

Grupo de Pesquisa em Álgebras de Clifford e Processamento de Sinais e Imagens

**Edjair Aguiar Gomes Filho – Representante Discente pelo Centro Acadêmico de Engenharia
da Computação**

Universidade Federal do Vale do São Francisco

Minicurso de Linguagem Python

Aula 1 - Introdução

Sumário:

1) INTRODUÇÃO	2
2) TIPOS EM PYTHON	2
2.1) Tipo <i>string</i>	3
2.2) Caracteres especiais em strings.....	3
2.3) Operações com strings.....	4
2.4) Métodos úteis com strings	4
3) OPERADORES ARITMÉTICOS	5
4) MODO INTERATIVO	6
5) ENTRADA E SAÍDA DE DADOS	7
5.1) Função <i>print()</i>	7
5.2) Função <i>input()</i>	7
6) EXERCÍCIOS	8

1) INTRODUÇÃO

A linguagem Python é uma linguagem de programação descrita pelas seguintes características:

- Alto-nível: sua sintaxe se aproxima mais da linguagem humana e se distancia mais da linguagem de máquina;
- Imperativa: as variáveis são mudadas através de comando imperativos, que expressam ordem;
- Orientada a objeto;
- Funcional: utiliza funções matemáticas para avaliar a mudança de variáveis;

Uma das características da linguagem é a fácil leitura do seu código, que permite a escrita de códigos mais enxutos, mais simples e com menos linhas se comparado ao mesmo programa escrito em outra linguagem. É muito utilizada para processamento de textos, dados científicos, computação gráfica de páginas da web e no ramo da inteligência artificial.

[!] Curiosidade: O nome *Python* tem origem no grupo britânico *Monty Python*, criador do programa [*Monty Python's Flying Circus*](#) – um programa de humor que foi ao ar pela primeira vez na televisão no ano de 1969.

2) TIPOS EM PYTHON

A parte interessante do Python é que essa é uma linguagem de tipos dinâmicos: ou seja, a declaração de tipo da variável não precisa ser feita. A própria linguagem é capaz de escolher que tipo usar para cada variável, podendo alterá-lo durante a execução do programa.

Aqui, teremos quatro tipos numéricos:

- Inteiro (int);
- Ponto flutuante (float);
- Booleano (bool);
- Complexo (complex);

Todos esses tipos suportam as operações algébricas comuns (adição, subtração, multiplicação e divisão) e também podem se relacionar. As figuras de 1-4 mostram alguns exemplos do funcionamento de tipos em Python.

```
>>> a = 1
>>> type(a)
<type 'int'>
```

Figura 1 – Tipo int

```
>>> a = 1.0
>>> type(a)
<type 'float'>
```

Figura 2 – Tipo float

```
>>> a = True
>>> type(a)
<type 'bool'>
```

Figura 3 – Tipo bool

```
>>> a = 4+3j
>>> type(a)
<type 'complex'>
```

Figura 4 – Tipo complex

2.1) Tipo *string*

Uma string é um vetor de caracteres. Quando dizemos que uma variável tem o tipo string, é porque nela contém um caractere, uma frase ou até mesmo um texto.

Em Python, strings podem ser escritas usando aspas simples ou aspas duplas. Ao usar aspas simples, torna-se mais fácil incluir aspas dentro da string.

- 'Isso é uma string em Python'
- "Isso também é uma string em Python"
- 'Ele disse "Alô?" e desligou o telefone'
- "Gota d'água"

2.2) Caracteres especiais em strings

Caracteres especiais são alguns caracteres que tornam mais fácil a manipulação de strings. Por exemplo:

\n	Quebra a linha da string
\t	Adiciona um <i>tab</i> à string

Tabela 1 – Caracteres especiais

Com isso, podemos tornar nossas strings mais esteticamente bonitas e/ou organizadas.

2.3) Operações com strings

Por se tratar de um vetor de caracteres, podemos muito bem manipular as strings. Para isso, temos alguns operadores e métodos especiais. Os principais são:

- **Concatenação (+):** Une duas strings em uma só;
- **Comprimento (len(nome_da_string)):** Devolve o tamanho da string que está sendo manipulada;
- **Repetição (*):** Repete o que está entre as aspas um número determinado de vezes.
- **Indexação e slicing ([]):** Por se tratar de um vetor, cada caractere da string tem uma indexação associada. Por exemplo, na string 'Sinais', temos 6 caracteres. A posição [0] é a letra 'S', enquanto a posição [4] é a letra 'i'.

A Tabela 2 nos dá alguns exemplos de operações e o que é armazenado na variável.

s = 'computational'	'computational'
t = 'linguistics'	'linguistics'
cl = s + ' ' + t	'computational linguistics'
l = len(cl)	25
u = '-' * 6	-----
c = s[3]	p
x = cl[11:16]	al li
y = cl[20:]	stics

Tabela 2 – Operações com strings

2.4) Métodos úteis com strings

Podemos também alterar uma string ao utilizar alguns métodos simples. Podemos, por exemplo, procurar uma sílaba, contar quantas vezes uma sílaba aparece na string e muito mais. A Tabela 3 mostra alguns exemplos de uso desses métodos.

s = 'example'	'example'
s = s.capitalize()	'Example'
t = s.lower()	'example'
s = s.replace('amp', 'M')	'exMle'
i = t.find('xa')	1
n = t.count('e')	2

Tabela 3 – Métodos úteis com strings

3) OPERADORES ARITMÉTICOS

Os operadores são os símbolos que atuam sobre as variáveis. Citamos:

- Operadores aritméticos:

+	Adição
-	Subtração
*	Multiplicação
/	Divisão
%	Módulo
**	Exponenciação
+=	Adiciona um valor à variável
-=	Subtrai um valor da variável

Tabela 4 – Operadores aritméticos

- Operadores comparativos:

>	Maior que
<	Menor que
==	Igual a
>=	Maior ou igual a
<=	Menor ou igual a
!=	Diferente que

Tabela 5 – Operadores comparativos

- Operadores lógicos;
- Operador de atribuição (=).

4) MODO INTERATIVO

Primeiramente, é possível configurar o interpretador Python para ser chamado diretamente pelo prompt de comando.

Baixe o Python 3.8 [aqui](#).

Você pode realizar a configuração do path através [desse tutorial](#).

Ao chamar o interpretador Python diretamente pelo prompt de comando (simplesmente digitando *python*), você entrará no modo interativo. Esse modo nos possibilita “conversar” diretamente com o prompt de comando, declarando variáveis, funções e tudo o que você pode fazer num programa interessante.

Podemos começar inicialmente pelo modo interativo, pois ele é muito interessante para aprendermos do zero.

```
C:\Users\USUARIO>python
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul  8 2019, 19:29:22) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> a = 4
>>> b = 10
>>> c = a + b
>>> c
14
>>> myname = "Edjair"
>>> myname
'Edjair'
```

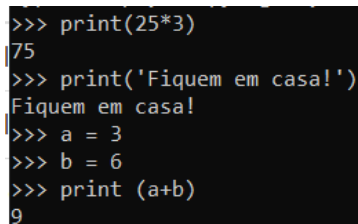
Figura 5 – Exemplo de programa simples no modo interativo

5) ENTRADA E SAÍDA DE DADOS

A entrada e a saída de dados referem-se à dados do mundo externo sendo expressos digitalmente e vice-versa, respectivamente.

5.1) Função *print()*

A grosso modo, a função `print()` serve para imprimir os argumentos dentro dela para a saída do programa. Ao passarmos valores ou variáveis para a função `print`, elas são mostradas no terminal. Um exemplo simples é mostrado na Figura 6.



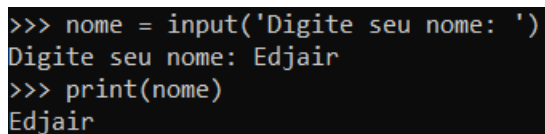
```
>>> print(25*3)
75
>>> print('Fiquem em casa!')
Fiquem em casa!
>>> a = 3
>>> b = 6
>>> print (a+b)
9
```

Figura 6 – Função *print()*

5.2) Função *input()*

A função `input()` faz uma pausa no programa e espera que o usuário digite algo para continuar. Para ler a entrada do usuário, a função `input` espera que, após digitada a entrada, o usuário tecele *enter*.

O `input` então lê essa entrada. A Figura 7 mostra um exemplo simples da função `input`.



```
>>> nome = input('Digite seu nome: ')
Digite seu nome: Edjair
>>> print(nome)
Edjair
```

Figura 7 – Função *input()*

Entretanto, é importante frisar que a função `input()` lê a entrada como uma string. Caso queiramos que a entrada seja um número inteiro ou float, precisamos usar as funções de conversão `int()` e `float()`. É necessário que façamos isso, uma vez que as operações aritméticas só podem ser realizadas com tipos numéricos. A Figura 8 dá um exemplo dessa conversão.

```
>>> idadeA = int(input('Digite a primeira idade: '))
Digite a primeira idade: 21
>>> idadeB = int(input('Digite a segunda idade: '))
Digite a segunda idade: 19
>>> media = (idadeA + idadeB)/2
>>> print(media)
20.0
```

Figura 8 – Utilizando `input()` com a função de conversão `int()`

6) EXERCÍCIOS

1. Faça um Programa que peça um número e então mostre a mensagem “O *número informado foi [número]*”.
2. Faça um Programa que peça dois números e imprima sua soma.
3. Faça um Programa que peça o nome do aluno, 4 notas e mostre na saída o aluno e sua média.
4. Faça um Programa que peça a temperatura em graus Fahrenheit, transforme e mostre a temperatura em graus Celsius. *Dica: $C = (5 * (F-32) / 9)$.*
5. Faça um Programa que peça o raio de um círculo, calcule e mostre sua área. *Dica: $A = \pi r^2$*