Python logging is a module that allows you to track events that occur while your program is running. You can use logging to record information about errors, warnings, and other events that occur during program execution. logging is a useful tool for debugging, troubleshooting, and monitoring your program.

**Python Logging Level**

Python logging levels are used to indicate the severity of a message. There are five logging levels in Python:

* DEBUG,
* INFO,
* WARNING,
* ERROR,
* CRITICAL.

The basicConfig() method can be used to configure the logging system in a number of ways. By default, it will configure the logger to output to the standard output stream, with a default logging level of INFO.

logging.basicConfig(filename='example.log', level=logging.DEBUG)

There are a variety of parameters that can be used with the basicConfig() function in Python.

**Commonly Used Parameters:**

**filename**: This parameter is used to specify the filename that the logging messages will be written to.

**filemode:** This parameter is used to specify the mode that the file will be opened. The default mode is 'a', which means that messages will be appended to the end of the file.

**Format:** This parameter is used to specify the format of the logging messages. The default format is '%(levelname)s: %(message)s', which means that the level name and message will be printed.

**datefmt**: This parameter is used to specify the date format of the logging messages.

**Level**: This parameter is used to specify the minimum level of messages that will be logged. The default level is 'WARNING', which means that only messages with a level of 'WARNING' or higher will be logged.

import logging

logging.warning('Working on it!') # will print a message to the console

logging.info('Doing Much More') # will not print anything

**Example 2:**

**import** **logging**

logger = logging.getLogger(\_\_name\_\_)

logging.basicConfig(filename='example.log', encoding='utf-8', level=logging.DEBUG)

logger.debug('This message should go to the log file')

logger.info('So should this')

logger.warning('And this, too')

logger.error('And non-ASCII stuff, too, like Øresund and Malmö')

Ex3:

import logging

import logging.config

logging.config.fileConfig('logging.conf')

# create logger

logger = logging.getLogger('simpleExample')

# 'application' code

logger.debug('debug message')

logger.info('info message')

logger.warning('warn message')

logger.error('error message')

logger.critical('critical message')