

# Movimento FILO em Python

**Mário Leite**

Junto de outros movimentos, o movimento **FILO** (**F**irst **I**n **L**ast **O**ut - "Primeiro a Entrar, Último a Sair") é o comportamento clássico de uma **pilha** (*stack*); neste caso o primeiro elemento inserido é o último a ser retirado. Em Python, isto pode ser implementado usando listas, que possuem métodos como **append()** para empilhar e **pop()** para desempilhar. É um conceito importante no contexto das pilhas, uma estrutura de dados fundamental em ciência da computação.

Isso descreve o comportamento da pilha, onde o primeiro elemento que é colocado nela é o último a ser retirado. Em outras palavras, em uma pilha, o último item a ser adicionado é sempre o primeiro a ser removido. Esse é um padrão de acesso bastante comum, utilizado por exemplo, em navegadores de internet (quando você navega pelas páginas, a última página visitada fica no topo da pilha).

## Como funciona:

1. **Push (inserção)**: Adiciona um elemento no topo da pilha. Esse item será o próximo a ser retirado se a pilha for "desempilhada".
2. **Pop (remoção)**: Remove o item que está no topo da pilha, ou seja, o último elemento adicionado.
3. **Peek/Top**: Permite visualizar o item no topo da pilha, sem removê-lo.

Um exemplo prático pode ser mostrado com uma de pratos, onde você coloca o prato no topo da pilha. Quando precisa de um prato, o prato que está no topo (o último colocado) é retirado.

- **Estado inicial:** (vazia)

**Pilha**

- Coloca 3 pratos na pilha, na ordem: **1, 2, 3**. A pilha fica assim:

**Pilha**

3

2

1

- Pega um prato (**pop**); o **3**, pois foi o último a ser colocado.

**Pilha**

2

1

Ou seja, o **último** elemento colocado foi o **primeiro** a sair.

A figura 1 mostra o resultado da execução do programa “**MovimentoFILO**”.

---

The screenshot shows the PyCharm interface with the 'Run' tab selected. The title bar says 'MovimentoFILO'. The run output window displays the following text:

```
C:\Users\Usuario\PycharmProjects\pythonProject8\venv\Scripts\python.exe "D:\Livros\Livro11\Códigos\Nível 3\\MovimentoFILO.py"
'Item 1' empilhado.
'Item 2' empilhado.
'Item 3' empilhado.
Pilha atual: ['Item 1', 'Item 2', 'Item 3']
'Item 3' desempilhado.
'Item 2' desempilhado.
Pilha atual: ['Item 1']
'Item 1' desempilhado.
A pilha está vazia. Nada para desempilhar.

Process finished with exit code 0
```

**Figura 1 - Saída do programa “MovimentoFILO”**

---

**Nota1:** Postagem baseada no livro: “**1001 Programas em Python Para Você Aprender Praticando - Volume 3: Nível Avançado**”. Publicado pelo autor na “Amazon” e no “Clube de Autores”  
<https://www.amazon.com.br/Curso-B%C3%A1sico-Programa%C3%A7%C3%A3o-Teoria-Pr%C3%A1tica/dp/8539908700>

**Nota2:** Acesse o *link* abaixo para ver meus mais recentes livros de Python publicado pelo “Clube de Autores”, no formato impresso, da coleção “**1001 Programas em Python Para Você Aprender Praticando**”:  
Volume1: Nível Básico (500 programas)  
Volume2: Nível Intermediário (300 programas)  
Volume3: Nível Avançado (201 programas)  
<https://clubedeautores.com.br/livros/autores/mario-leite>

Para adquirir PDF dos livros: [marleite@gamil.com](mailto:marleite@gamil.com)

---

```
'''  
MovimentoFILO.py
```

```
Cria uma classe para mostrar o movimento do tipo FILO (First In Last Out), onde o  
primeiro elemento inserido é o último a ser retirado.
```

```
'''  
  
class ClsPilha:  
    def __init__(self):  
        self.LstItens = [] #cria lista para armazenar os elementos da pilha  
  
    def Empilhar(self, item):  
        #Adiciona um item ao topo da pilha.  
        self.LstItens.append(item)  
        print(f'{item} empilhado.')  
  
    def Desempilhar(self):  
        #Remove e retorna o item do topo da pilha.  
        if(not self.VerificarVazia()):  
            item = self.LstItens.pop()  
            print(f'{item} desempilhado.')  
            return item  
        else:  
            print("A pilha está vazia. Nada para desempilhar!")  
            return None  
  
    def Checar(self):  
        "Retorna o item do topo sem removê-lo.  
        if(not self.VerificarVazia()):  
            return self.LstItens [-1]  
        else:  
            return None  
  
    def VerificarVazia(self):  
        #Verifica se a pilha está vazia.  
        return len(self.LstItens) == 0  
  
    def ExibirPilha(self):  
        """Exibe os itens da pilha."""  
        print(f'Pilha atual: {self.LstItens}')  
  
#-----  
  
def main():  
    Pilha = ClsPilha()  
    #Empilha os itens  
    Pilha.Empilhar("Item 1")  
    Pilha.Empilhar("Item 2")  
    Pilha.Empilhar("Item 3")  
  
    #Exibe a pilha  
    Pilha.ExibirPilha()  
  
    #Desempilha os itens em ordem FILO  
    Pilha.Desempilhar()  
    Pilha.Desempilhar()  
  
    #Exibe a pilha após desempilhar  
    Pilha.ExibirPilha()  
  
    #Tenta desempilhar de uma pilha vazia  
    Pilha.Desempilhar()  
    Pilha.Desempilhar()  
  
#=====  
#Programa principal  
if(__name__ == "__main__"):  
    main()  
#Fim do programa "MovimentoFILO" -----
```

