

Wstęp do programowania w języku Python

Python Turtle – grafika żółwia

KLASA 6

ŚCIĄGAWKA

Na początku programu MUSI być:

```
import turtle
```

```
t = turtle.Turtle()
```

(t to nazwa “naszego” żółwia)

Podstawowe ruchy żółwia

Podstawowe operacje, to przemieszczanie się naszego żółwia. Oto gotowe przykłady najważniejszych funkcji:

<code>t.fd(100)</code>	żółw t przesuwa się do przodu o 100 „kroków” (pikseli)
<code>t.bk(50)</code>	żółw t cofa się w tył o 50 „kroków”
<code>t.rt(90)</code>	żółw t obraca się w prawo o 90 stopni
<code>t.lt(45)</code> <code>t.home()</code>	żółw t obraca się w lewo o 45 stopni żółw t przesuwa się na środek okienka (bez obrotu, ciągle rysując)
<code>t.pu()</code>	„podnieś pisak” – powoduje, że żółw przemieszcza się bez rysowania
<code>t.pd()</code>	„opuść pisak” - żółw znowu rysuje

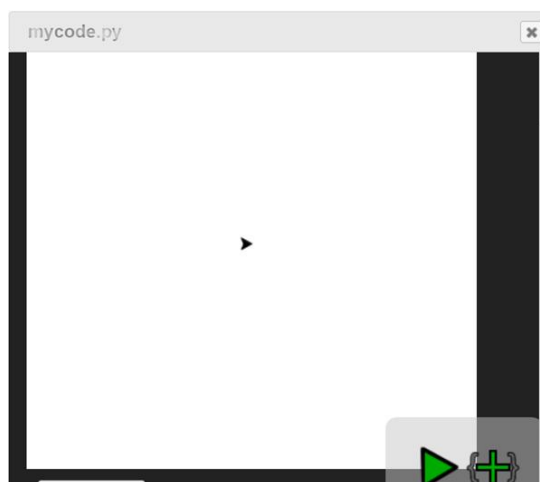
Kolory i rozmiar

<code>t.color('red')</code>	ustawienie czerwonego koloru dla linii, którą rysuje żółw
-----------------------------	---

	(jako x możemy podać takie kolory jak np. 'red','blue','green','yellow','black', 'white'
<code>t.pensize(8)</code>	ustawia grubość linii na 8 „pixeli”

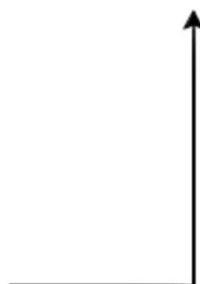
Lokalizacja żółwia

Przykład okienka dla żółwia - tutaj kwadrat 400x400 – żółw zawsze na początku jest w jego środku, „głową” skierowany w prawo.



Przykłady programów:

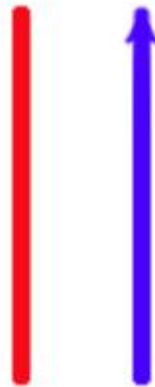
```
import turtle  
t=turtle.Turtle()  
t.fd(100)  
t.lt(90)  
t.fd(150)
```



```
import turtle
t=turtle.Turtle()
t.fd(150)
t.rt(120)
t.fd(50)
t.rt(60)
t.fd(100)
t.rt(60)
r.fd(50)
```



```
import turtle
t=turtle.Turtle()
t.lt(90)
t.color('red')
t.pensize(5)
t.fd(120)
t.bk(120)
t.rt(90)
t.pu()
t.fd(40)
t.pd()
t.lt(90)
t.color('blue')
t.fd(120)
```



Okręgi

Okrąg tworzymy takim rozkazem:

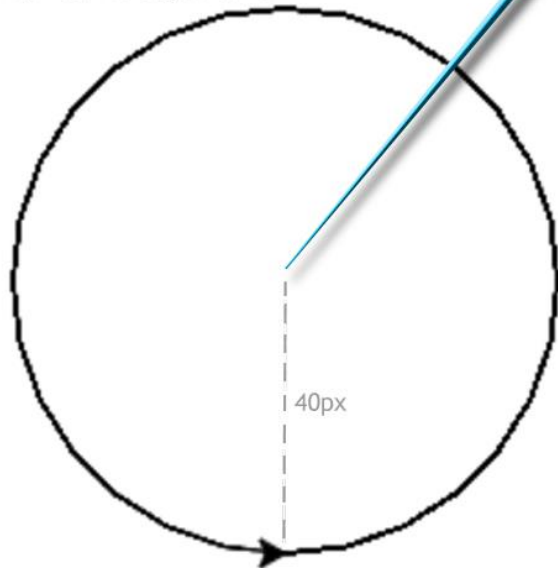
t.circle(40) w nawiasie podajemy promień okręgu
(w tym przykładzie to 40 pikseli)

Tak rysujemy okrąg cyrklem:
- wybieramy środek
- potem rysujemy okrąg



Tak rysuje okrąg żółw w Pythonie:
po rozkazie t.circle(40) żółw *tak jak stoi*,
tak zaczyna rysować okrąg (w lewo, patrząc z góry)

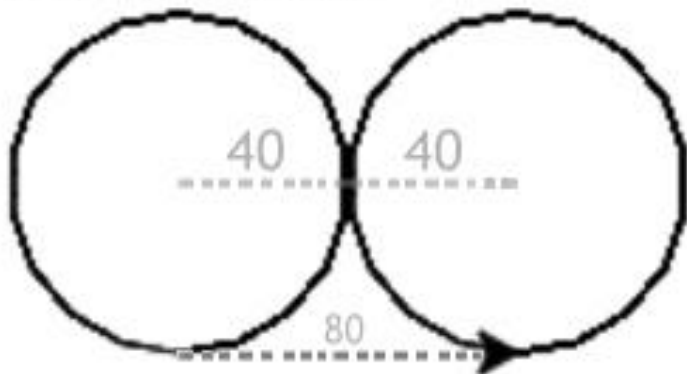
Musisz sobie więc sam (-a) wyobrazić, gdzie jest
środek okręgu (w tym przykładzie będzie gdzieś tutaj),
i że żółw przemieszcza się *pilnując* odległości
40 pikseli od tego środka.



Przykład programu:

```
import turtle  
t=turtle.Turtle()  
t.circle(40)  
t.pu()  
t.fd(80)  
t.pd()  
t.circle(40)
```

Dwa styczne okręgi o promieniu 40
(zauważ, że aby te okręgi stykały się,
to żółw po komendzie t.pu() musiał
przesunąć się w *powietrzu* o 80 pikseli
- czyli dwa promienie)



Losowanie liczby z danego przedziału:

najpierw po `import turtle` trzeba zaimportować nową bibliotekę:

import random

a w samym programie np. tak:

`los=random.randrange(0,100)` losuje liczbę naturalną z przedziału od 0 do 99 (100 jest tutaj *granicą*)

Przykład użycia:

Kolejne linie programu	Wyjaśnienie „co się dzieje”
<code>import turtle</code> <code>import random</code>	Wczytanie bibliotek: grafiki żółwia i losowania liczb
<code>t = turtle.Turtle()</code>	nasz żółw ma nazwę – to t
<code>los1 = random.randrange(0,100)</code>	losowanie liczby z przedziału od 0 do 99 (wylosowana liczba jest zapamiętywana pod nazwą los1)
<code>los2 = random.randrange(0,360)</code>	losowanie liczby z przedziału od 0 do 359 (wylosowana liczba jest zapamiętywana jako zmienna los2)
<code>t.fd(los)</code>	żółw rysuje odcinek o wylosowaną ilość pikseli
<code>t.rt(los2)</code>	żółw obraca się w prawo o wylosowany kąt Oto efekt – żółw przesunął się o wylosowane 82 piksele i obrócił o wylosowane 253 stopnie 