

# Python - Płatek Kocha

© Copyright by 3bird Projects 2022, <http://edukacja.3bird.pl>

## Uwagi ogólne

Płatek tworzony jest za pomocą rekurencji (funkcja wywołuje sama siebie, aż do spełnienia określonego warunku).

Uwaga! Wcięcia w kodzie, mają dla Pythona znaczenie (są konieczne w odpowiednich miejscach)! Python do pobrania (dla Windows): <https://www.python.org/downloads/windows/>

Uruchamianie skryptu:

```
C:\> python naszSkrypt.py
```

Uwaga: Nigdy nie wolno kopiować kodu z PDF-a, gdyż zawiera on niewidoczne znaki końca linii i tzw. twarde odstępny. Kod należy przepisać ze zrozumieniem.

## Kod skryptu - wersja podstawowa

```
from turtle import *
def narysujPlatek(dlugoscBoku, poziomZaglebienia):
    if poziomZaglebienia == 0:
        forward(dlugoscBoku)
    else:
        dlugoscBoku = dlugoscBoku / 3.0
        poziomZaglebienia = poziomZaglebienia - 1
        narysujPlatek(dlugoscBoku, poziomZaglebienia)
        left(60)
        narysujPlatek(dlugoscBoku, poziomZaglebienia)
        right(120)
        narysujPlatek(dlugoscBoku, poziomZaglebienia)
        left(60)
        narysujPlatek(dlugoscBoku, poziomZaglebienia)

dlugoscBoku = 300.0
penup()
goto(0,180)
backward(dlugoscBoku / 2.0)
pendown()

for kazdyElement in range(3):
    narysujPlatek(dlugoscBoku, 3)
    right(120)

mainloop()
```

## Kod skryptu - wersja rozbudowana

```
#!/usr/bin/env python
# Powyższa linia, tylko dla systemu Linux.

# Płatek Kocha będzie rysowany za pomocą biblioteki Turtle wbudowanej w Windows,
# ale w Linuksie może wymagać doinstalowania za pomocą: pip install turtle --user.
# Biblioteka Turtle korzysta z menedżera okien TKinter.
from turtle import *

# Zmienna "dlugoscBoku" jest bazową długością krawędzi trójkąta.
# Zmienna "poziomZaglebienia" to parametr odpowiadający za dokładność rysowanego płątka,
# za głębokość zagłębienia w kolejne poziomy szczegóły.
# Poniżej, funkcja rysująca _\_/ (czyli rozbija płatek na trójkąty).
def narysujPłatek(dlugoscBoku, poziomZaglebienia):
    # Kiedy poziomZaglebienia osiągnie 0, rysuj długość boku:
    if poziomZaglebienia == 0:
        # Ruch do przodu z opuszczonym rysikiem (rysowanie krawędzi):
        forward(dlugoscBoku)
    else:
        # Zmniejszenie długości boku (podział na 3):
        dlugoscBoku = dlugoscBoku / 3.0
        # Zmniejszenie poziomu o 1 (rysowanie kolejnego kształtu):
        poziomZaglebienia = poziomZaglebienia - 1
        narysujPłatek(dlugoscBoku, poziomZaglebienia) #wywołanie rekurencyjne
        left(60) #obrót w lewo o 60 stopni
        narysujPłatek(dlugoscBoku, poziomZaglebienia) #wywołanie rekurencyjne
        right(120) #obrót w prawo o 120 stopni
        narysujPłatek(dlugoscBoku, poziomZaglebienia) #wywołanie rekurencyjne
        left(60) #obrót w lewo o 60 stopni
        narysujPłatek(dlugoscBoku, poziomZaglebienia) #wywołanie rekurencyjne

setup(800,800) # Rozmiar okna
bgcolor("white") # Kolor tła płótna
pencolor("blue") # Kolor rysika
pensize(3) # Grubość rysika
hideturtle() # Ukryj żółwia (kursor)
# Ustawienie prędkości rysowania (wyższa wartość = szybciej):
speed(10000000)
# Ustawienie bazowej długości krawędzi na 600, ale wpływa to też na współrzędne płątka:
dlugoscBoku = 600.0
# Podniesienie rysika:
penup()
# Przesunięcie żółwia o 180px do góry (początkowa pozycja to środek):
goto(0,180)
# Przesunięcie kursora do tyłu (w lewo) tak aby płatek był wyśrodkowany na ekranie w poziomie:
backward(dlugoscBoku / 2.0)
# Opuszczenie rysika:
```

## **pendown()**

*# Rozpoczęcie i powtarzanie procesu, ponieważ przy każdym obrocie pętli budowane są dwa półramiona oraz jedno pełne ramię:*

**for** każdyElement **in** range(3):

*# Wywołanie funkcji rysującej i ustalenie poziomu zagłębienia:*

**narysujPlatek**(dlugoscBoku, 4)

*# Obrócenie kursora o 120 stopni ( $1/3 * 360$ ):*

**right**(120)

*# Utrzymanie okienka programu w celu wyświetlania efektu końcowego. Funkcja pochodzi od menedżera okien TKinter i polega na "nasłuchiwaniu" i oczekiwaniu na działania użytkownika:*

**mainloop()**

Ostatnia aktualizacja: 6 grudnia 2022.