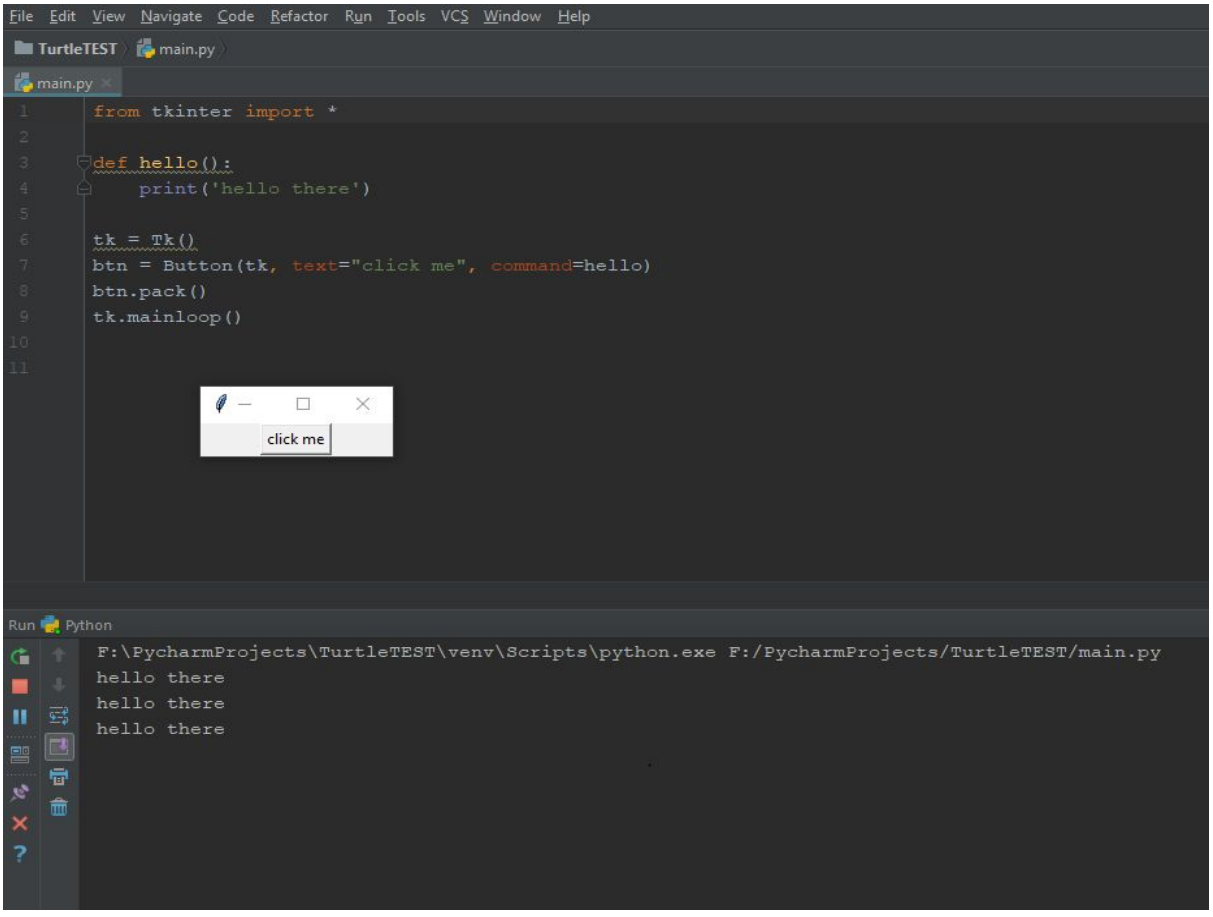
Przycisk “click me” aktualnie nic nie robi. Możesz na niego klikać cały dzień i nic się nie wydarzy. Pierwsze co musimy zrobić to funkcję, która np. wypisuje tekst.

```
def hello():
    print('hello there')
```

Teraz modyfikujemy nasz przykład.

```
from tkinter import *
tk = Tk()
btn = Button(tk, text="click me", command=hello)
btn.pack()
```

Zauważ, że wprowadziliśmy tylko niewielką zmianę do poprzedniej wersji kodu. Dodaliśmy parametr **command**, które informuje Pythona, aby używał funkcji hello po kliknięciu przycisku. Teraz, gdy klikniesz przycisk zobaczysz napis “hello there” napisany w konsoli twojego kompilatora. Pojawi się za każdym razem gdy klikniesz przycisk.



Nazwane parametry

Nazwane parametry są normalnymi parametrami, z wyjątkiem tego, że zamiast używać określonej kolejności podawania parametrów, jawnie nazywamy wartości, aby mogły być definiowane w dowolnej kolejności. Czasami funkcje mają wiele parametrów i nie zawsze musimy podawać wartość dla każdego z nich. Załóżmy, że mamy funkcję o nazwie osoba, która przyjmuje dwa parametry: wysokość i szerokość.

```
def osoba(szerokosc, wysokosc):
    print('Mam {} cm szerokości i {} cm wysokości.'.format(szerokosc,
wysokosc))
```

Normalnie wywołujemy funkcję w taki sposób:

```
osoba(73, 183)
```

Używając nazwanych parametrów możemy wywołać tę funkcję w taki sposób:

```
osoba(wysokosc=183, szerokosc=73)
```

Zasady

---

1. Używaj gita

Naucz się jak używać gita. Opcjonalnie można używać zewnętrznej strony np. GitHub do budowania własnego portfolio.

2. Zrozum problem

Naucz się jak przeprowadzać własny research. Jest to niezwykle przydatna umiejętność, która może zostać użyta w każdej innej dziedzinie życia. Jeśli problem okaże się trudny - nie poddawaj się, rozwiąż go. Narysuj/napisz/zrób cokolwiek jest potrzebne, aby to rozwiązać.

3. Zaimplementuj

Nie bój się pisać brzydkiego kodu i nie zamartwiaj się jak najlepszą implementacją. Napisz coś po raz pierwszy, potem zrefaktoryzuj, uporządkuj itd.

4. Napisz to jeszcze raz

Teraz kiedy masz już doświadczenie i już miałeś do czynienia z dokumentacją - możesz zaszaleć. Dodaj nowe rzeczy, spróbuj innego podejścia, zrób obsługę wielu języków. Baw się kodem. Spróbuj go zrefaktoryzować i zoptymalizować.

5. Nie przestawaj się uczyć

Teraz kiedy doszedłeś już tak daleko, nie przestawaj stawiać sobie nowych wyzwań. Rozbudowuj swoje portfolio.

6. Źródła

Wikipedia, Stack Overflow i Google.

Poważnie, to jest wszystko czego potrzebujesz. Jeśli nie jesteś w stanie znaleźć rozwiązania, to znaczy, że nie szukasz w odpowiedni sposób i musisz poprawić swoje umiejętności researchu. Z tymi trzema stronami jesteś w stanie znaleźć odpowiedź na twoje pytanie, a jeśli go nie ma to możesz utworzyć wątek na stack overflow i zapytać innych o pomoc. Tworząc taki wątek również pomagasz innym - jeśli ktoś będzie miał podobny problem znajdzie twoje pytanie.
Pamiętaj, jeśli tworzysz gdzieś pytanie dla problemu - zaznacz jaka była poprawna odpowiedź.