

Python: do básico à autonomia intermediária

aula 3

Bernardo Alves Furtado

February 11, 2022

Menu do dia

Turtle

```
Review: functions do_four.py  
for_loop.py:  
my_object0.py  
my_first_object.py  
my_second_object.py
```



Meet Maria, the turtle

- ▶ `import turtle`
- ▶ `maria = turtle.Turtle()`
- ▶ `maria.fd(100)`
- ▶ `maria.lt(90)`
- ▶ `maria.fd(100)`
- ▶ `maria.lt(90)`
- ▶ `maria.fd(100)`
- ▶ `maria.lt(90)`
- ▶ if in the script (not in console): `turtle.mainloop()`



Aplique o já conhecido

- ▶ `function`
- ▶ `dir`
- ▶ Quais são as outras funções de turtle?

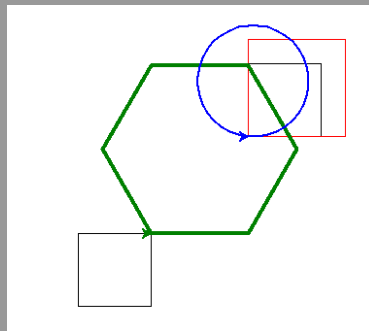


Don't repeat yourself: DRY

- ▶ Crie uma função simples, usando `for`
- ▶ Adicione um segundo parâmetro – `length` – que determina o comprimento da square
- ▶ Adicione um terceiro parâmetro – `n` – número de lados de um polígono
- ▶ Utilizando a linha
`angle = 360 / n`
Alterando o número de vezes de `range(4)` para `range(n)` e
Substituindo o `'90'`, é possível

Jose and Maria

```
▶ if __name__ == '__main__':  
    maria = turtle.Turtle()  
    jose = turtle.Turtle()  
    maria.color('red')  
    square(maria, 100)  
    square(jose, 75)  
    jose.penup()  
    jose.setpos(-100, -100)  
    jose.pendown()  
    square(jose, -75)  
    jose.color('green')  
    jose.pensize(4)  
    make_polygon(jose, 100, 6)  
    maria.color('blue')  
    maria.pensize(2)  
    make_polygon(maria, 10, 36)  
    turtle.mainloop()
```



Analysis

- ▶ **Generalization:** inserir parâmetros ao invés de valores fixos
- ▶ **Encapsulation:** uma função, mais de um objeto
- ▶ **Avoided repetition** (for loop)
- ▶ Go to my_third_object.py

Exercício

- ▶ Pense em um 'motif' e faça um desenho usando `turtle`
- ▶ Use as funções elaborados no slide DRY
- ▶ Deve conter pelo menos duas tartarugas desenhadoras
- ▶ Deve conter pelo menos a repetição de 4 formas
- ▶ Deve conter pelo menos duas cores e tamanhos de linha
- ▶ Deve conter pelo menos um 'salto' entre as formas desenhadas
- ▶ Leia o Chapter 4