



# Algoritmos Computacionais

**Conteúdo:** Módulos  
Prof. Dsc. Giomar Sequeiros  
[giomar@eng.uerj.br](mailto:giomar@eng.uerj.br)

# Módulos

# Módulos: tipos de importação

Em Python, existem três tipos de declarações de importação, que são:

**1. Importação simples:** A importação simples é usada quando você deseja importar um módulo inteiro. É a maneira mais comum de importar um módulo. A sintaxe é:

```
import modulo
```

**2. Importação com alias:** Podemos usar um alias para renomear um módulo quando importá-lo. Isso pode ser útil se o nome do módulo for muito longo ou se houver um conflito de nome. A sintaxe é:

```
import modulo as alias
```

**3. Importação seletiva:** Podemos importar apenas partes específicas de um módulo, em vez de todo o módulo. Isso pode economizar tempo e espaço em memória, pois apenas as partes necessárias são importadas. A sintaxe é:

```
from modulo import objeto1, objeto2, ...
```

# Módulo math

---

- O módulo **math** é um módulo embutido em Python que fornece **funções matemáticas** para realizar várias operações matemáticas. Algumas das funções disponíveis no módulo math são:
- **sqrt**: Retorna a raiz quadrada de um número.
- **pow**: Retorna a potência de um número.
- **log**: Retorna o logaritmo natural de um número.
- **exp**: Retorna o valor da exponencial de um número.
- **pi**: Retorna o valor de pi (3.141592653589793).
- **sin, cos, tan**: Retorna o seno, cosseno e tangente de um ângulo em radianos, respectivamente.

# Módulo math: exemplos

- Execute o código a seguir:

```
import math

# Retorna a raiz quadrada de um número
print(math.sqrt(16)) # Saída: 4.0

# Retorna a potência de um número
print(math.pow(2, 3)) # Saída: 8.0

# Retorna o logaritmo natural de um número
print(math.log(10)) # Saída: 2.302585092994046

# Retorna o valor da exponencial de um número
print(math.exp(2)) # Saída: 7.3890560989306495

# Retorna o valor de pi
print(math.pi) # Saída: 3.141592653589793
```

# Módulo math: exemplos

- Execute o código a seguir:

```
from math import *  
  
# Retorna o seno, cosseno e tangente de um ângulo em radianos  
angulo = pi/4  
print(sin(angulo)) # Saída: 0.7071067811865476  
print(cos(angulo)) # Saída: 0.7071067811865476  
print(tan(angulo)) # Saída: 0.9999999999999999
```

# Módulo math: exemplos

- Execute o código a seguir:

```
import math

def calcula_circunferencia(raio):
    '''Retorna a circunferencia de raio r
    float -> float '''
    circunferencia = 2 * math.pi * raio
    return circunferencia
```

```
# Teste
raio = 5
circunferencia = calcula_circunferencia(raio)
print(f"A circunferência de um círculo com raio {raio} é {circunferencia:.2f}")
```



# Módulo math: exemplos

- Execute o código a seguir:

```
import math as m

def calcula_hipotenusa(c1, c2):
    '''Retorna a hipotenusa de um triangulo de catetos c1 e c2
    float, float -> float'''
    hipotenusa = m.sqrt(c1 ** 2 + c2 ** 2)
    return hipotenusa
```

```
# Teste
c1 = 3
c2 = 4
hipotenusa = calcula_hipotenusa(c1, c2)
print(f"A hipotenusa de um triângulo retângulo com catetos {c1} e {c} é {hipotenusa:.2f}")
```



# Módulo math: exemplos

- Execute o código a seguir:

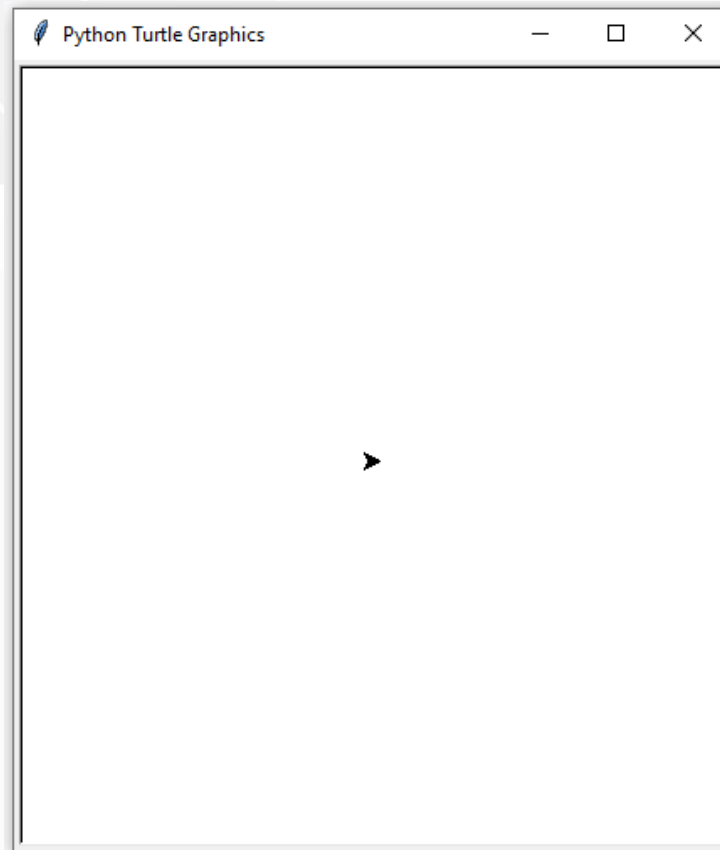
```
from math import pi

def area_circulo(r):
    '''Retorna a area de um circulo de raio r
    float -> float'''
    area = pi * (r ** 2)
    return area
```

```
# Teste
raio = 5
area = area_circulo(raio)
print(f"A área de um círculo com raio {raio} é
{area:.2f}")
```

# Módulo Turtle

- O módulo **turtle** é uma biblioteca gráfica que permite desenhar imagens em uma janela usando uma tartaruga (ou turtle, em inglês) que se move em torno da tela.



# Módulo Turtle: algumas funções

- Para usar o módulo turtle basta importar.

```
import turtle as t

#anda para frente 200 pixels
t.forward(200)
# gira 60 graus para a esquerda
t.left(60)
#anda para frente 50 pixels
t.forward(50)
# gira 90 graus para a direita
t.right(90)
#anda para tras 100 pixels
t.backward(100)
```

# Módulo Turtle: algumas funções

- Continua...

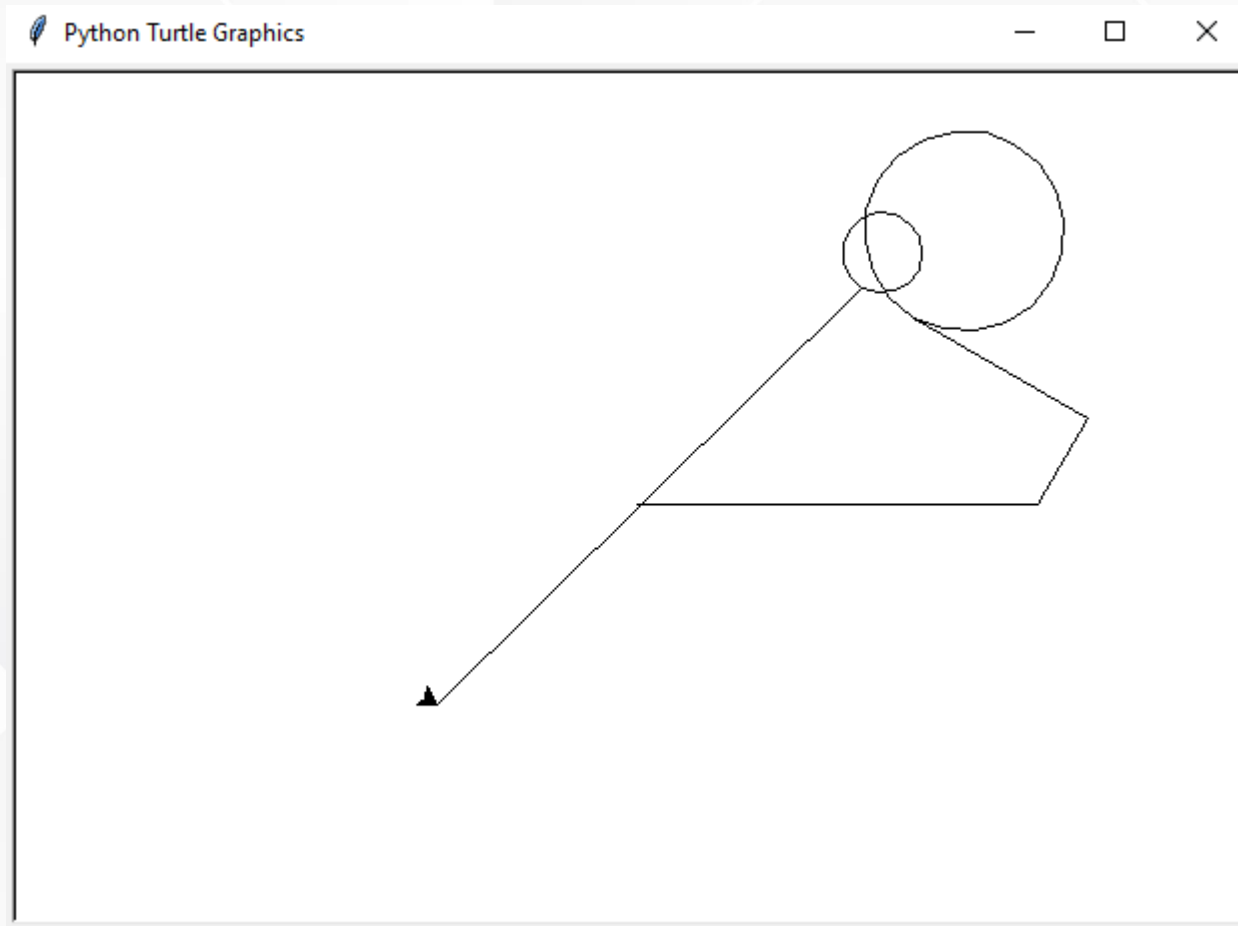
```
#desenha um círculo de raio 50
t.circle(50)

#levanta a caneta
t.up()
#anda para tras 30 pixels
t.backward(30)
#abaixa a caneta
t.down()
t.circle(20)

# va para as coordenadas x, y
t.goto(-100,-100)

# permite que a janela seja fechada
t.done()
```

## Saída



# Módulo Turtle: Exemplo 1

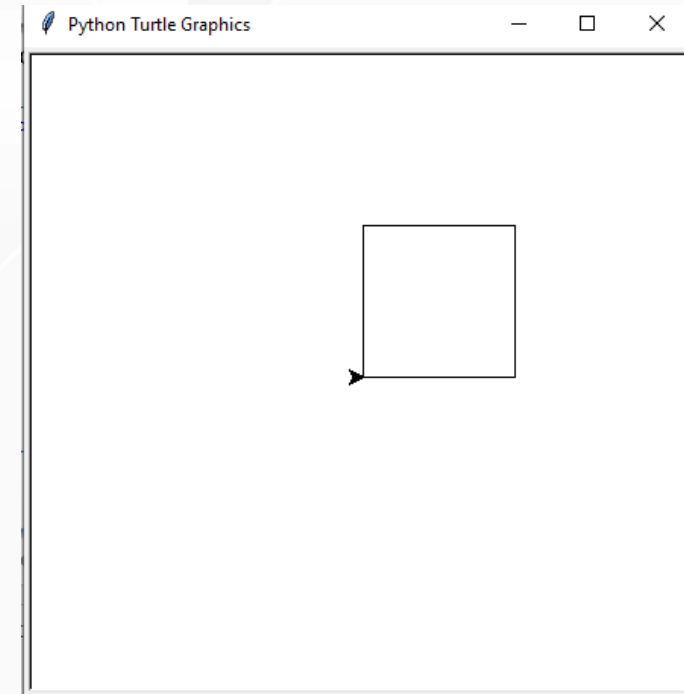
- Crie um arquivo **desenhos.py** e escreva o código a seguir e realize testes:

```
import turtle as t

def desenha_quadrado(x):
    '''desenha um quadrado de lado x'''
    t.forward(x)
    t.left(90)
    t.forward(x)
    t.left(90)
    t.forward(x)
    t.left(90)
    t.forward(x)
    t.left(90)
```

Execução:

```
>>> desenha_quadrado(100)
```



# Módulo Turtle: Exemplo 1

- Acrescente a função a seguir ao arquivo anterior e realize vários testes:

```
def desenha_figura(x, ang):  
    '''desenha uma figura de lado x  
    e angulo ang '''  
    desenha_quadrado(x)  
    t.left(ang)  
    desenha_quadrado(x)  
    t.left(ang)  
    desenha_quadrado(x)  
    t.left(ang)  
    desenha_quadrado(x)  
    t.left(ang)  
    desenha_quadrado(x)  
    t.left(ang)
```

## Execução:

```
>>> desenha_figura(100, 30)
```

