**Python Logging**

By Sovratica (<https://www.youtube.com/watch?v=g8nQ90Hk328> )

Purpose: to record progress or problems while execution of code

Levels: debug, info, warning, error, critical

Basic code:

import logging

logging.basicConfig(filename = “E:\\logfile.log”)

logger = logging.getLogger() #-- logger object

logger.info(“Our first message.”)

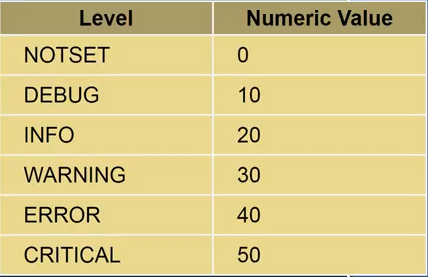
Upon executing this code the log file is created but there is no message inside it.

This is because of the way in which the logger was configured. By printing the following

print (logger.level)

The answer is 30, because by default the logger is set to level ‘WARNING’ which is of level 30 and since the logger.info is of level 20 it did not output anything. Note: for a logger to output a message it must be of a level ***equal to*** or ***greater than*** the set level.

Different levels of logging with their integer values



In the example logger.info is of level 20, while the logging.basicConfig was of level 30.

logging.basicConfig(filename = “E:\\logfile.log”, level = logging.Debug)

By doing so all the log messages will be recorded because Debug is of level 10.

In order to include other information we need to include a format to the logger object while creating it. For example we will include the time:

LOG\_FORMAT = “%(Levelname)s %(asctime)s - %(message)s”

logging.basicConfig(filename = “E:\\logfile.log”, level = logging.Debug, format = LOG\_FORMAT)

Example while finding the roots of a quadratic equation:

import math

import logging

LOG\_FORMAT = “%(Levelname)s %(asctime)s - %(message)s”

logging.basicConfig(filename = “E:\\logfile.log”, level = logging.Debug, format = LOG\_FORMAT, filemode = ‘w’)

logger = logging.getLogger()

def quadratic\_formula(a, b, c):

logger.info(“quadratic\_formula({0}, {1}, {2})”.format(a,b,c))

logger.debug(“ compute the discriminant)

disc = b\*\*2 – 4\*a\*c

logger.debug(“compute the roots)

root1 = (-b + math.sqrt(disc)) / (2\*a)

root2 = (-b - math.sqrt(disc)) / (2\*a)

logger.debug(“return the roots”)

return (root1, root2)

roots = quadratic\_formula(1, 0, 2)

print(roots)