## Python Logging

The Python logging facility contains several parts.

**Logger** - provides the programmer's interface to configure logging (configuration can also be done using a config file) and for creating log messages. Logger invoke Handlers to process the log messages.

The Python documentation states:

*"Loggers should NEVER be instantiated directly, but always through the function logging.getLogger(name). Multiple calls to getLogger with the same name always return a reference to the same Logger object."*

**LogRecord** contains the message text, timestamp, name of the function that invoked the logger, and all other info related to the logging event. LogRecord is passed as a parameter to various components of the logging package.

**Formatter** - convert a LogRecord object into a formatted string for printing to passing to an external system

**Filter** - decides whether or not a LogRecord should be output or not.

**Handler** - is responsible for sending log messages to their destination. It uses a Filter to decide whether the message should be logged and a Formatter to format the message. There are many kinds of handlers, such as

StreamHandler - send messages to a stream like sys.stdout (standard output)

FileHandler - write messages to a file

HTTPHandler - send messages to a server via the HTTP protocol

SMTPHandler - send messages via the SMTP protocol to a mail server

Handler is the base class for all handlers.

Logging Example

import logging

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

logger.setLevel(logging.DEBUG) # base logging threshold

# Create handlers

shandler = logging.StreamHandler()

shandler.setLevel(logging.WARNING) # handler logging threshold

fhandler = logging.FileHandler('demo.log')

# specify log format for each handler

format1 = logging.Formatter('%(levelname)s : %(message)s')

shandler.setFormatter(format1)

format2 = logging.Formatter(

'%(asctime)s %(funcName)s:%(levelname)s: %(message)s')

fhandler.setFormatter(format2)

# Add handlers to the logger

logger.addHandler(shandler)

logger.addHandler(fhandler)

# which of these messages will be output by each logger?

logger.info('This is an info message')

logger.error('This is an error message')

## Where are LogRecords Used?

To answer this, look at the methods of the Handler, Filter, and Formatter classes.

## How are LogRecords Created?

Look at the methods of the Logger class. Do you see any suggestive names?