**Aula Inaugural: Python e Lógica de Programação para Iniciantes**

**Parte 1: O que é Programar? (A Lógica por Trás de Tudo)**

Imagine que você precisa ensinar um robô a fazer um sanduíche. Você não pode simplesmente dizer "faça um sanduíche". O robô não entende isso. Você precisa dar instruções **extremamente precisas e em uma ordem específica**:

1. Pegue uma fatia de pão.
2. Coloque a fatia de pão em um prato.
3. Pegue a faca.
4. Abra o pote de maionese.
5. Passe a faca na maionese.
6. Espalhe a maionese na fatia de pão.
7. ... e assim por diante.

**Programar é exatamente isso:** dar uma sequência de instruções lógicas e ordenadas para que um computador execute uma tarefa. A **"Lógica de Programação"** não é uma linguagem, mas sim a *habilidade de pensar* dessa forma estruturada, quebrando um grande problema em pequenos passos lógicos.

Os pilares da lógica de programação são:

* **Sequência:** As instruções são executadas uma após a outra, na ordem em que foram escritas. (Como na receita do sanduíche).
* **Variáveis:** Pense em variáveis como "caixas" com etiquetas. Você usa essas caixas para guardar informações que vai precisar usar depois. Por exemplo, uma caixa chamada nome\_do\_usuario para guardar o nome "Maria".
* **Decisões (Condicionais):** O programa precisa tomar decisões. "Se estiver chovendo, pegue o guarda-chuva. Senão, pegue os óculos de sol." Isso permite que seu programa se comporte de maneiras diferentes dependendo da situação.
* **Repetição (Loops):** Às vezes, você precisa repetir uma tarefa várias vezes. Em vez de escrever a mesma instrução 100 vezes, você diz ao computador: "Repita esta ação 100 vezes".

**Parte 2: O que é Python? (A Ferramenta)**

Se a Lógica de Programação é a *receita*, as linguagens de programação (como o Python) são o *idioma* em que você escreve essa receita para o computador.

**Python** é uma linguagem de programação extremamente popular, e por ótimos motivos para quem está começando:

* **Sintaxe Simples e Legível:** Escrever em Python parece muito com escrever em inglês. É limpo e fácil de entender, sem símbolos complexos e regras rígidas que outras linguagens exigem.
* **Versátil:** Com Python, você pode criar sites, analisar dados, automatizar tarefas, desenvolver inteligência artificial, criar jogos e muito mais.
* **Comunidade Gigante:** Se você tiver uma dúvida, é quase certeza que alguém já teve a mesma dúvida e a resposta está a uma busca no Google de distância.

**Parte 3: Mão na Massa! Seu Primeiro Programa**

Vamos unir a Lógica e o Python na prática. Não se preocupe em instalar nada agora. Usaremos uma ferramenta online chamada **Google Colab**. É gratuita e roda direto no seu navegador.

**Acesse aqui:** <https://colab.research.google.com/> e clique em "Novo notebook".

**1. "Olá, Mundo!" - A Sequência**

O primeiro programa de todo programador. É uma tradição! No Colab, digite o seguinte na célula de código e aperte **Shift + Enter** (ou clique no ícone de "play" ▶️):

Python

print("Olá, Mundo!")

**O que aconteceu aqui?**

* print() é uma **função** do Python. Funções são como "comandos" pré-prontos que executam uma ação. A função print() tem uma única tarefa: exibir na tela o que estiver dentro dos parênteses.
* "Olá, Mundo!" é o texto que queremos exibir. Em programação, textos são chamados de **strings** e quase sempre são colocados entre aspas.

**2. Guardando Informações - Variáveis**

Vamos criar nossas "caixas" para guardar dados.

Python

# Criando uma variável para guardar um nome (texto/string)

nome = "Ana"

# Criando uma variável para guardar uma idade (número inteiro)

idade = 30

# Agora, vamos usar essas variáveis com a função print()

print("Olá, meu nome é")

print(nome)

print("e eu tenho")

print(idade)

print("anos.")

**O que aconteceu aqui?**

* nome = "Ana": Criamos uma "caixa" (variável) chamada nome e guardamos o texto "Ana" dentro dela.
* idade = 30: Criamos uma variável idade e guardamos o número 30 nela. Note que para números não usamos aspas.
* Quando usamos print(nome), o Python não exibe a palavra "nome", mas sim o **conteúdo** que está guardado dentro da variável nome.

**3. Tomando Decisões - Condicionais (if/else)**

Vamos criar um programa que decide se uma pessoa é maior de idade.

Python

idade = 25

# A mágica acontece aqui!

if idade >= 18:

print("Você é maior de idade. Pode entrar na festa!")

else:

print("Você é menor de idade. Não pode entrar.")

*Experimente mudar o valor da variável idade para 15 e rodar o código de novo para ver o que acontece.*

**Desvendando o código:**

* if idade >= 18:: Aqui está a lógica. **SE** (if) o valor dentro da variável idade for maior ou igual (>=) a 18, **então** execute o código que está abaixo e com um recuo (o print com a mensagem para maiores de idade).
* else:: **SENÃO** (else), ou seja, se a condição do if não for verdadeira, execute o código que vem depois dele (o print para menores de idade).
* A **indentação** (o espaço antes do print) é MUITO importante em Python. É ela que diz ao programa qual código pertence ao if e qual pertence ao else.

**4. Pedindo Informações ao Usuário**

Nosso programa fica mais legal se ele interagir com a pessoa que o está usando. Para isso, usamos a função input().

Python

# A função input() mostra uma mensagem e espera o usuário digitar algo

nome\_usuario = input("Qual é o seu nome? ")

# A função input() SEMPRE retorna texto. Precisamos converter a idade para número.

idade\_texto = input("Qual é a sua idade? ")

idade\_numero = int(idade\_texto) # int() transforma o texto em número inteiro

print(f"Olá, {nome\_usuario}! Que legal que você tem {idade\_numero} anos!")

# Agora vamos usar a decisão com os dados do usuário

if idade\_numero >= 18:

print("Pela sua idade, você já pode dirigir.")

else:

print("Pela sua idade, você ainda não pode dirigir.")

**O que há de novo?**

* input("Qual é o seu nome? "): Exibe a pergunta na tela e o que o usuário digitar será guardado na variável nome\_usuario.
* int(idade\_texto): Como o input sempre trata a resposta como texto, não podemos fazer comparações matemáticas ("15" é diferente de 15). A função int() converte o texto "15" no número inteiro 15.
* print(f"Olá, {nome\_usuario}!"): Essa é uma "f-string". É um jeito fácil de misturar texto e variáveis. O f antes das aspas diz ao Python para substituir {nome\_usuario} pelo valor guardado nessa variável.

**Resumo da Aula e Próximos Passos**

Parabéns! Você acabou de aprender os conceitos mais fundamentais da programação e escreveu seus primeiros códigos em Python.

* **Lógica de Programação** é a forma de pensar, de quebrar problemas em passos.
* **Python** é a linguagem que usamos para escrever esses passos para o computador.
* Vimos **Sequência** (código executado em ordem), **Variáveis** (caixas para guardar dados), **Condicionais** (tomada de decisão com if/else) e como interagir com o usuário (input).

**O que vem agora?** O segredo para aprender a programar é **praticar**.

1. **Modifique os códigos:** Mude os nomes, as idades, as mensagens. Crie um condicional que verifica se um número é maior que 100.
2. **Explore a repetição:** O próximo passo na lógica seria aprender sobre **loops** (for e while) para repetir tarefas. Pesquise por "loop for em Python para iniciantes".
3. **Crie pequenos desafios:**
   * Faça um programa que pede duas notas de um aluno e diz se ele foi aprovado (média maior que 7).
   * Crie um programa que pergunta o nome do usuário e, se for o seu nome, ele exibe uma mensagem especial.

A programação é uma jornada de aprendizado contínuo. Seja curioso, não tenha medo de errar (os erros ensinam muito!) e divirta-se criando. Você deu o primeiro e mais importante passo!