

Aula 2

Lógica de Programação e Algoritmos

Prof. Vinicius Pozzobon Borin

1

Conversa Inicial

2

- O objetivo desta aula é darmos os primeiros passos em programação
- Construção de nossos primeiros algoritmos

3

- A estrutura de conteúdos desta aula é a seguinte:
 - Onde programar em Python?
 - Quais partes compõem um algoritmo computacional?

4

- Como escrevemos algo na tela?
- Como manipulamos dados e variáveis?
- Como realizamos operações aritméticas?
- Como lemos informações do teclado?

5

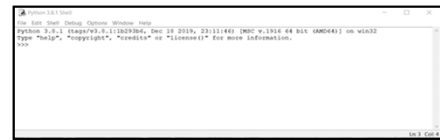
Ambientes de desenvolvimento

6

- Python requer instalação em ambientes Windows
 - <<https://www.python.org/downloads/>>
- Nativamente instalado em ambientes Linux

7

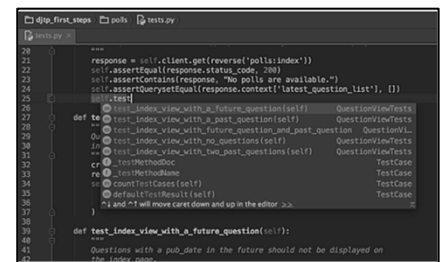
- Junto ao Python vem instalado o IDLE



IDLE Python Shell 3.8 em ambiente Windows

8

- PyCharm – Ambiente de Desenvolvimento Integrado (*Integrated Development Environment* [IDE])
 - <<https://www.jetbrains.com/pt-br/pycharm/>>



Fonte: JetBrains, [5.4]

9

10

- Google Colab – Jupyter Notebook
 - <<https://colab.research.google.com/>>



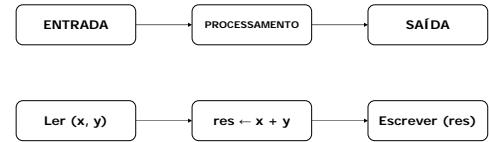
11

- Agora vamos à ferramenta

12

Ciclo de processamento de dados

13



Fonte: Borlin, 2020

14

O primeiro programa

- ▀ Vamos praticar no Python

15

Atenção ao programar

- ▀ Sempre verifique cada caractere digitado
- ▀ Caracteres maiúsculos e minúsculos são distintos
- ▀ Veja o exemplo no Google Colab

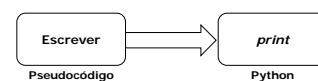
16

- ▀ Abriu aspas? Feche
- ▀ Abriu parênteses? Feche
- ▀ Cuidado com os espaços

17

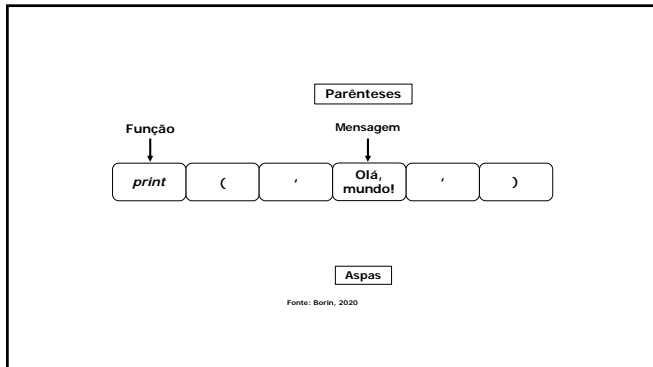
Função de saída

- ▀ **Print**: comando, instrução, função



Fonte: Borlin, 2020

18

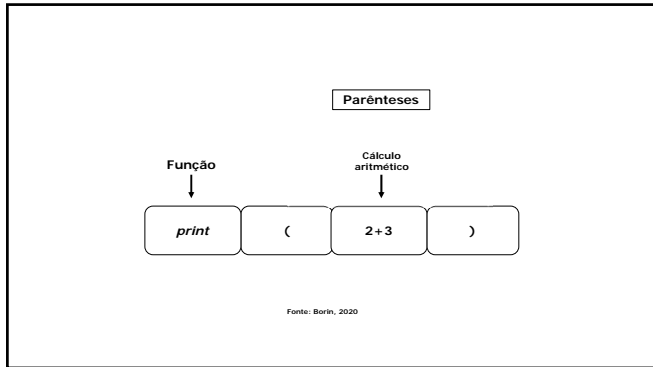


19

Função de saída

- Vamos praticar no Python

20



21

- Vamos praticar no Python

22

Operadores e operações matemáticas

Pseudocódigo	Python	Operação
+	+	Adição
-	-	Subtração
*	*	Multiplicação
/	/	Divisão (com casas decimais)
Não existe	//	Divisão (somente a parte inteira)
MOD	%	Módulo/resto da divisão
^	**	Exponenciação ou potenciação

23

- Atenção à ordem de precedência dos operadores

$$10 \times \left(\frac{5 + 7}{4} \right)$$

24

- Vamos praticar no Python

25

Variáveis, dados e seus tipos

26

Dados

- “Sequência de símbolos quantificados ou quantificáveis” (Puga, 2009, p. 18)
- Valores fornecidos via entrada e manipulados ao longo do programa

27

Variável

- Nome dado a uma região da memória do programa
- Sempre que você invocar o nome de uma variável no programa, seu bloco de memória será automaticamente carregado da RAM

28

Atribuição

- Lê-se: a variável “nota” recebe o dado 8,5

`nota = 8.5`



Símbolo de atribuição

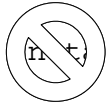
29

- Vamos praticar no Python

30

Regras para nomes de variáveis

- Nunca inicie o nome de uma variável com um número



31

Regras para nomes de variáveis

- Inicie o nome de uma variável com uma letra ou sublinha

nota ♥ _nota ♥

32

- Números, letras e sublinhas podem ser empregados à vontade no meio

nota5 ♥ _n_o_t_a_ ♥

!nota ⓧ

33

- Python permite o uso de letras com acentuação
- Não é recomendado

Python 3 Python 2
preço ♥ preço ⓧ

34

Tipos primitivos de dados

- Numérico (inteiro e ponto flutuante)
- Caractere
- Literal/booleana

35

Variáveis numéricas

- Quando queremos realizar operações aritméticas
- Inteiro (int) – números sem casas decimais
- Ponto flutuante (*float*) – números com casas decimais

36

Variáveis lógicas/booleanas

- Armazenam um *bit*. Dois estados:
 - Verdadeiro ou falso
 - Nível lógico alto ou baixo
 - 1 ou 0

37

- Lista de operadores lógicos em Python e em pseudocódigo

Python	Pseudocódigo	Operação
==	=	Igualdade
>	>	Maior que
<	<	Menor que
>=	>=	Maior ou igual a
<=	<=	Menor ou igual a
!=	<>	Diferente

38

- Vamos praticar no Python

39

Variáveis de cadeias de caracteres (*strings*)

- Armazenam conjuntos de símbolos encadeados, incluindo acentuação, pontuação etc.

40

Tabela ASCII

- 8 bits

Dec	Hx	Oct	Chr	Dec	Hx	Oct	Chr
32	20	040	Space	64	40	100	@
33	21	041	!	65	41	101	A
34	22	042	"	66	42	102	B
35	23	043	#	67	43	103	C
36	24	044	\$	68	44	104	D
37	25	045	%	69	45	105	E
38	26	046	&	70	46	106	F
39	27	047	'	71	47	107	G
40	28	050	(72	48	110	H
41	29	051)	73	49	111	I
42	2A	052	*	74	4A	112	J
43	2B	053	+	75	4B	113	K
44	2C	054	,	76	4C	114	L

Fonte: ASCII Table, [5.d.]

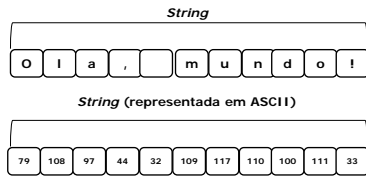
41

- UNICODE
- 32 bits



42

Variáveis de cadeias de caracteres (*strings*)



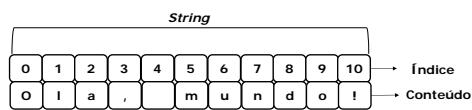
Fonte: Borlin, 2020

43

- Índice do caractere – número inteiro que indica onde o caractere está dentro da *string*
- A contagem do índice inicia sempre em zero

44

Variáveis de cadeias de caracteres (*strings*)



Fonte: Borlin, 2020

45

- Vamos praticar no Python

46

Manipulações com *strings*

47

Concatenação

- Juntar/somar *strings*
- Vamos praticar no Python

48

Composição

- Juntar diferentes variáveis e *strings*
- Vamos praticar no Python

49

Composição

- Lista de marcadores de posição

Marcador	Tipo
%d ou %i	Números inteiros
%f	Números de ponto flutuante
%s	<i>Strings</i>

50

- Vejamos novamente no Python

51

Composição moderna

'Você tirou %d na disciplina de %s'
% (nota, disciplina)

'Você tirou {} na disciplina de {}'
.format(nota, disciplina)

52

Fatiamento

- Podemos recortar/fatiar um pedaço da *string*
- Vamos praticar no Python

53

Tamanho (*length*)

- Podemos descobrir o tamanho da cadeia de caracteres com uma função chamada *len*
- Vamos praticar no Python

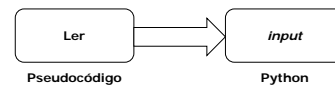
54

Função de entrada e fluxo de execução do programa

55

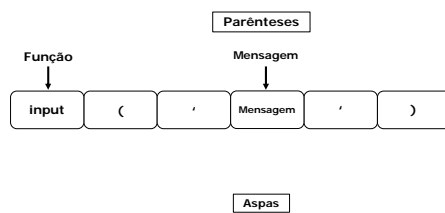
Função de entrada

- **input**: comando, instrução, função



Fonte: Borlin, 2020

56



Fonte: Borlin, 2020

57

- **Vamos ao Python**

58

Convertendo dados de entrada

- O **input** sempre retorna um dado do tipo *string*
- Se quisermos um dado numérico, utilizamos a função *int* ou *float* antes do *input*

59

Fluxo de execução do programa

- Como se dá a execução de um programa em Python?
- Vejamos no código

60

Exercício

- Desenvolva um algoritmo que solicite ao usuário dois números inteiros. Imprima a soma destes dois números na tela

Pseudocódigo

Algoritmo Exercício 2.1

Var x, y, res: inteiro

Início

Escrever("Digite um número inteiro: ")

Ler(x)

Escrever("Digite outro número inteiro: ")

Ler(y)

res = x + y

Escrever("O resultado da soma de ", x, " com ", y, " é ", res)

Fim

Fluxograma

