

Programação em Python

Desenhando com *turtle*

Prof. Daniel Di Domenico

https://github.com/danidomenico/gex003_algprog

Desenhando com *turtle*

- **Turtle** é uma biblioteca Python que funciona como um quadro de desenho;
 - Permite criar desenhos a partir de linhas e círculos, como se estivesse utilizando uma **caneta** em um **quadro**;
- Para criar um programa que faça desenhos, deve-se:
 - Importar a biblioteca *turtle*;
 - Criar um controlador para o *turtle* (caneta);
 - Desenhar através dos métodos da *turtle*;
 - Executar o método *turtle.done()*;

```
#importa a biblioteca turtle  
import turtle
```

```
des = turtle.Turtle() #cria o controlador para turtle (caneta)  
des.forward(100) #desenha uma linha de 100 pixels  
turtle.done() #mantém a tela aberta para ver o desenho
```

Desenhando com *turtle*

- Instalação da biblioteca **turtle**:
 - Deve ser realizada caso ainda não esteja instalada no sistema;
- Comando para instalação em sistema Linux (Debian/Ubuntu):

```
#Instalação do pacote com a turtle para Python 3  
sudo apt-get install python3-tk
```

- Demais sistemas e versões do Python:
 - Acessar documentação:

<http://www.tkdocs.com/tutorial/install.html>

Desenhando com *turtle*

- **Desenhando linhas:**
 - Um desenho sempre começa com a **caneta** no centro da tela (coordenada 0.0, 0.0).

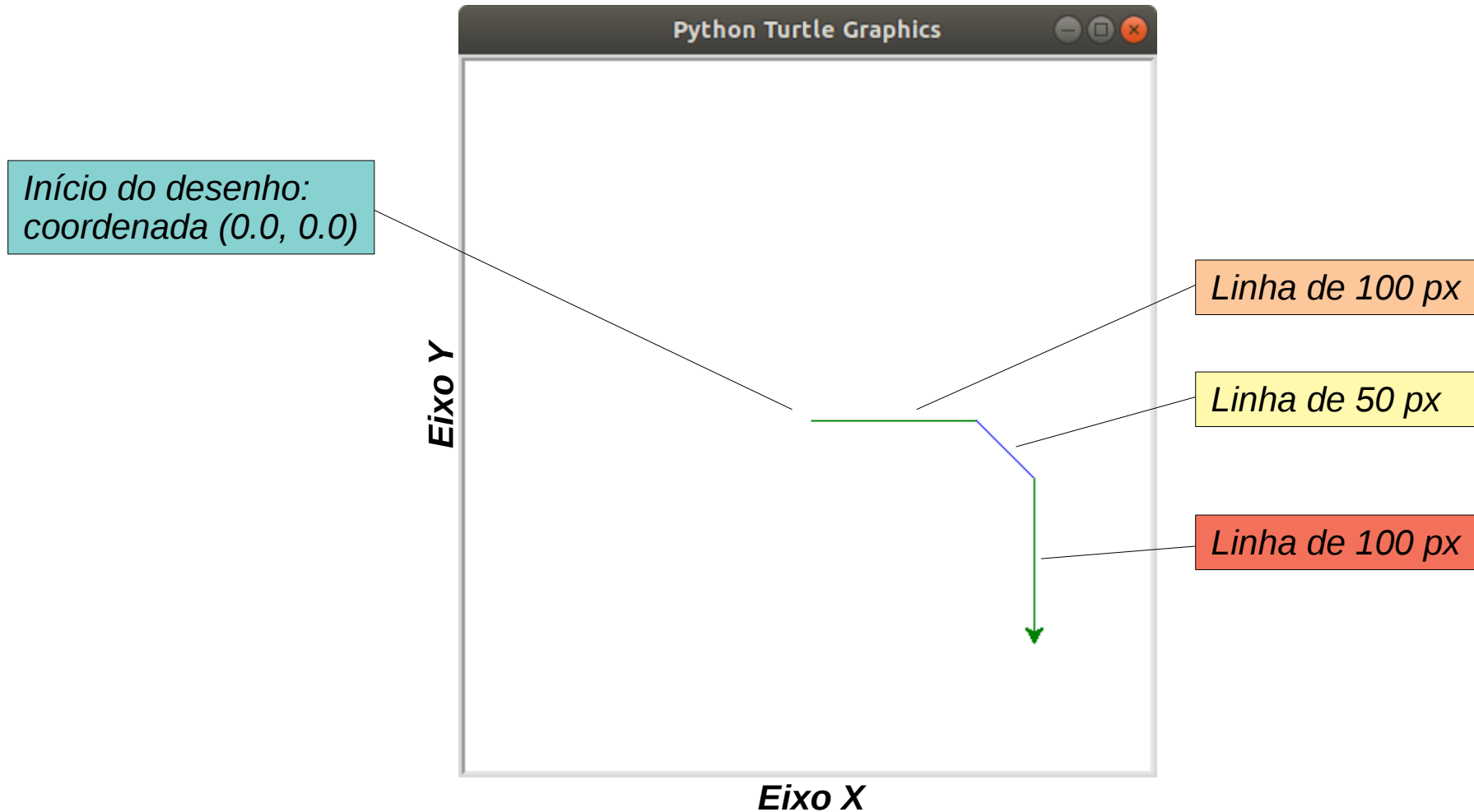
```
import turtle

des = turtle.Turtle() #cria o controlador para turtle (caneta)
des.color('green') #muda a cor da caneta
des.forward(100) #desenha uma linha de 100 pixels
des.right(45) #muda o ângulo para continuar o desenho em 45°
des.color('blue')
des.forward(50)
des.right(45)
des.color('green')
des.forward(100)

turtle.done()
```

Desenhando com *turtle*

- Desenhando linhas:



Desenhando com *turtle*

- **Desenhando círculos:**
 - Deve-se definir o tamanho do raio:

```
import turtle
```

```
des = turtle.Turtle(shape="turtle") #cria a caneta, forma de tartaruga  
des.circle(50) #desenha um círculo com raio de 50 px
```

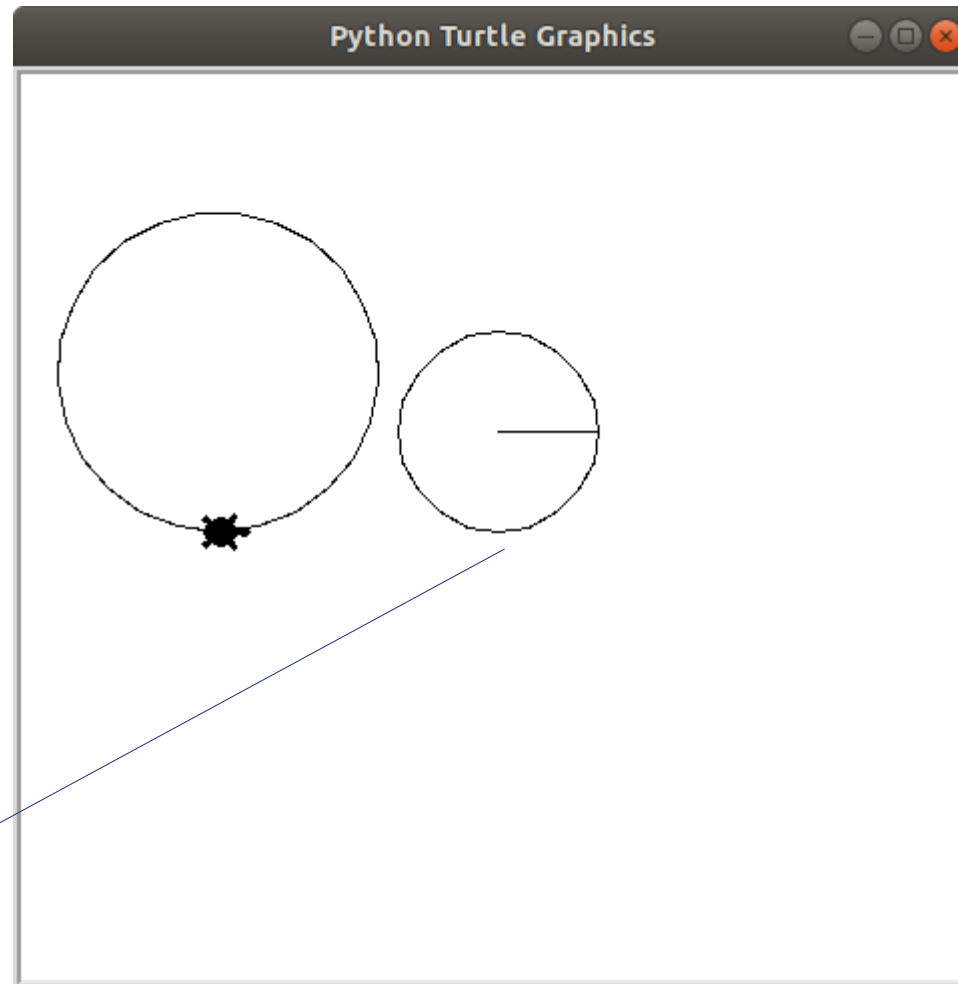
```
des.penup() #tira (levanta) a caneta do quadro  
des.setposition(0, 50) #muda a posição da caneta para x=0.0, y=50.0  
des.pendown() #encosta a caneta no quadro  
des.forward(50) #desenha uma reta de 50 px (marca o raio do círculo)
```

```
des.penup() #tira (levanta) a caneta do quadro  
des.setposition(-140, 0) #muda a posição da caneta para x=-140.0, y=0.0  
des.pendown() #encosta a caneta no quadro  
des.circle(80) #desenha um círculo com raio de 80 px
```

```
turtle.done()
```

Desenhando com *turtle*

- Desenhando círculos:



*Início do desenho:
coordenada (0.0, 0.0)*

Desenhando com *turtle*

- **Alguns métodos principais (caneta):**

Método	Ação
right(x)	Gira a caneta X graus para a direita
left(x)	Gira a caneta X graus para a esquerda
color('x')	Muda a cor da caneta
forward(x)	Move a caneta X pixels para frente
backward(x)	Move a caneta X pixels para trás
setpos(x, y)	Move a caneta para as coordenadas X, Y
circle(x)	Desenha um círculo com raio de X pixels
position()	Retorna a posição atual da caneta (coordenadas X e Y)
penup()/pendown()	Movimenta a caneta para cima e para baixo

Desenhando com *turtle*

- **Exercício:**
 - Utilizando a biblioteca *turtle*, desenhe um quadrado com lado de 100 px.

Desenhando com *turtle*

- **Exercício:**

- Utilizando a biblioteca *turtle*, desenhe um quadrado com lado de 100 px.

```
import turtle

des = turtle.Turtle(shape='circle') #cria a caneta, forma de círculo
des.forward(100)
des.right(90)
des.forward(100)
des.right(90)
des.forward(100)
des.right(90)
des.forward(100)
des.right(90)

turtle.done()
```

Desenhando com *turtle*

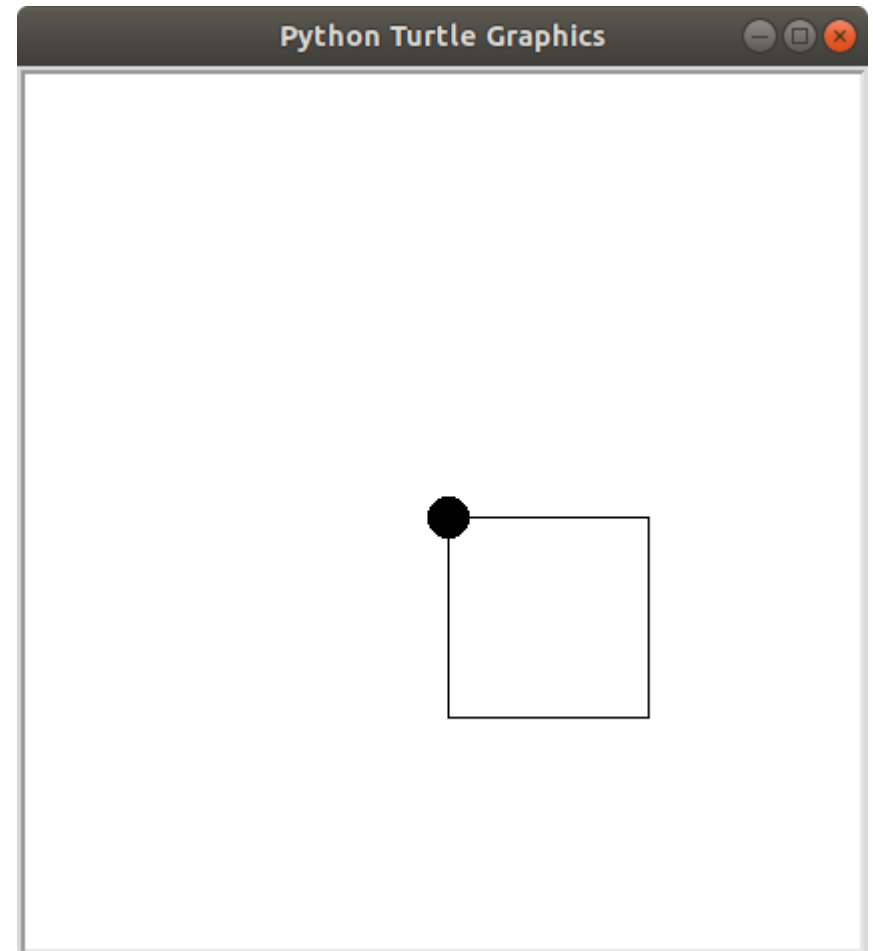
- **Exercício:**

- Utilizando a biblioteca *turtle*, desenhe um quadrado com lado de 100 px:
 - Versão utilizando laço:

```
import turtle

des = turtle.Turtle(shape='circle')
for i in range(4):
    des.forward(100)
    des.right(90)

turtle.done()
```



Desenhando com *turtle*

- **Controle da tela (quadro de desenho):**
 - Utilizar o método `turtle.Screen()`:

```
import turtle
```

```
tela = turtle.Screen() #Acessa o controle da tela (quadro de desenho)  
tela.setup(500, 500) #Define o tamanho do quadro de desenho (em pixels)
```

```
des = turtle.Turtle()  
des.left(180)  
des.forward(50)
```

```
tela.exitonclick() #A tela será fecha quando houver um clique
```

Desenhando com *turtle*

- **Exemplo avançado 1:**
 - Espiral:

```
import turtle

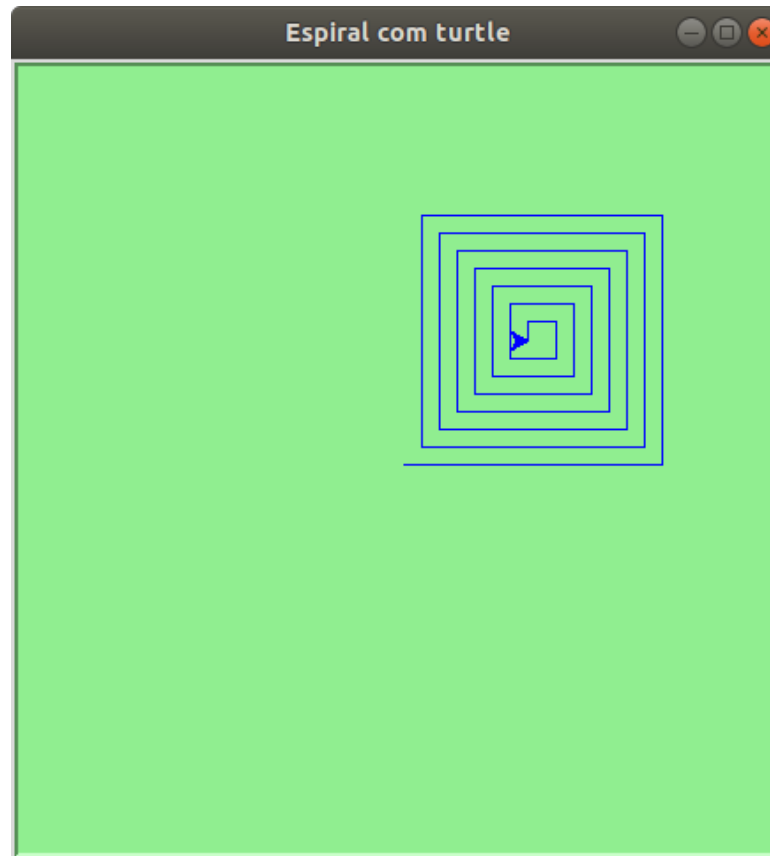
tela = turtle.Screen()
tela.bgcolor("light green") #Define a cor do quadro de desenho pra verde
tela.title("Espiral com turtle") #Define o título do quadro de desenho
des = turtle.Turtle()
des.color("blue") #Define a cor da caneta

for tam in 146, 126, 106, 86, 66, 46, 26:
    tam2 = tam
    for i in range(4):
        des.forward(tam2)
        des.left(90)
        tam2 = tam2-5

tela.exitonclick()
```

Desenhando com *turtle*

- **Exemplo avançado 1:**
 - Espiral:



Desenhando com *turtle*

- **Exemplo avançado 2:**
 - Anéis olímpicos (parte 1):

```
import turtle
```

```
posX = [0, -120, 60, -60, -180]
```

```
posY = [0, 0, 60, 60, 60]
```

```
cores = ['blue', 'purple', 'red', 'yellow', 'green']
```

```
tela = turtle.Screen() #Acessa o controle da tela (quadro de desenho)
```

```
tela.setup(500, 500) #Define o tamanho do quadro de desenho (em pixels)
```

```
tela.title("Olimpíadas") #Define o título do quadro de desenho
```

```
#continua....
```

Desenhando com *turtle*

- **Exemplo avançado 2:**
 - Anéis olímpicos (parte 2):

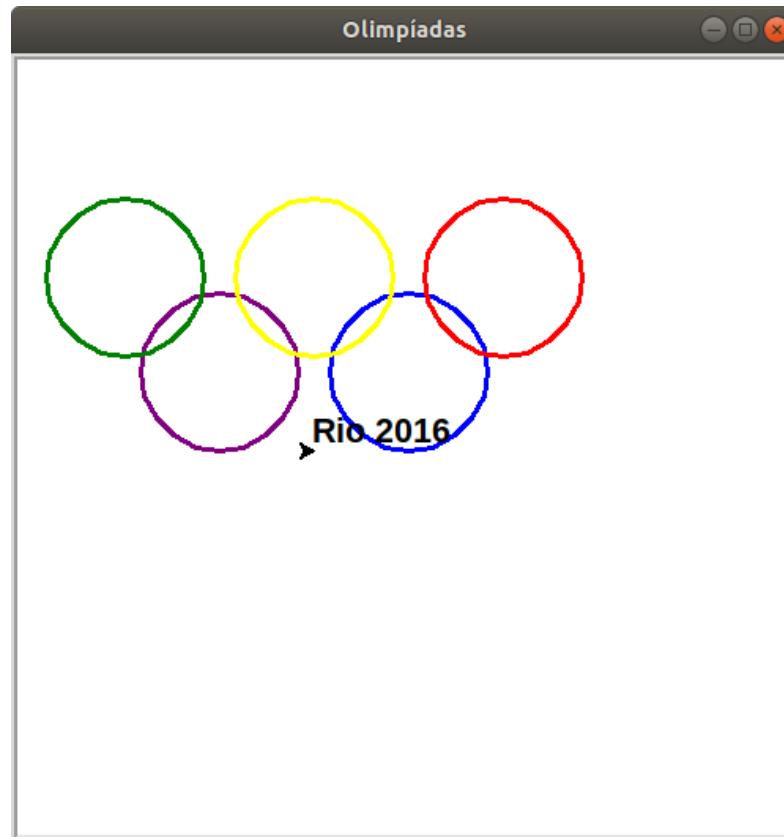
```
des = turtle.Turtle()
des.pensize(3) #Muda o tamanho da caneta para 3
for i in range(len(posX)):
    des.penup() #Tira (levanta) a caneta do quadro
    des.setposition(posX[i], posY[i]) #Define a posição da caneta
    des.pendown() #Encosta a caneta no quadro
    des.color(cores[i])
    des.circle(50)

des.penup()
des.setposition(-60, 0)
des.pendown()
des.color("black")
des.write("Rio 2016", font=("Arial", 16, "bold")) #Escreve no quadro

tela.exitonclick()
```


Desenhando com *turtle*

- **Exemplo avançado:**
 - Anéis olímpicos:



Desenhando com *turtle*

- Mais recursos, exemplos, outras funcionalidades e documentação sobre a biblioteca *turtle*:

<https://docs.python.org/3.6/library/turtle.html#module-turtle>

<https://www.geeksforgeeks.org/turtle-programming-python/>

Desenhando com *turtle*

- **Exercícios:**

- 1) Desenhe 2 quadrados um dentro do outro
- 2) Desenhe 10 quadrados um dentro do outro usando um laço de repetição;
- 3) Construa 10 triângulos de diferentes tamanhos usando um laço;

ATENÇÃO: utilize sua criatividade!