



INFINITY SCHOOL

VISUAL ART CREATIVE CENTER

AULA 01 – Algoritmo e Variáveis

O QUE IREMOS APRENDER

01

INTRODUÇÃO

02

ALGORITMO

03

CONHECENDO O VS CODE

04

VARIÁVEIS

05

FUNÇÕES INTERNAS

06

OPERADORES ARITMÉTICOS

07

F STRING

IN

Introdução



A introdução à lógica da programação ensina como os computadores resolvem problemas através de uma sequência organizada de instruções.

É sobre pensar e separar problemas complexos em partes menores de forma estruturada para criar algoritmos eficientes.

Essa prática melhora as habilidades de resolução de problemas, fundamentais para o sucesso na programação.

Importância da lógica

É a base sobre a qual todas as linguagens de programação são construídas. Ela envolve a aplicação de princípios de raciocínio lógico para criar algoritmos que resolvem problemas específicos de maneira sistemática e estruturada.

Exemplo Prático de Problema Simples:

Problema: Fazer um suco de laranja.

Passos:

1. Pegar laranjas.
2. Cortar as laranjas.
3. Espremer as laranjas.
4. Coar o suco.
5. Servir o suco.



DINÂMICA

Objetivo: Praticar a decomposição de um problema cotidiano em passos lógicos.

Como vocês acham que é a logica de dividir o processo de fazer uma xícara de chá?
Escreva no bloco de notas esses passos de maneira detalhada.

Observação: Não esqueçam de compartilhar o resultado de vocês.

Algoritmo

Um algoritmo é uma sequência finita de instruções claras e definidas, projetadas para realizar uma tarefa específica ou resolver um problema, com cada passo sendo executável em um tempo finito.

Os algoritmos são cruciais na programação porque:

Estruturação de Soluções

Abordagem estruturada para resolver problemas complexos.

Reutilização de Código

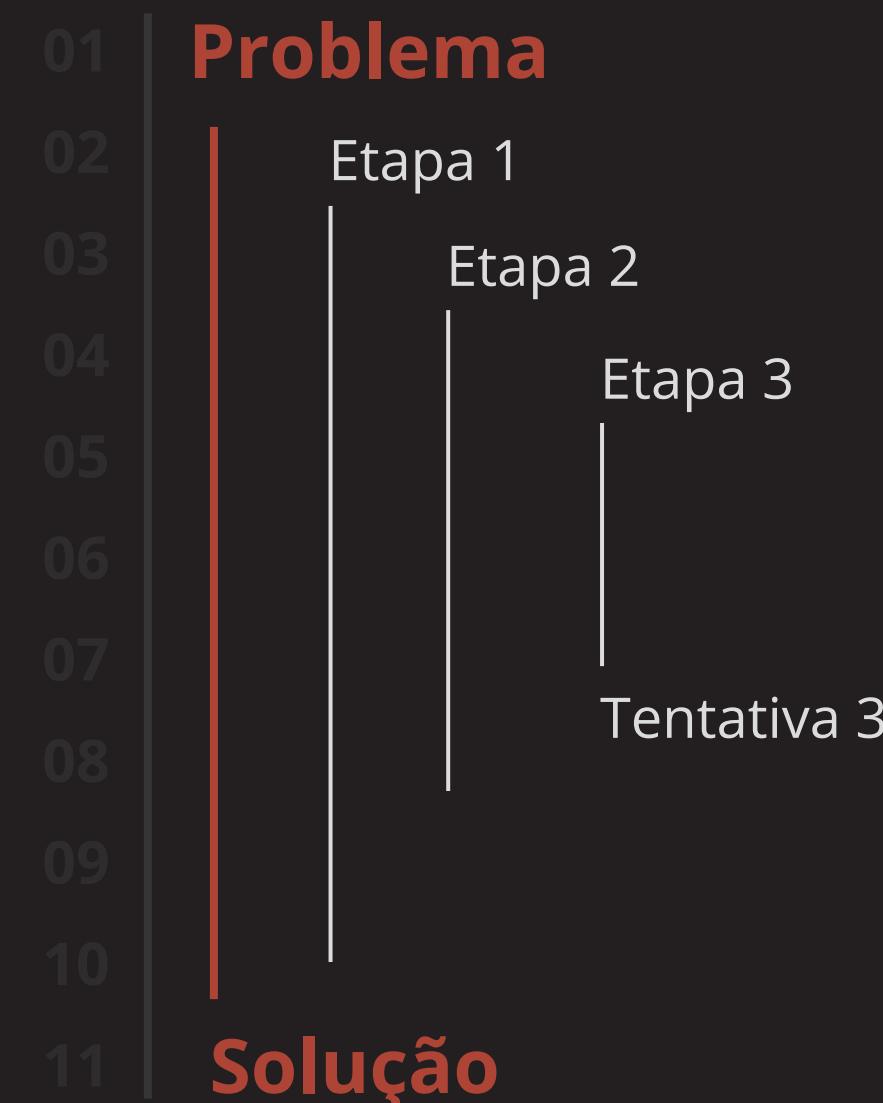
Algoritmos bem projetados podem ser reutilizados, economizando tempo e esforço.

Otimização

A eficiência do programa depende da eficácia dos algoritmos.

Claridade e Manutenção

Algoritmos bem definidos tornam o código mais claro e fácil de manter, facilitando a colaboração.



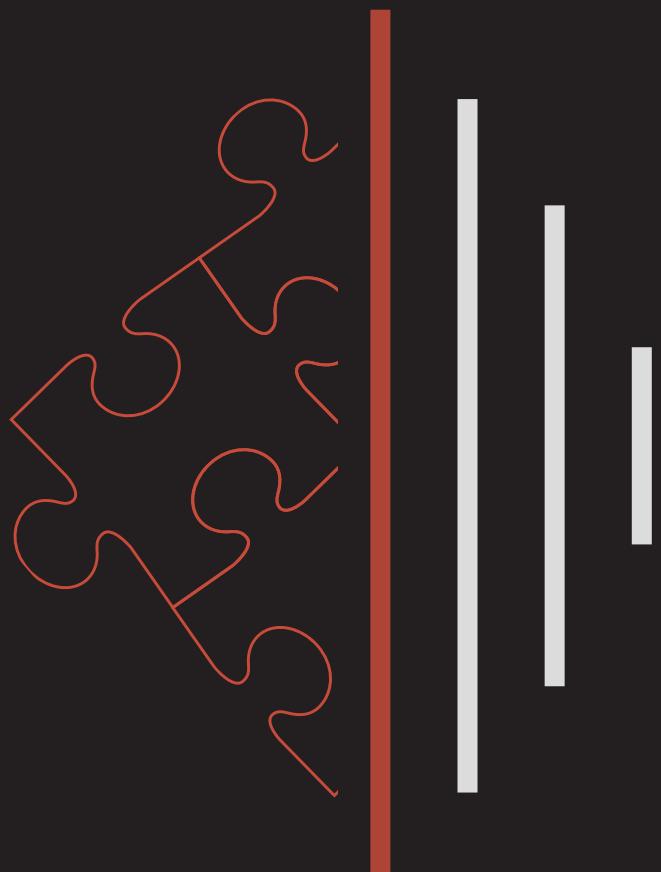
Lógica X Algoritmo

Enquanto a lógica fornece a base teórica para a resolução de problemas, os algoritmos são a aplicação prática dessa lógica em passos bem definidos.

Lógica

Refere-se ao processo de raciocínio estruturado e sistemático para resolver problemas.

Envolve a formulação de premissas e a derivação de conclusões válidas a partir dessas premissas.



Algoritmo

É uma sequência finita de instruções ou passos bem definidos, projetados para realizar uma tarefa específica ou resolver um problema com base na lógica.

Relação entre Lógica e Algoritmo



Planejamento

A lógica é usada para planejar como um problema será resolvido. Isso envolve identificar os passos necessários e a ordem em que devem ser executados.



Execução

O algoritmo é a execução desse plano. Ele transforma o raciocínio lógico em uma série de passos que podem ser seguidos por um computador.



Otimização

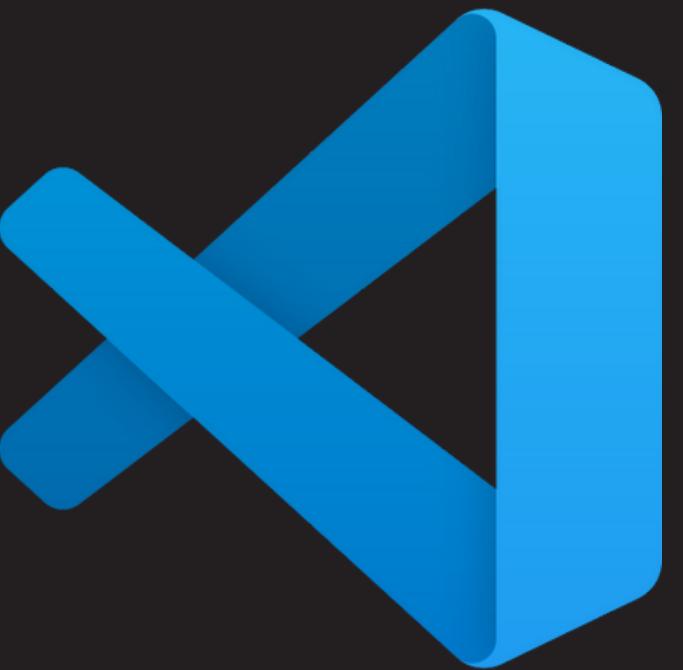
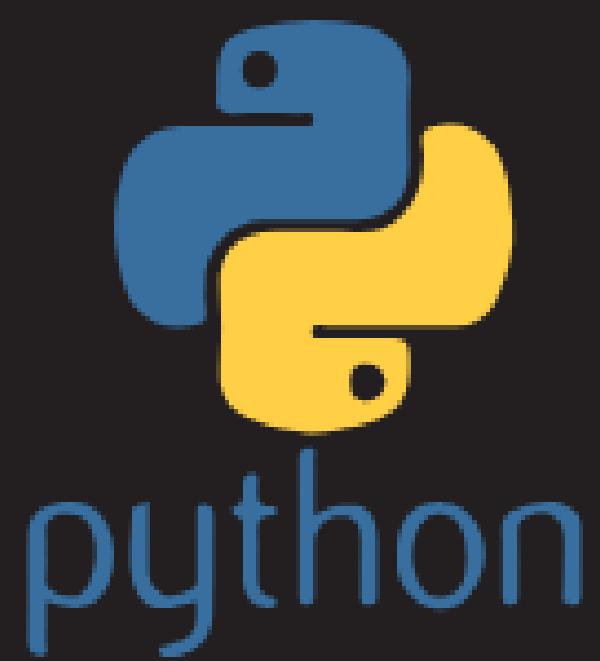
Algoritmos podem ser otimizados usando princípios lógicos para melhorar sua eficiência e desempenho.

Codando Python no VScode

Python é popular em diversas áreas como web, ciência de dados e IA, é apreciada por sua versatilidade, ampla biblioteca padrão e forte comunidade de desenvolvedores.

O **Visual Studio Code (VsCode)** é um editor de código gratuito e versátil, desenvolvido pela Microsoft. Popular entre desenvolvedores, suporta várias linguagens e oferece extensões que aumentam a produtividade, sendo essencial para programadores modernos.

A **extensão Python no VsCode** oferece recursos como autocompletar, depuração e linting, tornando o VsCode um ambiente de desenvolvimento poderoso para Python.



Download do VScode e a extensão Python no VScode

Como Baixar e Instalar Python:

- Passo 1: Acessar o Site Oficial do Python;
- Passo 2: Baixar o Instalador do Python;
- Passo 3: Executar o Instalador;
- Passo 4: Verificar a Instalação;

Como Baixar e Instalar Python:

- Passo 1: Acessar o Site Oficial do VSCode;
- Passo 2: Baixar o Instalador do VSCode;
- Passo 3: Executar o Instalador;
- Passo 4: Configurar o VSCode;
- Passo 5: Testar a Instalação;

Como Baixar e Instalar Python:

- Passo 1: Abrir o Visual Studio Code;
- Passo 2: Acessar a Guia de Extensões;
- Passo 3: Buscar a Extensão do Python;
- Passo 4.1: Instalar a Extensão do Python;
- Passo 4.2: Python, Pylance e Python Debugger.
- Passo 5: Configurar a Extensão do Python;



DINÂMICA

Objetivo: Conhecer a ferramenta VScode

Abra o Visual Studio Code (VScode)

Crie uma pasta para seu Código

Crie um arquivo “Aula01.py”

Observação: Fique atento aos próximos passos,
pois iremos aprender algo bem importante!

Variáveis

As variáveis são essenciais na programação, permitindo armazenar e manipular dados durante a execução de um programa.

Uma variável é um **espaço de armazenamento na memória do computador, identificado por um nome**, que pode conter dados que mudam durante o programa.

Em Python, as variáveis são dinamicamente tipadas, ou seja, **o tipo de dado é determinado automaticamente pelo valor atribuído**.

01

Tipo A

02

Tipo B

03

Tipo C

04

Tipo D

Tipos de Variáveis

Os diferentes tipos de variáveis é essencial para a programação eficiente. Cada tipo de variável tem suas próprias características e usos específicos.

Strings (str)

Representam sequências de caracteres, usadas para armazenar texto.

```
nome = "Carlos" #String (str)
```

Inteiros (int)

Representam números inteiros, positivos ou negativos, sem parte decimal.

```
idade = 30 # Inteiro (int)
```

Flutuantes (float)

Representam números que possuem parte decimal.

```
altura = 1.85 # Ponto Flutuante ou decimal (float)
```

Booleanos (bool)

Representam valores lógicos, sendo 'True' ou 'False'.

```
maior_de_idade = True # Boolean (bool)
```

Construindo Variáveis

Declaração de Variáveis

Em Python, para declarar uma variável, basta atribuir um valor a um nome de variável.

Atribuição de Valores

Para atribuir um valor a uma variável, usa-se o operador de atribuição.

```
msg      =  'Exemplo de mensagem!'  
variável    atribuição    dado armazenado
```

```
numero   =  3.141592653589931  
variável    atribuição    dado armazenado
```

Funções Internas

As funções internas do Python são ferramentas poderosas para realizar operações essenciais sem escrever código adicional. Nesta seção, os alunos aprenderão sobre algumas das primeiras funções internas frequentemente usadas: '**print**', '**type**', e '**input**'.

print: Função usada para exibir informações na tela.

print("variável")

input: Função usada para receber entrada do usuário.

input("digite aqui: ")

type: Função usada para verificar o tipo de dado de uma variável.

type(10) #int
type("10") #str
type(10.5) #float

Funções Internas (print)

Conceito: A função '**print()**' é uma das funções mais usadas no Python, sendo utilizada para imprimir valores na tela. A sintaxe da função é muito simples, basta escrever "**print()**" seguido dos valores dentro do parenteses que você deseja imprimir

Uso: Exibe uma mensagem ou valor na tela.

Sintaxe: `print(valor)`

Exemplo:



```
1 print("Olá, Mundo!")  
2 print(42)
```

Terminal:

```
Olá, Mundo!  
42
```

Funções Internas (type)

Conceito: **type()** método retorna o tipo de classe do argumento (objeto) passado como parâmetro.
A função **type()** é usada principalmente para fins de depuração.

Uso: Retorna o tipo de dado de uma variável.

Sintaxe: **type(variavel)**

Exemplo:



```
1 print(type(42))  
2 print(type(3.14))  
3 print(type("texto"))
```

Terminal:

```
# Saída: <class 'int'>  
# Saída: <class 'float'>  
# Saída: <class 'str'>
```



Funções Internas (input)

Conceito: A função '**input**' exibe um prompt ao usuário, que pode digitar uma resposta e pressionar Enter. O texto digitado é então retornado e pode ser armazenado em uma variável, como '**nome**'.

Uso: Recebe uma entrada do usuário como uma string.

Sintaxe: `input(mensagem)`

Exemplo:



```
1 nome = input("Digite seu nome: ")  
2 print("Olá,", nome)
```

Terminal:

→ # Saída: Olá, "Seu nome"

ATIVIDADE PRÁTICA

Objetivo: Praticar a declaração e uso de variáveis de diferentes tipos, além de aprender a verificar e imprimir seus tipos em Python.

Atividade 01: Crie uma variavel chamada “saudacao”, em seguida coloque uma atribuição e dentro de dado armazenado escreva “Hello World”

Observação: Dizem que quando você aprende a programar, esse é o primeiro código que você deve criar, caso não crie esse código o programador terá azar

ATIVIDADE PRÁTICA

Objetivo: Praticar a declaração e uso de variáveis de diferentes tipos, além de aprender a verificar e imprimir seus tipos em Python.

Atividade 02: Crie um programa que peça ao usuário para digitar:

1. Seu nome;
2. Sua idade;
3. Sua altura;
4. Em seguida, imprima esses valores e seus respectivos tipos.

Observação: Conhecer e utilizar diferentes tipos de variáveis é fundamental para manipular dados de maneira eficaz.

Operadores aritméticos

Os operadores aritméticos realizam operações matemáticas em variáveis e valores, sendo essenciais em qualquer linguagem de programação.

Eles permitem cálculos básicos e complexos. Compreender esses operadores permite resolver diversos problemas e prepara para diversas situações.

+	Soma	$3+2=5$
-	Subtração	$3-2=1$
*	Multiplicação	$3*2=6$
/	Divisão	$3/2=1.5$
%	Módulo	$3\%2=1$
**	Exponenciação	$3^{**}2=9$
//	Divisão Inteira	$3//2=1$

Operadores Básicos

A prática com esses operadores ajuda a solidificar o entendimento e prepara os alunos para aplicar esses conceitos em situações mais complexas no futuro.

Adição (+)

Soma de dois valores.

Subtração (-)

Subtrai um valor de outro.

Multiplicação (*)

Multiplica dois valores.

Divisão (/)

Divide um valor pelo outro, resultando em um número de ponto flutuante.

Divisão Inteira (//)

Divide um valor pelo outro, resultando no inteiro da divisão.

Exponenciação (**)

Eleva um valor à potência de outro.



```
1 num1 = 62
2 num2 = 11
3
4 print("Adição: ", num1 + num2) #73
5 print("Subtração: ", num1 - num2) #51
6 print("Multiplicação: ", num1 * num2) #682
7 print("Divisão: ", num1 / num2) #5.63636363...
8 print("Divisão Inteira: ", num1 // num2) #5
9 print("Módulo: ", num1 % num2) #7
10 print("Exponenciação: ", num1 ** num2)
#52036560683837093888
```

F string

F-strings, ou formatted string literals, são uma forma moderna e conveniente de formatar strings em Python. Elas permitem incluir expressões dentro de strings de forma mais legível e concisa.

Sintaxe de F-Strings

- Para usar uma f-string, você simplesmente coloca um `f` ou `F` antes das aspas da string e insere as expressões entre chaves `{}` dentro da string.

Exemplo:



```
1 nome = "Infinity School"
2 saudacao = (f"Olá, eu sou {nome} e dou as boas-vindas a todos vocês!")
3
4 print(saudacao)
```

Características das F-Strings

Interpolação de Variáveis:

- As variáveis são inseridas diretamente na string.



```
1 produto = "maçã"
2 preço = 1.99
3 print(f'O preço da {produto} é R${preço}')
```

Formatando Valores:

- F-strings suportam formatação avançada, como especificar o número de casas decimais.



```
1 pi = 3.14159
2 print(f'O valor de pi com duas casas decimais é {pi:.2f}.')
```

Avaliação de Expressões:

- Você pode incluir expressões dentro das chaves, e elas serão avaliadas.



```
1 largura = 5
2 altura = 10
3 área = (f'A área do retângulo é {largura * altura}.' )
4
5 print(área)
```

Vantagens das F-Strings

Legibilidade:

F-strings são mais fáceis de ler e escrever, especialmente quando comparadas a métodos de formatação mais antigos.

Desempenho:

F-strings são geralmente mais rápidas do que os métodos anteriores de formatação de strings.

Flexibilidade:

A capacidade de incluir expressões e chamar funções diretamente dentro das chaves {} torna as f-strings muito poderosas.



Pratique e Aprenda

Objetivo: Criar um Programa que Peça as 4 Notas Bimestrais e Mostre a Média

Instruções:

1 - Solicitar as Notas do Usuário:

Use a função `input()` para pedir ao usuário que insira cada uma das quatro notas bimestrais. Converta a entrada do usuário de string para um número (`float`) para permitir cálculos.

2 - Calcular a Média das Notas:

Some as quatro notas e divida o resultado por quatro para obter a média.

3 - Mostrar a Média Calculada para o Usuário:

Use a função `print()` para exibir a média das notas calculada.

Benefícios

Fortalecer o entendimento de conceitos básicos e incentivar o pensamento lógico.

Promover a prática com a sintaxe de Python, desenvolvimento de programas interativos e aprimorar habilidades de depuração, preparando para conceitos avançados.

Pratique e Aprenda

Objetivo: Faça um programa que pergunte quanto você ganha por hora e o número de horas trabalhadas no mês, calcule o salário total e exiba o resultado (Considere que você trabalha 20 dias no mês).

Instruções:

1 - Solicitar o Salário Mensal:

Use a função `input()` para pedir ao usuário que insira quanto ele ganha por mês. Converta a entrada do usuário de string para um número (`float`) para permitir cálculos.

2 - Solicitar o Número de Horas Trabalhadas na Semana:

Use a função `input()` para pedir ao usuário que insira o número de horas trabalhadas na semana. Converta a entrada do usuário de string para um número (`float`) para permitir cálculos.

3 - Calcular o Total de Horas Trabalhadas no Mês:

Multiplique o número de horas trabalhadas na semana por 4 para obter o total de horas trabalhadas no mês.

4 - Calcular o Salário por Hora:

Divida o salário mensal pelo total de horas trabalhadas no mês para obter o salário por hora.

5 - Mostrar o Salário por Hora Calculado para o Usuário:

Use a função `print()` para exibir o salário por hora calculado.

Pratique e Aprenda

Objetivo: Peça ao usuário para digitar seu nome, idade e cidade natal. Use uma f-string para formatar e exibir uma mensagem com essas informações.

Instruções:

1 - Solicitar o Nome do Usuário:

Use a função input() para pedir ao usuário que insira seu nome.

2 - Solicitar a Idade do Usuário:

Use a função input() para pedir ao usuário que insira sua idade. Converta a entrada do usuário de string para um número (int).

3 - Solicitar a Cidade Natal do Usuário:

Use a função input() para pedir ao usuário que insira sua cidade natal.

4 - Formatar e Exibir a Mensagem com f-string:

Use uma f-string para formatar a mensagem com as informações fornecidas pelo usuário e exiba a mensagem usando a função print().

Benefícios

Praticar a criação e uso de f-strings para formatar strings e incluir variáveis e expressões.

SE LIGA NO CONTEÚDO DA PRÓXIMA AULA!

AULA 02 DE LÓGICA DA PROGRAMAÇÃO.
ESTRUTURA CONDICIONAL (IF, ELSE E ELIF)



INFINITY SCHOOL
VISUAL ART CREATIVE CENTER

Aula 02 - Condicionais

O QUE SÃO OPERADORES

INDENTAÇÃO

ESTRUTURA CONDICIONAL

TIPOS DE ESTRUTURAS CONDICIONAIS

CONDICIONAIS ANINHADAS

OPERADORES LÓGICOS (AND, OR, NOT)



INFINITY SCHOOL

VISUAL ART CREATIVE CENTER

AULA 01 – Algorítimo e Variáveis