



# INFINITY SCHOOL

VISUAL ART CREATIVE CENTER

AULA 06 – Revisão da Logica da Programação

# VEREMOS HOJE

**01**

REVISÃO DE VARIÁVEIS

**02**

REVISÃO DE ALGORITMO

**03**

REVISÃO DE CONDICIONAIS

**04**

REVISÃO DE ESTRUTURAS DE REPETIÇÃO

# Revisão da Aula Anterior

- Definição de Inteligência Artificial
- História da IA
- Tipos de Inteligência Artificial
- Introdução ao ChatGPT



# Revisão de Variáveis

## Algoritmo Conceito:

Uma variável é um espaço na memória que armazena um valor que pode mudar durante a execução do programa.

Variáveis são identificadas por nomes e têm tipos de dados, como inteiros, floats, strings e booleanos.

01

Tipo A

02

Tipo B

03

Tipo C

04

Tipo D

# Revisão de Variáveis

Uma variável é um espaço na memória que armazena um valor que pode mudar durante a execução do programa.

## Tipos de Variáveis:

Inteiros (int): Números sem parte decimal, ex: 5, -3.

**idade** = 25 # Inteiro (int)

Floats (float): Números com parte decimal, ex: 3.14, -2.0.

**altura** = 1.75 # Ponto Flutuante ou decimal (float)

Strings (str): Sequências de caracteres, ex: "Hello", "Python".

**nome** = "Infinity" #String (str)

Booleanos (bool): Verdadeiro ou falso, ex: True, False.

**Ativo** = True # Boolean (bool)

# ATIVIDADE PRÁTICA

## **Conversão de Unidades:**

Crie um programa que converta metros para centímetros.  
Peça ao usuário para digitar um valor em metros, armazene em uma variável e converta para centímetros.

## **Cálculo de Área:**

Crie um programa que calcule a área de um retângulo.  
Peça ao usuário para digitar a largura e a altura, armazene em variáveis e calcule a área.

# ATIVIDADE PRÁTICA

## **Cálculo de IMC:**

Crie um programa que calcule o Índice de Massa Corporal (IMC). Peça ao usuário para digitar seu peso e altura, armazene em variáveis e calcule o IMC.

## **Cálculo de Juros Simples:**

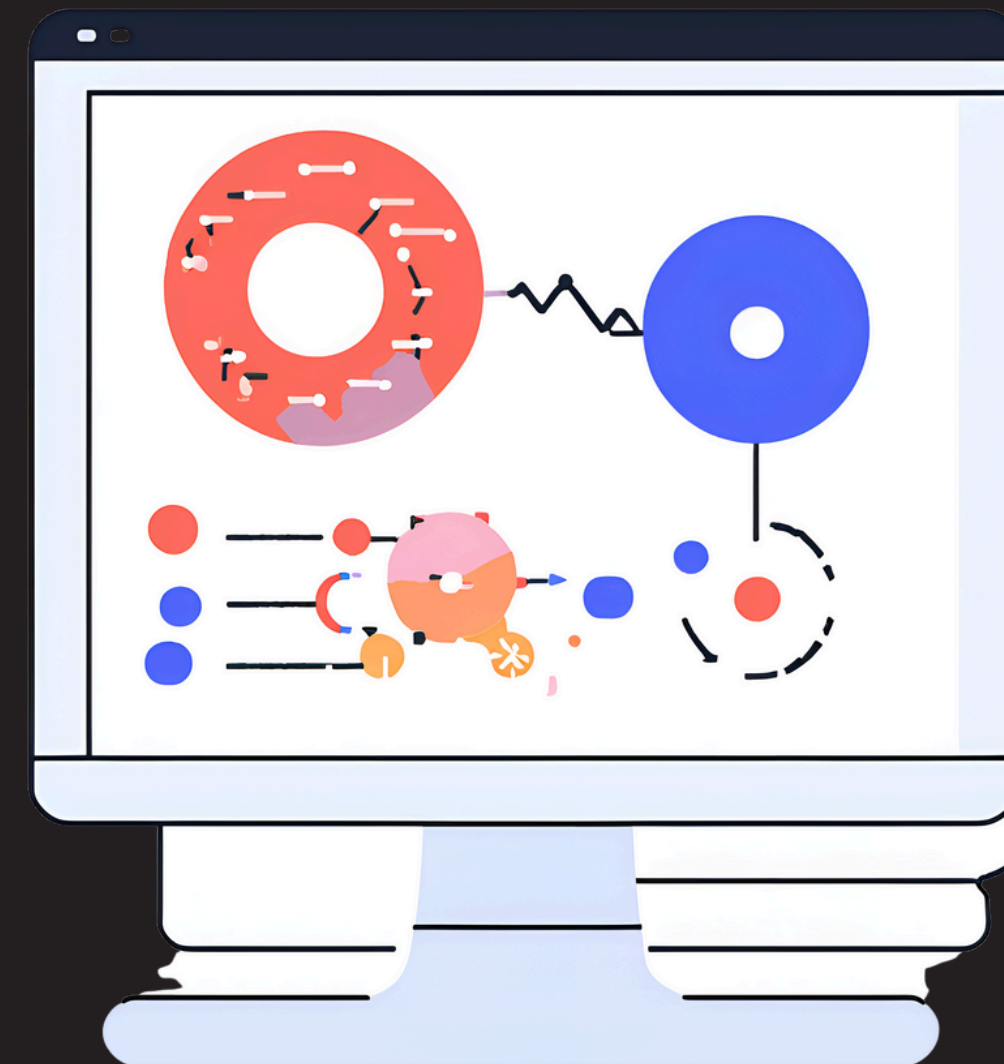
Crie um programa que calcule o valor futuro de um investimento usando a fórmula de juros simples. Peça ao usuário para digitar o capital inicial, a taxa de juros e o tempo de aplicação.

# Revisão de Algoritmo

## Algoritmo Conceito:

Um algoritmo é uma sequência finita de instruções bem definidas e ordenadas que, quando seguidas, resolvem um problema específico.

Exemplos de algoritmos incluem busca, ordenação e cálculo de médias.





# Revisão de Algoritmo

## Algoritmo Conceito:

**Definição:** Um conjunto de passos ordenados e finitos para resolver um problema.

**Características:** Clareza, finitude e definição precisa.



```
1  a = 5
2  b = 10
3  c = 15
4  soma = a + b + c
5  media = soma / 3
6  print(media)  # Saída: 10.0
```

## Exemplo Simples:

Algoritmo para calcular a média de três números:

# ATIVIDADE PRÁTICA

## **Algoritmo de Cálculo de Desconto:**

Crie um algoritmo que peça quatro notas de um aluno, calcule a média e exiba se o aluno foi aprovado ou reprovado (média  $\geq 6$ ).

## **Algoritmo de Cálculo de Desconto:**

Desenvolva um algoritmo que calcule o preço de um produto após aplicar um desconto. Solicite o preço original e o percentual de desconto.

# ATIVIDADE PRÁTICA

## **Algoritmo de Conversão de Tempo:**

Desenvolva um algoritmo que converta uma quantidade de segundos fornecida pelo usuário em horas, minutos e segundos.

## **Algoritmo de Conversão de Temperatura:**

Crie um algoritmo que converta uma temperatura de Celsius para Fahrenheit. Solicite ao usuário a temperatura em Celsius e exiba o resultado em Fahrenheit.

# Revisão de Condicionais

## Estrutura if:

Executa um bloco de código se a condição especificada for verdadeira.



```
1  if condição:
2      # bloco de código a ser executado se a condição for verdadeira
```

## Estrutura elif:

Significa "else if" e permite verificar múltiplas condições.



```
1  if condição1:
2      # bloco de código a ser executado se condição1 for verdadeira
3  elif condição2:
4      # bloco de código a ser executado se condição2 for verdadeira
```

## Estrutura else:

Executa um bloco de código se nenhuma das condições anteriores for verdadeira.



```
1  if condição1:
2      # bloco de código
3  elif condição2:
4      # bloco de código
5  else:
6      # bloco de código se nenhuma condição for verdadeira
```

# Revisão de Condicionais

As condicionais em programação são instruções que controlam o fluxo de execução de um programa.

As principais estruturas condicionais incluem **if**, **elif** e **else**.

```
1  numero = float(input("Digite um número: "))
2
3  if numero > 0:
4      print("O número é positivo.")
5  elif numero < 0:
6      print("O número é negativo.")
7  else:
8      print("O número é zero.")
```

O código acima para verificar se um número é positivo, negativo ou zero com base na entrada do usuário.

# ATIVIDADE PRÁTICA

## **Categoria de Idade:**

Desenvolva um programa que peça a idade do usuário e informe se ele é criança, adolescente, adulto ou idoso.

## **Classificação de Notas:**

Crie um programa que solicite uma nota de 0 a 100 e informe o conceito (A, B, C, D, F) com base na nota.

# ATIVIDADE PRÁTICA

## **Verificar Signo:**

Escreva um programa que peça o dia e o mês de nascimento do usuário e informe o signo correspondente.

## **Sistema de Login:**

Desenvolva um programa que simule um sistema de login.

O programa deve pedir o nome de usuário e a senha e verificar se correspondem a um usuário pré-cadastrado. Exiba mensagens apropriadas para login bem-sucedido ou falha.

# Revisão de Estruturas de Repetição

## Laço while:

Executa um bloco de código enquanto uma condição especificada for verdadeira.  
A condição é verificada antes de cada iteração.



```
1  while condição:  
2      # bloco de código a ser repetido
```

## Laço for:

Itera sobre uma sequência (como uma lista, tupla ou string) e executa um bloco de código para cada item na sequência.



```
1  for variável in sequência:  
2      # bloco de código a ser repetido
```



# Revisão de Estruturas de Repetição

## Uso do while:

O laço while é útil quando o número de iterações não é conhecido de antemão e depende de uma condição dinâmica.



```
1 contador = 1
2 while contador <= 5:
3     print(contador)
4     contador += 1
5 # Saída: 1, 2, 3, 4, 5
```

## Uso do for:

O laço for é ideal para iterar sobre elementos de uma sequência e quando o número de iterações é conhecido ou definido pela sequência.



```
1 frutas = ["maçã", "banana", "cereja"]
2 for fruta in frutas:
3     print(fruta)
4 # Saída: maçã, banana, cereja
```

# ATIVIDADE PRÁTICA

## **Contagem Regressiva:**

Desenvolva um programa que use um laço while para exibir uma contagem regressiva de 10 até 1 e, em seguida, exiba "Feliz Ano Novo!".

## **Contagem Regressiva:**

Desenvolva um programa que use um laço for para exibir uma contagem regressiva de 10 até 1 e, em seguida, exiba "Feliz Ano Novo!".

# ATIVIDADE PRÁTICA

## **Soma de Números Pares:**

Crie um programa que use um laço while para somar todos os números pares de 1 a 100 e exiba o resultado.

## **Tabuada de um Número:**

Faça um programa que solicite um número ao usuário e use um laço for para exibir a tabuada desse número (de 1 a 10).

# ATIVIDADE PRÁTICA

## **Verificação de Palíndromo:**

Escreva um programa que solicite uma palavra ao usuário e use um laço while para verificar se a palavra é um palíndromo (lê-se da mesma forma de trás para frente).

## **Sistema de Login com Tentativas Limitadas:**

Desenvolva um programa que simule um sistema de login.

O programa deve solicitar o nome de usuário e senha até que o usuário insira as credenciais corretas ou até que o número máximo de tentativas seja atingido. Use um laço while com uma condicional para verificar as credenciais e limitar as tentativas.



# SEU MÓDULO TERMINA AQUI!

RESPONDA NOSSA  
PESQUISA DE SATISFAÇÃO



IN

**INFINITY SCHOOL**  
VISUAL ART CREATIVE CENTER



# INFINITY SCHOOL

VISUAL ART CREATIVE CENTER

AULA 06 – Revisão da Logica da Programação