

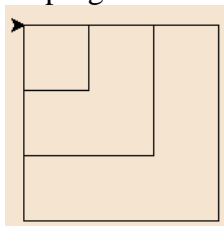
PYTHON – TURTLE 04 – FUNKCJE (20)

*Funkcja jest fragmentem programu, który możemy zapamiętać pod określoną nazwą.
Za pomocą funkcji dzielimy złożony program na logiczne części.
Funkcję można wykonywać wiele razy używając nazwy.
Do „wnętrza” funkcji możemy przekazywać wartości; funkcja może je przeliczać i zwracać wyniki.*

Kwadrat (2)

- Wpisz definicję funkcji kwadrat i uruchom program

```
def kwadrat(bok):  
    for i in range(4):  
        fd(bok);rt(90)  
  
kwadrat(50)  
kwadrat(100)  
kwadrat(150)
```



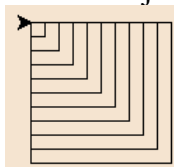
*Funkcja kwadrat ma jeden parametr o nazwie bok
powtarzamy 4 razy
do przodu o bok
obrót w prawo o 90*

- Wklej do ramki zrzut ekranu z programem i oknem żółwia

Kwadraty (2)

- Usuń trzy polecenia kwadrat(...) lub weź w komentarz
nie usuwaj definicji kwadrat(bok)
- Wpisz pod definicją funkcji zestaw instrukcji i uruchom program

```
bok=100  
for i in range(10):  
    kwadrat(bok)  
    bok=bok-10
```



*do zmiennej bok przypisz liczbę 100
pętla (i) powtarza 10 razy
rysowanie kwadratu o boku bok
zmniejszanie boku o 10*

Rysowanych jest 10 kwadratów. Bok pierwszego jest równy 100 i każdy następny bok jest o 10 mniejszy od poprzedniego

- Wklej do ramki zrzut ekranu z programem i oknem żółwia

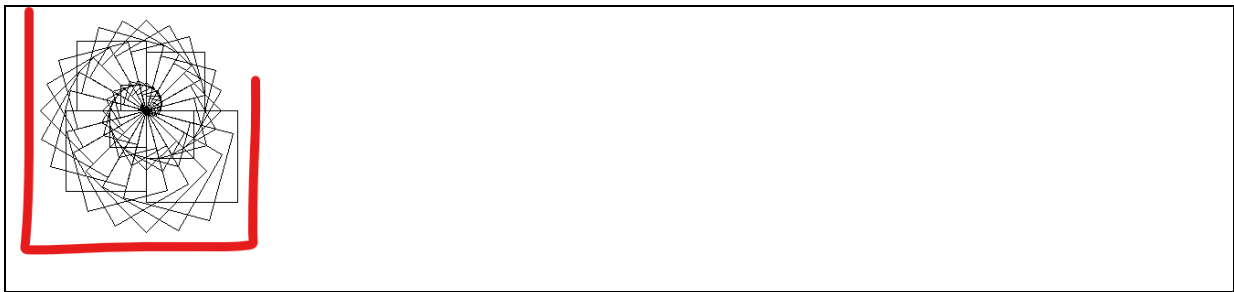
Ślimak (2)

- Przekształć poprzednie instrukcje (lub wpisz od nowa) w definicję funkcji

```
def slimak(bok,ile,kat,skok):  
    for i in range(ile):  
        kwadrat(bok)  
        bok=bok-skok  
        rt(kat)
```

*funkcja slimak() potrzebuje czterech parametrów:
bok - długość największego kwadratu
ile - ile kwadratów rysować
kat - o jaki kąt obracamy kolejny kwadrat
skok - o jaką długość zmniejszamy kolejny kwadrat*

- Uruchom funkcję **slimak()** z następującymi parametrami: **bok=100, ile=50, kat=15, skok=2**
- Wklej do ramki zrzut ekranu z programem i oknem żółwia



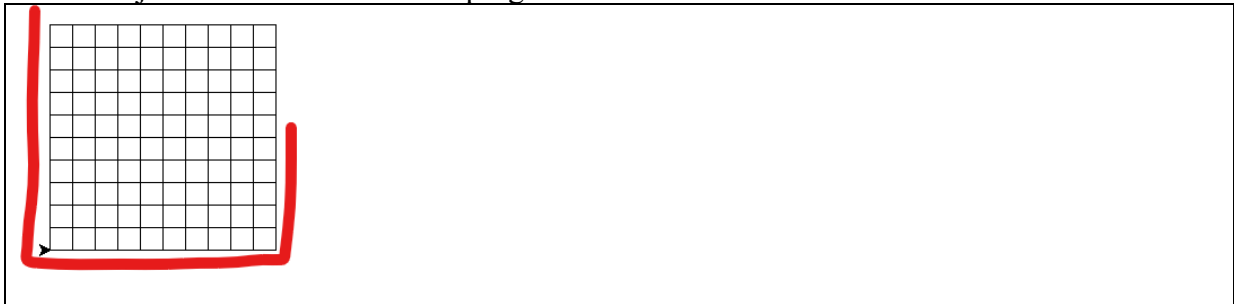
Wiersz (2)

- Napisz funkcję o nazwie **wiersz** z dwoma parametrami: **bok, ile**
Funkcja wykonuje następujące polecenia:
pętla for powtarza **ile** razy
rysowanie kwadratu za pomocą funkcji **kwadrat** z parametrem **bok**
ruch **do przodu** o **bok**
- Uruchom polecenie `wiersz(20,10)`
- Wklej do ramki zrzut ekranu z programem i oknem żółwia



Szachy (2)

- Napisz funkcję o nazwie **szachy** z dwoma parametrami: **bok, ile**
Funkcja wykonuje następujące polecenia:
pętla for powtarza **ile** razy
rysowanie kwadratów za pomocą funkcji **wiersza** z parametrami **bok,ile**
skok do nowego wiersza: `bk(bok*ile);rt(90);fd(bok);lt(90)`
- Uruchom polecenie `szachy(20,10)`
- Wklej do ramki zrzut ekranu z programem i oknem żółwia

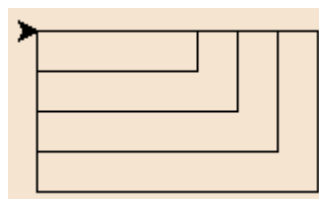


Prostokąt (2)

- Wpisz definicję prostokąta i uruchom program

```
def prost(b1,b2):
    for i in range(2):
        fd(b1);rt(90)
        fd(b2);rt(90)

prost(80,20)
prost(100,40)
prost(120,60)
prost(140,80)
```



Funkcja ma dwa parametry: *b1* i *b2*, które są bokami prostokąta powtarzamy 2 razy

do przodu o parametr funkcji *b1*
obróć w prawo o 90
do przodu o parametr funkcji *b2*
obróć w prawo o 90

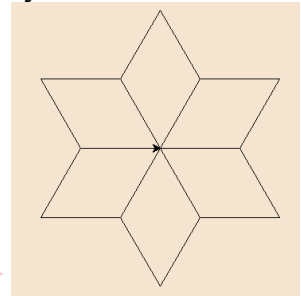
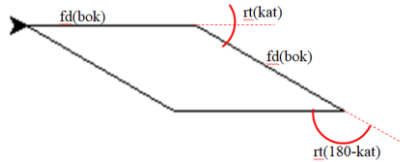
- Wklej do ramki zrzut ekranu z programem i oknem żółwia

Romb (2)

- Wpisz definicję funkcji rysującej romby o boku **bok** i kącie wewnętrznym **kat**

```
def romb(bok, kat):
    for i in range(2):
        fd(bok); rt(kat)
        fd(bok); rt(180-kat)

    for i in range(6):
        romb(100, 60)
        rt(60)
```



Funkcja ma dwa parametry: bok i kat

powtarzamy 2 razy

do przodu o parametr funkcji bok

obrót w prawo o parametr funkcji kat

do przodu o parametr funkcji bok

obrót w prawo o 180-kat

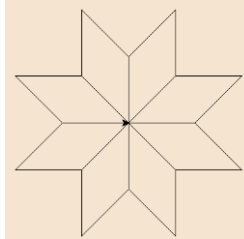
Program rysuje 6 razy romb o kącie 60 stopni i po narysowaniu rombu żółw się obraca o 60 stopni

- Wklej do ramki zrzut ekranu z programem i oknem żółwia

Gwiazda (2)

- Wpisz definicję funkcji rysującą gwiazdę z rombów

```
def gwiazda(bok, ile):
    kat=360/ile
    for i in range(ile):
        romb(bok, kat)
        rt(kat)
```



Parametrami funkcji są bok rombu i liczba rombów ile

Kąt rombu i jednocześnie kąt obrotu to $360^\circ / \text{liczba rombów}$.

przypisz do zmiennej kat iloraz 360 i parametru ile

powtarzamy ile razy

rysowanie rombu - parametry: bok i kat

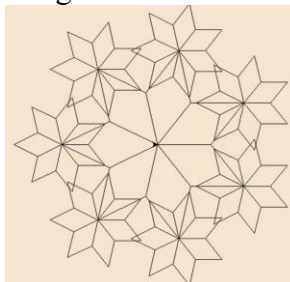
obrót w prawo o kat

- Dopisz instrukcję rysującą ośmioramienną gwiazdę o boku 100
- Wklej do ramki zrzut ekranu z programem i oknem żółwia

Wianek1 (2)

- Wpisz definicję funkcji rysującą wianek z gwiazdek

```
def wianek(bok, ile):
    kat=360/ile
    gwiazda(bok, ile)
    for i in range(ile):
        fd(bok); rt(kat); fd(bok)
        gwiazda(bok/2, ile)
        bk(bok); lt(kat); bk(bok)
        rt(kat)
```



Funkcja ma dwa parametry: bok, ile

przypisz do zmiennej kat iloraz 360 i parametru ile

rysuj gwiazdę z parametrami bok i ile

powtarzamy ile razy

do przodu o bok żółw na koniec ramienia gwiazdy
 obrót w prawo o kąt
 do przodu o bok
 rysowanie gwiazdy - parametry: bok dwa razy mniejszy i kat
 wstecz o bok żółw na środek gwiazdy
 obrót w lewo o kąt
 wstecz o bok
 obrót w lewo o kat nowe położenie żółwia

Środek małej gwiazdy znajduje się na końcu ramienia dużej gwiazdy

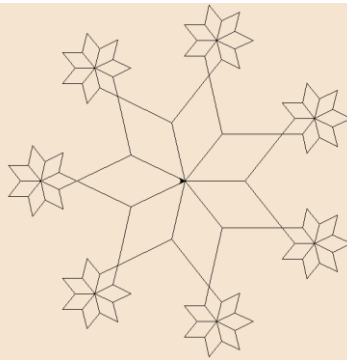
Mała gwiazda jest dwa razy mniejsza od dużej

- Dopisz instrukcję rysującą **siedmioramienny wianek o boku 100**
- Wklej do ramki zrzut ekranu z programem i oknem żółwia

Wianek2 (2)

- Wpisz funkcję, która rysuje wianek, w którym końce ramion się stykają

```
def wianeks(bok, ile):
    kat=360/ile
    b=bok/3
    gwiazda(bok, ile)
    for i in range(ile):
        pu(); fd(bok); rt(kat); fd(bok)
        fd(b); lt(kat); fd(b); pd()
        rt(180)
        gwiazda(b, ile)
        rt(180)
        pu(); bk(b); rt(kat); bk(b)
        bk(bok); lt(kat); bk(bok); pd()
        rt(kat)
```



Małe gwiazdki styczne do ramion dużej gwiazdy - należy przenieść żółwia do nowego „wirtualnego” środka. Aby gwiazdy o nieparzystej liczbie ramion były styczne, należy obrócić żółwia o 180 stopni.

- Dopisz instrukcję rysującą **siedmioramienny wianek o boku 100**
- Wklej do ramki zrzut ekranu z programem i oknem żółwia