

# Python Turtle

## Spis treści

UWAGI WSTĘPNE .....	1
TURTLE - METODY .....	BŁĄD! NIE ZDEFINIOWANO ZAKŁADKI.
PRZESUWANIE I RYSOWANIE ŻÓŁWIA .....	1
STAN ŻÓŁWIA.....	1
STEROWANIE PIÓREM I WYPEŁNIENIEM .....	2
WYGLĄD ŻÓŁWIA.....	2
RYSOWANIE TEKSTU .....	1
STEROWANIE ANIMACJĄ.....	2
INSTRUKCJE WEJŚCIOWE.....	2
OBSŁUGA ZDARZEŃ .....	3
USTAWIENIA, KONFIGURACJA .....	3
OKNO I EKRAN, OBIEKTY .....	3
STEROWANIE PROGRAMEM .....	3

## Uwagi wstępne

oznaczenia parametrów:

n,m → liczby całkowite  
x,y,r → liczby rzeczywiste, współrzędne, rozmiary, promień  
-n,-x → liczba ujemna  
k, kąt → kąt w stopniach lub radianach (zależnie od ustawień)  
↺, ↻ → kierunek kątowny  
kolor → ustawienie koloru:  
nazwa → "red"  
kod HEX → "#FF88a2"  
w zależności od **colormode()**:  
r,g,b → 0.5, 0.9, 1 lub 128, 200, 255  
(r,g,b) → (0.5, 0.9, 1) lub (128, 200, 255)  
nazwa → nazwa ustalona w bibliotece

## Przesuwanie i rysowanie żółwia

<b>fd(n)</b> forward(n)	przesuń do przodu o n
<b>bk(n)</b> backward(n) back(n)	przesuń do tyłu o n
<b>rt(kąt)</b>	obróć o kąt w prawo

right(kąt)	
<b>lt(kąt)</b> left(kąt)	obróć o kąt w lewo
<b>goto(x,y)</b> setpos(x,y) setposition(x,y)	przenieś do pozycji (x,y). nie zmieniaj orientacji żółwia.
<b>setx(n)</b>	przesuń w poziomie
<b>sety(n)</b>	przesuń w pionie
<b>seth(kąt)</b> setheading()	ustaw kąt żółwia
<b>home()</b>	ustaw żółwia w stan początkowy
<b>circle(r)</b> <b>circle(-r)</b> <b>circle(r,kąt)</b> <b>circle(r,kąt,ile)</b>	narysuj okrąg o promieniu r ↺ narysuj okrąg o promieniu r ↻ narysuj łuk (część okręgu) jw. z podaniem ilości odcinków
<b>dot()</b> <b>dot(r)</b> <b>dot(r, "kolor")</b>	narysuj okrągłą kropkę jw. o promieniu r jw. w podanym kolorze
<b>stamp()</b>	zostaw stempel żółwia, zwraca numer stempla
<b>clearstamp(n)</b>	usuń stempel o numerze n
<b>clearstamps()</b> <b>clearstamps(n)</b> <b>clearstamps(-n)</b>	usuń wszystkie stemple usuń pierwsze n stempli usuń ostatnie n stempli
<b>undo()</b>	cofnij ostatnią akcję
<b>begin_poly()</b>	rozpocznij rejestrowanie wierzchołków wielokąta. Obecna pozycja żółwia to pierwszy wierzchołek wielokąta.
<b>end_poly()</b>	zatrzymaj rejestrowanie wierzchołków wielokąta. Bieżąca pozycja żółwia to ostatni wierzchołek wielokąta. Zostanie połączony z pierwszym wierzchołkiem.
<b>get_poly()</b>	zwróć ostatnio zarejestrowany wielokąt

## Rysowanie tekstu

<b>write(txt,</b> <b>move=False,</b> <b>align='left',</b> <b>font='Arial', 8, 'normal')</b>	napisz tekst txt na pozycji żółwia move → czy przenieść żółwia align → wyrównanie ("left", "center", "right") font → czcionka: nazwa, rozmiar, typ
--	---

## Stan żółwia

<b>st()</b> showturtle()	pokaż żółwia
<b>ht()</b> hideturtle()	ukryj żółwia
<b>isvisible()</b>	czy żółw widoczny
<b>pos()</b> position()	zwróć bieżącą pozycję żółwia (x,y)
<b>towards(x,y)</b>	zwróć kąt linii od żółwia do (x,y)

<b>xcor()</b>	zwróć współrzędną x
<b>ycor()</b>	zwróć współrzędną y
<b>heading()</b>	zwróć bieżący kurs (kąt)
<b>distance()</b>	zwróć odległość od żółwia do (x,y)

## Sterowanie piórem i wypełnieniem

<b>pd()</b> pendown(),down()	pisak w dół
<b>pu()</b> penup(),up()	pisak w górę
<b>width(x)</b> pensize(x)	ustaw rozmiar pióra
<b>isdown()</b>	czy pióro opuszczone
<b>filling()</b>	zwraca stan wypełnienia (True/False)
<b>begin_fill()</b>	rozpoczęcie rysowania z wypełnieniem
<b>end_fill()</b>	wypełnij kształt narysowany po ostatnim <b>begin_file()</b>
<b>color()</b>	zwraca kolory pióra i wypełnienia
<b>color(kolor)</b>	ustaw kolor jednakowy dla pióra i wypełnienia
<b>color(kolorp,kolorf)</b>	ustaw kolory dla pióra i wypełnienia
<b>pencolor(kolor)</b>	jw., tylko kolor pióra
<b>fillcolor(kolor)</b>	jw., tylko kolor wypełnienia

## Wygląd żółwia

<b>shape()</b> <b>shape(nazwa=None)</b>	zwróć nazwę kształtu żółwia Ustaw kształt żółwia ("arrow", "turtle", "circle", "square", "triangle", "classic")
<b>resizemode()</b> <b>resizemode(tryb)</b>	zwróć tryb zmiany rozmiaru żółwia ustaw tryb zmiany rozmiaru żółwia ("auto", "user", "noresize")
<b>shapeseize()</b> stretch_wid=None, stretch_len=None, outline=None) turtle.size()	<b>zmiana rozmiaru żółwia</b> stretch_wid – rozciąganie po szerokości stretch_len – rozciąganie po wysokości zarys – szerokość konturu
<b>shearfactor(wsp)</b>	ustaw wsp. ścinania kształtu żółwia. Jest to tg kąta ścinania
<b>tilt()</b>	obróć kształt żółwia o kąt
<b>settiltangle()</b>	obróć kształt żółwia do podanego kąta
<b>tiltangle()</b>	ustaw lub zwróć aktualny kąt nachylenia
<b>shapetransform(t11,t12,t21,t22)</b>	ustaw lub zwróć bieżącą macierz transformacji kształtu żółwia (t11,t12,t21,t22), $t_{11} \cdot t_{22} - t_{12} \cdot t_{21} < > 0$
<b>get_shapepoly()</b>	zwróć bieżący wielokąt kształtu jako krotkę par współrzędnych. Można to wykorzystać do zdefiniowania nowego kształtu lub składników kształtu złożonego.

## Sterowanie animacją

Metoda	opis
<b>speed(n)</b>	ustaw szybkość 0 - bez opóźnień, najszybciej 1→10: wolno → szybko
<b>delay(czas)</b>	ustaw lub zwróć opóźnienie rysowania w ms. (Jest to w przybliżeniu odstęp czasu między dwiema kolejnymi aktualizacjami kanwy)
<b>tracer(n=None, delay=None)</b>	włącz/wyłącz animację żółwia i ustaw opóźnienie aktualizacji rysunków. Jeśli podano n to wykonywana jest tylko każda n-ta regularna aktualizacja ekranu. Po wywołaniu bez argumentów, zwraca aktualnie przechowywaną wartość n. delay - patrz delay()
<b>update()</b>	aktualizuj TurtleScreen. Stosowane, gdy tracer wyłączony.

## Instrukcje wejściowe

<b>textinput(tytuł, monit)</b>	zwraca tekst wprowadzony w oknie dialogowym
<b>numinput(tytuł, monit, default=None, min=None, max=None)</b>	zwraca liczbę wprowadzoną w oknie dialogowym

# Obsługa zdarzeń

<b>mainloop()</b> done()	rozpoczyna pętlę zdarzeń. Musi być ostatnią instrukcją w programie.
<b>listen()</b>	ustaw fokus na TurtleScreen
<b>onclick(fun, btn=1, add=None)</b>	ustawienie obsługi kliknięcia, fun - funkcja z dwoma argumentami: fun(x,y) btn - 1-LPM add=True - dodanie nowej obsługi, False - zastąpienie istniejącej
<b>onclick(None)</b>	usunięcie obsługi kliknięcia
<b>onrelease()</b>	zwolnienie przycisku myszy, parametry jw
<b>ondrag()</b>	przeciąganie myszą, parametry jw
<b>onkey(fun, "klawisz")</b> onkeyrelease()	ustaw zwolnienie klawisza fun - funkcja bezargumentowa lub None (usunięcie) klawisz - znak lub symbol klawisza
<b>onkeypress()</b>	ustaw naciśnięcie klawisza, parametry jw
<b>ontimer(fun, czas)</b>	cykliczne uruchamianie funkcji fun - funkcja bezargumentowa czas w ms
<b>exitonclick()</b>	połącz metodę bye() z kliknięciami myszy na ekranie

## Ustawienia, konfiguracja

<b>degrees(kąt=360)</b>	ustaw jednostkę miary kątów
<b>radians()</b>	ustaw jednostki kąta radian
<b>mode(tryb)</b>	ustaw tryb żółwia (kurs początkowy i kierunek dodatni kąta) "standard" - →↻ "logo" - ↻↑ "world" - własne ustawienia
<b>pen(par=None)</b>	odczytaj lub ustaw parametry stanu pióra par - słownik zawierające kilka lub wszystkie parametry: "shown": True/False "pendown": True/False "pencolor": kolor "fillcolor": kolor "pensize": x "speed": 0, 1..10 "resizemode": "auto" or "user" or "noresize" "stretchfactor": (n, m) "outline": n "tilt": kąt
<b>colormode(tryb)</b>	ustaw tryb kodowania koloru tryb - jedna z wartości 1 lub 255
<b>setundobuffer(n)</b>	ustaw rozmiar bufora dla undo
<b>undobufferentries()</b>	zwróć rozmiar bufora
<b>getshapes()</b>	zwróć nazwy dostępnych kształtów żółwia

<b>addshape("plik")</b> <b>addshape("nazwa", kształt)</b> <b>addshape("nazwa", obiekt)</b> register_shape(...)	rejestracja nowego kształtu żółwia plik - graficzny, nie obraca się nazwa - zarejestrowana nazwa kształtu kształt - krotka par współrzędnych obiekt - typu Shape
<b>turtles()</b>	zwróć listę żółwi na ekranie

## Okno i ekran, obiekty

<b>title("tytuł")</b>	ustaw tytuł okna
<b>window_height()</b>	zwróć wysokość okna
<b>window_width()</b>	zwróć szerokość okna
<b>bgcolor()</b>	zwróć kolor tła
<b>bgcolor(kolor)</b>	ustaw kolor tła
<b>bgpic(plik)</b>	ustaw tło obrazkowe, nopic - usuń
<b>screensize(dx, dy, kolor=None)</b>	ustaw rozmiary płótna (i tło)
<b>clone()</b>	utwórz i zwróć klon żółwia o tych samych parametrach
<b>getturtle()</b> getpen()	zwraca własny obiekt żółwia
<b>getscreen()</b>	zwróć obiekt TurtleScreen, na którym rysuje żółw.
<b>getcanvas()</b>	zwróć płótno należące do TurtleScreen
<b>setworldcoordinates(x1,y1,x2,y2)</b>	ustaw współrzędne użytkownika, włącza tryb "world". x1,y1 - lewy dolny róg x2,y2 - prawy górny róg Kąty mogą być zniekształcone
<b>setup(width=..., height=..., startx=..., starty=...)</b>	ustaw rozmiar i położenie okna głównego. Domyślne wartości w pliku turtle.cfg. width, height: domyślnie 50% i 70% ekranu - liczba całkowita - rozmiar w pikselach, - liczba rzeczywista - ułamek ekranu, startx, starty: współrzędne pozycji początkowej (home) - jeśli >0, w px od lewej/górnej krawędzi ekranu, - jeśli <0, w px od prawej/dolnej krawędzi ekranu, - jeśli NONE, wyśrodkowanie

## Sterowanie programem

<b>clearscreen()</b>	kasuj ekran i wszystkie żółwie
<b>resetscreen()</b>	resetuj wszystkie żółwie do ich stanu początkowego
<b>reset()</b>	usuń rysunki żółwia z ekranu, ponownie wyśrodkuj żółwia i ustaw zmienne na wartości domyślne.
<b>clear()</b>	usuń rysunki żółwia z ekranu. Nie ruszaj żółwia. Nie ma to wpływu na stan i pozycję żółwia oraz rysunki innych żółwi.
<b>bye()</b>	zamknij okno programu