



O seguinte código exemplifica como configurar o log para deixar a formatação das mensagens como quiser. O código, é apresentado o texto que é o exemplo de como ficará a saída no arquivo de log.

Desativar Numeração

```
1 import logging
2
3 #Criação e configuração do objeto logger
4 FORMATACAO_MSG = "%(asctime)s | %(levelname)s -> %(message)s"
5
6 logging.basicConfig(filename = 'logs.log',
7 level = logging.DEBUG,
8 format = FORMATACAO_MSG)
9 logger = logging.getLogger()
10
11 #Testando o logger
12 logger.debug('depuração')
13 logger.info('informação')
14 logger.warning('aviso')
15 logger.error('erro')
16 logger.critical('critico')
```

Agora confira o texto de como ficará a saída no arquivo de log:

```
2021-05-10 20:36:38,996 | DEBUG -> depuração
2021-05-10 20:36:38,997 INFO -> informação
2021-05-10 20:36:38,997 | WARNING -> aviso
2021-05-10 20:36:38,997 ERROR -> erro
2021-05-10 20:36:38,997 CRITICAL -> critico
```

Como você pode notar, a formatação do log foi definida na linha 4, o %(asctime)s adiciona o hora e a data da n (levelname)s adiciona a gravidade da mensagem e %(message)s adiciona a mensagem em si.

Na linha 8, utilizamos a variável 'FORMATACAO\_MSG', que contém a formatação do log, como argumento para format no basicConfig(). Assim, você pode definir o formato da mensagem de registro, facilitando a leitura do identificação de algum erro ocorrido.

Na próxima página, você conhecerá outras configurações que pode fazer no arquivo de log.