



IN

INFINITY SCHOOL

V I S U A L A R T C R E A T I V E C E N T E R

LP – Algoritmos & Variáveis

- 01 Algoritmos
- 02 Variáveis
- 03 Operadores aritméticos e relacionais
- 04 Saída de dados e F – String



[illegible]

LP – Algoritmos & Variáveis



Os 3 principais requisitos para ser um bom programador

- Ter um bom raciocínio lógico.
- Gostar de resolver problemas.
- Gostar de aprender.

LP – Algoritmos & Variáveis



PROGRAMAÇÃO

De forma geral, a **programação** é um processo de escrita, de testes e de manutenção de programas de computadores. É um processo de criação de um **conjunto de instruções** que dizem ao computador como realizar uma tarefa. Esse **conjunto de instruções**, por sua vez, chamamos de **Algoritmos**.

LP – Algoritmos & Variáveis

O que são Algoritmos?

Algoritmo pode ser definido como uma sequência finita de passos para a solução de um problema.

Em nosso dia a dia, executamos diversos **Algoritmos** para realização de tarefas:

- Receita de bolo;
- Trocar uma lâmpada;
- Fazer um sanduíche;
- Ir para a escola.



LP – Algoritmos & Variáveis

O que são Algoritmos?

Para cada um dos exemplos anteriores sabemos que as ações são finitas e que temos que segui-las de modo ordenado. Assim, ao final da execução de uma receita de bolo, por exemplo, teremos o bolo como produto final do nosso **Algoritmo**.

Um software, por sua vez, funciona da mesma maneira, pois executa uma série de instruções descritas (**Algoritmo**) para realização de uma tarefa.



LP – Algoritmos & Variáveis

Exemplo de como funciona um Algoritmo

#1

Coloque um fio de azeite ou óleo sobre a frigideira e, após, coloque o ovo.

#2

Na sequência, leve a frigideira ao fogo baixo e deixe fritar até o ponto de sua preferência.

#3

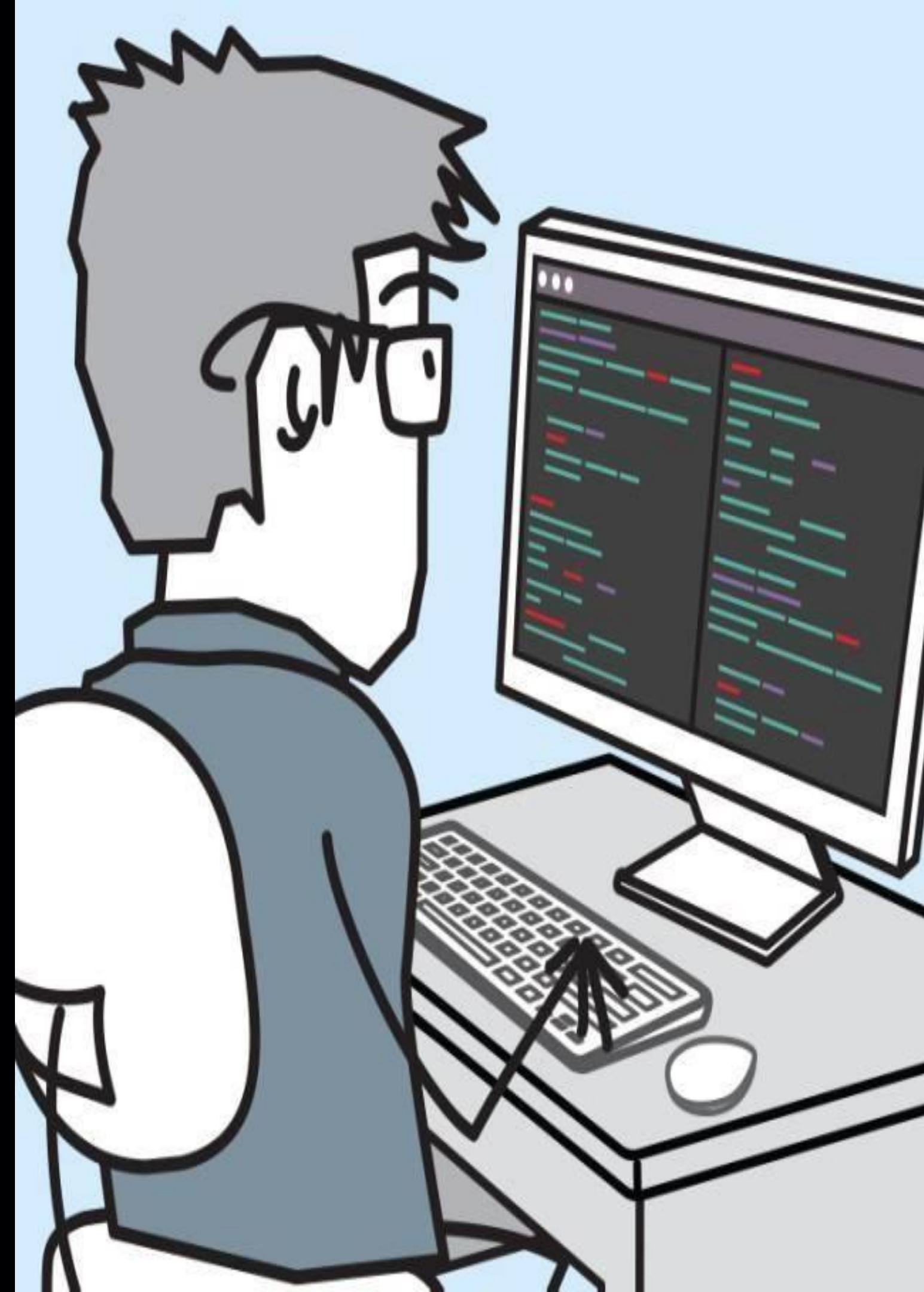
Finalmente, retire o ovo frito da frigideira, coloque-o sobre um prato e saboreie.



LP – Algoritmos & Variáveis

O que são Variáveis?

As **variáveis** são estruturas importantes em qualquer programação, independentemente da linguagem que se utilize para implementar os códigos ou programas. Elas podem ser entendidas como uma caixinha onde os **dados** são armazenados temporariamente ou em definitivo. Ainda, as **variáveis** são manipuladas durante a execução do programa (Algoritmo).



LP – Algoritmos & Variáveis



O **dado** que a **variável** armazena, por sua vez, possui um **tipo**, e conhecê-lo é fundamental para criação e implementação dos Algoritmos.

Em computação, temos quatro tipos de dados elementares:

- Dados do tipo *string* (*str*) - todo e qualquer texto.
- Dados do tipo inteiro (*int*) - números inteiros.
- Dados do tipo decimal (*float*) - números decimais.
- Dados do tipo lógico (*bool*) - verdadeiro ou falso.

LP – Algoritmos & Variáveis

Exemplos Variáveis em Python



```
1 # A variável nome recebe um tipo de dado string (str)
2 nome = 'Natanael'
3 # Printamos o tipo de dado atribuído à variável
4 print(type(nome))
5
6 # A variável idade recebe um tipo de dado inteiro (int)
7 idade = 12
8 # Printamos o tipo de dado atribuído à variável
9 print(type(idade))
10
11 # A variável peso recebe um tipo de dado decimal (float)
12 peso = 34.5
13 # Printamos o tipo de dado atribuído à variável
14 print(type(peso))
15
16 # A variável brasileiro recebe um tipo de dado lógico (bool)
17 brasileiro = True
18 # Printamos o tipo de dado atribuído à variável
19 print(type(brasileiro))
```



LP – Algoritmos & Variáveis

Operadores Aritméticos e Relacionais

O computador é uma máquina que lida com dados. Por essa razão, além de compreender quais os tipos desses dados, é necessário manipulá-los para que sejam criados os **Algoritmos**. Para tanto, dispomos de operadores aritméticos (matemáticos) e relacionais (comparação).



LP – Algoritmos & Variáveis

Operadores Aritméticos

Operador	Descrição
+	Adição
-	Subtração
*	Multiplicação
/	Divisão
//	Divisão inteira
**	Potenciação
%	Resto da divisão

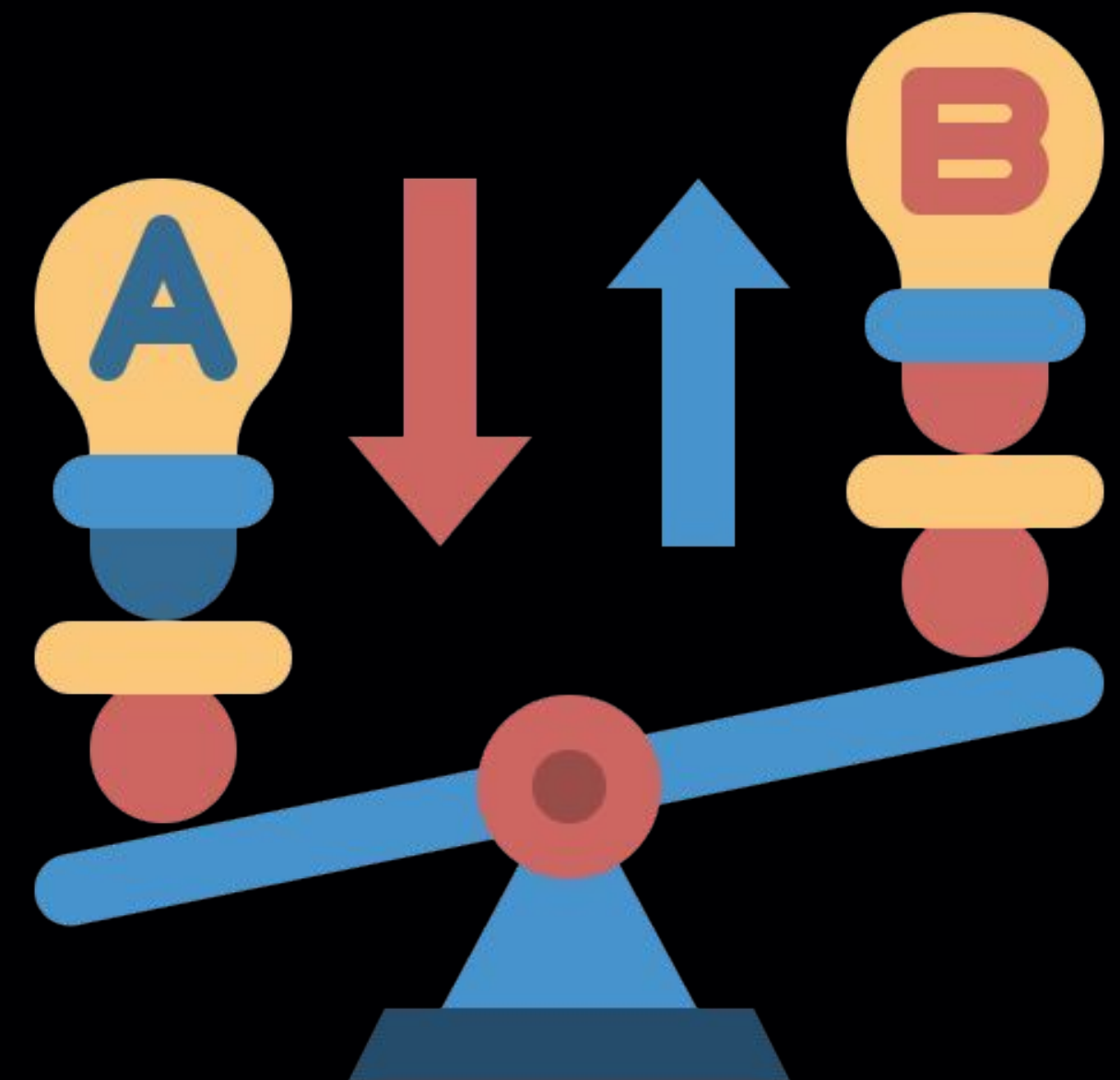


LP – Algoritmos & Variáveis



```
1  # == IGUAL: Verifica se um valor é igual o outro.
2  print(5 == 5) # // True
3
4  # != DIFERENTE: Verifica se o valor é diferente do outro.
5  print(5 != 5) # // False
6
7  # > MAIOR QUE: Verifica se um valor é maior que o outro.
8  print(5 > 1) # // True
9
10 # < MENOR QUE: Verifica se o valor é menor que o outro.
11 print(5 < 1) # // False
12
13 # >= MAIOR IGUAL: Verifica se um valor é maior ou igual o outro.
14 print(5 >= 5) # // True
15
16 # <= MENOR IGUAL: Verifica se o valor é menor ou igual o outro.
17 print(5 <= 4) # // False
```

Operadores Relacionais



LP – Algoritmos & Variáveis



```
1 # A variável nome recebe um dado lido do teclado, ou seja
2 # digitado pelo usuário.
3 nome = input("Informe seu nome: ")
4
5 # Aqui utilizamos o print() para exibir o nome digitado
6 print(nome)
```

Enquanto o comando **input()** é utilizado para entrada de dados, o comando **print()** é utilizado para saída de dados. No caso, a entrada do dado é o nome digitado pelo usuário, e a saída a exibição desse nome.

Atribuindo e imprimindo dados utilizando funções

Note que também é possível pedirmos ao usuário que informe o dado que será atribuído à variável. Nesse caso, a função **input()** permite ler dados digitados no teclado.

LP – Algoritmos & Variáveis

Saída de dados e a função `print()`

Ao utilizarmos a função `print()`, até então, exibimos apenas o valor atribuído a uma variável. No entanto, e se precisarmos exibir um texto combinado ao valor dessa variável? Em Python, podemos utilizar o recurso **F-strings**, ou “strings literais formatadas” (*formatted string literals*). Com esse recurso, é possível personalizar ainda mais as saídas de dados.

Saída de dados com F-STRING



```
1  # A variável nome recebe um dado lido do teclado
2  nome = input("Informe seu nome: ")
3
4  # Aqui utilizamos o F-STRING para exibir uma
5  # mensagem e o nome atribuído à variável.
6  print(f"Seu nome é {nome}.")
```

F-strings são *strings* com a letra **f** no início e chaves **{ }** para realizar a interpolação de expressões.

LP – Algoritmos & Variáveis

Antes de praticar...

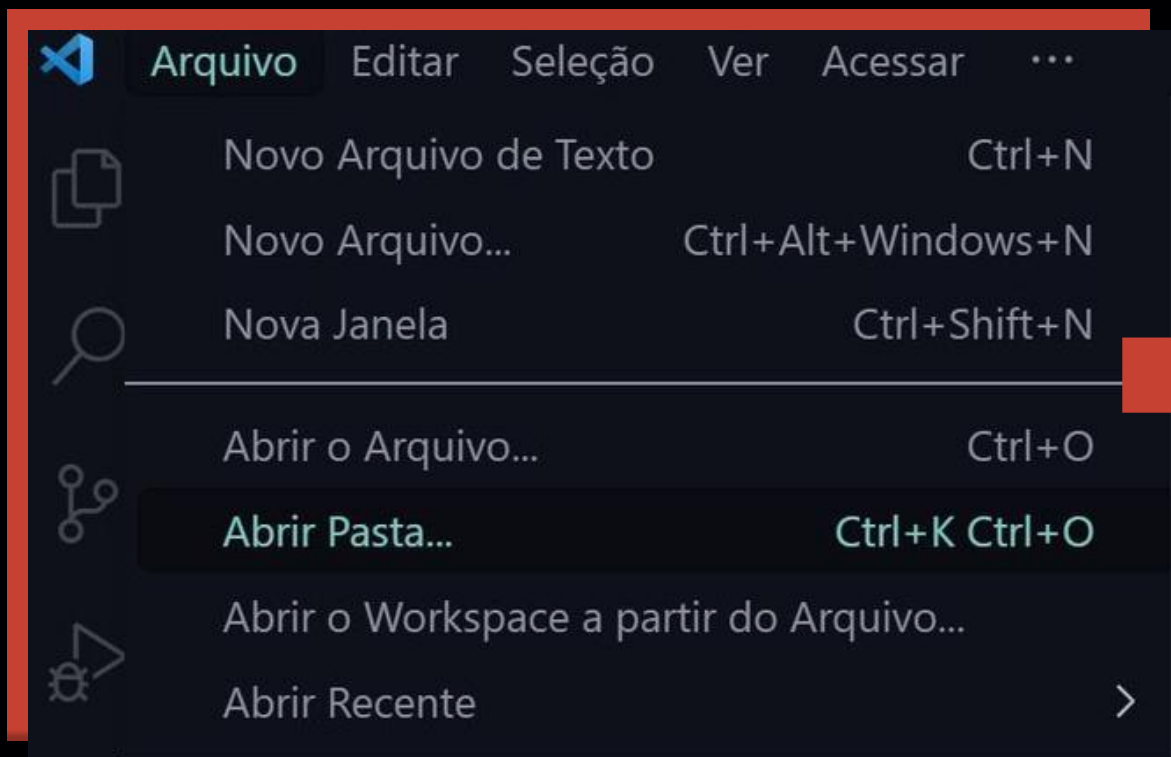
Conhecendo o VSCode

VS Code é um editor de código-fonte gratuito e de código aberto desenvolvido pela Microsoft. Ele é usado por desenvolvedores de software para escrever, depurar e refatorar código em várias linguagens de programação. Além de ser um editor altamente configurável e extensível, o VS Code é multiplataforma, e pode ser usado no Windows, no macOS e no Linux.

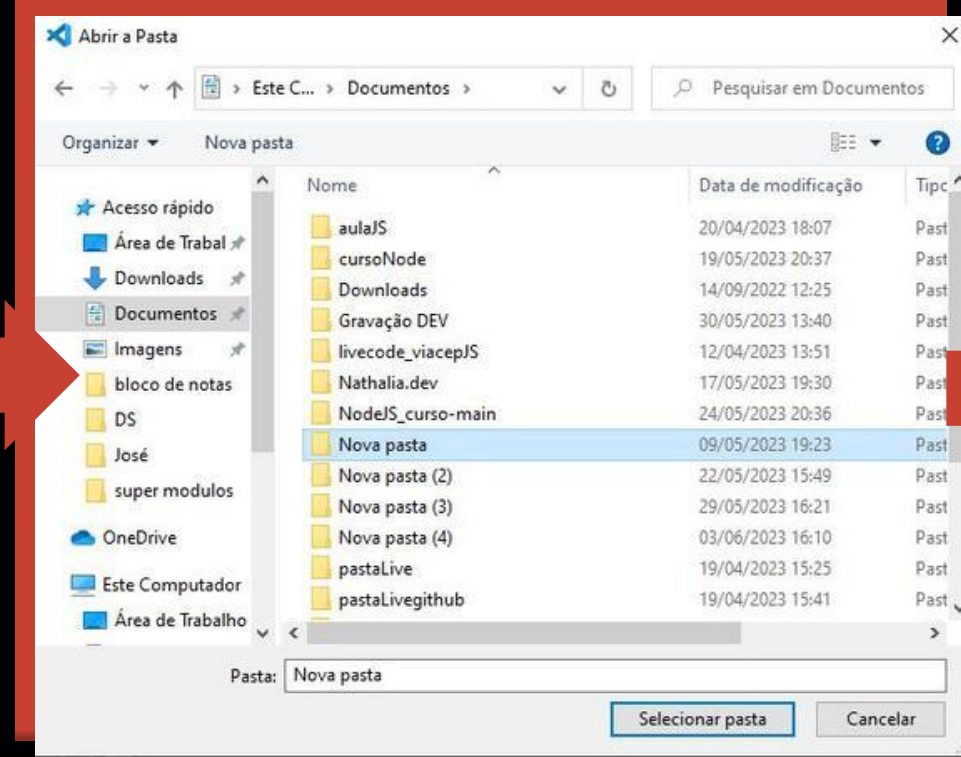


LP – Algoritmos & Variáveis

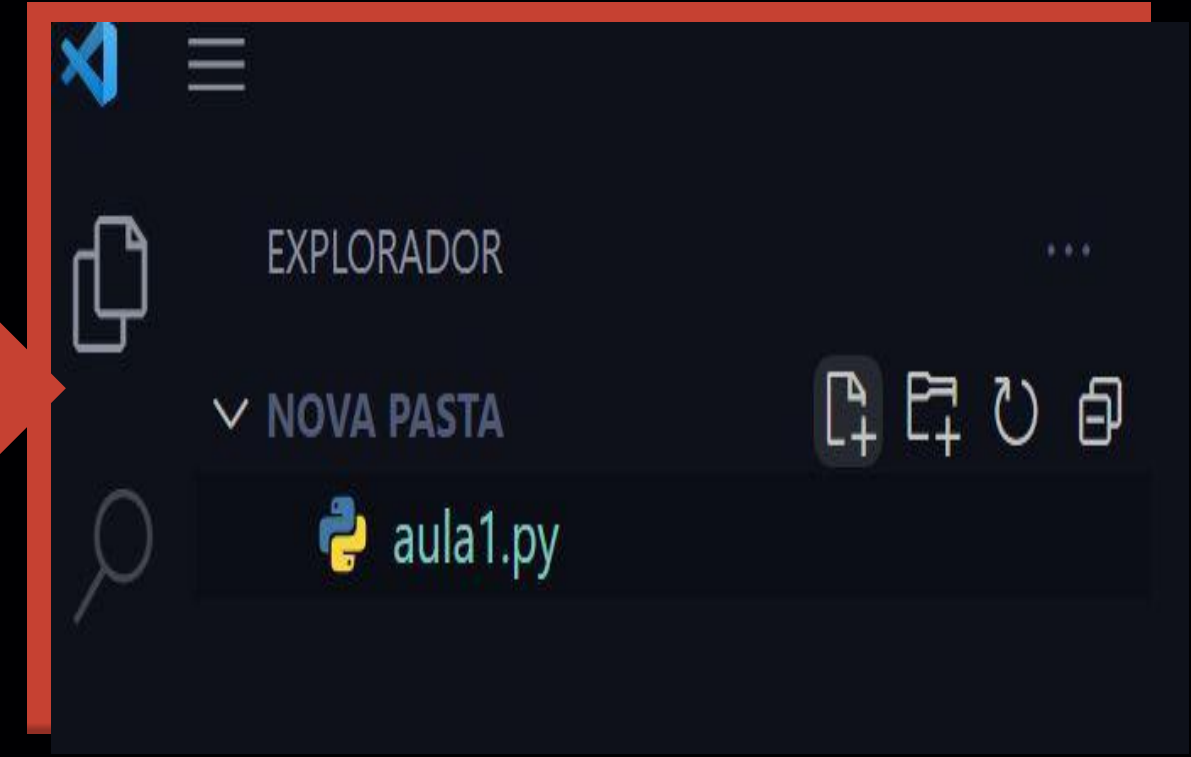
Passo a passo para configurar o VSCode



1. Abra o VS Code e clique em "File" (Arquivo) no menu superior. Clique em "open folder" (Abrir Pasta).



2. No gerenciador de pasta, selecione ou crie uma pasta para seu projeto ou aula.



3. Digite o nome do arquivo com a extensão .py (por exemplo, "aula1.py") e pressione Enter.

LP – Algoritmos & Variáveis

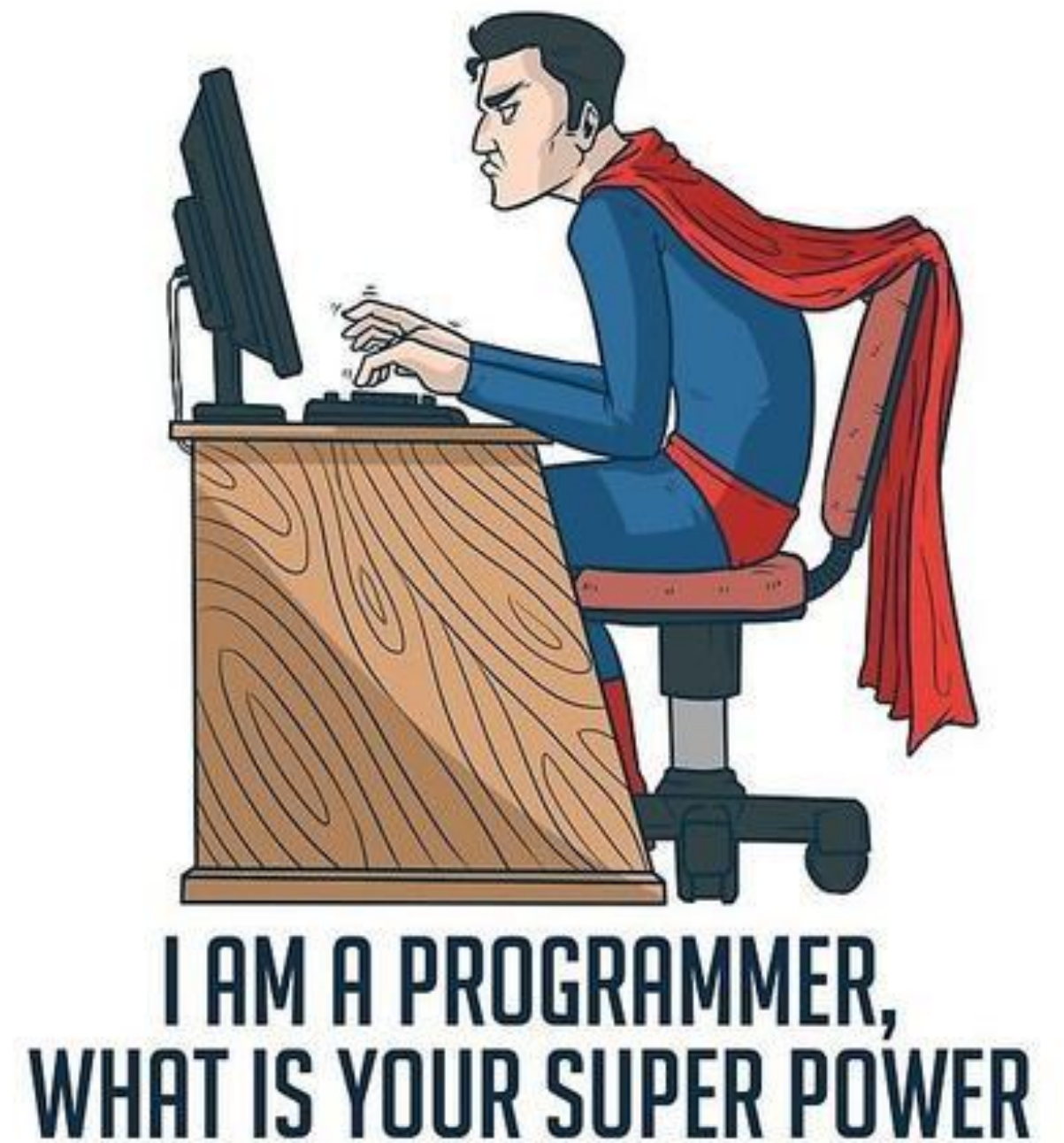
1. Crie um algoritmo que exiba na tela (terminal) a mensagem "Olá Mundo".
2. Crie um algoritmo que exiba na tela (terminal) seu nome. Por exemplo: "Olá, João Paulo."
3. Crie um algoritmo que exiba na tela (terminal) sua idade. Por exemplo: 17.
4. Crie um algoritmo que exiba na tela (terminal) (em linhas separadas) seu nome e sua idade.
5. Crie um algoritmo que exiba na tela (terminal) (na mesma linha) seu nome e sua idade.
6. Crie um algoritmo que exiba na tela (terminal) a soma de dois números inteiros.
7. Crie um algoritmo que exiba na tela (terminal) a subtração de dois números inteiros.
8. Crie um algoritmo que exiba na tela (terminal) a multiplicação de dois números inteiros.
9. Crie um algoritmo que exiba na tela (terminal) a divisão de dois números inteiros.
10. Crie um algoritmo que exiba na tela (terminal) o resto da divisão de dois números inteiros.



**I AM A PROGRAMMER,
WHAT IS YOUR SUPER POWER**

LP – Algoritmos & Variáveis

11. Crie um algoritmo que exiba na tela (terminal) apenas a parte inteira de uma divisão de dois números inteiros.
12. Crie um algoritmo que exiba na tela (terminal) o quadrado de um número inteiro.
13. Crie um algoritmo que exiba na tela (terminal) o valor lógico de uma variável qualquer.
14. Crie um algoritmo que exiba na tela (terminal) o tipo de uma variável cujo valor atribuído é "jpaulo@infinityschool.com.br".
15. Crie um algoritmo que exiba na tela (terminal) o tipo de uma variável cujo valor atribuído é "c".
16. Crie um algoritmo que exiba na tela (terminal) o tipo de uma variável cujo valor atribuído é 12.
17. Crie um algoritmo que exiba na tela (terminal) o tipo de uma variável cujo valor atribuído é 1.87.
18. Crie um algoritmo que exiba na tela (terminal) o tipo de uma variável cujo valor atribuído é False.
19. Crie um algoritmo que concatene duas strings ao ser executado. Exiba a string concatenada na tela (terminal).
20. Crie um algoritmo que concatene dois números inteiros ao ser executado. Exiba os números concatenados na tela (terminal).





IN

INFINITY SCHOOL

V I S U A L A R T C R E A T I V E C E N T E R