

Python, Thonny a

E 3011

Jan Böhm

RECETOX

February 25, 2025

Co nás dnes čeká

1 

2 

3 $\square()$ a $\triangle()$

4 

```
1 import math
2
3 help(math)
```

Odpovězte na otázky:

- 1 Vyčíslete $\frac{45 \cdot \frac{73}{39} - 15}{72^3 - 5701}$
- 2 Porovnejte 1.01^{100} a e
- 3 Spočítejte $\sin(\frac{13}{7}\pi)$ a $\tan(12345^\circ)$
- 4 Ověřte, že \triangle o stranách 9; 40; 41 je pravoúhlý. Pak dopočítejte zbylé vnitřní úhly.
- 5 Vydělte se zbytkem $2024 \div 17$
- 6 Zaokrouhlete $\sqrt{2}$ na celé číslo dolů, nahoru a na 2 desetinná místa.
- 7 Co vrátí Python na dotazy $\frac{1}{0}$, $\log(-1)$, $\sqrt{-1}$?

Co nás dnes čeká

1 

2 

3 $\square()$ a $\triangle()$

4 

```
1 import turtle
2
3 Leonardo = turtle.Turtle() # creates turtle named Leonardo
4
5 Leonardo.forward(200)
6 Leonardo.left(90)
7 Leonardo.backward(200)
8 Leonardo.right(-90)
9 Leonardo.forward(100)
10 Leonardo.penup()
11 Leonardo.forward(100)
12 Leonardo.right(90)
13 Leonardo.pendown()
14 Leonardo.forward(200)
15
16 turtle.exitonclick() #from library turtle use exitonclick()
```



Pomocí želvy nakreslete čtverec o straně 250 px.



Pomocí želvy nakreslete symbol Relikvií smrti.

Co nás dnes čeká

1



2



3

$\square()$ a $\triangle()$

4



```
import turtle

# funkce square nakresli ctverec o delce strany side
def square(side):
    Michelangelo = turtle.Turtle()
    Michelangelo.forward(side)
    Michelangelo.left(90)
    Michelangelo.forward(side)
    Michelangelo.left(90)
    Michelangelo.forward(side)
    Michelangelo.left(90)
    Michelangelo.forward(side)
    Michelangelo.left(90)

square(250)
```



Upravte kód kterým jste vytvořili čtverec tak, aby vznikla funkce `square(side)`, která nakreslí čtverec o délce strany `side`.

Poznámka

Dva přístupy jak tvořit funkce s želvou:

```
1 import turtle
2
3 def square(side):
4     Mike = turtle.Turtle() # turtle created inside function
5     Mike.forward(side)
6     Mike.left(90)
7     ...
8
9 square(100)
```

```
1 import turtle
2
3 def square(turtle, side): # turtle called as input of function
4     turtle.forward(side)
5     turtle.left(90)
6     ...
7
8 Mike = turtle.Turtle() # turtle created outside of function
9 square(Mike, 100) # turtle Mike is passed to square as input
```

Druhý přístup můžeme doplnit o return želvy...

```
1 import turtle
2
3 def square(turtle, side):
4     turtle.forward(side)
5     turtle.left(90)
6     ...
7     return(turtle)
8
9 Mike = turtle.Turtle()
10 Mike = square(Mike, 100)
11 Mike.forward(100)
12 Mike = square(Mike, 150)
```

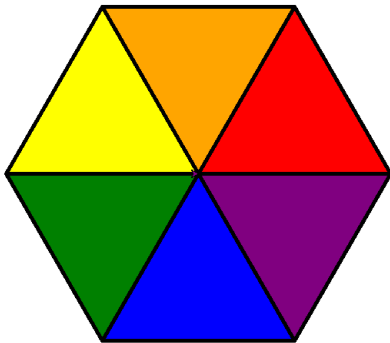
Funkce $\Delta()$

Napište funkci `triangle(ninja, side, fillColor, lineColor)` která pomocí želvy se jménem `ninja` nakreslí rovnostranný trojúhelník o délce strany `side`, vyplněný barvou `fillColor` a okrajem barvy `lineColor`.

```
1 import turtle
2
3 # popis, co ma funkce za argumenty a co dela
4 def triangle(ninja, side = 100, fillColor = "white", lineColor
   = "black"):
5     pass
6
7 Donatello = turtle.Turtle()
8 triangle(Donatello)
9 triangle(ninja = Donatello, side = 200, fillColor = "green",
   lineColor = "green")
10 triangle(Donatello, side = 1500, "red", "blue")
```



Pomocí své funkce triangle nakreslete "kytičku"



Co nás dnes čeká

1



2



3

$\square()$ a $\triangle()$

4



Domácí úkol [2 body]



Napište funkci `kanjiX(ninja, color = "black")`, kde `X` je placeholder pro anglický název znaku, který jste dostali na 1. cvičení. Pomocí této funkce želva `ninja` nakreslí daný znak barvou `color`. Další požadavky:

- Výsledný znak se vejde do čtverce 250×250 pixelů a současně ho rozumně vyplní (s nějakým okrajem pro čitelnost).
- Želva začíná v levém dolním rohu a hledí vpravo.
- Želva končí opět v levém dolním rohu a hledí vpravo.
- Řešení (*.py soubor) nahrajte do odevzdáárny `hw1` do neděle 9. března (včetně).

Hodnotí se:

- Samotná kvalita výsledku (google lens check).
- Splnění všech pokynů v zadání.