



Rafael Vieira Coelho

rafaelvc2@gmail.com



Logging

O que seria Logging?

- Logging é uma forma de rastrear eventos que ocorreram com um software quando o mesmo está rodando.
- O desenvolvedor de software adiciona chamadas de log em seu código que indica o que ocorreu naquele trecho do código.
- Um evento é descrito por uma mensagem que descreve o evento e pode conter dados do programa.
- Cada evento pode estar associado por um nível de severidade.

Funções de Logging

- **debug()**: informação detalhada que pode ser usada para diagnóstico de problemas.
- **info()**: informação de que algo está funcionando corretamente.
- **warning()**: aviso sobre algo inesperado mas que não atrapalha o funcionamento. Pode gerar erros futuros.
- **error()**: informa um erro que está ocorrendo e atrapalhando o funcionamento de parte do *software*.
- **critical()**: relata um erro grave que faz com que o programa termine sua execução.

Valores de Níveis de Log

Level	Numeric value
CRITICAL	50
ERROR	40
WARNING	30
INFO	20
DEBUG	10

Definindo o Arquivo de Log de DEBUG

```
ex1_logging.py x ex2_logging.py x ex3_logging.py x ex4_logging.py x
1 import logging
2
3 logging.basicConfig(filename='file.log', level=logging.DEBUG)
4 logging.debug('This message should go to the log file')
5 logging.info('So should this')
6 logging.warning('And this, too')
```

```
ex1_logging.py x file.log x ex2_logging.py x info.log x ex3_logging.py x ex4_logging.p
1 DEBUG:root:This message should go to the log file
2 INFO:root:So should this
3 WARNING:root:And this, too
4
```

Definindo o Arquivo de Log de Informação

```
ex1_logging.py x ex2_logging.py x ex3_logging.py x ex4_logging.py x
1  import logging
2
3  def main():
4      logging.basicConfig(filename='info.log', level=logging.INFO)
5      logging.info('Started')
6      #...
7      logging.info('Finished')
8
9  if __name__ == '__main__':
10     main()
```

```
ex1_logging.py x file.log x ex2_logging.py x info.log x ex3_logging.py x ex4_logging.p
1  INFO:root:Started
2  INFO:root:Finished
3
```

Mostrando as Informações de Log na Tela

```
ex1_logging.py x ex2_logging.py x ex3_logging.py x ex4_logging.py x
1 import logging
2
3 logging.basicConfig(format='%(levelname)s:%(message)s', level=logging.DEBUG)
4 logging.debug('This message should appear on the console')
5 logging.info('So should this')
6 logging.warning('And this, too')
```


Adicionando Data e Hora a Mensagem

```
ex1_logging.py x file.log x ex2_logging.py x info.log x ex3_logging.py x ex4_logging.py x data.log x
1 import logging
2
3 logging.basicConfig(filename='data.log', format='%(asctime)s %(message)s')
4 logging.warning('is when this event was logged.')
5
```

```
ex1_logging.py x file.log x ex2_logging.py x info.log x ex3_logging.py x ex4_logging.py x data.log x
1 2019-05-24 08:07:08,538 is when this event was logged.
2
```

Tarefas

1. Modifique a sua aplicação criada até o momento para o trabalho prático para que mostre mensagens de erro, informação e aviso através do módulo logging de Python.