

Python, zajęcia 25.04.20

# Plan zajęć

- Rysowanie figur 3D w turtle'u
- Robienie aplikacji graficznej – tkinter –  
podpięcie do niej rysowania figur 3D

# Turtle - przypomnienie

```
zolw = turtle.Turtle()  
zolw.color('red')  
zolw.pensize(3)  
zolw.speed(3)
```

← Tak inicjalizujemy żółwia

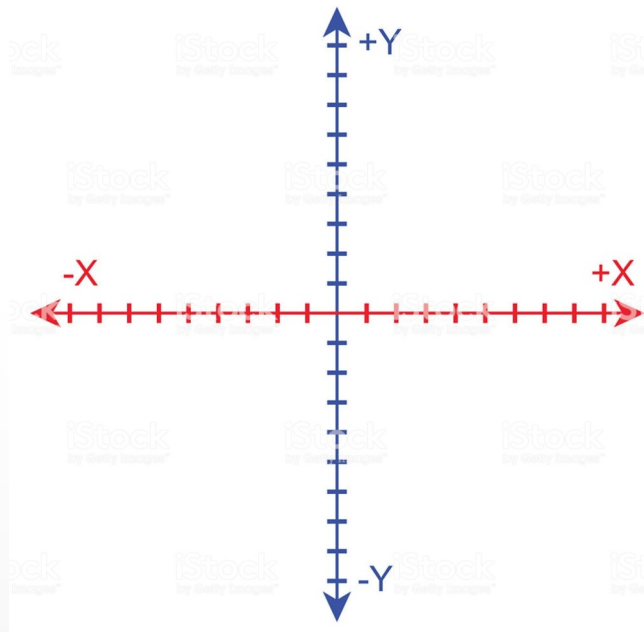
```
#najważniejsze metody  
zolw.forward(100) # idź do przodu o sto pikseli  
zolw.right(90) # skręć w prawo o 90 stopni  
zolw.left(90) # skręć w lewo o 90 stopni  
zolw.penup() # podnieś pióro (przestań pisać)  
zolw.pendown() # obniż pióro (zaczynij pisać)
```

# Turtle – współrzędne

Chcemy rysować figury w 3D. Tym razem będą nam potrzebne współrzędne!

Przypomnienie:

- współrzędna  $x$  to położenie w poziomie (lewo-prawo)
- współrzędna  $y$  to położenie w pionie (górze-dół)



# Turtle – współrzędne cd.

Turtle zawsze zaczyna we współrzędnej (0, 0),

```
zolw.xcor() # zwraca nam X żółwia  
zolw.ycor() # zwraca nam Y żółwia  
zolw.pos()  # zwraca nam X i Y żółwia
```

↑  
Sposoby na dowiedzenie się, jaką pozycję ma żółw.

```
zolw.setpos(10, 10) # każemy iść żółwiowi na (10, 10)
```

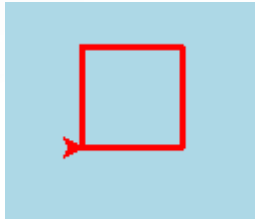
↑  
Sposób na przemieszczenie żółwia na pozycję X, Y  
(w tym wypadku 10, 10)

# Turtle – figury 3D

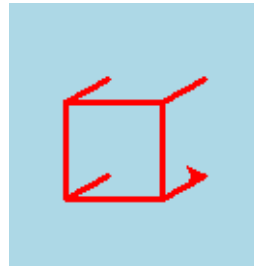
Jak narysować kwadrat w 3D?

Musimy wykonać 3 kroki:

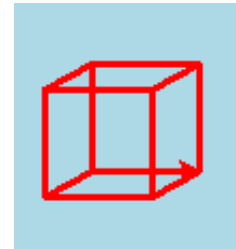
1. Narysuj zwykły kwadrat.



2. Dorysuj cztery ukośne linie.



3. Narysuj drugi raz zwykły kwadrat. Voilà!



# Turtle – figury 3D cd.

Ale skąd mamy wiedzieć, gdzie dokładnie narysować ukośne linie?

**Uwaga: kwadrat ma bok o długości  $a$ .**

Następny punkt o współrzędnych  $x$  i  $y+a$

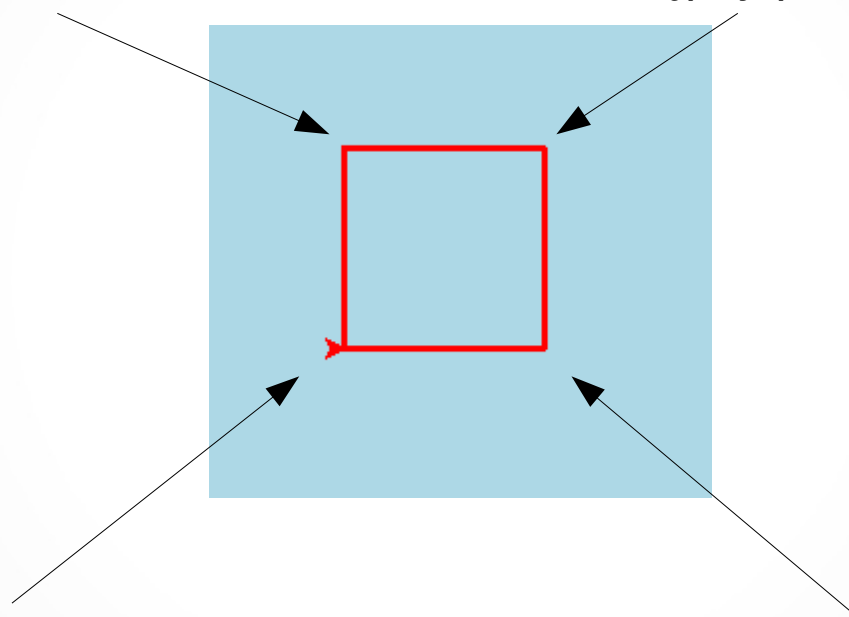
Następny punkt o współrzędnych  $x+a$  i  $y+a$

Bo  $y$  zmienia się o  $a$ !

Punkt początkowy o współrzędnych  $x$  i  $y$

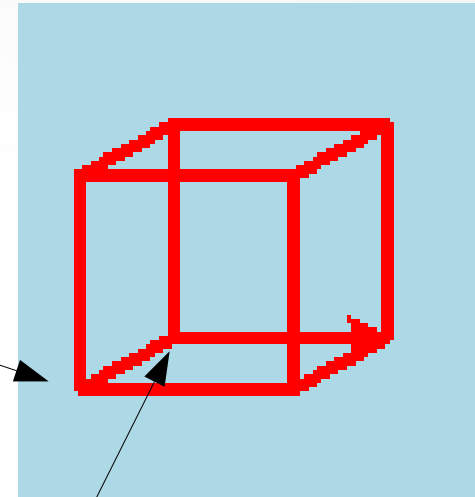
Następny punkt o współrzędnych  $x+a$  i  $y$

Bo  $x$  zmienia się o  $a$ ! →



# Turtle – figury 3D cd.

Kwadrat rysujemy tu...



I tu!

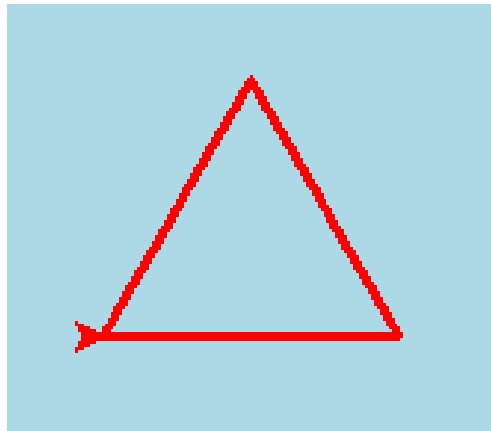
**Czyli: kwadrat rysujemy w punkcie, w którym zaczynamy  
i w punkcie po pierwszej skośnej linii.  
Schemat działania będzie podobny dla innych figur.**



# Turtle – figury 3D cd.

Jakie będą współrzędne skośnych linii w przypadku trójkąta równobocznego?

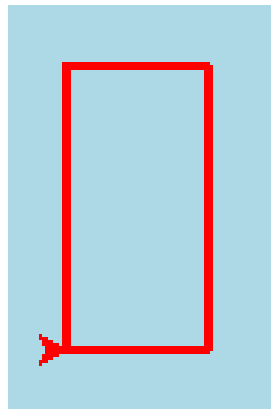
Długość boku: **a**. Wysokość:  $h = \frac{\sqrt{3}}{2}a$



# Turtle – figury 3D cd.

Jakie będą współrzędne skośnych linii w przypadku prostokąta?

Długość boków: **a** i **b**.



# Jak zrobić skośne linie?

Najpierw musimy sprawdzić, w jakiej pozycji znajduje się żółw:

```
startowe_x = zolw.xcor() # współrzędna x żółwia  
startowe_y = zolw.ycor() # współrzędna y żółwia
```

Teraz, zamiast robić osiem zmiennych, żeby zapamiętać **x** i **y**, damy je do dwóch list:

```
#ze wskazówkami od dolnego lewego rogu  
kolejne_x = [startowe_x, startowe_x, startowe_x+a, startowe_x+a]  
kolejne_y = [startowe_y, startowe_y+a, startowe_y+a, startowe_y]
```

Potem w pętli przeglądniemy te listy. Pamiętajcie, że:

**kolejne\_x[0]** – pierwszy element

**kolejne\_x[1]** – drugi element

...

**kolejne\_x[n]** – n-ty element

# Jak zrobić skośne linie? Cd.

Więc rysowanie skośnych linii będzie wyglądać tak:

```
i=0
while i<4:
    zolw.penup()
    zolw.left(30) # skierujemy go w dobra strone do rysowania ukośnych linii
    zolw.setpos(kolejne_x[i], kolejne_y[i])
    self.zolw.pendown()
    self.zolw.forward(25)
    self.zolw.right(30) #skierujemy go z powrotem na proste linie
```

No i pamiętajmy o tym, żeby po narysowaniu pierwszej skośnej linii narysować kwadrat. Czyli musimy sprawdzić **if**-em, czy **i** nie jest równe **0**.

# Zadania part 1

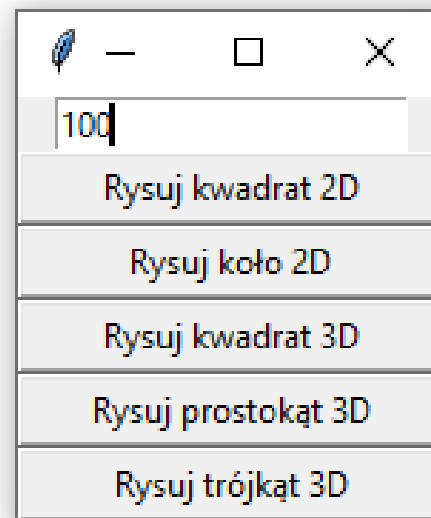
Rysowanie figur 3D:

Plik *figury\_3D.py* na githubie: [github.com/jerzyklos/zajecia](https://github.com/jerzyklos/zajecia)  
folder *zajecia\_25\_04*

Discord: <https://discord.gg/9WUezj> - jeśli link nie będzie działał, nowy będzie na czacie

# Tkinter – szkielet programu

Chcemy zrobić taką aplikację:



A więc podpiąć rysowanie figur pod odpowiednie guziki.


# Tkinter – szkielet programu cd.

```
guzik1 = Button(window, text="Rysuj kwadrat 2D", command=RysujKwadrat, width=20)  
guzik1.grid(column=0, row=1)
```

W którym wierszu będzie guzik,  
czyli który od góry będzie



Jaka funkcja zostanie wywołana,  
jak naciśniemy guzik



# Zadania part 2

Zrobienie apki do rysowania figur 3D:

Plik *guziki\_zadanie.py* na githubie:

[github.com/jerzyklos/zajecia](https://github.com/jerzyklos/zajecia) folder *zajecia\_25\_04*

Discord: <https://discord.gg/9WUezj> - jeśli link nie będzie działał, nowy będzie na czacie



# Listbox - wstęp

Jeśli chcemy wyświetlić listę elementów:

```
from tkinter import *

window = Tk()

listbox = Listbox(window)
listbox.pack()

listbox.insert(END, "element listy") # dodajemy element do listy

for item in ["pierwszy element", "drugi element", "trzeci element"]:
    listbox.insert(END, item)

mainloop()
```

# Zadania part 3

Wyświetlanie listy elementów z pliku:

Plik *listbox\_zadanie.py* na githubie:

[github.com/jerzyklos/zajecia](https://github.com/jerzyklos/zajecia) folder *zajecia\_25\_04*

Discord: <https://discord.gg/9WUezj> - jeśli link nie będzie działał, nowy będzie na czacie