

# Introdução

## Revisão de Python – Prática

Jan K. S. – [janks@puc-rio.br](mailto:janks@puc-rio.br)

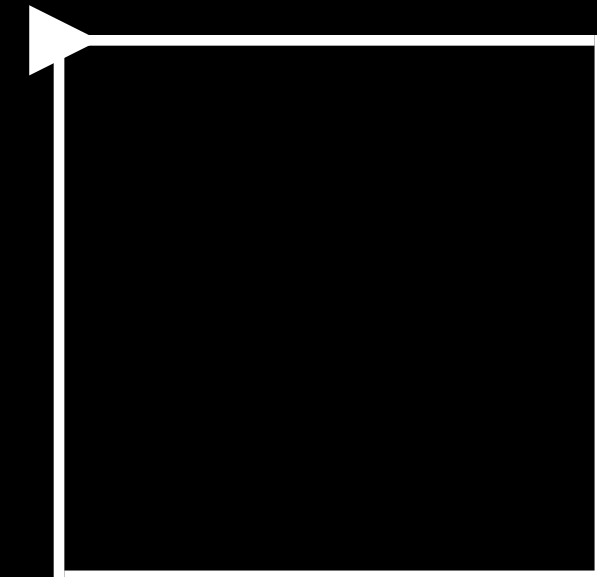
ENG 1419 – Programação de Microcontroladores

Forward 20 steps! Now turn  
right, by 45 degrees! Now  
go back 40 steps! Turn  
right, 90 degrees!



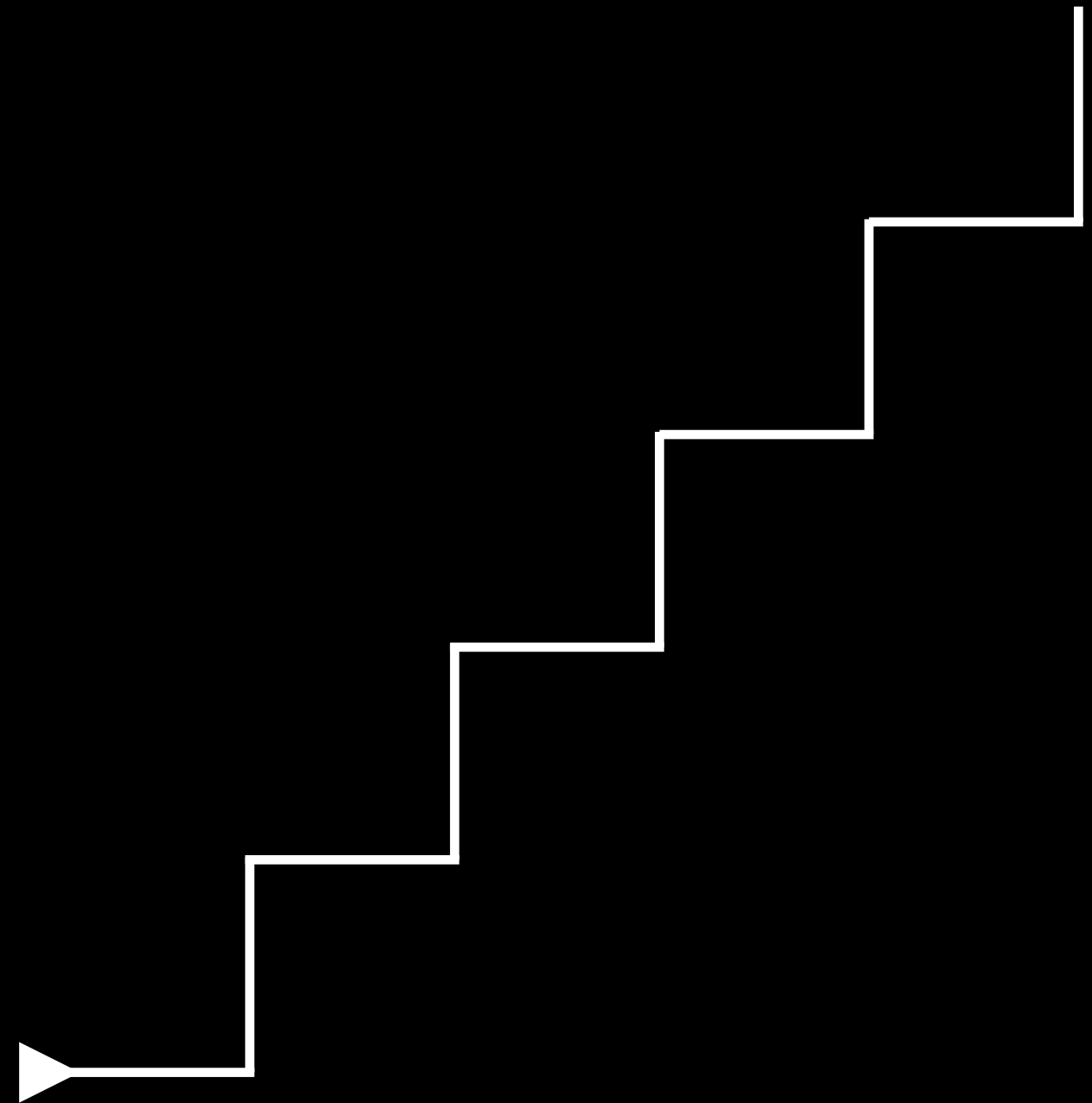
visited To The Sea.com

```
>>> from turtle import *
>>> forward(100)    # avance 100 pontos de distância
>>> right(90)       # vire 90° à direita
>>> forward(100)
>>> right(90)
>>> forward(100)
>>> right(90)
>>> forward(100)
```



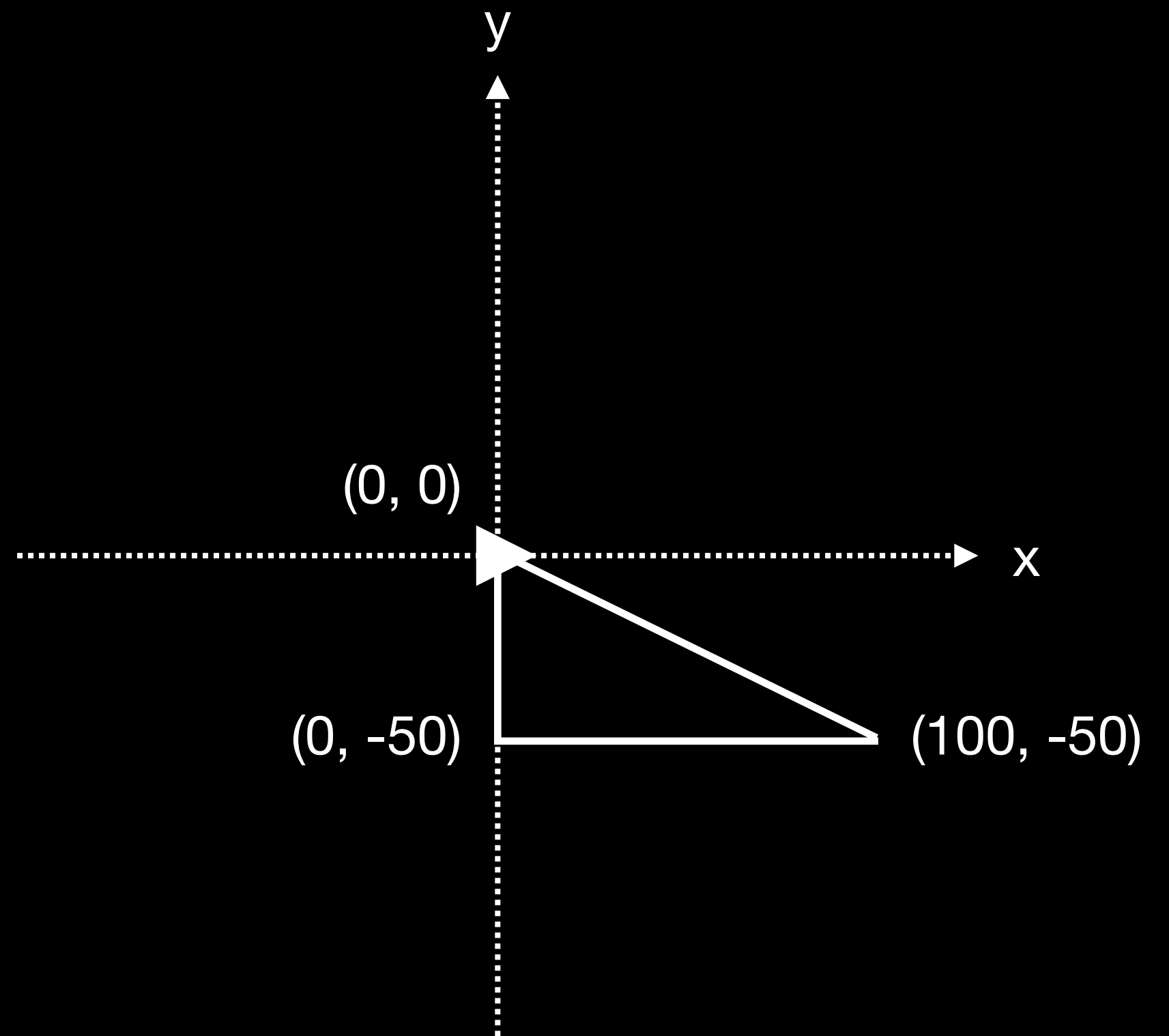
Comandos para Avançar e Rotacionar

```
>>> from turtle import *
>>> for i in range(0, 7):
...     forward(50)
...     left(90)
...     forward(50)
...     right(90)
... 
```



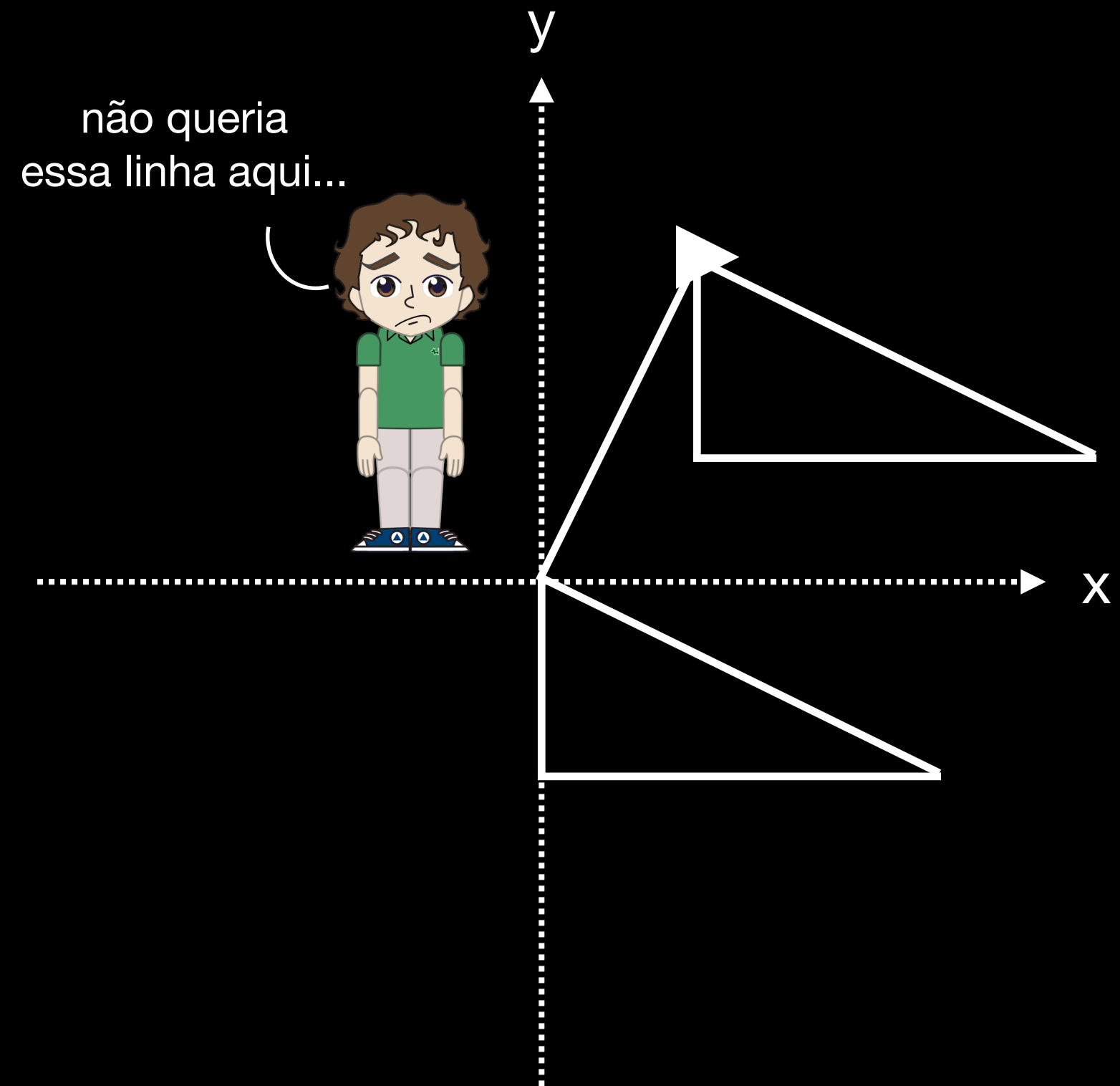
Rotação para Esquerda e para a Direita

```
>>> from turtle import *  
>>> goto(0, -50)  
>>> goto(100, -50)  
>>> goto(0, 0)
```



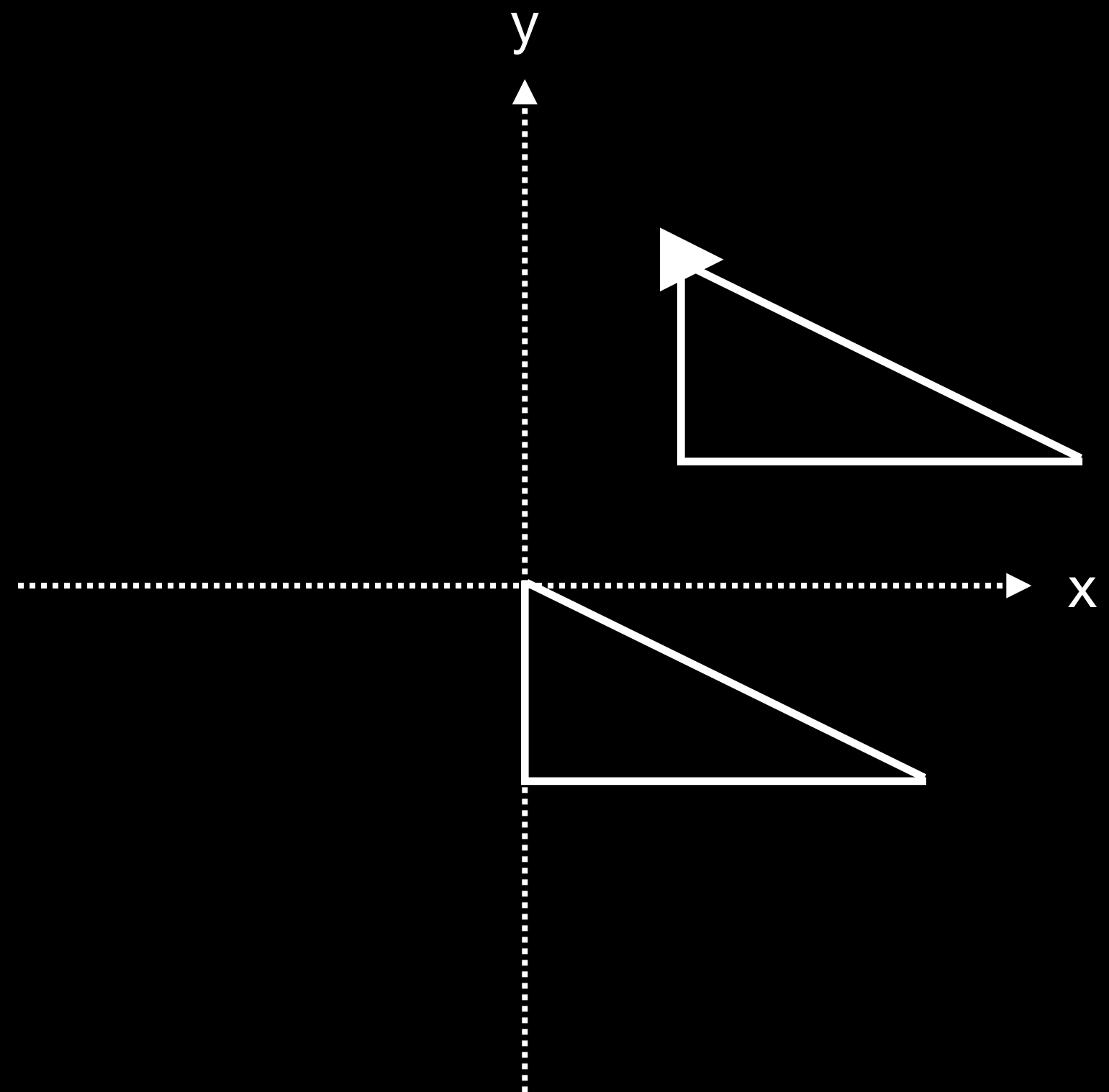
Deslocamento para um Ponto Específico

```
>>> from turtle import *
>>> goto(0, -50)
>>> goto(100, -50)
>>> goto(0, 0)
>>> goto(40, 80)
>>> goto(40, 30)
>>> goto(140, 30)
>>> goto(40, 80)
```



Desenho de uma Segunda Figura

```
>>> from turtle import *
>>> goto(0, -50)
>>> goto(100, -50)
>>> goto(0, 0)
>>> penup()
>>> goto(40, 80)
>>> pendown()
>>> goto(40, 30)
>>> goto(140, 30)
>>> goto(40, 80)
```



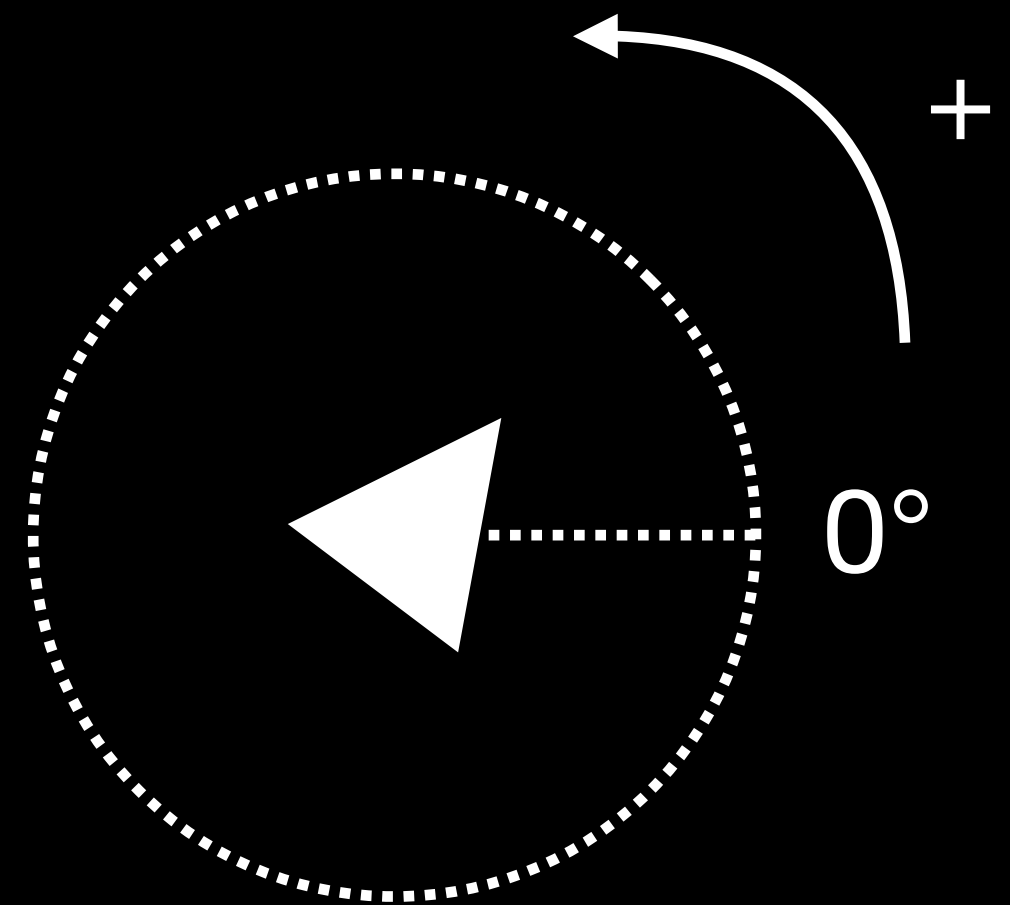
Desenho de uma Segunda Figura "Levantando" e "Abaixando" a Caneta

```
>>> from turtle import *
```

```
>>> ...
```

```
>>> setheading(0)
```

define ângulo como  $0^\circ$ ,  
independentemente do ângulo inicial

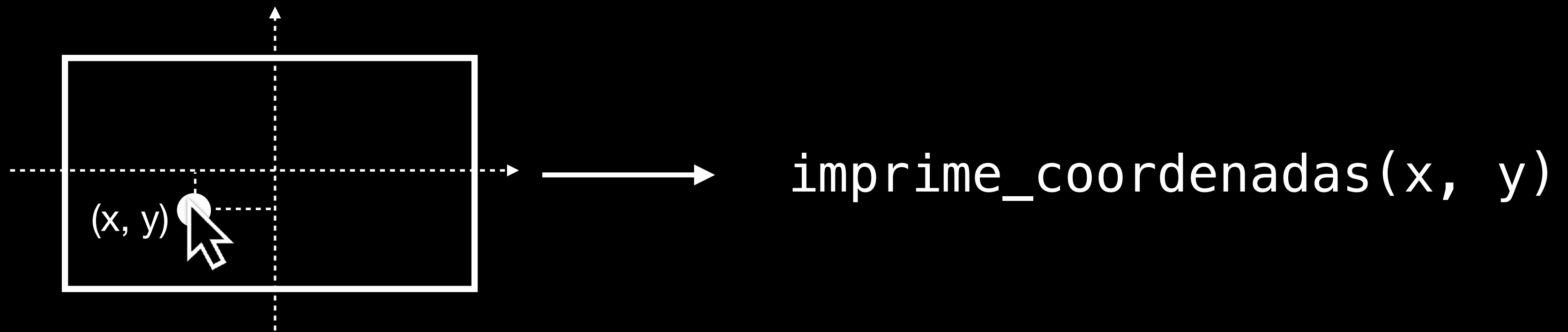


Definição Absoluta do Ângulo



▶ Olá, mundo!

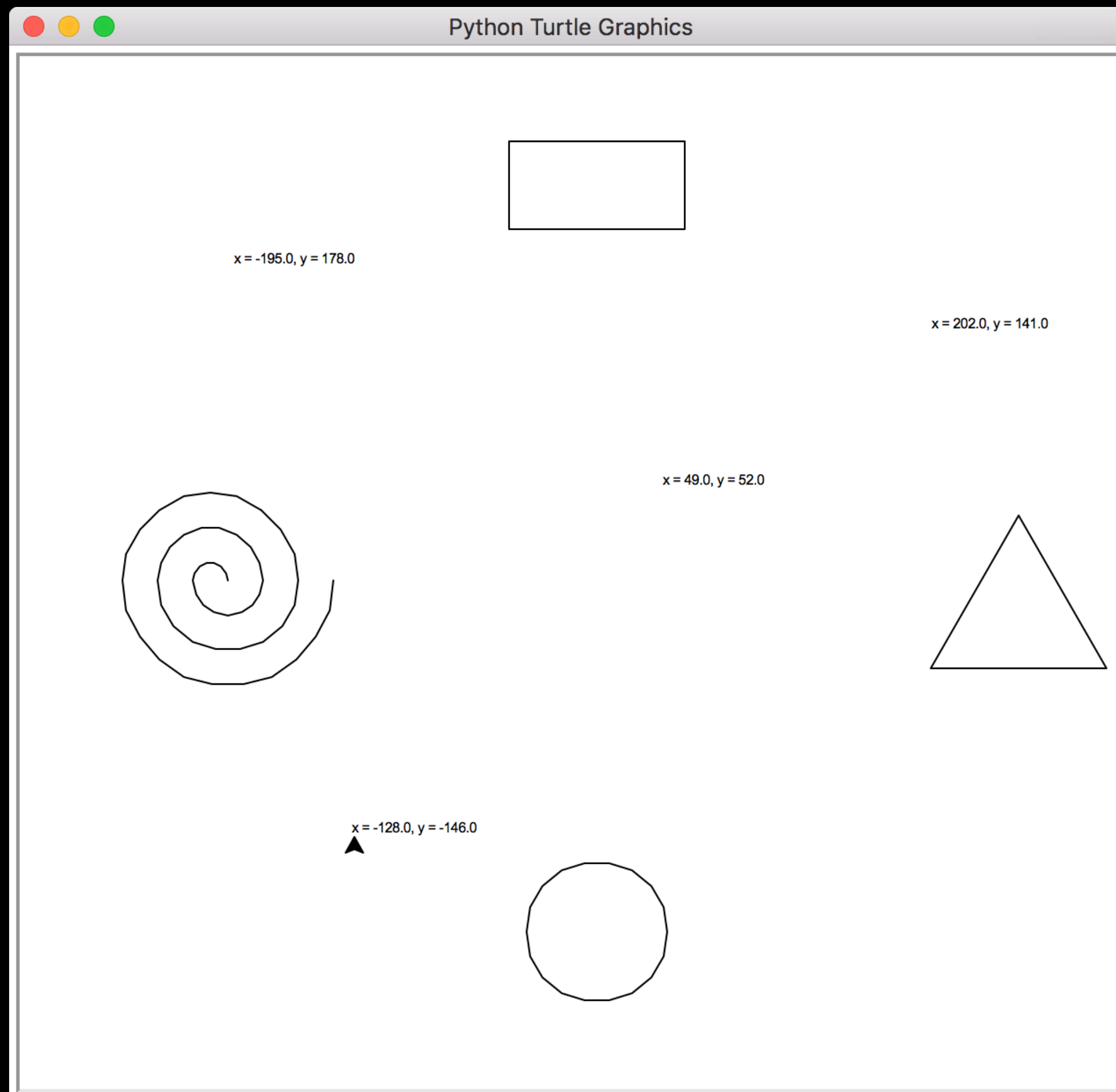
```
>>> write("Olá, mundo!")
```



```
>>> def imprime_coordenadas(x, y):  
...     print("x = ", x)  
...     print("y = ", y)  
...  
>>> onscreenclick(imprime_coordenadas)
```

*não coloque parênteses no final!*

# Testes Iniciais



Objetivo dos Testes Iniciais



## Testes Iniciais

**Desenhe o retângulo 100x50 no topo.**

↳ DICA: adapte o exemplo do quadrado dos slides.

**Desenhe um triângulo equilátero na direita.**

↳ DICA: use a função `setheading` para definir o ângulo.

**Desenhe o círculo na parte de baixo.**

↳ DICA: consulte o [site da documentação oficial](#).

**Desenhe a espiral na esquerda.**

↳ DICA: desenhe semi-círculos aumentando o raio.

Ao clicar em um ponto da tela, vá para esse local e **desenhe um texto** com o valor das coordenadas `x` e `y`, no formato "`x = -32.0, y = 231.0`".

↳ DICA: use a função `write` e a concatenação de string com números.

# Implementação



Desenho de Bandeiras

```
>>> from turtle import *  
>>> forward(100)  
>>> right(90)  
>>> forward(100)  
>>> right(90)  
>>> forward(100)  
>>> right(90)  
>>> forward(100)
```



Exemplo Anterior do Quadrado



```
>>> from turtle import *
>>> fillcolor("blue")    # define cor de preenchimento
>>> begin_fill()         # inicia preenchimento
>>> forward(100)
>>> right(90)
>>> forward(100)
>>> right(90)
>>> forward(100)
>>> right(90)
>>> forward(100)
>>> end_fill()           # finaliza preenchimento
```



Exemplo Anterior do Quadrado com Preenchimento

Thonny - /Users/Jan/Library/Mobile Documents/com~apple~CloudDocs/PUC/Microncontroladores/Introdução/Introducao/implementacao.py @ 31 : 82

implementacao.py

```
from turtle import *

def desenha_retangulo(x, y, comprimento, altura, cor):
    return

def desenha_circulo(x, y, raio, cor):
    return

def desenha_poligono(lista_pontos, cor):
    return

# Bandeira 1
desenha_retangulo(0, 40, 100, 20, 'blue')
desenha_retangulo(0, 20, 100, 20, 'white')
desenha_retangulo(0, 0, 100, 20, 'red')

# Bandeira 2
desenha_retangulo(0, 140, 100, 20, 'orange')
desenha_retangulo(0, 120, 100, 20, 'white')
desenha_retangulo(0, 100, 100, 20, 'green')
desenha_circulo(50, 110, 10, 'orange')

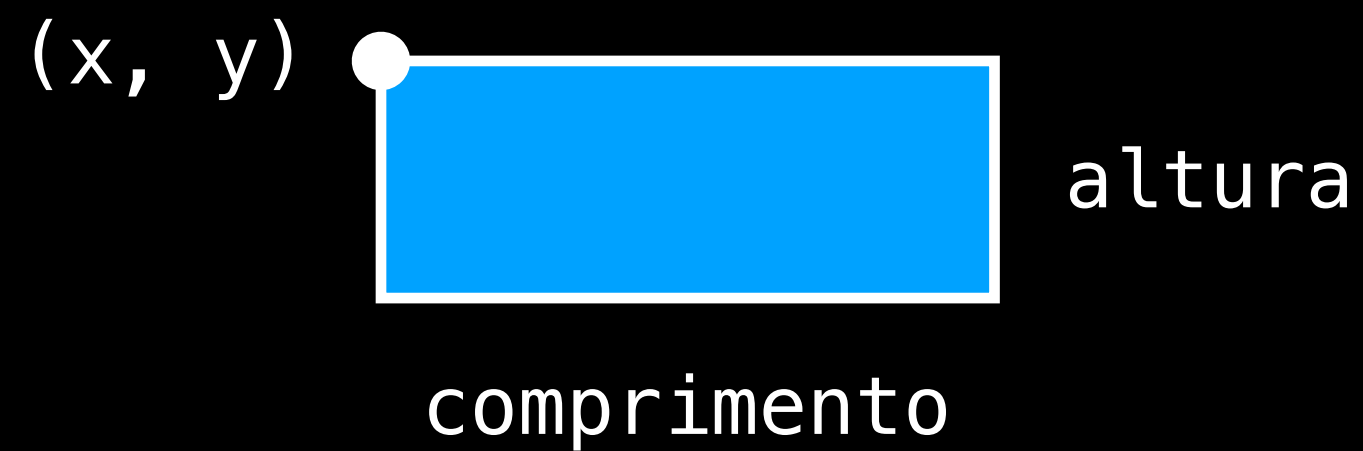
# Bandeira 3
desenha_retangulo(0, 230, 100, 30, 'white')
```

Shell

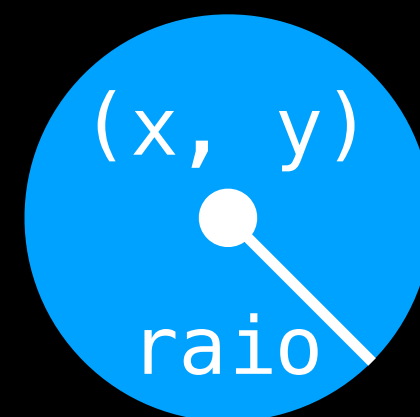
```
>>>
```

Objetivo: Implementar Funções Auxiliares de Desenho das Bandeiras

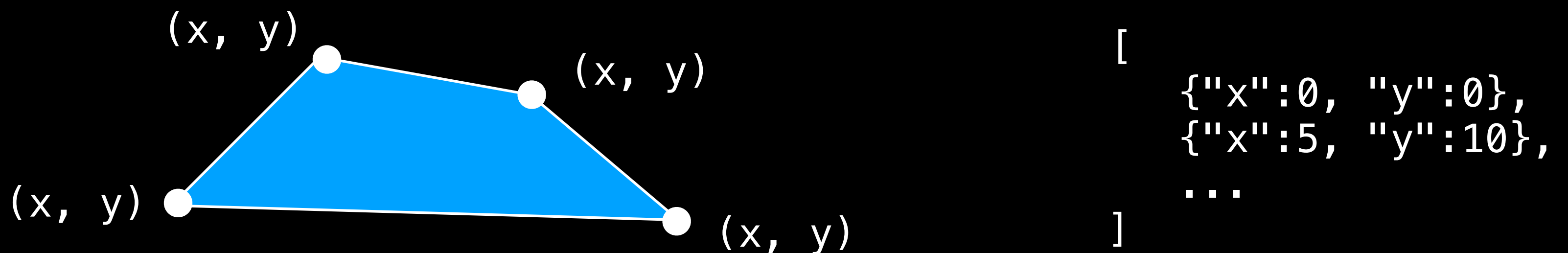
`desenhar_retangulo(x, y, comprimento, altura, cor)`



`desenhar_circulo(x, y, raio, cor)`



`desenhar_poligono(lista_de_pontos, cor)`



Descrição das Funções



## Implementação

Implemente a função **desenha\_retangulo**.

↳ DICA: lembre-se que as coordenadas  $x$  e  $y$  representam o canto esquerdo superior.

Implemente a função **desenha\_circulo**.

↳ DICA: tome cuidado ao definir a posição inicial do desenho em relação ao centro.

Implemente a função **desenha\_poligono**.

↳ DICA: vá para o primeiro ponto, passe pelos demais e depois volte ao primeiro.

Aperfeiçoamento

```

{
  "nome": "Brasil",
  "elementos": [
    {
      "tipo": "retângulo",
      "x": -50,
      "y": 30,
      "comprimento": 100,
      "altura": 60,
      "cor": "green"
    },
    {
      "tipo": "polígono",
      "pontos": [
        {"x": 0, "y": 25},
        {"x": -45, "y": 0},
        {"x": 0, "y": -25},
        {"x": 45, "y": 0}
      ],
      "cor": "yellow"
    },
    {
      "tipo": "círculo",
      "x": 0,
      "y": 0,
      "raio": 10,
      "cor": "blue"
    }
  ]
}

```



Thonny - /Users/Jan/Desktop/00c\_aperfeicoamento.py @ 13 : 23

```
1 from json import load
2 from turtle import *
3
4 # Copie as funções da Implementação aqui
5
6
7
8 # Implemente a função abaixo
9 def desenha_bandeira(dicionario_do_pais):
10     return
11
12 dicionarios_de_paises = load(open('paises.json', encoding="UTF-8"))
13 desenha_bandeira(dicionarios_de_paises[0])
14
15
16 # Ao clicar na tela, solicitar o nome de um país, busque-o na lista de
17
18
```

Shell x

Python 3.7.7 (bundled)

>>> |

## Parte 1: Implementação da Função desenha\_bandeira

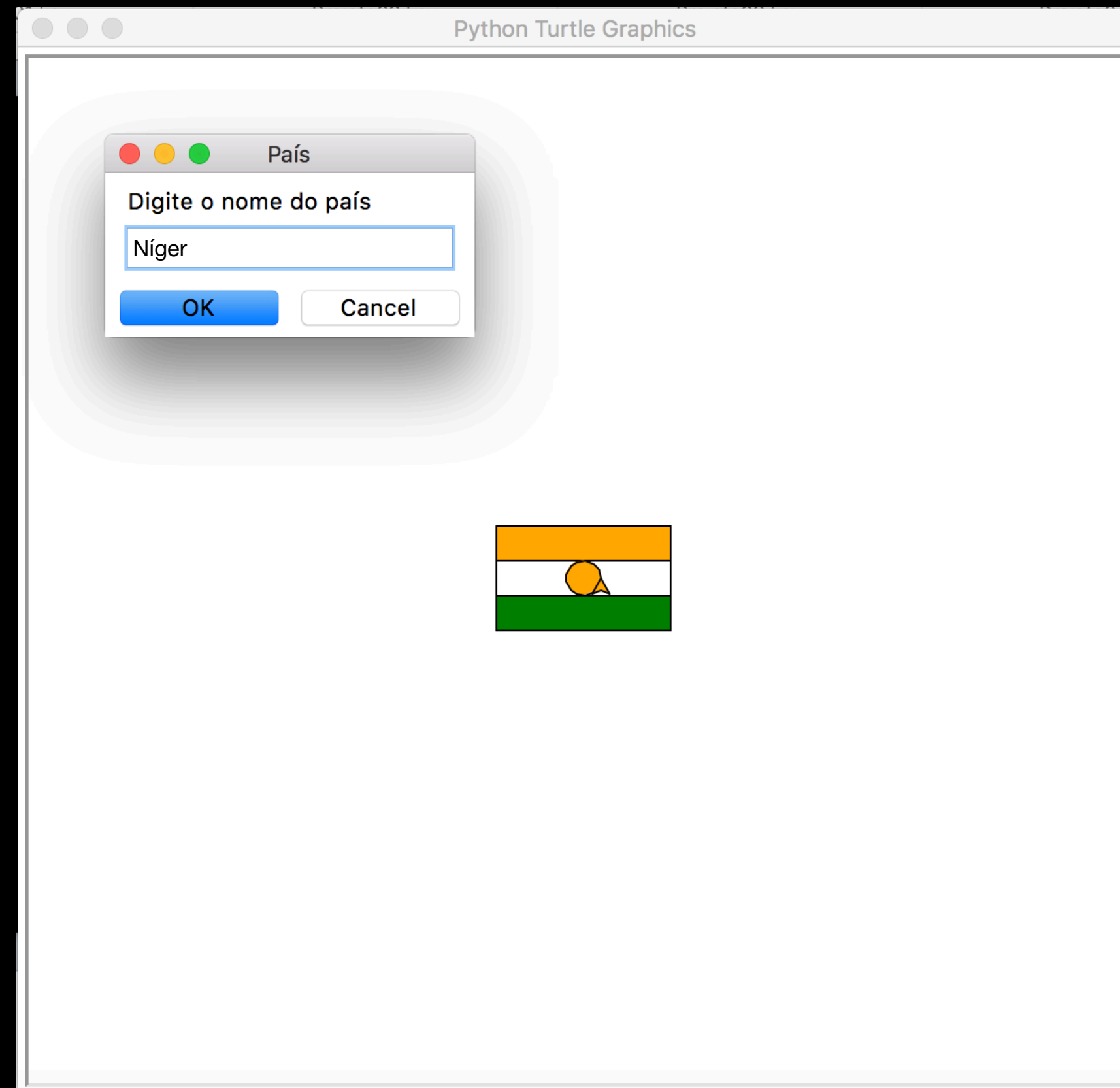


paises.json



```
[
  {
    "nome": "Brasil",
    "elementos": [
      {
        "tipo": "retângulo",
        ...
      },
      {
        "tipo": "polígono",
        ...
      },
      ...
    ]
  },
  {
    "nome": "Níger",
    "elementos": [
      {
        "tipo": "retângulo",
        ...
      },
      ...
    ]
  },
  ...
]
```





Parte 2: Escolha do País a Ser Desenhado



## Aperfeiçoamento

Copie as 3 funções da etapa anterior, e chame-as ao implementar a função `desenha_bandeira`.

↳ DICA: acesse o campo "elementos" do dicionário recebido.

Associe o clique na tela a uma função sua. Nesta função, chame a `textinput` do Turtle (pesquise no Google) para solicitar o nome de um país. Em seguida, busque esse país na lista de dicionários e desenhhe-o usando a função anterior.

↳ DICA: percorra os elementos da lista global `lista_de_paises`, comparando a chave "nome" de cada um com o país solicitado.

Desafio



Desafio

Adicione a bandeira da Africa do Sul no arquivo JSON e teste seu desenho.

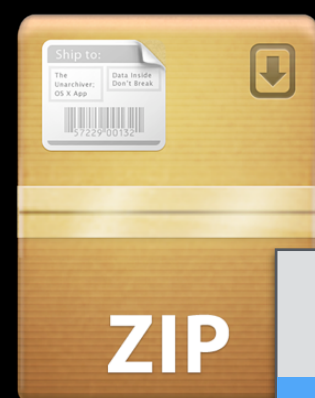


Material da Introdução

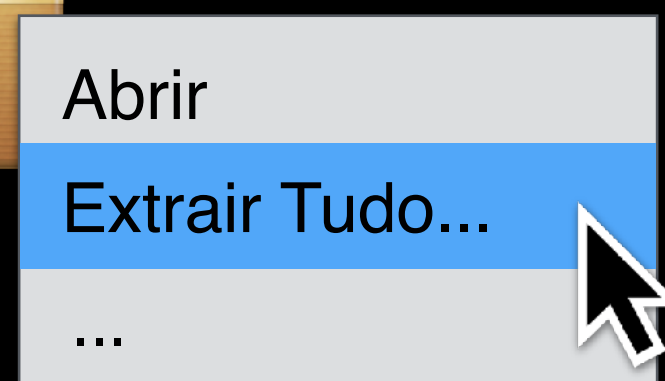


[janks.link/micro/introducao.zip](https://janks.link/micro/introducao.zip)

Material da Introdução



ZIP



Introdução



00a\_testes\_iniciais.py



00b\_implementacao.py



00c\_aperfeicoamento.py



países.json



extra00

Extração de Arquivos do Exercício