

# Apostila 2: Python - Manipulação de variáveis

## 1. Manipulação de Variáveis e Constantes

Em Python, variáveis são usadas para armazenar valores. As constantes não são uma estrutura nativa, mas por convenção, usamos nomes em letras maiúsculas para indicá-las.

```
idade = 25 # Variável  
PI = 3.14159 # Constante (por convenção)
```

## 2. Tipo de Variáveis

Os principais tipos são:

- `int`: Números inteiros
- `float`: Números decimais
- `str`: Cadeias de caracteres
- `bool`: Valores booleanos (True ou False)

```
numero = 10  
preco = 10.5  
nome = "Python"  
ativo = True
```

## 3. Manipulação de Strings

Podemos acessar caracteres individuais, concatenar e manipular strings.

```
texto = "Python"  
print(texto[0]) # P  
print(texto.upper()) # PYTHON  
print(texto.lower()) # python
```

## 4. Operadores de Atribuição

Usados para atribuir valores a variáveis.

```
x = 10
```

```
x += 5 # x = x + 5
x -= 3 # x = x - 3
```

## 5. Operadores Aritméticos

```
soma = 10 + 5
subtracao = 10 - 5
multiplicacao = 10 * 5
divisao = 10 / 5
modulo = 10 % 3 # Resto da divisão
exponenciacao = 2 ** 3 # 2^3 = 8
```

## 6. Operadores de Comparação

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

## 7. Operadores Lógicos - Tabela Verdade

- **and**: Retorna **True** se ambas as condições forem verdadeiras.
- **or**: Retorna **True** se pelo menos uma condição for verdadeira.
- **not**: Inverte o valor lógico.

```
print(True and False) # False
print(True or False) # True
print(not True) # False
```

## 8. Operadores de Identidade

Verificam se duas variáveis são o mesmo objeto na memória.

```
a = [1, 2, 3]
b = a
c = [1, 2, 3]
print(a is b) # True
print(a is c) # False
```

## 9. Operadores de Associação

Verificam se um valor está presente em uma sequência.

```
lista = ["boi", "gato"]
print("boi" in lista) # True
print("cachorro" not in lista) # True
```

## 10. Estruturas de Decisão: if, elif, else

```
idade = 18
if idade < 18:
    print("Menor de idade")
elif idade == 18:
    print("Tem exatamente 18 anos")
else:
    print("Maior de idade")
```

## 11. Estrutura de Repetição for

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

## 12. Estrutura de Repetição while

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

## 13. Comandos de Controle de Fluxo: break, continue e pass

```
for i in range(10):
    if i == 5:
        break # Interrompe o loop
    print(i)
for i in range(10):
    if i % 2 == 0:
        continue # Pula para a próxima iteração
    print(i)
def funcao():
    pass # Placeholder para futuras implementações
```

# Exercícios

1. Crie uma variável chamada `nome` e atribua seu nome a ela. Em seguida, imprima a variável.
2. Escreva um programa que receba dois números e imprima a soma deles.
3. Verifique se um número é positivo, negativo ou zero.
4. Crie uma lista com três elementos e verifique se um determinado valor está na lista.
5. Utilize um loop `for` para imprimir os números de 1 a 10.
6. Utilize um loop `while` para contar até 5.
7. Escreva um programa que pare de executar quando um número aleatório gerado for maior que 90 (use a biblioteca `random`).
8. Utilize `break`, `continue` e `pass` em um exemplo prático.
9. Ler dois números inteiros, executar e mostrar o resultado das seguintes operações: adição, subtração, multiplicação e divisão.
10. Efetuar o cálculo da quantidade de litros de combustível gasto em uma viagem, utilizando um automóvel que faz 12 Km por litro. Para obter o cálculo, o usuário deve fornecer o tempo gasto na viagem e a velocidade média durante ela. Desta forma, será possível obter a distância percorrida com a fórmula  $DISTANCIA = TEMPO * VELOCIDADE$ . Tendo o valor da distância, basta calcular a quantidade de litros de combustível utilizada na viagem, com a fórmula:  $LITROS\_USADOS = DISTANCIA / 12$ . O programa deve apresentar os valores da velocidade média, tempo gasto na viagem, a distância percorrida e a quantidade de litros utilizada na viagem