



CENTRO UNIVERSITÁRIO ITOP
UNITOP

AGRONOMIA

Introdução à Informática

Prof. M.Sc. **Paulo Augusto**

Palmas, 22 de abril de 2025

Programação

❑ Computador

- É extremamente eficiente e rápido para realizar operações e cálculos complexos;
- Não raciocina ou reflete a intenção do programador;
- Lenda da programação...
- Interpreta de maneira literal. (como criança);
- Sempre que inicia uma programação começa do **Zero**.

Programação

Lógica de Programação (como funciona)

Envolve pensar de forma estruturada e sequencial para resolver um problema utilizando um computador. Ela é a base para escrever códigos que o computador consiga entender e executar.

- **Entrada (Input):** Dados ou informações que o programa recebe.
- **Processamento:** Ações ou cálculos que o programa faz com os dados.
- **Saída (Output):** Resultado que o programa gera a partir do processamento.

Programação

➤ Pseudocódigos;

- Não é uma linguagem;
- Sintaxe;

```
1 programa
2 {
3     funcao inicio()
4     {
5         inteiro a=1, b=0
6         se (a<b)
7             {
8
9             }
10    }
11 }
```

Programação

Lógica de Programação

➤ Algoritmos;

- Conjunto de passos;
- Sequência lógica;
- Organização coerente das instruções.

Instrução:

- Se a porta estiver aberta:
- Pegar o lixo;
- Levar o lixo para fora.
- Se a porta estiver aberta:
- Pegue a chave;
- Enfie na fechadura;
- Gire no sentido... até parar;
- Abra a porta;
- Pegue o lixo;
- Leve o lixo para fora;

Programação

❑ Lógica de Programação

- Variáveis e Constantes;
 - É uma espaço reservado na memória para guardar algum dado ou informação;

Programação

❑ Lógica de Programação

➤ algoritmos;

Algoritmo CalcularAreaRetangulo

var

base, altura, area: real

Inicio

Escreva("Digite o valor da base: ")

Leia(base)

Escreva("Digite o valor da altura: ")

Leia(altura)

area <- base * altura

Escreva("A área do retângulo é: ", area)

Fim

```
import java.util.Scanner
```

```
public class CalcularAreaRetangulo {
```

```
    public static void main(String[] args) {
```

```
        Scanner sc = new Scanner(System.in);
```

```
        double base, altura, area;
```

```
        System.out.print("Digite o valo da base: ");
```

```
        base = sc.nextDouble();
```

```
        System.out.print("Digite o valo da altura: ");
```

```
        altura = sc.nextDouble();
```

```
        System.out.println("A área do retângulo é: " +area);
```

```
        sc.close();
```

```
    }
```

```
}
```

Programação

Lógica de Programação (estrutura básica)

Essas estruturas são fundamentais para controlar o fluxo do programa. As principais são:

- **Sequência;**
- **Seleção (Condicionais);**
- **Repetição (Laços).**

Programação

Estrutura (Sequência)

A sequência é a execução das instruções uma após a outra, na ordem em que são escritas no código.

Ex.:

1. Receber um número;
2. Multiplicar o número por 2;
3. Exibir o resultado.

```
1  Início
2  Receber número
3  resultado ← número * 2
4  Escreva resultado
4  Fim
```

Programação

Estrutura (Seleção (Condicionais))

Permite que o programa execute diferentes ações dependendo de uma condição. A estrutura mais comum é o **if**.

```
# Se o número for maior que 10, imprime "Maior"  
# Caso contrário, imprime "Menor ou igual"
```

```
numero = 12  
if numero > 10:  
    print("Maior")  
else:  
    print("Menor ou igual")
```

Programação

Estrutura (Seleção (Condicionais))

Permite que o programa execute diferentes ações dependendo de uma condição. A estrutura mais comum é o **if**.

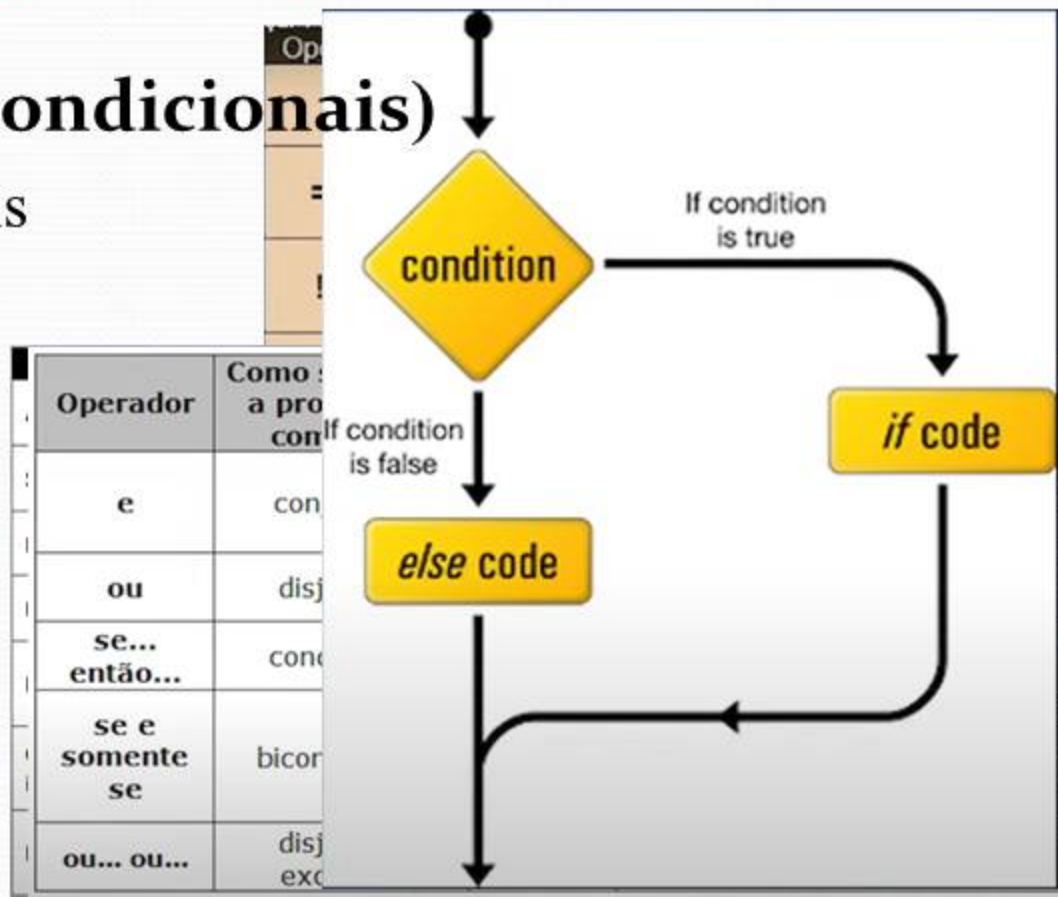
```
1 Início
2   Leia número
3   Se número > 10 então
4       Escreva "O número é maior que 10."
5   senão
6       Escreva "O número é menor ou igual a 10."
7 Fim
```

Programação

Estrutura (Seleção (Condicionais))

➤ Operadores e Condicionais

- Operações matemáticas;
- Comparações;
- Lógicas;
- Fluxos condicionais.



Programação

Estrutura (Repetição, Laços)

Laços (ou loops) permitem repetir um bloco de código várias vezes, **for** e **while**.

```
# Imprimir números de 1 a 5
```

```
for i in range(1, 6):  
    print(i)
```

```
# Imprimir números de 1 a 5
```

```
i = 1  
while i <= 5:  
    print(i)  
    i = i + 1
```

```
1 Início  
2 Para i de 1 até 5 faça  
3     Escreva "O número: ", i  
4 Fim Para  
5 Fim
```

```
1 Início  
2 i = 1  
3 Enquanto i <= 5 faça  
4     Escreva "O número: ", i  
5     i = i + 1  
6 Fim Enquanto  
7 Fim
```


Programação

Variáveis e Constantes

- É um espaço reservado na memória para guardar algum dado ou informação;
- Tipos mais comuns:
 - **Inteiro (int)**: Números inteiros (ex: 1, -5, 20)
 - **Ponto flutuante (float)**: Números com casas decimais (ex: 3.14, -0.5)
 - **Texto (string)**: Sequências de caracteres (ex: "Olá, Mundo!")
 - **Booleano (bool)**: Verdadeiro ou falso (ex: True, False)

Programação

Variáveis e Constantes

```
idade = 25          # Variável do tipo inteiro
nome = "Carlos"     # Variável do tipo string
altura = 1.75       # Variável do tipo float
ativo = True        # Variável do tipo booleano
```

Programação

Linguagem de programação

- A verdadeira (e única) linguagem de computador é a linguagem de máquina (binária), mas para a maioria das pessoas essa linguagem é ininteligível.
- Os programadores precisam de um intermediário entre eles e a máquina. Uma linguagem que facilite o processo de programar computadores mas que no final seja traduzida para a linguagem de máquina.

Programação

Linguagem de programação

- Uma linguagem de programação permite desenvolver os conjuntos de instruções que constituem o programa de computador.
- Existem muitas linguagens de programação diferentes, cada uma com seu vocabulário, gramática e usos exclusivos.

Programação

Linguagem de programação

- Linguagens de Baixo nível
 - Linguagem de máquina
 - ASSEMBLY
- Linguagens de Alto nível
 - COBOL
 - JAVA
 - C
 - PYTHON
- Linguagens Visuais
 - Programação em blocos

Linguagem de
Baixo Nível
ASSEMBLY

Programação

Este programa calcula a média de uma lista de números.

```
DOSSEG
.MODEL          SMALL
EXTRN           _IntDivide:PROC
.CODE
PUBLIC          _Average
_Average PROC
    push        bp
    mov         bp, sp
    les         bx, [bp+4]
    mov         cx, [bp+8]
    mov         ax, 0
AverageLoop:
    add         ax, es: [bx]
    add         bx, 2
    loop        AverageLoop
    push        WORD PRG [bp+8]
    push        ax
    call        _IntDivide
    add         sp, 4
    pop         bp
    ret
_Average      ENDP
END
```

Linguagem de
Alto Nível

Linguagem C

Programação

```
#include <stdio.h>

int main() {
    float num, den;
    printf("Digite o numerador: ");
    scanf("%f", &num);
    printf("Digite o denominador: ");
    scanf("%f", &den);
    if (den == 0) {
        printf("Numerador não pode ser zero\n");
        return 0;
    }
    printf("Resultado da divisão: %.2f", num/den);
    return 0;
}
```

Este programa calcula a divisão
de dois números quaisquer

Linguagem de programação em Blocos - Scratch

<https://scratch.mit.edu>



Este programa calcula a área de um retângulo.

Programação

Algoritmo para dividir dois números

1. Iniciar o programa
2. Definir variáveis
3. Apresentar mensagem para o usuário digitar o dividendo
4. Ler o dividendo
5. Apresentar mensagem para o usuário digitar o divisor
6. Ler o divisor
7. Se divisor igual a zero então apresentar mensagem e terminar o programa
8. Apresentar o resultado da divisão do dividendo pelo divisor
9. Terminar o programa

Programação

Python – *O que é?*

Python é uma linguagem de programação de alto nível, fácil de aprender, e muito poderosa. É uma das linguagens mais populares no mundo da programação devido à sua simplicidade e versatilidade.

- **Sintaxe simples:** Não precisa de símbolos complexos como `{ }` ou `;` para estruturar o código.
- **Usos variados:** é usada para desenvolver websites, automações, jogos, inteligência artificial, ciência de dados, e muito mais.

Programação

Python – *Escrevendo seu Primeiro Código Python*

<https://www.jdoodle.com/>

JDoodle.com

Write code online.

Write your code in 100+ languages and frameworks with no local setup.

Start Coding

1

```
public class MyClass {  
    public static void main(String args[]) {  
        int x=10;  
        int y=25;  
        int z=x+y;  
  
        System.out.println("Sum of x+y = " + z);  
    }  
}
```

2

3

Create a new Project

Pick a language/framework

Search IDE Compiler or Terminal

Python3

C++

AWK

Programação

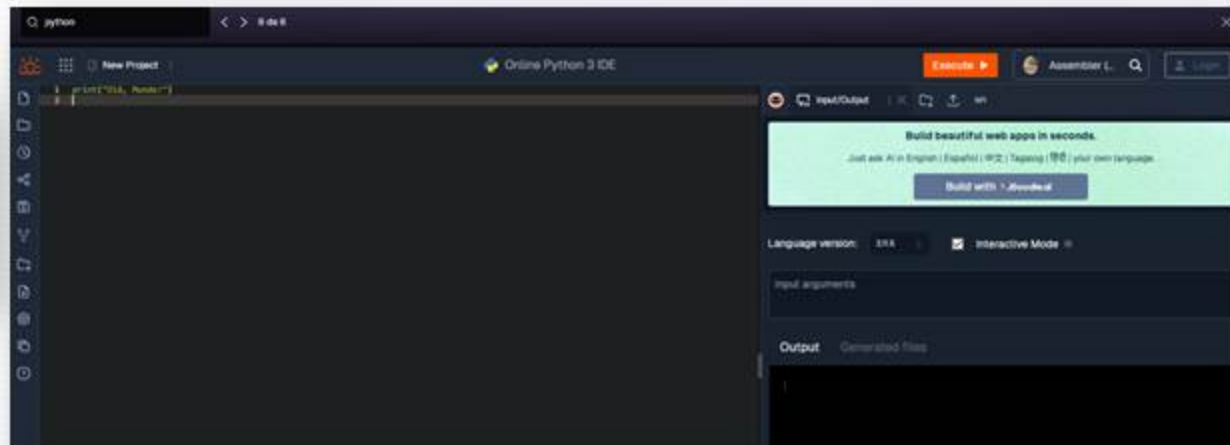
Python – *Escrevendo seu Primeiro Código Python*

No Python, você pode usar qualquer editor de texto para escrever seu código (como o **Bloco de Notas**, **Visual Studio Code**, **PyCharm**, dentre outros).

Ex.: **Olá, Mundo!**

```
print("Olá, Mundo!")
```

```
1 #include <stdio.h>
2
3 int main() {
4     printf("Olá, Mundo!\n");
5     return 0;
6 }
```



Programação

Python – *Variáveis e Tipos de Dados*

Em Python pode armazenar valores em variáveis. Uma variável é como um "recipiente" para armazenar dados.

Tipos de dados comuns:

- **Inteiros** (`int`): números inteiros, como 5, 100, -7
- **Pontos flutuantes** (`float`): números com ponto decimal, como 3.14, -0.001
- **Strings** (`str`): texto, como "Olá", "Python"
- **Booleanos** (`bool`): valores True ou False

```
nome = "João"           # String
idade = 25               # Inteiro
altura = 1.75            # Float
is_estudante = True      # Booleano

print(nome)              # Vai exibir "João"
print(idade)              # Vai exibir 25
```

Programação

Python – *Operações Matemáticas*

Python permite realizar operações matemáticas facilmente.

```
a = 10
b = 5

soma = a + b
subtracao = a - b
multiplicacao = a * b
divisao = a / b

print(soma)
print(subtracao)
print(multiplicacao)
print(divisao)
```

Programação

Python – *Estruturas de Controle*

As estruturas condicionais são usadas para tomar decisões no seu código.

Condicionais (if, else)

```
idade = 18

if idade >= 18:
    print("Você é maior de idade!")
else:
    print("Você é menor de idade!")
```


Programação

Python – *Laços de Repetição*

Os laços de repetição permitem executar um bloco de código várias vezes.

Laços de repetição (for, while):

```
# Exemplo de laço 'for'
```

```
for i in range(5): # Repete de 0 a 4  
    print(i)
```

```
# Exemplo de laço 'while'
```

```
contador = 0  
while contador < 5:  
    print(contador)  
    contador += 1 # Incrementa contador
```

Programação

Python – *Funções*

São blocos de código que você pode reutilizar. Elas são definidas com a palavra-chave `def`.

```
def saudacao(nome):  
    print(f"Olá, {nome}!")  
  
saudacao("Maria")  # Chama a função e imprime "Olá, Maria!"
```

Programação

Python – *Função input()*

Em Python a interação com o usuário é feita principalmente com a função `input()`. Ela permite capturar o que o usuário digita no teclado. O valor capturado sempre é do tipo `string` (texto), então, se for necessário outro tipo de dado (como números), será necessário convertê-lo.

```
# Recebe o nome do usuário
nome = input("Qual é o seu nome? ")

# Exibe uma mensagem para o usuário
print(f"Olá, {nome}!")
```

Programação

Python – *Função input() – Trabalhando com números*

Como mencionado no slide anterior, a função `input()` sempre retorna uma string. Se quisermos trabalhar com números (inteiros ou decimais), precisamos converter a entrada para o tipo apropriado usando `int()` (para números inteiros) ou `float()` (para números decimais)..

```
# Recebe dois números do usuário
numero1 = input("Digite o primeiro número: ")
numero2 = input("Digite o segundo número: ")

# Converte as entradas para números inteiros
numero1 = int(numero1)
numero2 = int(numero2)

# Realiza uma operação (soma)
soma = numero1 + numero2

# Exibe o resultado
print(f"A soma de {numero1} e {numero2} é {soma}.")
```

Programação

Python – *Funções*

São blocos de código que você pode reutilizar. Elas são definidas com a palavra-chave `def`.

```
def saudacao(nome):  
    print(f"Olá, {nome}!")  
  
nome = input("Qual é o seu nome? ")  
  
saudacao(nome)
```


- Fazer um programa em python que faça uma divisão de dois números.

```
dividendo = float(input("Digite o dividendo: "))
divisor = float(input("Digite o divisor: "))

if divisor == 0:
    print("Divisor não pode ser zero!!!")
else:
    resultado = dividendo / divisor
    print("Resultado da divisão: {:.2f}".format(resultado))
```

- Fazer um programa em python que peça ao usuário para digitar um número, em seguida mostre a tabuada de multiplicar do número digitado.

```
1  # Programa de Tabuada em Python
2  numero = int(input("Digite um número para ver sua tabuada: "))
3
4  print(f"\nTabuada do {numero}:")
5  print("-" * 15)  # Linha divisória
6
7  for i in range(1, 11):  # De 1 a 10
8      resultado = numero * i
9      print(f"{numero} x {i:2} = {resultado:3}")  # :2 e :3 para alinhamento
10
11 print("-" * 15)  # Linha divisória
```