



# Geek University

**Evolua seu lado geek!**

[www.geekuniversity.com.br](http://www.geekuniversity.com.br)



# Concorrência em Python





# Concorrência em Python

Python não nasceu “concorrente”.

A linguagem Python, que já tem cerca de 30 anos, vem se modernizando com o passar do tempo e de acordo com as necessidades.



# Concorrência em Python

Python não nasceu “concorrente”.

A linguagem Python, que já tem cerca de 30 anos, vem se mordenizando com o passar do tempo e de acordo com as necessidades.

1.5+

threading





# Concorrência em Python

Python não nasceu “concorrente”.

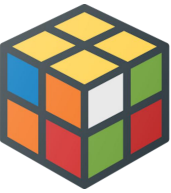
A linguagem Python, que já tem cerca de 30 anos, vem se modernizando com o passar do tempo e de acordo com as necessidades.

1.5+



threading

Com threading permitiu que pudéssemos usar a linguagem Python para criar threads nativas no sistema operacional podendo executar código de forma concorrente. Um detalhe aqui é que na implementação padrão da linguagem Python, conhecida como Cpython, as threads estão limitadas com um dispositivo chamado GIL - Global Interpreter Lock, fazendo com que o código seja executado em série e não em paralelo.



# Concorrência em Python

Python não nasceu “concorrente”.

A linguagem Python, que já tem cerca de 30 anos, vem se mordenizando com o passar do tempo e de acordo com as necessidades.





# Concorrência em Python

Python não nasceu “concorrente”.

A linguagem Python, que já tem cerca de 30 anos, vem se mordenizando com o passar do tempo e de acordo com as necessidades.



Com multiprocessing permitiu que pudéssemos usar a linguagem Python para criar subprocessos nativos no sistema operacional ao invés de threads, podendo executar código de forma concorrente e paralela evitando o problema com o GIL.



# Concorrência em Python

Python não nasceu “concorrente”.

A linguagem Python, que já tem cerca de 30 anos, vem se mordenizando com o passar do tempo e de acordo com as necessidades.



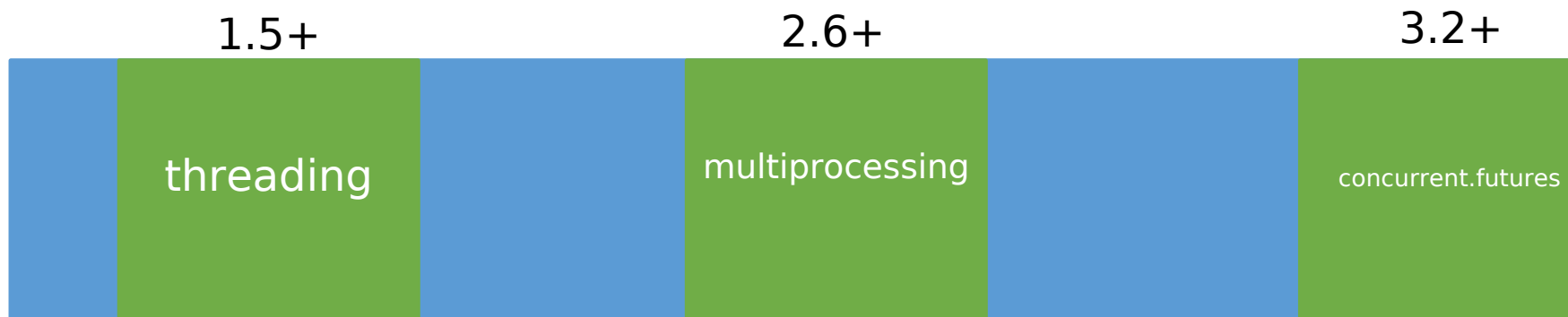




# Concorrência em Python

Python não nasceu “concorrente”.

A linguagem Python, que já tem cerca de 30 anos, vem se mordenizando com o passar do tempo e de acordo com as necessidades.



Com `concurrent.futures` passamos a ter uma implementação abstrada de concorrência permitindo de forma fácil fazer uso (ou alternar) de concorrência usando threads ou multiprocessos.

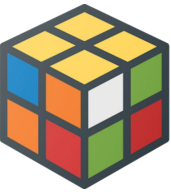


# Concorrência em Python

Python não nasceu “concorrente”.

A linguagem Python, que já tem cerca de 30 anos, vem se mordenizando com o passar do tempo e de acordo com as necessidades.





# Concorrência em Python

Python não nasceu “concorrente”.

A linguagem Python, que já tem cerca de 30 anos, vem se mordenizando com o passar do tempo e de acordo com as necessidades.



Com asyncio passamos finalmente a ter suporte á programação assíncrona.



# Concorrência em Python

Python não nasceu “concorrente”.

A linguagem Python, que já tem cerca de 30 anos, vem se mordenizando com o passar do tempo e de acordo com as necessidades.



Com asyncio passamos finalmente a ter suporte á programação assíncrona.

Na próxima aula iremos revisar sobre o Global Interpreter Lock - GIL...



# Geek University

**Evolua seu lado geek!**

[www.geekuniversity.com.br](http://www.geekuniversity.com.br)