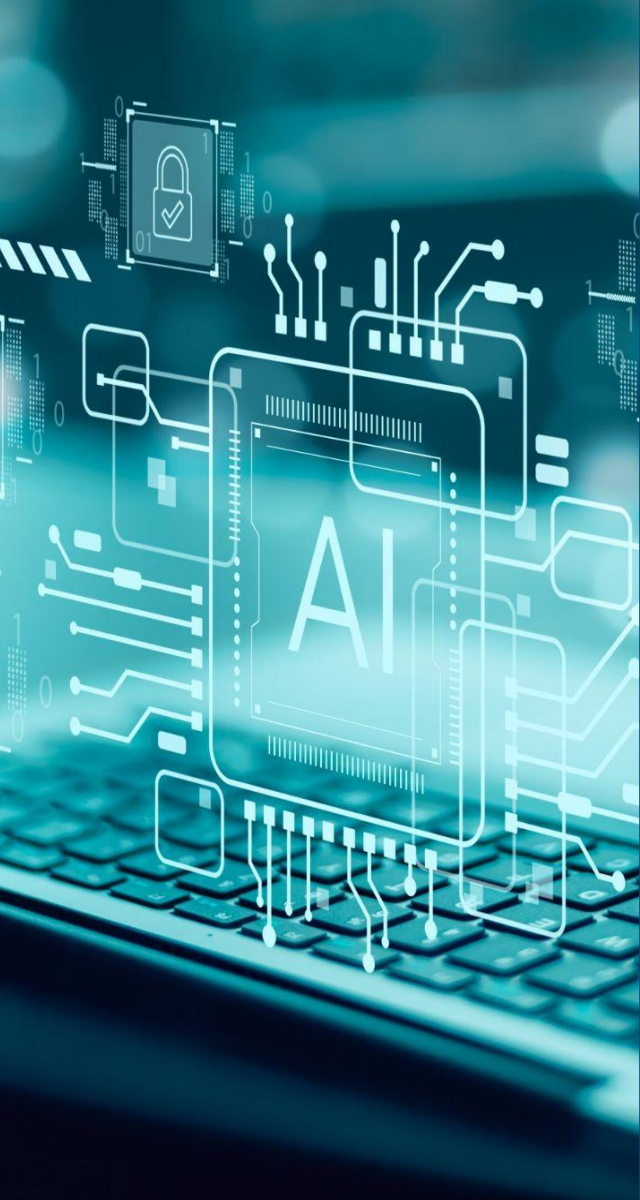




Pró-reitora de Pós-Graduação, Pesquisa e Inovação Especialização em Ciências de Dados e Analytics

Programação para Ciência de Dados
Aula 1

Prof. Rodrigo Cesar Lira da Silva
rcls@ecomp.poli.br



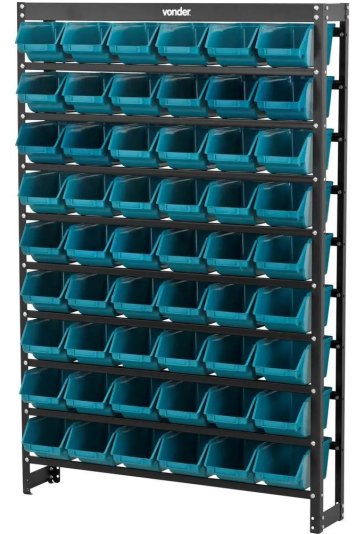
Variáveis e Entrada de Dados

Objetivos

- Apresentar o conceito de variáveis e tipos de dados
- Apresentar a função de entrada de dados em Python

Variáveis

- Em programação, variáveis são usadas para armazenar valores e para dar nome a uma área de memória do computador onde armazenamos dados.
- Podemos imaginar a memória do computador como uma grande estante, em que cada compartimento tem um nome.



Variáveis

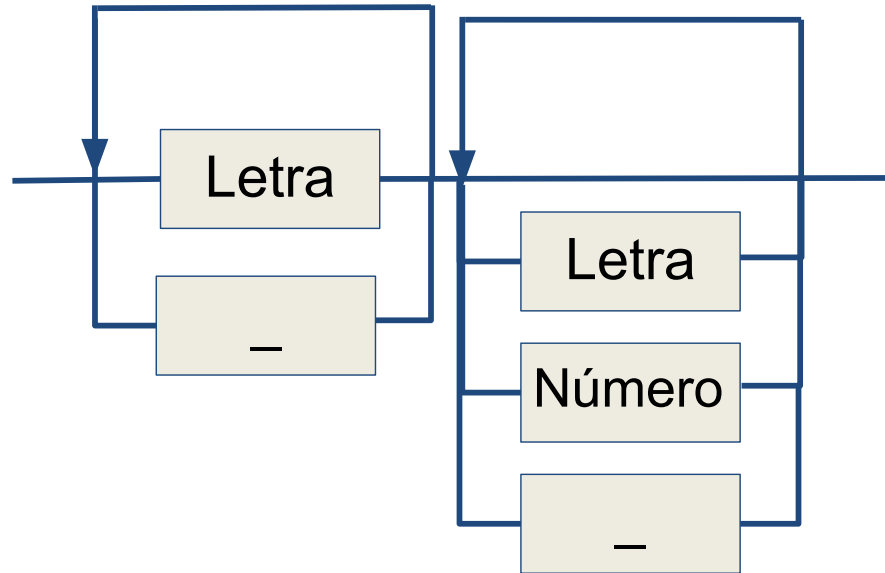
- Para armazenar um valor numa variável, nós usamos o símbolo de igualdade (=) entre o nome do compartimento e o valor que queremos armazenar.

numero = 120

Chamaremos essa operação de **atribuição**.

Nomes de Variáveis

- Em Python, nomes de variáveis devem iniciar obrigatoriamente com uma letra ou underline (_) mas também podem conter números.



Nomes de Variáveis

Nome	Válido?
a1	Sim
velocidade	Sim
velocidade90	Sim
salário_médio	Sim
salario_medio	Sim
salario medio	Não
_b	Sim
1a	Não
__2	Sim

Nomes de Variáveis

```
>>> 1a = 2
```

```
SyntaxError: invalid syntax  
.
```


Conteúdo de Variáveis

- O tipo da variável define a natureza dos dados que a variável armazena.
- Python tem vários tipos de dados, mas os mais comuns são int, float, strings e bool.
- Tipos de Dados Primitivos
 - Cada linguagem define um conjunto de tipos de dados

Variáveis Numéricas

- Dizemos que uma variável é numérica quando armazena números inteiros (int) ou ponto flutuante (float).
- Inteiros não possuem parte decimal
- Pontos flutuantes possuem partes decimais
 - Usamos . e não vírgula como separador entre a parte inteira e a fracionária.

valor = 10.2

Operadores Aritméticos

Operador	Operação
+	Adição
-	Subtração
*	Multiplicação
/	Divisão (com resultado fracionário)
//	Divisão (com resultado inteiro)
%	Módulo ou resto
**	Exponenciação ou potenciação

Operadores Aritméticos

- **Teste no interpretador interativo:**

- $2^{**}5$
- $10 / 3$
- $10 // 3$
- $10 \% 3$
- $10 - 3 * 4$

Precedência! Usem parênteses

Variável do tipo lógico

- Muitas vezes queremos armazenar um conteúdo simples: verdadeiro ou falso em uma variável. Nesse caso, usamos o tipo lógico ou booleano, o tipo **bool** em Python.
 - Em Python, **True** representa o verdadeiro e **False**, falso.

```
resultado = True  
aprovado = False
```

Operadores Relacionais

- Ao realizarmos comparações lógicas, utilizamos operadores relacionais.

Operador	Operação
==	igualdade
>	maior que
<	menor que
!=	diferente
>=	maior igual
<=	menor igual

Operadores Relacionais

- Testar:

a	=	1
b	=	5
c	=	2
d	=	1

a == b

b > a

a < b

a == d

b >= a

c <= b

d != a

d != b

Quais são os resultados das operações relacionais?

Operadores Lógicos

- Cada operador obedece a um conjunto de regras expresso pela tabela verdade desse operador.

Operador	Operação
not	não
and	e
or	ou

Precedência

Operadores Lógicos

- Primeiro as operações aritméticas, depois as de comparação, por fim as lógicas
- Expressões em parênteses são realizadas primeiro e depois:

Nível	Categoria	Operadores
7 (alto)	exponenciação	**
6	multiplicação	* / // %
5	adição	+ -
4	relação	== != <= >= > <
3	negação	not
2	conjunção	and
1 (baixo)	disjunção	or

$$3 + 4 ** (2**3)$$

Operador NOT

- Tabela Verdade

valor	not valor
True	False
False	True

Operador OR

- Tabela Verdade

a	b	a or b
True	True	True
False	True	True
True	False	True
False	False	False

Operador AND

- Tabela Verdade

a	b	a and b
True	True	True
False	True	False
True	False	False
False	False	False

Expressões Lógicas

- Operadores podem ser combinados em expressões lógicas

idade = 20 salario = 100

salario > 1000 and idade > 18

?

Entrada de Dados

- A função input é utilizada para solicitar dados do usuário. Ela recebe um parâmetro que é a mensagem a ser exibida, e retorna o valor digitado pelo usuário.

```
num = input("Digite um número: ")  
print(num)
```

```
nome = input("Digite um nome: ")  
print(nome)
```

Conversão de Entrada de Dados

- A **input** sempre retorna valores do tipo string, ou seja, não importa se digitamos apenas números, o resultado é sempre uma string.

Para resolver esse problema, utilizamos funções de conversão.

```
a = "2"
```

```
type(a)
```

```
novo_a = int(a)
```

```
type(novo_a)
```

float() , bool() e str()

Leitura Complementar

<https://realpython.com/python-data-types/>

<https://docs.python.org/3/library/stdtypes.html>

<https://www.journaldev.com/23435/python-complex-numbers-cmath>