

# AULA 1: FUNDAMENTOS DO PYTHON E PRIMEIROS PASSOS

**Objetivo:** Introduzir o Python de forma prática, abordando seus fundamentos essenciais, desde a instalação até a manipulação de variáveis e operadores.

## 1. O que é Python e por que ele é tão popular?

Python é uma linguagem de programação interpretada, de alto nível e com tipagem dinâmica. Seus principais diferenciais são:

- **Simplicidade e legibilidade:** Sua sintaxe é clara e objetiva, o que facilita o aprendizado e a escrita de código.
- **Versatilidade:** Pode ser utilizada para desenvolvimento web, automação, ciência de dados, IA, entre outras áreas.
- **Grande comunidade:** O suporte da comunidade é um diferencial para quem está aprendendo.
- **Multiplataforma:** Python pode ser executado em Windows, macOS e Linux sem alterações significativas no código.

## 2. Instalando e Configurando o Ambiente

Para começar a programar em Python, podemos utilizar:

- **Google Colab:** Ambiente online baseado em Jupyter Notebook. Ideal para iniciantes, pois não exige instalação.
- **PyCharm:** IDE completa, voltada para desenvolvimento profissional.
- **Visual Studio Code:** Leve, com excelente suporte para Python.
- **Terminal Python:** Para quem prefere rodar scripts diretamente pelo terminal do sistema operacional.

## 3. Variáveis e Tipos de Dados

Variáveis são "caixas" que armazenam valores. Python tem tipagem dinâmica, ou seja, podemos reatribuir valores de tipos diferentes sem precisar declará-los previamente.

**Tipos de Dados Principais:**

- **Inteiros (int):** Representam números inteiros. Exemplo:

idade = 25

- **Float (float):** Números com casas decimais. Exemplo:

altura = 1.75

- **Strings (str):** Sequências de caracteres. Exemplo:

```
nome = "João"
```

- **Booleanos (bool):** Verdadeiro ou falso. Exemplo:

```
ativo = True
```

### **Exemplo prático:**

Criando uma variável para representar um produto em uma loja virtual:

```
produto = "Notebook Gamer"
```

```
preco = 4999.90
```

```
quantidade_estoque = 15
```

```
disponivel = True
```

```
print(f"Produto: {produto}")
```

```
print(f"Preço: R$ {preco:.2f}")
```

```
print(f"Quantidade em estoque: {quantidade_estoque}")
```

```
print(f"Disponível para compra? {disponivel}")
```

## **4. Entrada e Saída de Dados**

### **Função print()**

Usada para exibir mensagens na tela.

```
print("Olá, mundo!")
```

### **Função input()**

Permite capturar dados do usuário.

```
nome = input("Digite seu nome: ")
```

```
print(f"Bem-vindo, {nome}!")
```

## **5. Operadores em Python**

### **Operadores Aritméticos:**

Realizam cálculos matemáticos.

```
soma = 10 + 5
```

```
subtracao = 10 - 5
```

```
multiplicacao = 10 * 5
```

```
divisao = 10 / 5
```

```
divisao_inteira = 10 // 3
```

```
resto = 10 % 3
```

```
potencia = 2 ** 3
```

### **Operadores de Comparação:**

Utilizados para comparação entre valores.

```
print(10 > 5) # True
```

```
print(10 < 5) # False
```

```
print(10 == 10) # True
```

```
print(10 != 5) # True
```

### **6. Estruturas Condicionais (if, elif, else)**

Permitem a execução condicional de blocos de código.

#### **Exemplo prático:**

```
idade = int(input("Digite sua idade: "))
```

```
if idade < 18:
```

```
    print("Você é menor de idade.")
```

```
elif idade >= 18 and idade < 60:
```

```
    print("Você é adulto.")
```

```
else:
```

```
    print("Você é idoso.")
```

# AULA 2: ESTRUTURAS DE CONTROLE E REPETIÇÃO

**Objetivo:** Ensinar loops, listas, tuplas e dicionários, essenciais para manipulação de dados em Python.

## 1. Laços de Repetição (Loops)

Loops permitem repetir blocos de código.

### 1.1. Loop for

Usado para iterar sobre sequências.

```
i = 2
```

```
for i in range(5):
```

```
    print(f"Iteração número {i}")
```

### 1.2. Loop while

Executa enquanto a condição for verdadeira.

```
contador = 0
```

```
while contador < 5:
```

```
    print(f"Contador: {contador}")
```

```
    contador += 1
```

## 2. Listas e Tuplas

Estruturas para armazenar múltiplos valores.

### Listas (Mutáveis)

```
frutas = ["maçã", "banana", "laranja"]
```

```
print(frutas[0]) # maçã
```

```
frutas.append("uva") # Adiciona um novo item
```

### Tuplas (Imutáveis)

```
cores = ("vermelho", "azul", "verde")
```

```
print(cores[1]) # azul
```

## 3. Dicionários

Coleção de pares chave-valor.

```
produto = {  
    "nome": "Notebook",  
    "preco": 2500.00,  
    "estoque": 10  
}  
  
print(produto["nome"]) # Notebook  
produto["preco"] = 2300.00 # Atualiza o preço
```