Plano de Aula 3: Lógica de Programação em Python

Objetivos da Aula:

- Aviso sobre as atividades (exibir exemplo de como realizar o upload)
- Aviso sobre o repositório de exemplos
- Relembrar rapidamente sobre a aula anterior
- Explorar operadores relacionais e lógicos.
- Compreender e aplicar as estruturas de decisão `if`, `else`, e `elif` em Python.
- Aplicar estruturas de repetição com `for` e `while`.
- Desenvolver habilidades para resolver problemas práticos utilizando lógica de programação.

Tópico 1: Estruturas de Decisão

As estruturas de decisão permitem que o programa tome diferentes caminhos com base em condições específicas.

Exemplo 1: Avaliação de notas

```
nota = float(input("Digite sua nota: "))

if nota >= 8.5:
    print("Você tirou um A.")
elif nota >= 7.0:
    print("Você tirou um B.")
elif nota >= 5.5:
    print("Você tirou um C.")
else:
    print("Você está reprovado.")
```

Aqui, usamos `elif` para criar diferentes faixas de notas. A ideia é demonstrar como podemos ter mais de duas possibilidades de saída.

Exemplo 2: Classificação de temperatura

```
temperatura = float(input("Digite a temperatura em graus Celsius:
"))

if temperatura > 35:
    print("Está muito quente!")
elif temperatura > 25:
    print("Clima agradável.")
elif temperatura > 15:
    print("Clima ameno.")
else:
```

```
    print("Está frio!")
```

Este exemplo traz uma situação real onde a condição da temperatura determina a mensagem exibida.

Tópico 2: Operadores Relacionais e Lógicos

Os operadores relacionais comparam valores, enquanto os lógicos combinam expressões booleanas.

Exemplo 1: Comparação de números

```
a = 7
b = 10
print(a > b)  # False
print(a <= b)  # True
```

Esse exemplo simples reforça o uso de operadores relacionais como `<`, `>`, `<=` e `>=`.

Exemplo 2: Verificação de múltiplas condições

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

if idade >= 18 and idade <= 60:
    print("Você está em idade ativa para trabalhar.")
elif idade > 60 or idade < 18:
    print("Você está fora da faixa de idade ativa.")-
```

Aqui, o uso de `and` e `or` é exemplificado para verificar múltiplas condições simultâneas.

Tópico 3: Estruturas de Repetição

Exemplo 1: Uso do `for` para somar números pares

```
soma = 0
for i in range(1, 101):
    if i % 2 == 0:
        soma += i
print("A soma dos números pares de 1 a 100 é:", soma)
```

Este código usa um `for` para percorrer os números de 1 a 100 e soma apenas os números pares.

Exemplo 2: Validação de senha com `while`

```
senha_correta = "abc123"
tentativa = ""

while tentativa != senha_correta:
    tentativa = input("Digite a senha: ")
    if tentativa != senha_correta:
        print("Senha incorreta. Tente novamente.")
print("Senha correta! Acesso concedido.")
```

O `while` é utilizado aqui para continuar solicitando a senha até que a correta seja digitada.

Exemplo 3: Contagem regressiva

```
contador = 10

while contador > 0:
    print(contador)
    contador -= 1
print("Feliz Ano Novo!")
```

Aqui, mostramos uma contagem regressiva usando um laço `while`, com a mensagem final quando o laço termina.

Atividade Prática:

1. Crie um programa que:
    - Solicite ao usuário a idade.
    - Use estruturas de decisão para classificar a idade em categorias como "Criança", "Adolescente", "Adulto" e "Idoso".
2. Modifique o exemplo da soma de números pares para que o usuário possa escolher o intervalo de números a serem somados.
3. Escreva um código que simule um sistema de login. O usuário deve fornecer o nome de usuário e a senha corretos para acessar o sistema. Use estruturas de repetição e condições para validar as credenciais.

Conclusão da Aula

Nesta aula, os alunos aprenderam a usar estruturas de decisão e repetição para resolver problemas lógicos simples e interagir com o usuário. No final da aula, eles estarão prontos para aplicar esses conceitos em cenários do mundo real, como validação de entrada de dados e execução de operações repetitivas.

Dicas Finais:

- Pratique os exemplos para entender melhor o funcionamento das condições e dos loops.
- Teste variações dos códigos para diferentes entradas e veja como o comportamento do programa muda.