

Introdução à programação:

- ferramenta p/ resolver problemas ambientais
 - engenheiros
- o que são linguagens de programação
 - executáveis (compiladas versus interpretadas)
 - arquivos texto .py
 - softwares (windows x linux x macbook)
 - muveem (colab) → python + texto → notebook (.ipynb)
- python 3:
 - "calculadora"
 - aulas Afonso.
 - variáveis com números (exemplo no final da aula)
2+3, a+b,
- algoritmos:
 - receita de bolo
 - manuais técnicos
 - guias de montagem
 - programas de computador

Aula 02: Afonso

Aula 03: Afonso : aula de
revisão.

- lógica ⊕ criatividade ∇
- exercitar ∇

Revisão aulas Prof. Afonso:

- algoritmo: \rightarrow propriedades:
 - garantia de término
 - exatidão (não ser ambígua)
 - efetividade (dê para resolver)
 - eficiente (há diferentes formas) $\left\{ \begin{array}{l} \text{criatividade} \\ \text{experiência} \end{array} \right.$

- estrutura de algoritmo:
 - sequencial
 - condicional: (se ..., então ...)
 - de repetição (enquanto, ..., faça ...)

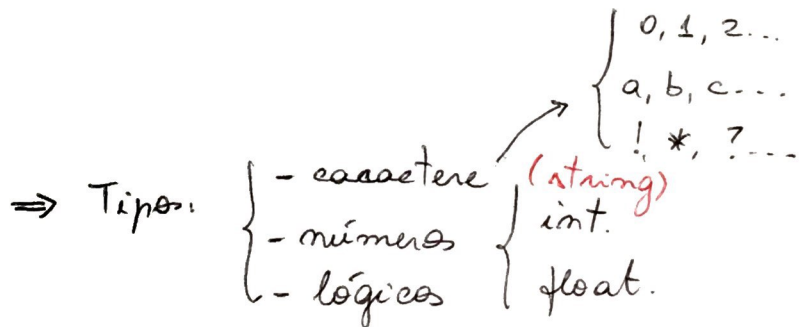
algoritmo \Rightarrow código

ex) problemas de eng. ambiental que são ou não algoritmos

- fluxograma (projeto).

- Python:
 - google colab
 - game
 - terminal

- Dados:
 - nomes
 - números
 - datas
 - V/F (lógica)



constante vs. variável

- Memória: ~~variáveis~~ dados (rótulo, valor)

\hookrightarrow identificador: regras p/ cada linguagem.

$n = 10$ (diferentes formas em python)

exemplo no colab

- comentários: #, """ ... """

Entrada e saída de dados:

- `x = input()`
- `print(x)`
- `type(x)`

atualizar:

`a, b = int(a), int(b)`

conversão → $\left\{ \begin{array}{l} \text{int}(x) \\ \text{float}(x) \\ \text{str}(x) \end{array} \right.$

- operadores aritméticos (similar à calculadora)
 - prioridade: parêntesis (associativa e distributiva)
 - exemplo colab*

- - funções embutidas (`int()`, `round()`, `abs()`)
- bibliotecas: `math` (`sin`, `cos`, `pi`)
(módulos)

- operadores relacionais: (`==`, `>`, `<`, ...) } retorna TRUE / FALSE
- lógicas: `not`, `and`, `or`
- de atribuição: `+=`, `-=`

