



FILAS EM PYTHON

Filas são estruturas de dados fundamentais em programação. Nesse ebook você vai aprender a gerenciar filas da maneira certa em Python!

Crie Ebooks técnicos incríveis em minutos com IA

Conheça a 1ª IA Especializada na criação de Ebooks **com código!**



Chega de formatar código no Google Docs



Deixe que nossa IA faça o trabalho pesado

 Syntax Highlight

 Adicione Banners Promocionais

 Edite em Markdown em Tempo Real

 Infográficos feitos por IA

TESTE AGORA 

Salve salve Pythonista!

Hoje vamos mergulhar em um tópico extremamente relevante e útil no universo Python: as **filas**.

Quem já trabalhou com processamento de tarefas em série sabe o quanto é importante entender o gerenciamento de filas.

Estaremos utilizando a biblioteca **queue**, uma grande aliada quando temos tarefas que precisam ser processadas de forma ordenada.

Obter um controle mais aprimorado nas operações de fila em Python permite aprimorar o desempenho do seu código, especialmente em aplicações que fazem muito uso de filas para tarefas como gerenciamento de *threads*, gerenciamento de operações em tempo real, entre outros.

Introdução a Filas

As filas são estruturas de dados essenciais que obedecem ao princípio de operação FIFO (First-In-First-Out), ou em português, “Primeiro a entrar, primeiro a sair”.

Basicamente, a ideia é que o primeiro elemento que é adicionado à fila será o primeiro a ser removido dela.

É como uma fila de supermercado, cada cliente que chega vai para o fim da fila e o cliente no início da fila é o próximo a ser atendido.

```
fila = ['Cliente 1', 'Cliente 2', 'Cliente 3']

fila.append('Cliente 4')
# A fila agora é ['cliente 1', 'cliente 2', 'cliente 3', 'cliente 4']

atendido = fila.pop(0)
# atendido agora é 'Cliente 1'
```

Filas em Python

Nativa no Python, basta importá-la com a declaração `import queue`. O módulo `queue` do Python oferece três tipos de classes: `Queue`, `LifoQueue` e `PriorityQueue`:

- **Queue**: É o primeiro tipo de fila. Ele segue a propriedade FIFO onde o primeiro elemento inserido é o primeiro a ser extraído.
- **LifoQueue**: Esta é a fila que segue a propriedade LIFO (Last-In-First-Out) onde o último elemento inserido é o primeiro a ser extraído.
- **PriorityQueue**: Não segue nenhuma das propriedades acima, FIFO ou LIFO. Esta fila segue o conceito de “ordem de prioridade”. Se os elementos da fila têm prioridades diferentes, então o elemento com alta prioridade é extraído primeiro.

Por exemplo, essa seria a sintaxe básica para cada tipo de fila:

```
import queue

fila_fifo = queue.Queue()
fila_lifo = queue.LifoQueue()
fila_prioridade = queue.PriorityQueue()
```

Gerenciamento de Filas

Com a biblioteca `queue`, pode-se adicionar elementos na fila utilizando o método `put` e remover utilizando o método `get`.

```
import queue

fila = queue.Queue()
fila.put("Primeiro da fila")
fila.put("Segundo da fila")

print(fila.get())
```

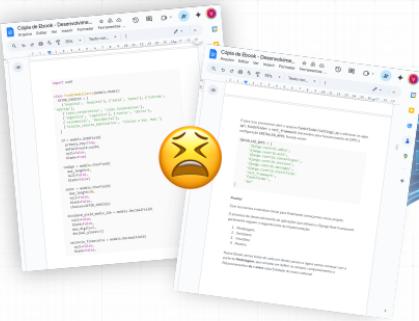
E a saída será:

```
Primeiro da fila
```

Falando em filas: O **DevBook** usa filas de processamento para gerar ebooks técnicos de forma escalável. É uma plataforma que criei para criar conteúdo profissional com IA — código formatado, infográficos e exportação em PDF. Conhece lá!

Crie Ebooks técnicos incríveis em minutos com IA

Conheça a 1ª IA Especializada na criação de Ebooks **com código!**



Chega de formatar código no Google Docs



Deixe que nossa IA faça o trabalho pesado

 Syntax Highlight

 Adicione Banners Promocionais

 Edite em Markdown em Tempo Real

 Infográficos feitos por IA

TESTE AGORA! PRIMEIRO CAPÍTULO 100% GRÁTIS 

FIFO, LIFO, Priority Queue

Como já discutimos, quando utilizamos filas, temos três abordagens principais: **FIFO, LIFO e Priority Queue**. Veja a diferença entre elas:

```
# Exemplo de FIFO
fila_FIFO = queue.Queue()
fila_FIFO.put("Primeiro da fila")
fila_FIFO.put("Segundo da fila")
print(fila_FIFO.get())
# Saída será: Primeiro da fila

# Exemplo de LIFO
fila_LIFO = queue.LifoQueue()
fila_LIFO.put("Primeiro da fila")
fila_LIFO.put("Segundo da fila")
print(fila_LIFO.get())
# Saída será: Segundo da fila

# Exemplo de PriorityQueue
fila_prioridade = queue.PriorityQueue()
fila_prioridade.put((2, "Primeiro da fila"))
fila_prioridade.put((1, "Segundo da fila"))
print(fila_prioridade.get())
# Saída será: (1, 'Segundo da fila')
```

Conclusão

Neste artigo, abordamos os conceitos de fila, como elas são implementadas em Python através do módulo `queue` e como gerenciá-las.

Também discutimos os princípios FIFO, LIFO e Priority Queue.

Espero que esses conhecimentos te ajudem a arrasar no que diz respeito ao gerenciamento de filas.

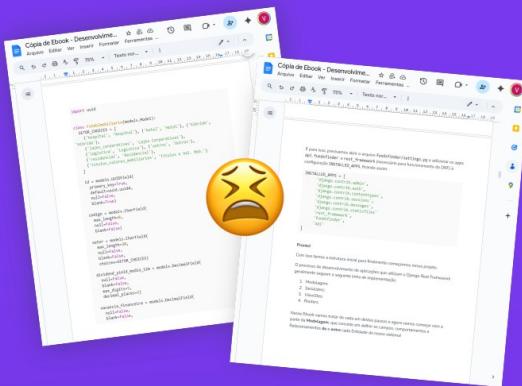
Não esqueça de compartilhar com seus amigos programadores.

E que a força do Python esteja com você! 



Crie Ebooks técnicos em minutos com IA

Conheça a 1ª IA Especializada na criação de Ebooks **com código!**



Chega de formatar código no Google Docs



Arquitetura de Software Moderna

A arquitetura de software alvo é profissional contendo o e-mail e produções de software para arquiteturas modernas. Oferece recursos como interface gráfica com interface de usuário.

```
import python
import python

class Arquitetura_de_Software_Moderna:
    ...
    def share(self):
        pass
    ...
    return "Arquitetura de NeXt", "arquitetura_moderna"
}

def __init__(self):
    if user_authenticated():
        self.user_authenticated = user_authenticated()
        self.user_email = self.user_authenticated['email']
        self.user_name = self.user_authenticated['name']
    ...
    # Envie AI para gerar um código
    return type
}
resource_available
```

AI-generated system

A arquitetura com propósito é a arquitetura moderna. Seus componentes incluem interface de usuário, banco de dados e outros sistemas externos. Chama-se de sistema gerado por IA.

Clean layout

O layout é limpo e organizado, facilitando a leitura e compreensão do código gerado.



</> Syntax Highlight

Infográficos feitos por IA

Adicione Banners Promocionais

Deixe que nossa IA faça o trabalho pesado

Edite em Markdown em Tempo Real

TESTE AGORA



PRIMEIRO CAPÍTULO 100% GRÁTIS