

Roteiro de Aula: Introdução a Variáveis e Estruturas de Dados em Python

Objetivo da Aula:

Apresentar os conceitos de variáveis, operações básicas, e estruturas de dados, utilizando exemplos práticos em Python.

avisos sobre as atividades da aula 1

Tópicos Principais:

1. Tipos de Variáveis
2. Operações com Variáveis
3. Estruturas de Dados (Listas e Tuplas)
4. Algoritmos com Variáveis (alguns exemplos)

prática extra:

5. Mostrar opção da <https://judge.beecrowd.com/pt/categories> como mais uma alternativa para praticar
6. Mostrar turtle do python também como alternativa de praticar

1. Introdução às Variáveis em Python

- **Explicação inicial:**
 - Variáveis são fundamentais para o armazenamento e manipulação de dados.
 - Exemplos básicos de variáveis: Inteiros, Flutuantes, Strings, Booleanos.

Exemplo prático (Código no Python):

python

```
# Exemplo de variáveis em Python
num_inteiro = 10
num_decimal = 5.5
nome = "Maria"
ativo = True

print(num_inteiro, num_decimal, nome, ativo)
```

- **Objetivo:** Mostrar como criar e utilizar variáveis de diferentes tipos.
-

2. Operações com Variáveis

- **Operadores matemáticos básicos:**
 - Soma, subtração, multiplicação, divisão e exponenciação.

Exemplo prático de adição e multiplicação:

```
python
x = 100
y = 50
soma = x + y
multiplicacao = x * y

print("Soma:", soma)
print("Multiplicação:", multiplicacao)
```

- **Objetivo:** Mostrar como realizar operações simples em Python e exibir os resultados.

Exemplo prático com operadores compostos:

```
python
a = 3.5
b = 4
c = 6
x = 5
y = 1
resultado = a * b + c / (x + y)
print("Resultado:", resultado)
```

- **Objetivo:** Demonstrar o uso de vários operadores em uma única expressão.

Operações com Strings:

```
nome = "Jose"
sobrenome = "Silva"
nome_completo = nome + " " + sobrenome
print(nome_completo)
```

```
fruta = "Limão"
fruta_repetida = (fruta + " ") * 3
```

```
print(fruta_repetida)
```

3. Estruturas de Dados: Listas e Tuplas

- **Introdução às Listas:**

- Uma lista pode conter diferentes tipos de elementos e é mutável.

Exemplo prático com lista:

python

```
lista = [1, 2, 3, 'quatro', 5.0]
print(lista)
lista.append(6) # Adiciona um novo elemento à lista
print(lista)
```

- **Objetivo:** Mostrar como criar e manipular uma lista em Python.
- **Introdução às Tuplas:**
 - Diferente das listas, as tuplas são imutáveis.

Exemplo prático com tupla:

python

```
tupla = (1, 2, 3, 4, 5)
print(tupla)
```

- **Objetivo:** Mostrar a diferença entre listas e tuplas.

```
# Lista de números inteiros
numeros = [10, 20, 30, 40, 50]
```

```
# Soma de todos os elementos
soma_total = sum(numeros)
print("Soma total:", soma_total)
```

```
# Acessando elementos
primeiro = numeros[0] # Primeiro elemento
ultimo = numeros[-1] # Último elemento
print("Primeiro número:", primeiro)
print("Último número:", ultimo)
```

```
# Adicionando um número à lista
numeros.append(60)
```

```
print("Lista após adicionar 60:", numeros)

# Removendo o último número
numeros.pop()
print("Lista após remover o último número:", numeros)
```

4. Algoritmos Utilizando Variáveis em Python

Exemplo 1: Calcular a média de três notas

```
python
# Solicitar as três notas ao usuário
nota1 = float(input("Digite a primeira nota: "))
nota2 = float(input("Digite a segunda nota: "))
nota3 = float(input("Digite a terceira nota: "))

# Calcular a média das notas
media = (nota1 + nota2 + nota3) / 3

# Exibir a média
print("A média das três notas é:", media)
```

- **Objetivo:** Ensinar o uso de variáveis e operações matemáticas básicas com entrada do usuário.

Exemplo 2: Converter horas em segundos

```
python

horas = float(input("Digite o número de horas: "))
segundos = horas * 3600
print("Total de segundos:", segundos)
```

- **Objetivo:** Reforçar o uso de operadores matemáticos e entradas do usuário.

Exemplo 3: Calcular o quadrado de dois números

```
python
Copiar código
num1 = float(input("Digite o primeiro número: "))
num2 = float(input("Digite o segundo número: "))
```

```
quadrado1 = num1 ** 2
quadrado2 = num2 ** 2

print(f"O quadrado de {num1} é: {quadrado1}")
print(f"O quadrado de {num2} é: {quadrado2}")
```

-

Conclusão e Exercícios Práticos

- **Resumo dos conceitos abordados:**
 1. Variáveis, operações e estruturas de dados.
 2. Como manipular listas e tuplas.
 3. Criação de algoritmos simples.
- **Exercícios para fixação:**
 1. Crie um programa que calcule o IMC de uma pessoa, dado o peso e a altura.
 2. Escreva um código que peça ao usuário o nome de três frutas e as exiba em uma lista.
 3. Faça um programa que peça ao usuário dois números e exiba o maior deles.

Contexto do problema:

Suponha que temos uma lista de produtos com seus respectivos preços e quantidades, e queremos calcular o total das compras e aplicar um desconto de 10% se o total for superior a R\$ 200,00.

Exemplo de código:

```
# Lista de compras: Cada produto é representado por uma tupla (nome,
preço unitário, quantidade)
```

```
compras = [

    ("Arroz", 20.50, 2),

    ("Feijão", 7.80, 3),

    ("Macarrão", 4.50, 4),
```

```
( "Óleo", 9.30, 2),  
  ("Leite", 4.20, 12)  
]  
  
# Função para calcular o total das compras  
def calcular_total(compras):  
    total = 0  
  
    for item in compras:  
        nome, preco, quantidade = item # Desempacotar a tupla  
  
        total += preco * quantidade      # Multiplicar preço pela  
        quantidade e somar ao total  
  
    return total  
  
# Função para aplicar o desconto  
def aplicar_desconto(total, limite=200, desconto=0.10):  
    if total > limite:  
        return total * (1 - desconto) # Aplica 10% de desconto  
  
    return total  
  
# Calcular o total das compras  
total_compras = calcular_total(compras)  
  
# Verificar e aplicar desconto  
total_final = aplicar_desconto(total_compras)
```

```
# Exibir os resultados

print(f"Total sem desconto: R$ {total_compras:.2f}")

print(f"Total com desconto (se aplicável): R$ {total_final:.2f}")
```

Explicação:

1. **Lista de compras:**
 - Cada produto é representado por uma **tupla**, contendo o nome do produto, o preço unitário e a quantidade.
 - Exemplo: `("Arroz", 20.50, 2)` significa que o **Arroz** custa **R\$ 20.50** e foram compradas **2 unidades**.
2. **Função `calcular_total`:**
 - Recebe a lista de compras e percorre cada tupla. Para cada item, ela multiplica o preço pela quantidade e soma ao total.
 - Exemplo de uso de **desempacotamento de tupla** para pegar os valores diretamente dentro do laço `for`.
3. **Operador matemático:**
 - A multiplicação é usada para calcular o custo total de cada item (`preço * quantidade`).
4. **Função `aplicar_desconto`:**
 - Aplica um desconto de **10%** se o valor total das compras for superior a **R\$ 200,00**.
 - Utiliza uma simples operação de **multiplicação e subtração** para calcular o valor com desconto.
5. **Saída:**
 - O algoritmo imprime o total sem desconto e, se aplicável, o total com desconto.

TURTLE PY - Exemplo 1: Desenhando um quadrado (Conceito de laços)

Este exemplo mostra como usar um **laço `for`** para repetir instruções e criar um quadrado:

python

[Copiar código](#)

```
import turtle

# Criar uma janela de desenho

window = turtle.Screen()

# Criar um objeto tartaruga

t = turtle.Turtle()

# Desenhar um quadrado

for _ in range(4):

    t.forward(100)  # Move 100 unidades à frente

    t.right(90)      # Gira 90 graus para a direita


# Fechar a janela ao clicar

window.exitonclick()
```

Conceitos ensinados:

- **Laço `for`** para repetir blocos de código.
- **Movimento e direção** com `forward` e `right`.

Exemplo 2: Desenhando um triângulo equilátero (Conceito de ângulos)

Neste exemplo, os alunos aprendem sobre ângulos e como calcular rotações:

```
import turtle

# Criar a tartaruga

t = turtle.Turtle()

# Desenhar um triângulo equilátero
```



```
for _ in range(3):  
    t.forward(100)  # Move 100 unidades  
    t.left(120)     # Gira 120 graus à esquerda  
turtle.done()
```

Conceitos ensinados:

- **Cálculo de ângulos:** o ângulo externo de um triângulo equilátero é 120°.
 - Reforça o uso de **laços**.
-

Exemplo 3: Desenhando uma estrela (Funções e loops)

Neste exemplo, os alunos aprendem a dividir o código em **funções** e usar laços para desenhos mais complexos:

```
import turtle  
  
# Função para desenhar uma estrela  
def draw_star(size):  
    for _ in range(5):  
        t.forward(size)  
        t.right(144)  # Ângulo para criar uma estrela  
  
# Configurar a tartaruga  
t = turtle.Turtle()  
  
# Desenhar uma estrela de tamanho 100  
draw_star(100)  
  
turtle.done()
```

Conceitos ensinados:

- Uso de **funções** para modularizar o código.
 - Laços e geometria simples para desenhar figuras.
-

Exemplo 4: Espiral colorida (Laços aninhados e variáveis)

Aqui, os alunos aprendem a usar **variáveis** para modificar dinamicamente o comportamento do código:

```
import turtle

# Configurar a tartaruga

t = turtle.Turtle()

t.speed(10) # Aumentar a velocidade de desenho

colors = ['red', 'blue', 'green', 'yellow', 'purple', 'orange']

# Desenhar uma espiral colorida

for i in range(50):

    t.pencolor(colors[i % len(colors)]) # Alternar as cores

    t.forward(i * 5)

    t.right(144) # Ângulo de 144 graus

turtle.done()
```

Conceitos ensinados:

- **Laços aninhados** e uso de **módulo (%)** para alternar cores.
- Introdução ao conceito de **parâmetros variáveis** (mudança de **i** a cada iteração).

