Introdução à Computação em Python

PY101

Prof. Rodrigo L. Alvarino [1]

[1] Instituto de Física
Universidade de São Paulo

April 10, 2024

Mapa de Blocos

Introdução
Algoritmos
Intro Python
Condicionais
Loops
Listas
Matrizes

Dicionários
Funções
Arquivos

Classes
Precisão
Execução Associada

IFUSP
Instituto de Física da USP

Bloco 1: Introdução

- Um pouco de História
-
-
-
-

IFUSP
Instituto de Física da USP

Um pouco de História

IFUSP
Instituto de Física da USP

As primeiras Máquinas de Calcular

IFUSP
Instituto de Física da USP

Bloco 2: Algoritmos

- O que é um Algoritmo

IFUSP
Instituto de Física da USP

O que é um Algoritmo

IFUSP
Instituto de Física da USP

O que são Algoritmos?

Chamamos de algoritmo toda sequência de ordens/comandos dada a um operador/computador para ser executada.

- Quando estamos aprendendo a somar e subtrair, aprendemos o algoritmo da soma
- Quando estamos aprendendo a dividir, aprendemos o algoritmo da divisão
- Quando vamos assar um bolo, seguimos o algoritmo: “Receita”.

IFUSP
Instituto de Física da USP

Algoritmo da Soma

Para somar dois números inteiros de A e B algarismos, aplicamos o seguinte algoritmo:

```
1 disponha os numeros, um sobre o outro, de modo que os
      algarismos da unidade estejam alinhados (isso alinha todas
      as outras ordens automaticamente)
2 some os algarismos da unidade.
3 escreva o algarismo da unidade do resultado da soma anterior na
       casa da unidade.
4 Se houver algarismo da dezena escreva sobre os demais
      algarismos da dezena na parte superior do dispositivo.
5 Repita o procedimento com os algarismos na coluna da ordem
      seguinte, grafando unidade abaixo e, quando houver,
      grafando algarismo da ordem superior do resultado acima da
      proxima ordem.
```

Mas isso não parece tão prático! Vamos melhorá-lo.

IFUSP Instituto de Física da USP

```
1  input: int A, int B
   output: int
3  datatype: int i ← 1, int C ← 0

5  resultado ← 0
   while i ≤ min(A,B)
7      if (A.pos(i) + B.pos(i) + C).pos(i+1) != 0
           resultado.pos(i) ← (A.pos(i) + B.pos(i) + C).pos(i)
9          C ← (A.pos(i) + B.pos(i)+C).pos(i+1)
       else
11         resultado.pos(i) ← A.pos(i) + B.pos(i) + C
           C ← 0
13     end if
       i ← i + 1
15 end while
   if C != 0
17     resultado.pos(i) ← A.pos(i) + B.pos(i) + C
   end if
19 return resultado
```

Algorithm 2.1: Algoritmo da Soma

IFUSP Instituto de Física da USP

Pseudo-Código

- Chamamos de pseudo-código um algoritmo escrito com formatação de código mas cujos termos não provém de nenhuma linguagem de programação.
- Feito exclusivamente para explicar o procedimento a ser adotado na execução do algoritmo.
- O algoritmo da soma visto anteriormente é um exemplo de pseudo-código.

IFUSP
Instituto de Física da USP

Vamos definir algumas funções básicas universais a todas às linguagens e que, inclusive, serão utilizadas em nossos pseudo-códigos.

begin – end

Comando de início (entrada) e finalização (saída) de programas, funções e métodos.

input – output

comando de definição para variáveis que serão importadas/recebidas (input) e exportadas (output) pelo programa

while

loop que recorre *enquanto* alguma asserção booleana é verdadeira.

if – else

Condicionais gerais '*Se*' e '*Caso contrário*' que executam o código aninhado neles.

IFUSP
Instituto de Física da USP

return

essa ordem pode ser interpretada de muitas formas em se tratando de pseudo-códigos:

- imprimir um resultado
- salvar um valor
- ativar/desativar uma variável booleana, etc. . .

function

Define um método/função que recebe (ou não) certas variáveis e devolve (ou não) certos valores executando o código designado a ela.

call

Chamamos um método pré-definido em algum local do código ou externo a ele.

Veremos mais sobre essas ordens quando tratarmos de funções.

IFUSP
Instituto de Física da USP

Sugestões de Exercícios Programas

- Escreva um algoritmo e pseudo-código de uma receita de bolo
- Faça um pseudo-código do algoritmo da subtração
- Escreva o algoritmo da divisão
- Faça um pseudo-código para o algoritmo da divisão

IFUSP
Instituto de Física da USP

Bloco 3: Introdução ao Python

- Sintaxe Básica
- Primeiras Funções

IFUSP
Instituto de Física da USP

Sintaxe Básica

IFUSP
Instituto de Física da USP

Variável

Uma variável é um local (pointer) de memória dedicado a armazenar, temporariamente, um valor.

Valor

Qualquer pacote de informação dentre os tipos:

- Integer (Number)
- Float (Number)
- Boolean (True/False)
- String (Texto)
- Array (List)
- Dictionary (Keyed List)
- ...

IFUSP
Instituto de Física da USP

Exemplos de aplicação

```
a = 1
b = 2.0
c = 5.75124258
isso_eh_uma_variavel = 1.0
d = True
e = False
text1 = "texto de exemplo"
text2 = "1.0"
A = [1,2,3,4,5.0]
B = [a,b,c,d,e] # [1 , 2.0 , 5.75124258 , True , False]
C = ["a","b","c","d","e"]
dic = {k1:"banana" , k2:"cereja" , k3:"morango"}
```

Listing 2: Variáveis

IFUSP
Instituto de Física da USP

Conectivos Booleanos (Lógicos)

Usamos os comparadores, conectivos lógicos, para gerar valores lógicos baseados em outros valores.

Exemplo

```
a = 2
b = 3
c = a>b # false
d = a<b # true
e = a>=b # false
f = a<=b # true
g = a==b #false
h = a!=b #true
```

Listing 3: comparativos

Operadores Booleanos

```
A = True
B = False

not A # false
not B # true

A and B # false
A or B # true
```

Listing 4: Boolean Operators

IFUSP
Instituto de Física da USP

Primeiras Funções

IFUSP
Instituto de Física da USP

print

```
print("Hello World!")
```

```
a = "hello world!"
print(a)
```

```
a = 1.0
b = "Hello"
c = "World"
print(a,b,c)
```

IFUSP
Instituto de Física da USP

Se não quisermos o espaço entre os valores printados

```
a = "Ban"
b = "a"
c = "na"
print(a,b,c, sep="")
```

Se não quisermos a quebra de linha entre prints

```
print("Hellow ", end="")
print("World")
```

O símbolo \n serve para “new line”, ou *line break*

```
print("Hellow \n World!")
```

Input

```
1
        a = input("Digite seu input aqui")
        b = float(input("Digite um numero"))
3
        c = int(input("Digite um numero inteiro"))
        d = str(input("Digite uma string (texto)"))
5
```

IFUSP
Instituto de Física da USP

Exercícios

1. Escreva um código que receba ‘base’ e ‘altura’ e calcule a área de um retângulo (e mostre)
2. Escreva um código que recebe todos os valores necessários à fórmula da gravitação universal e retorne o valor da força gravitacional.

$$F_G = G \cdot \frac{m_1 \cdot m_2}{d^2} \qquad G = 6.67 \cdot 10^{-11}$$

3. Escreva um código que receba um valor de força e massa e retorne o valor da aceleração a que o corpo está submetido
4. Escreva um código que receba a aceleração e um intervalo de tempo Δt e retorne o deslocamento escalar ΔS do corpo.

IFUSP
Instituto de Física da USP

Bloco 4: Condicionais

- If Then Else
- Else If
- Nested Conditionals

IFUSP
Instituto de Física da USP

If Then Else

IFUSP
Instituto de Física da USP

A Função If ... Then: ... Else: ...

A mais simples aplicação de condicionais é a verificação de valores diretos

```
a = 2

if a == 2:
    print("Hello World")
else:
    print("a is not 2")
```

Listing 5: Função If

Perceba que utilizamos os operadores booleanos para chamar a função condicional. i.e. a função condicional é uma função booleana da forma: “Se (asserção) for verdadeira, faça isso. Se não, faça isso”

IFUSP Instituto de Física da USP

Else If

IFUSP
Instituto de Física da USP

Várias Condições

Podemos chamar a função condicional para avaliar várias possibilidades antes de “desistir”. i.e. executar o código aninhado em ‘else:’

```
a = "potato"

if a == 2:
    print("a is the number 2")
elif a == "potato":
    print("a is the vegetable potato")
else:
    print("I don't know what is a")
```

Listing 6: else if

IFUSP
Instituto de Física da USP

Omitindo Else

Como o critério ‘else’ executa o código caso qualquer exceção aos critérios especificados seja constatada, é possível que situações imprevisíveis e até mesmo erros passem por essa chamada de condicional despercebidos.

Por esse motivo, é aconselhável omitir o critério ‘else’ e substituí-lo por um ‘elif’ que especifique as exceções esperadas pelo desenvolvedor.

IFUSP
Instituto de Física da USP

Nested Conditionals

IFUSP
Instituto de Física da USP

Condicionais Aninhados

Podemos aninhar uma chamada da função condicional dentro de outra para avaliar os critérios como melhor nos convir.

Atenção, em vista de processamento, os condicionais sempre devem ser utilizados com sabedoria! Por isso, planeje muito bem o algoritmo a ser executado.

Devemos ser capazes de dissernir o momento de utilizar nesting e o momento de explicitar as combinações condicionais utilizando os operadores ‘AND’ e ‘OR’

IFUSP
Instituto de Física da USP

Exercício Programa

Estágios da Vida: Escreva uma cadeia if-elif-else que etermine o estáfio da vida de uma pessoa. Defina um valor para a variável *age* e então:

- Se a pessoa tiver menos de 2 anos, informe que ela é um bebê
- se tiver mais de 2 menos que 4 anos, criança
- mais de 4 menos de 13, garoto(a)
- mais de 13 menos de 20, adolescente
- mais de 20 menos de 65, adulto(a)
- mais de 65, idoso(a)

IFUSP
Instituto de Física da USP

Bloco 5: Loops

- Loops Simples
- Loops Aninhados

IFUSP
Instituto de Física da USP

Loops Simples

IFUSP
Instituto de Física da USP

O que são Loops?

Toda vez que precisamos repetir um comando ou uma série de comandos muitas vezes, é útil aninha-los em um loop.
Existem dois tipos principais de loops

while

Repete o código aninhado enquanto uma condição for atendida.

foreach

executa o código com as respectivas alterações iterativas associadas a um contador que percorre uma lista.

IFUSP
Instituto de Física da USP

```
integer :: a=10 , b=10 , res = 1
integer :: i = 1, j

! Imprimir Fatorial de 10
do while (i<=a)
    res = res*i
    i = i+1
end do
print *, 'O fatorial de 10 eh: ', res

res=0
! Imprimir Terminal de 10
do j = 1 , b
    res = res + j
end do
print *, 'O terminal de 10 eh: ',res
```

Listing 7: Loops em Fortran

IFUSP Instituto de Física da USP

```
a=10
b=10
res=1
i=1

while i <=a:
    res = res*i
    i = i + 1
print('O fatorial de 10 eh: ',res)

res=0
for j in range(1,11):
    res = res + j
print('O terminal de 10 eh: ',res)
```

Listing 8: Loops em Python

IFUSP
Instituto de Física da USP

Loops Aninhados

IFUSP
Instituto de Física da USP

Problema

Suponha que se deseja imprimir todas as combinações possíveis de elementos de dois conjuntos, a dizer A e B diferentes.
Sejam os conjuntos

$$A = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\} \qquad B = \{a, b, c, d, e, f, g, h, i, j\}$$

A solução deve parecer, de alguma forma com:

```
    a1,a2,a3,a4,a5,a6,a7,a8,a9,a10
    b1,b2,b3,b4,b5,b6,b7,b8,b9,b10
    c1,c2,c3,...
    ...
```

Listing 9: Print example

Para solucionar o problema, podemos utilizar um código simples como

```
A = [1,2,3,4,5,6,7,8,9,10]
B = ['a','b','c','d','e','f','g','h','i','j']

for i in B:
    for j in A:
        print(i,j,sep='',end=' ')
    print()
```

Listing 10: Nested Loop

Resultado

```
a1 a2 a3 a4 a5 a6 a7 a8 a9 a10
b1 b2 b3 b4 b5 b6 b7 b8 b9 b10
c1 c2 c3 c4 c5 c6 c7 c8 c9 c10
d1 d2 d3 d4 d5 d6 d7 d8 d9 d10
e1 e2 e3 e4 e5 e6 e7 e8 e9 e10
f1 f2 f3 f4 f5 f6 f7 f8 f9 f10
g1 g2 g3 g4 g5 g6 g7 g8 g9 g10
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10
i1 i2 i3 i4 i5 i6 i7 i8 i9 i10
j1 j2 j3 j4 j5 j6 j7 j8 j9 j10
```

IFUSP
Instituto de Física da USP

1. **Ingredientes para uma pizza**: Escreva um laço que peça ao usuário para fornecer uma série de ingredientes para uma pizza até que o valor ‘quit’ seja fornecido. À medida que cada ingrediente é especificado, apresente uma mensagem informando que você acrescentará esse ingrediente à pizza.
2. **Ingressos para o cinema**[2]: Um cinema cobra preços diferentes para os ingressos de acordo com a idade a pessoa. Se uma pessoa tiver menos de 3 anos de idade, o ingresso será gratuito; se tiver entre 3 e 12 anos, o ingresso custará 10 reais; se tiver mais de 12 anos, o ingresso custará 15 reais. Escreva um laço em que você pergunte a idade do usuário e, então informe o preço do ingresso do cinema.
3. Utilizando cinemática escalar, modele a frenagem de um carro. O código deve emitir prints periódicos com frequência de 1 Hz da posição e do tempo do carro até o repouso.

[2] Exercício fácil ao se utilizar condicionais, mas difícil somente com loops

IFUSP
Instituto de Física da USP

Bloco 6: Listas

- Listas no Python
- A biblioteca Numpy

IFUSP
Instituto de Física da USP

Listas no Python

IFUSP
Instituto de Física da USP

Como vimos anteriormente, uma lista é da forma

```
A = [a1,a2,a3,a4,a5,a6]
```

Podemos obter o comprimento de uma lista através da função len()

```
len(A) # essa funcao devolve o valor 6 nesse caso
```

podemos selecionar um elemento da lista especificando seu INDICE. Atenção, a contagem do índice da lista começa no 0.

```
B = ["potato" , "carrot" , "melon"]
B[0] # equivale ao primeiro elemento "potato"
B[1] # equivale ao primeiro elemento "carrot"
B[2] # equivale ao primeiro elemento "melon"
```

IFUSP Instituto de Física da USP

Podemos editar item a item de uma lista especificando seu índice. Por exemplo, na lista do item anterior

```
B = ["potato" , "carrot" , "melon"]
B[1] = "strawberry"
print(B)
```

O resultado seria algo na forma

```
potato , strawberry , melon
```

IFUSP
Instituto de Física da USP

No python, a função “Faça isso para cada elemento” é abreviada como FOR, e a utilizamos com a seguinte sintaxe

```
A = [1, 2, 3, "potato", True, "Sun", 10, 15]

for i in A:
    print(i)
```

```
1
2
3
potato
True
Sun
10
15
```

Adicionando elementos ao fim da lista

utilizamos a função .append() que é própria da classe de objetos “array”

```
A = []
A.append("Sun")
A.append(2)
A.append(True)
A.append("potato")
print(A)
```

```
Sun, 2, True, potato
```

IFUSP
Instituto de Física da USP

Inserindo elementos em posição específica

Utilizamos o método *.insert(i, value)* para inserir um valor na posição i e deslocar todos os outros elementos da lista à direita.

```
A = [1,2,3]
A.insert(1,"potato")
print(A)
```

```
1 , "potato" , 2 , 3
```

Removendo elementos pela posição

Podemos usar a função *del*

```
A = [1,2,3]
del A[2] # A = [1,2]
```

Também podemos utilizar o método *.pop(i)* especificando o índice, que é por padrão a ultima posição da lista.

```
A = [1,2,3,4,5]
x = A.pop()
# A = [1,2,3,4]
y = A.pop(1)
# A = [1,3,4]
```

Note que os valores “poppados” são armazenados em variáveis

IFUSP
Instituto de Física da USP

Removendo de acordo com valor

Podemos utilizar o método *.remove(value)*

```
1 A=["potato", "tomato", 1, 2, "melon"]

3 A.remove("tomato")
# A = ["potato", 1, 2, "melon"]
```

Ordenando Listas

Podemos utilizar a função sort() para ordenar uma array

```
A = [5,2,3,4,1]
A.sort()
print(A)
```

```
1,2,3,4,5
```

O método .sort() também aceita um parâmetro booleano chamado *reverse*

```
A = [5,2,3,4,1]
A.sort(reverse=True)
print(A)
```

```
5,4,3,2,1
```

IFUSP
Instituto de Física da USP

Lista de listas

podemos produzir uma lista cujos elementos sejam outras listas

```
A= [[1,2,3],["potato","banana","melon"]]

A[0] # eh o elemento [1,2,3] que eh uma lista
A[1] # eh o elemento ["potato","banana","melon"] que eh uma
    lista

A[0][0] # eh o elemento 1
A[0][2] # eh o elemento 3
A[1][1] # eh o elemento "banana"
```

IFUSP
Instituto de Física da USP

Exercício Programa

IFUSP
Instituto de Física da USP

Questões

```
    A = [1,2,3,4,"potato"]
```

Qual a diferença entre as duas linhas de código abaixo?

```
1 B = A
  B = A[:]
```

O que ocorre ao utilizar os operadores + e - com listas?

IFUSP
Instituto de Física da USP

A biblioteca Numpy

IFUSP
Instituto de Física da USP

Bloco 7: matrizes

IFUSP
Instituto de Física da USP

IFUSP
Instituto de Física da USP

Frame Title

IFUSP
Instituto de Física da USP

Bloco 8: Dicionários

IFUSP
Instituto de Física da USP

IFUSP
Instituto de Física da USP

Frame Title

IFUSP
Instituto de Física da USP

Bloco 9: Funções

IFUSP
Instituto de Física da USP

IFUSP
Instituto de Física da USP

Frame Title

IFUSP
Instituto de Física da USP

Bloco 10: Gerenciamento de Arquivos

IFUSP
Instituto de Física da USP

IFUSP
Instituto de Física da USP

Frame Title

IFUSP
Instituto de Física da USP

Bloco 11: Classes

Orientação a Objetos

IFUSP
Instituto de Física da USP

IFUSP
Instituto de Física da USP

Frame Title

IFUSP
Instituto de Física da USP

Orientação a Objetos

IFUSP
Instituto de Física da USP

Bloco 12: Precisão

IFUSP
Instituto de Física da USP

IFUSP
Instituto de Física da USP

Frame Title

IFUSP
Instituto de Física da USP

References I