

Atividades do Módulo 4 - QA NA PRÁTICA



Esse [repositório](#) é dedicado às atividades realizadas durante o Módulo 4 - M4 - LÓGICA DE PROGRAMAÇÃO COM PYTHON do curso de Quality Assurance oferecido pelo [Instituto JogaJunto](#).

📝 Descrição da 16ª Atividade: 🌟

⌚ INDIVIDUAL PARTE 1 USANDO IF: Construa um script para verificar se o usuário tem uma idade maior que 18 anos, se tiver, imprima na tela "Indivíduo possui idade mínima para dirigir"

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

if idade > 18:
    print("Indivíduo possui idade mínima para dirigir")
```

O arquivo dessa atividade está nesse repositório dentro da pasta Atividades: Atividades\Atividade16.py.

Integrantes da Squad:

| Beatriz Souza | Bruno Soares | Leanderson Lima | Rebeca Borges | Sara Cruz |

Atividades do Módulo 4 - QA NA PRÁTICA



Esse [repositório](#) é dedicado às atividades realizadas durante o Módulo 4 - M4 - LÓGICA DE PROGRAMAÇÃO COM PYTHON do curso de Quality Assurance oferecido pelo [Instituto JogaJunto](#).

📝 Descrição da 17ª Atividade: 🌟

⌚ INDIVIDUAL PARTE 2 USANDO ELSE: Complemente o script feito, imprimindo na tela "Indivíduo NÃO possui idade mínima para dirigir"

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

if idade > 18:
    print("Indivíduo possui idade mínima para dirigir")
```

O arquivo dessa atividade está nesse repositório dentro da pasta Atividades: Atividades\Atividade17.py.

Integrantes da Squad:

| Beatriz Souza | Bruno Soares | Leanderson Lima | Rebeca Borges | Sara Cruz |

Atividades do Módulo 4 - QA NA PRÁTICA



Esse [repositório](#) é dedicado às atividades realizadas durante o Módulo 4 - M4 - LÓGICA DE PROGRAMAÇÃO COM PYTHON do curso de Quality Assurance oferecido pelo [Instituto Jogajunto](#).

📝 Descrição da 18ª Atividade: 🌟

⌚ INDIVIDUAL USANDO ELIF: Complemente o script feito, imprimindo na tela "Indivíduo tem entre 17 e 18 anos e ainda NÃO está apto para dirigir"

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

if idade > 18:
    print("Indivíduo possui idade mínima para dirigir")
elif idade >= 17:
    print("Indivíduo tem entre 17 e 18 anos e ainda NÃO está apto para dirigir")
else:
    print("Indivíduo NÃO possui idade mínima para dirigir")
```

O arquivo dessa atividade está nesse repositório dentro da pasta Atividades: Atividades\Atividade18.py.

Integrantes da Squad:

| Beatriz Souza | Bruno Soares | Leanderson Lima | Rebeca Borges | Sara Cruz |

Atividades do Módulo 4 - QA NA PRÁTICA



Esse [repositório](#) é dedicado às atividades realizadas durante o Módulo 4 - M4 - LÓGICA DE PROGRAMAÇÃO COM PYTHON do curso de Quality Assurance oferecido pelo [Instituto Jogajunto](#).

💡 Descrição da 19ª Atividade: 🌟

💡 EM SQUADS Leiam o texto abaixo e resolvam. Na "FashionStyle", para um cliente obter 10% de desconto em suas compras, a compra deve ser de pelo menos R\$250,00 e para obter 30%, a compra deve ser acima de R\$500,00. Caso contrário, nenhum desconto é aplicado. No caixa, haverá uma tela voltada para o cliente. Ao passar o produto, caso cumpra o requisito da promoção, aparecerá a mensagem:

- Caso o cliente não cumpra o requisito, deve aparecer "POXA, FALTA POUCO PARA VOCÊ GANHAR 10% DE DESCONTO EM SUA COMPRA."
- Caso o cliente faça uma compra acima de R\$250,00 "PARABÉNS. VOCÊ GANHOU 10% DE DESCONTO, MAS PODE GANHAR 30% SE SUA COMPRA FOR ACIMA DE R\$500,00"
- Caso o cliente faça uma compra acima de R\$500,00 "PARABÉNS. VOCÊ GANHOU SUPER DESCONTO DE 30%"

```
valor_compra = float(input("Digite o valor da compra: R$"))

mensagem_desconto = ""

if valor_compra >= 500.0:
    mensagem_desconto = "Você ganhou 30% de desconto! PARABÉNS!"
    valor_compra = valor_compra * 0.7
elif valor_compra >= 250.0:
    mensagem_desconto = "Você ganhou 10% de desconto! PARABÉNS!"
    valor_compra = valor_compra * 0.9
else:
    mensagem_desconto = "Quase lá! Com mais R${:.2f}, você ganha 10% de desconto."
    % (250.0 - valor_compra)

print(mensagem_desconto)

print("Total a pagar: R${:.2f}" % valor_compra)
```

O arquivo dessa atividade está nesse repositório dentro da pasta Atividades: Atividades\Atividade19.py.

Integrantes da Squad:

| Beatriz Souza | Bruno Soares | Leanderson Lima | Rebeca Borges | Sara Cruz |

Atividades do Módulo 4 - QA NA PRÁTICA



Esse [repositório](#) é dedicado às atividades realizadas durante o Módulo 4 - M4 - LÓGICA DE PROGRAMAÇÃO COM PYTHON do curso de Quality Assurance oferecido pelo [Instituto Jogajunto](#).

💡 Descrição da 20ª Atividade: 🌟

💡 EM SQUADS Leiam o texto abaixo e resolvam. Na "JUNTOFIT", se um aluno tiver frequência de 21 vezes, sem interrupções, ele ganha um mês de aulas gratuitas para presentear um acompanhante. Caso contrário, ele não se qualifica para o benefício. Na catraca de acesso, haverá uma tela voltada para o cliente. Todos os dias, quando ele passar, deve aparecer a mensagem:

"VOCÊ ESTÁ PARTICIPANDO DA NOSSA PROMO TREINA JUNTO"

Quando ele completar 21 identificações seguidas, deve aparecer a mensagem:

"UHUU. AGORA VOCÊ PODE PRESENTEAR UM AMIGO OU AMIGA PARA TREINAR COM VOCÊ".

Caso o cliente tenha uma certa frequência, mas falte algum dia, quando retornar, deve aparecer:

"QUE BOM VER VOCÊ DE VOLTA. A PARTIR DE AGORA INICIAMOS MAIS UMA CONTAGEM DE 21 DIAS PARA A PROMO TREINA JUNTO."

```
frequencia = 0
dias_seguidos = 0

while True:

    input("Pressione Enter para registrar sua presença hoje: ")

    frequencia += 1

    if frequencia == 21:
        print("UHUU. AGORA VOCÊ PODE PRESENTEAR UM AMIGO OU AMIGA PARA TREINAR COM VOCÊ.")
        dias_seguidos = 0
    else:
        print("VOCÊ ESTÁ PARTICIPANDO DA NOSSA PROMO TREINA JUNTO")

    escolha = input("Deseja continuar treinando? (S para sim, qualquer outra tecla")
```

```
para sair): "").strip().lower()

if escolha != 's':
    break

if frequencia < 21:
    dias_seguidos += 1
    if dias_seguidos == 1:
        print("QUE BOM VER VOCÊ DE VOLTA. A PARTIR DE AGORA INICIAMOS MAIS UMA
CONTAGEM DE 21 DIAS PARA A PROMO TREINA JUNTO.")
    else:
        print(f"QUE BOM VER VOCÊ DE VOLTA. CONTINUE ASSIM! Mais {21 -
dias_seguidos} dias para a promoção.")
    dias_seguidos = 0
```

O arquivo dessa atividade está nesse repositório dentro da pasta Atividades: Atividades\Atividade20.py.

Integrantes da Squad:

| Beatriz Souza | Bruno Soares | Leanderson Lima | Rebeca Borges | Sara Cruz |

Atividades do Módulo 4 - QA NA PRÁTICA



Esse [repositório](#) é dedicado às atividades realizadas durante o Módulo 4 - M4 - LÓGICA DE PROGRAMAÇÃO COM PYTHON do curso de Quality Assurance oferecido pelo [Instituto Jogajunto](#).

💡 Descrição da 21ª Atividade: 🌟

💡 Faça uma pesquisa sobre ESTRUTURAS DE REPETIÇÃO E ITERAÇÃO, identificando: O que são estruturas de repetição e iteração? | Quando são usadas? | Quais os principais tipos de estruturas de repetição?

Estruturas de Repetição e Iteração

As estruturas de repetição e iteração são fundamentais na programação e são usadas para executar um conjunto de instruções repetidamente, com base em uma condição específica. Essas estruturas são utilizadas quando se deseja automatizar tarefas que precisam ser realizadas várias vezes ou quando se precisa percorrer elementos em uma coleção de dados, como uma lista ou um conjunto.

As estruturas de repetição podem ser usadas em uma variedade de situações, incluindo:

- **Processamento de Dados:** Para processar cada elemento de uma lista, arquivo ou conjunto de dados.
- **Iteração de Loops:** Para criar loops que executam um conjunto de instruções até que uma condição seja atendida.
- **Validação de Entradas:** Para garantir que o usuário insira dados corretos ou para verificar entradas em um formulário, repetindo até que sejam válidas.
- **Implementação de Algoritmos:** Para implementar algoritmos que envolvem repetição, como ordenação, busca e cálculos iterativos.

Principais Tipos de Estruturas de Repetição

For Loop

Utilizado quando você sabe antecipadamente quantas vezes deseja repetir um bloco de código. Em Python:

```
for i in range(5):
    print(i)
```

While Loop

Usado quando você deseja repetir um bloco de código enquanto uma condição for verdadeira. Em Python:

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

Do-While Loop (Não disponível em Python)

Este tipo de loop executa um bloco de código pelo menos uma vez e, em seguida, verifica a condição para continuar a execução.

Em Python, não existe uma estrutura de loop do-while incorporada como em algumas outras linguagens de programação, como C, C++, C#, etc. No entanto, você pode simular um loop do-while usando um loop while tradicional com uma condição que sempre seja verdadeira na primeira iteração e, em seguida, usar uma instrução break para sair do loop quando a condição desejada não for mais atendida.

Aqui está um exemplo de como simular um loop do-while em Python:

```
while True:
    print("Este é o bloco do loop do-while simulado.")

    continuar = input("Deseja continuar? (S para sim, qualquer outra tecla para sair): ").strip().lower()

    if continuar != 's':
        break
```

Em C#, por exemplo, que possui essa estrutura de repetição incorporada ficaria dessa forma:

```
using System;

class Program
{
    static void Main()
    {
        int contador = 0;

        do
        {
            Console.WriteLine($"Este é um loop do-while. Contador: {contador}");
            contador++;
        }
        while (contador < 5);
    }
}
```

```
}
```

Loop Aninhado

É possível usar loops dentro de outros loops para realizar tarefas complexas ou percorrer matrizes multidimensionais. Em Python:

```
for i in range(3):
    for j in range(2):
        print(i, j)
```

Loop Infinito

Um loop que executa indefinidamente até que seja explicitamente interrompido. Cuidado ao usar loops infinitos, pois eles podem causar travamentos. Em Python:

```
while True:
    print("Isso é um loop infinito")
```

As estruturas de repetição e iteração são elementos fundamentais na programação e são essenciais para controlar o fluxo de um programa, permitindo a automação de tarefas repetitivas e a manipulação de dados em coleções.

Integrantes da Squad:

| Beatriz Souza | Bruno Soares | Leanderson Lima | Rebeca Borges | Sara Cruz |