



Introdução à linguagem Python

Aula 1 - Primeiros passos com a linguagem

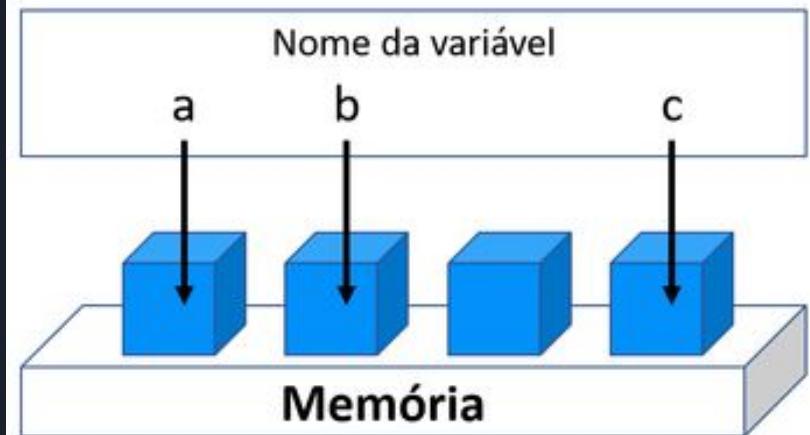
Por: Thiago Ribeiro da Silva

Graduando em Ciência da Computação

O que são variáveis?

- São espaços na memórias que podem receber valores, podendo serem alterados durante um algoritmo;
- Possuem nomes para diferenciação -> Identificadores;
- Para atribuir um valor a uma variável, utiliza-se o símbolo “=”

Representação de variáveis como “caixas” na memória do computador



O que são variáveis?

- Nomes podem ser compostos de números, letras ou _ (único caractere especial permitido);
- Não podem começar com números;
- Não podem ser iguais às palavras reservadas da linguagem (*print*, *input*, *string*, *len*).



```
1 nome = "Gustavo" #correto
2 2nome = "Pereira" #incorrecto
3 lugar01 = "" #correto
4 nome completo = "Gustavo Pereira" #incorrecto
5 _secreto = "" #correto
6 exclama! = "" #incorrecto
```



Entrada e saída de um programa

- Recebendo dados: `input()`
- Exibindo dados: `print()`

```
1 #entrada
2 nome = input("qual o seu nome?")
3 #saída
4 print(f"Olá, {nome}!")
```



Primeiro Exemplo

- Escreva um programa que recebe dois números inteiros e realiza a soma deles
- Primeira tentativa:



```
1 n1 = input("Primeiro número: ")
2 n2 = input("Segundo Número: ")
3 soma = n1 + n2
4 print(f"A soma é {soma}")
```



O que são Tipos Primitivos?

- Conjunto de tipos básicos a partir dos quais todos os outros são concebidos
- Quase todas as linguagens possuem 4 tipos básicos em comum:

NOME	Descrição
Integer (int)	Números inteiros (0, -1, 1000, 1892)
Float (float)	Números racionais (0, 0.123, 3.1415)
Boolean (bool)	Valores lógicos (False, True)
String (str)	Textos (“ola”, “A”, “”, “123”)

O que são Tipos Primitivos?

- Python é uma linguagem dinamicamente tipada, ou seja, o tipo de uma variável é definido em tempo de execução, podendo mudar a todo momento;
- Para converter um tipo em outro (se for possível), basta executar **nomedotipo(variável)**.

```
1 # tipos primitivos simples
2 inteiro = 124
3 racional = 3.1415
4 booleano = True
5 texto = 'Olá, mundo!'
6
7 #conversão
8 inteiro = str("124") #o texto 124 vira numero 124
```

Corrigindo o primeiro Exemplo

- Input manda valores em formato de string naturalmente, então é necessário converter para inteiro para realizar a soma corretamente:



```
1 n1 = int(input("Primeiro número: "))
2 n2 = int(input("Segundo Número: "))
3 soma = n1 + n2
4 print(f"A soma é {soma}")
```



Operadores aritméticos

- Operadores para realização de operações clássicas da aritmética

OPERADOR	DESCRIÇÃO	OPERADOR	DESCRIÇÃO
+	Soma	//	divisão inteira
-	Subtração	%	módulo (resto)
*	Multiplicação	**	potenciação
/	Divisão		

Operadores aritméticos

- Operadores para realização de operações clássicas da aritmética

```
1 n1 = 12
2 n2 = 34
3 soma = n1 + n2 #46
4 subtracao = n1 - n2 #-22
5 multiplicacao = n1 * n2 #408
6 divisao = n2 / n1 #2.833333
7 divisao_inteira = n2 // n1 #2
8 resto = n2 % n1 #10
9 potencia = n1 ** n2 # 4.9e36
```



Operadores de atribuição

- Formas reduzidas de atribuir um valor ao valor atual de uma variável

OPERADOR	EXEMPLO	EQUIVALENTE
<code>+=</code>	<code>x += 1</code>	<code>x = x + 1</code>
<code>-=</code>	<code>x -= 1</code>	<code>x = x - 1</code>
<code>*=</code>	<code>x *= 1</code>	<code>x = x * 1</code>
<code>/=</code>	<code>x /= 1</code>	<code>x = x / 1</code>
<code>%=</code>	<code>x %= 1</code>	<code>x = x % 1</code>

Manipulando Strings

- Strings em python são encadeamentos de caracteres, onde cada caractere e fatia da string pode ser manipulada por funções internas;
- É possível acessar um único caractere com **string[i]**, onde **i** se refere a posição do caractere;
- Para fatiar a string em um intervalo específico, basta usar **string[inicio:final:step]**, onde **step** se refere a quantos caracteres serão pulados, por padrão, 1.





Manipulando Strings



```
1 string = 'porco aranha'
2 print(string[3]) #c
3 print(string[6:]) #aranha
4 print(string[0:5]) #porco
5 print(string[0::3]) #pcan
```



Manipulando Strings

- Algumas das principais funções internas para manipulação de strings estão descritas aqui:

FUNÇÃO	DESCRIÇÃO
len(string)	retorna o tamanho
string.count(" ", início, fim)	conta a quantidade de ocorrências de uma outra string
string.find(" ")	Apresenta a posição de uma string dentro de outra
string.replace("termo1", "termo2")	Troc a um trecho da string por outro passado



Manipulando Strings



```
1 print(len(string)) #12
2 print(string.count('a')) #3
3 print(string.find('aranha')) #6
4 print(string.replace('porco', 'gato')) #gato aranha
```



Manipulando Strings

- Algumas das principais funções internas para manipulação de strings estão descritas aqui:

FUNÇÃO	DESCRIÇÃO
string.lower(), string.upper()	torna a string em minúscula / maiúscula
string.capitalize(), string.title()	torna string com primeiro caractere maiúsculo/ os primeiros caracteres de palavras maiúsculos
string.split("")	Quebra uma string a cada encontro com algum caractere/string -> retorna lista
"".join(string)	Une os elementos na lista por meio de um caractere/string



Manipulando Strings



```
1 print(string.lower()) #porco aranha
2 print(string.upper()) #PORCO ARANHA
3 print(string.capitalize()) #Porco aranha
4 print(string.title()) #Porco Aranha
5 lista = string.split(' ')
6 print(lista) #['porco', 'aranha']
7 print('-'.join(lista)) #porco-aranha
```

Obrigado!

