

Thread

1. Criar um código que exemplifique duas funções, onde cada uma execute um `while` que itere de 0 a 9.
2. Criar um código que exemplifique uma função que receba um nome para a thread, um valor para sleep e um limite para o `while` que será executado dentro dessa função, imprimindo o tempo (`time()`) a cada iteração.
3. Criar um código que exemplifique a criação de duas threads utilizando o módulo `_thread`, que executem a função acima, com nome, valor de sleep e limite de `while` distintos.
4. Criar um código que exemplifique a criação de duas threads usando `threading.Thread()` que executem a função mencionada, com nome, valor de sleep e limite de `while` distintos. A execução das threads deve levar em consideração a thread principal, ou seja, utilizando a condicional `if __name__ == "__main__":`.
5. Criar uma classe que herde de `threading.Thread` e que sobrecarregue os métodos `__init__()` (para receber o nome da thread, o valor de tempo e o limite para o `while`) e `run()`. Nesse mesmo exercício, imprimir as propriedades das threads criadas e dados gerais sobre as threads correntes (por meio de `threading.<atributos>`). A execução das threads deve ser feita considerando a thread principal, ou seja, dentro de `if __name__ == "__main__":`.
6. Criar um código que exemplifique a criação de duas funções, onde cada uma contenha um loop `for` sobre `range(1000000)` para incrementar uma variável global. Em seguida, modificar essas funções para que sejam executadas de maneira assíncrona e síncrona. Ambas as execuções deverão utilizar `threading.Thread()`, sendo chamadas

a partir de uma função principal que obedeça à condicional `if`
`__name__ == "__main__":`.