Cálculo Numérico

Laboratório | Tema 3: Interpolação

Alexandre Zabot

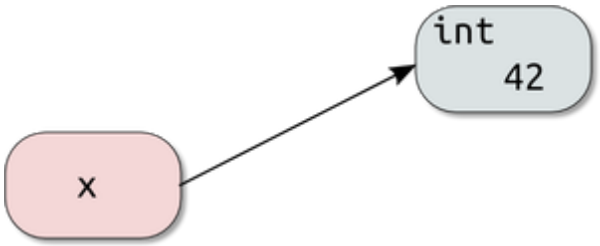
Índice

Nomes

Nomes em Python

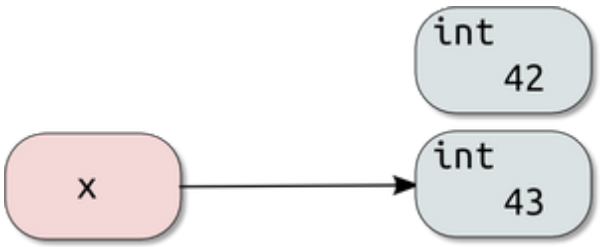
Em Python, uma variável é apenas um NOME que REFERENCIA a um OBJETO.

Nomes



$$x = 42$$

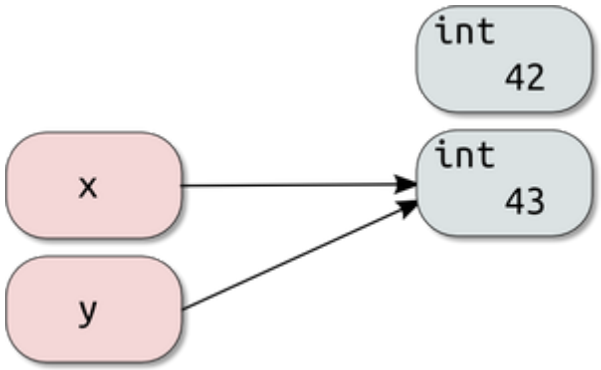
Nomes




```
x = 42
x = x + 1
```

Foi criado um novo objeto int e x foi referenciado a ele.

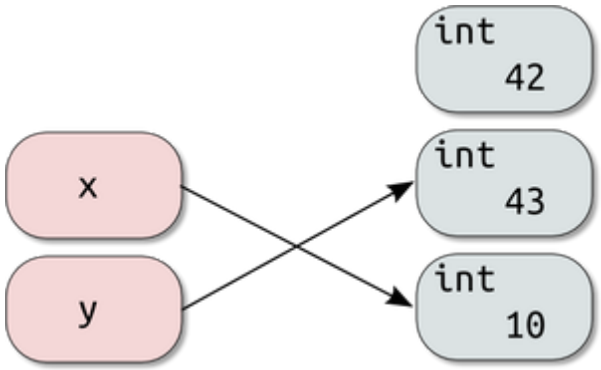
Nomes




```
x = 42
x = x + 1
y = x
```

y é um nome diferente de x, mas faz referência ao mesmo objeto.

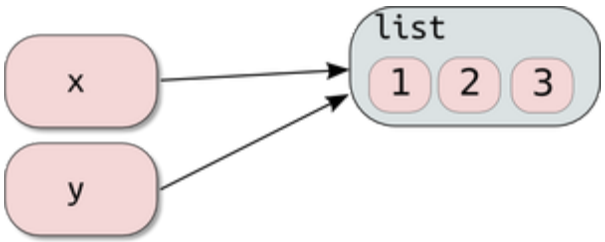
Nomes




```
x = 42
x = x + 1
y = x
x = 10
```

Agora x faz referência a outro objeto

Nomes




```
x = [1, 2, 3]
y = x
x.append(4)
print(x)
>>> [1, 2, 3, 4]
print(y)
>>> [1, 2, 3, 4]
```

Lembrando que listas são **mutáveis**, o que aconteceu agora?

Que é quem?

```
x = [1, 2, 3]
y = x
print( y is x )
>>> True
```

Copiando conteúdo

```
x = [1, 2, 3]
y = x[:] # cria uma nova lista com todo conteúdo de x e
    atribui a y
print( y is x )
>>> False
```

Copiando conteúdo

```
import copy

x = [1, 2, 3]
y = copy.copy(x)
print( y is x )
>>> True
```

Copiando conteúdo

```
import copy

x = [ [1, 2], [3, 4] ]
y = copy.copy(x)
print( y is x )
>>> False

print( y[0] is x[0] )
>>> True
```

Copiando conteúdo

```
import copy

x = [ [1, 2], [3, 4] ]
y = copy.deepcopy(x)
print( y[0] is x[0] )
>>> False
```

Índice

Nomes em funções

Procure entender o seguinte script em termos do que vimos antes

```
def func(x, y):
    x = x + 1
    y.append(4)

x = 1
y = [1, 2, 3]
func(x, y)
print('x:', x)
>>> x: 1
print('y:', y)
>>> y: [1, 2, 3, 4]
```

Nomes

Nomes e funções em Python

Para qualquer objeto passado para uma função, é feita uma cópia da referência do objeto para o escopo local da função

Ou seja, se passo um objeto para uma função, é criado um novo nome dentro da função que apontará para a referência do nome antigo. O comportamento desse novo nome dependerá do tipo de objeto: mutável ou imutável.

Nomes locais são destruídos após o *return*.

Índice

Ler arquivos de dados

```
import numpy as np

nome="dados.txt"

M = np.loadtxt(nome, dtype='float', comments='#')
print(M)

x,z = np.loadtxt(nome, dtype='float', comments='#', usecols
    =(0, 2), unpack=True)
print(x)
print(z)
```

http://docs.scipy.org/doc/numpy/reference/generated/numpy.loadtxt.html

Índice

Atividade

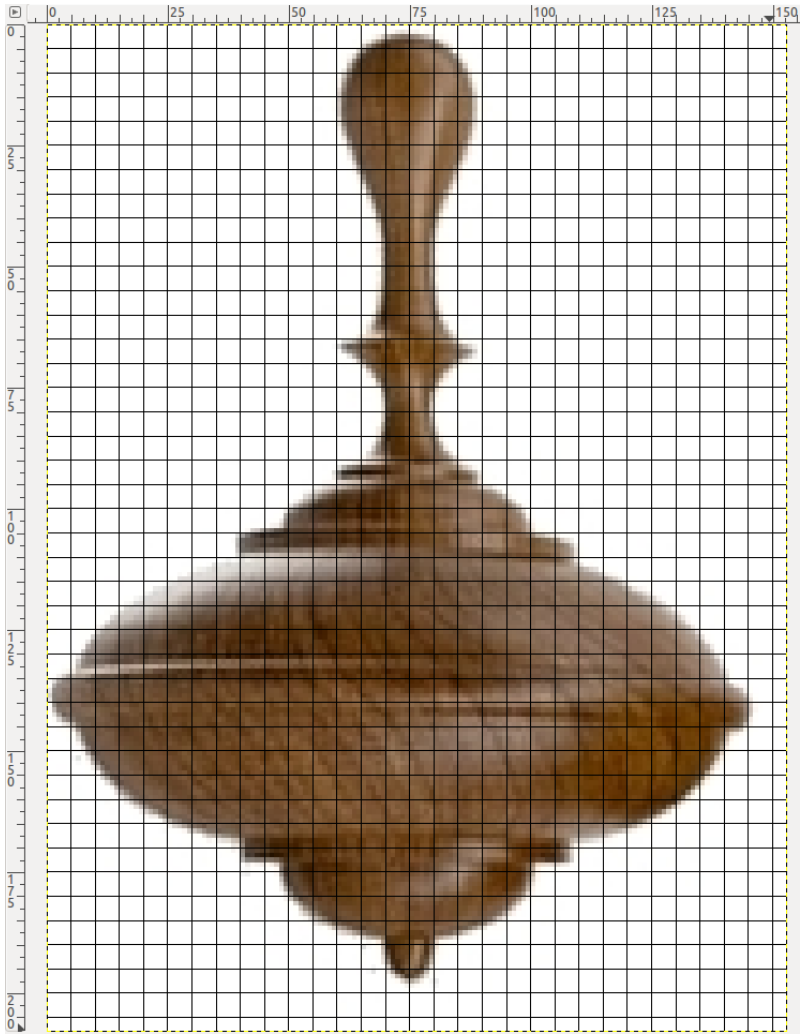


Figura: Calcule o volume do pião.

Passo a passo

- Use a imagem para encontrar um conjunto de pontos $\{(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)\}$ que descrevam a lateral do pião.
- Calcule uma função Spline Cúbica com condições de contorno livres ($S(y)$) que descreva o conjunto de pontos.
- Integre o sólido de revolução gerado pela rotação de $S(y)$ em torno do eixo vertical que passa pelo meio do pião, em $x = x_c$.

Base Teórica

A Spline Cúbica obtida será uma função de y. O volume do pião é obtido através da revolução em torno do eixo $x = x_c$, portanto,

$$V = \int_{y_{min}}^{y_{max}} \pi [S(y) - x_c]^2 dy \tag{1}$$

Programa

- Ler os pontos de um arquivo de dados em duas colunas: x e y
- Montar a matriz da Spline Cúbica com condições de contorno livres
- Resolver o SL usando o Método de Thomas
- Fazer o gráfico da Spline para ver se ficou bom
- Calcular a Integral por um dos métodos estudados

Construir o programa com base nos critérios de **EFICIÊNCIA**, **ROBUSTEZ** e **REUSABILIDADE**.