

# Aula 2: Estruturas Condicionais e Loops

## Introdução

Na aula anterior, aprendemos sobre variáveis, tipos de dados e como imprimir mensagens na tela com a função `print()`. Agora, vamos aprender a tomar decisões nos programas e a repetir tarefas automaticamente usando estruturas condicionais e loops.

---

## Estruturas Condicionais

As estruturas condicionais permitem que um programa tome decisões com base em determinadas condições. Em Python, usamos a instrução `if` para criar essas condições.

### Exemplo:

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

if idade >= 18:
    print("Você é maior de idade.")
else:
    print("Você é menor de idade.")
```

Nesse exemplo, se a idade for 18 ou maior, a mensagem "Você é maior de idade." será exibida; caso contrário, aparecerá "Você é menor de idade.".

### Estrutura `if-elif-else`

Podemos ter múltiplas condições com `elif` (else if).

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

if nota >= 7:
    print("Aprovado!")
elif nota >= 5:
    print("Recuperação!")
else:
    print("Reprovado!")
```

### Exercício 1:

Escreva um programa que solicite ao usuário um número e informe se ele é positivo, negativo ou zero.

---

## Operadores Lógicos

Os operadores lógicos ajudam a combinar condições:

- **and** (E): ambas as condições devem ser verdadeiras.
- **or** (OU): pelo menos uma condição deve ser verdadeira.
- **not** (NÃO): inverte a condição.

### Exemplo:

```
usuario = input("Digite seu nome de usuário: ")
senha = input("Digite sua senha: ")
```

```
if usuario == "admin" and senha == "1234":
    print("Acesso permitido!")
else:
    print("Usuário ou senha incorretos!")
```

### Exercício 2:

Crie um programa que solicite ao usuário dois números e informe se ambos são pares.

---

## Laços de Repetição (Loops)

Os loops são usados para repetir comandos várias vezes. Em Python, temos dois principais tipos de loops: **while** e **for**.

### Laço **while**

O **while** executa um bloco de código enquanto uma condição for verdadeira.

```
contador = 1
```

```
while contador <= 5:
    print("Contagem:", contador)
    contador += 1
```

### Exercício 3:

Crie um programa que solicite um número ao usuário e conte de 1 até esse número.

---

### Laço **for**

O **for** é usado para percorrer sequências, como listas ou intervalos numéricos.

```
for i in range(1, 6):  
    print("Valor de i:", i)
```

O comando **range(1, 6)** gera os números de 1 a 5.

### Exercício 4:

Crie um programa que percorra os números de 1 a 10 e exiba apenas os pares.

---

## Desafio Final

Agora que aprendemos sobre condições e loops, faça um programa que peça ao usuário um número e exiba a tabuada desse número (de 1 a 10).

Exemplo:

Entrada:

Digite um número: 5

Saída:

```
5 x 1 = 5  
5 x 2 = 10  
...  
5 x 10 = 50
```

---

## Conclusão

Nesta aula, aprendemos:

- Como usar estruturas condicionais (`if`, `elif`, `else`).
- Operadores lógicos (`and`, `or`, `not`).
- Laços de repetição (`while` e `for`).

Na próxima aula, aprenderemos sobre **funções**, que nos permitem organizar melhor o código e reutilizar comandos de forma eficiente!