

Documentação do Projeto

Lógica, algoritmos e programação de computadores

estruturas_condicao1.py

1 - Se o valor da variável `temperatura` for `menor que 30`, o código exibirá "A temperatura hoje está amena".

2 - Se a temperatura for `30 ou maior`, exibirá "Hoje está fazendo calor".

3 - Agora, se você alterar o valor da `variável para 31`, o script deve imprimir "Hoje está fazendo calor". Aqui está o comportamento esperado:

3.1 - Com `temperatura = 29`: "A temperatura hoje está amena"

3.2 - Com `temperatura = 31`: "Hoje está fazendo calor"

3.3 - Certifique-se de que o código esteja implementado corretam

estruturas_condicao2.py

1 - Condicional no `elif`: No caso, queremos verificar se o valor está entre 2 e 5, o que é feito com `elif 2 <= tempoExperiencia < 5`.

2 - Sinal de comparação: Para a primeira condição, era necessário verificar se a experiência é menor que 2, então usei `tempoExperiencia < 2` na condição do `if`.

3 - Agora o código se comporta conforme o esperado:

3.1 - Se `tempoExperiencia` for menor que 2, imprime "Nível de conhecimento júnior."

3.2 - Se estiver entre 2 e 5, imprime "Nível de conhecimento pleno."

3.3 - Caso contrário, imprime "Nível de conhecimento sênior."

estruturas_repeticao1.py

1 - Variável `entrada_idade`: Inicialmente atribuída com uma string vazia "".

2 - Estrutura `while`: O laço continua enquanto o valor da variável `entrada_idade` for diferente de `'0'`. O loop é controlado pela conversão da entrada para string (`str(entrada_idade)`).

3 - Entrada de dados com `input`: Solicita que o usuário insira um número, e atribui esse valor à variável `entrada_idade`.

4 - Condição para exibição: O número digitado é exibido apenas se não for igual a `'0'`, para evitar a impressão na saída.

Interrupção: Se o usuário digitar `'0'`, o loop é encerrado e o programa exibe a mensagem "Programa encerrado."

estruturas_repeticao2.py

1 - Iteração sobre `texto`:

1.1 - A variável `texto` contém a string `'Olá, laço for.'` .

1.2 - O laço `for item in texto`: itera sobre cada caractere da string, atribuindo cada um deles à variável `item`.

1.3 - A cada iteração, imprime o caractere precedido de `'Caractere: '`.

2.0 - Iteração sobre o intervalo de 1 a 10:

2.1 - O laço `for numero in range(1, 11)`: itera sobre o intervalo de 1 a 10. O `range(1, 11)` gera números de 1 até 10.

2.2 - A cada iteração, imprime o número precedido de `'Número do intervalo: '`. Como estamos concatenando uma string com um número, usamos `str(numero)` para converter o número em string.

funcoes1.py

1 - A função `imprimir_variavel` foi definida sem receber parâmetros.

2 - Dentro da função, a variável `texto` foi criada e atribuída a string `'Olá, funções em Python'`.

3 - A função `print()` exibe o valor de `texto quando` a função é chamada.

4 - Fora da função, fazemos a chamada de `imprimir_variavel()` para executar o que foi definido no escopo da função.

Funcoes2.py

1 - A função `loginUsuario` recebe um parâmetro chamado `perfil`.

2 - Dentro da função, utilizamos `perfil.lower()` para converter o valor passado para letras minúsculas, independentemente de como o usuário digitou (letras maiúsculas ou minúsculas).

3 - Se o valor de `perfil.lower()` for igual a `'admin'`, a função imprime `'Bem-vindo, Administrador'`, caso contrário, imprime `'Bem-vindo, Usuário'`.

4 - Fora da função, a chamamos com diferentes valores para verificar o comportamento.

calculadora_v2.py

1 - Funções de Operações: As funções `adicao`, `subtracao`, `multiplicacao` e `divisao` são responsáveis pelos cálculos matemáticos.

2 - Função `calculadora`: Ela recebe os dois números e a operação, verificando qual função de cálculo deve ser chamada.

3 - Estrutura de repetição `while`: O loop continua até que o usuário responda com "n" ou "N" para encerrar o programa.