PYTHON ACADEMY

FILAS EM PYTHON

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

PYTHONACADEMY.COM.BR

Crie ebooks profissionais incríveis em minutos com IA

Chega de formatar texto no Google Docs, Word ou ferramentas que só te fazem perder tempo...

E deixe que nossa IA faça o trabalho pesado!

- Capas gerados por IA
- Infográficos feitos por IA
- Edite em Markdown em Tempo Real
- Adicione Banners Promocionais

TESTE AGORA

PRIMEIRO CAPÍTULO 100% GRÁTIS

> ✅ ***Atualizado para Python 3.13*** *(Julho 2024) Queue, PriorityQueue, LifoQueue, threading e casos práticos.*

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 principio 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 **Ebookr.ai** usa filas de processamento para gerar ebooks de forma escalável. É uma plataforma que criei para criar ebooks profissionais com IA sobre qualquer tema — com infográficos automáticos e exportação em PDF. Conhece lá!*

FIFO, LIFO, Priority Queue

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

```python
# 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')
```

Caso Prático: Producer-Consumer com Threading

```python
import queue
import threading
import time
import random

def produtor(fila, id_produtor):
    """Produz itens e coloca na fila"""
    for i in range(5):
        item = f"Item-{id_produtor}-{i}"
        fila.put(item)
        print(f"Produtor {id_produtor} produziu: {item}")
        time.sleep(random.uniform(0.1, 0.5))

def consumidor(fila, id_consumidor):
    """Consome itens da fila"""
    while True:
        try:
            item = fila.get(timeout=2)
            print(f"Consumidor {id_consumidor} processou: {item}")
            time.sleep(random.uniform(0.2, 0.6))
            fila.task_done()
        except queue.Empty:
            break

# Criar fila compartilhada
fila_trabalho = queue.Queue(maxsize=10)

# Criar threads de produtores
produtores = [
    threading.Thread(target=produtor, args=(fila_trabalho, i))
    for i in range(2)
]

# Criar threads de consumidores
consumidores = [
    threading.Thread(target=consumidor, args=(fila_trabalho, i))
    for i in range(3)
]

# Iniciar produtores
for p in produtores:
```

```
    p.start()

# Aguardar produtores terminarem
for p in produtores:
    p.join()

# Iniciar consumidores
for c in consumidores:
    c.start()

# Aguardar todos os itens serem processados
fila_trabalho.join()

print("Todos os itens foram processados!")
```

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.

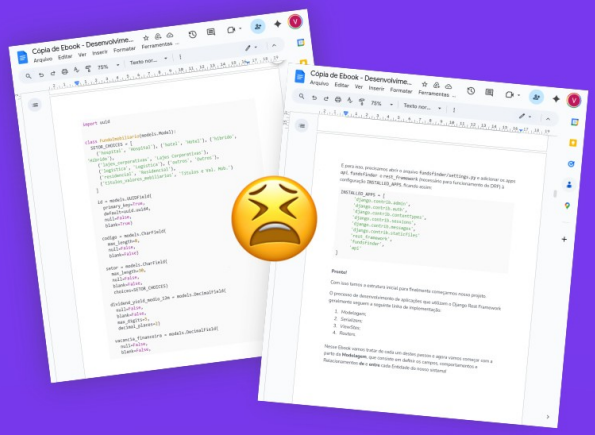
Não esqueça de compartilhar com seus amigos programadores.

E que a força do Python esteja com você! 👊

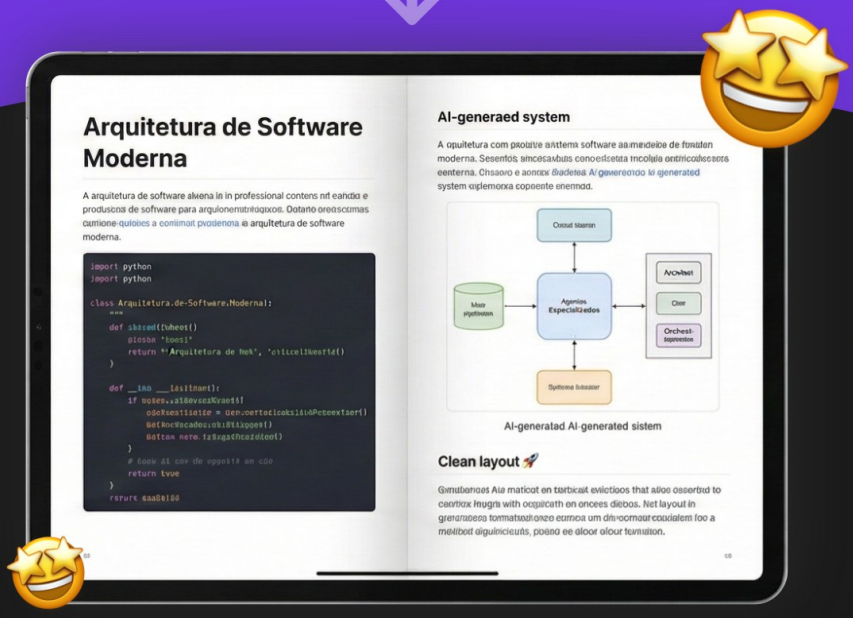
Não se esqueça de conferir!

Ebookr

Crie Ebooks profissionais em minutos com IA



Chega de formatar código no Google Docs ou Word



Capas gerados por IA

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