Logging

Python provides a standard and configurable logging facility. You can set up the collection of loggers & handlers separately from their actual *use* in your program.

In [1]:
```
import logging
logging.basicConfig()

log = logging.getLogger()
log
```

Out[1]:
```
<logging.RootLogger at 0x25a34d0>
```

In [2]:
```
log.log(logging.CRITICAL, 'This is a critical message')
log.log(logging.FATAL, 'This is a fatal message')
```

```
CRITICAL:root:This is a critical messageCRITICAL:root:This is a fatal message
```

In [3]:
```
logging.CRITICAL, logging.ERROR, logging.WARN, logging.INFO, logging.DEBUG
```

Out[3]:
```
(50, 40, 30, 20, 10)
```

In [4]:
```
log.critical('This is critical')
```

```
CRITICAL:root:This is critical
```

In [5]:
```
log.error('This is an error')
log.warn('This is a warning')
log.info('This is info')
log.debug('This is debug')
```

```
ERROR:root:This is an errorWARNING:root:This is a warning
```

In [6]:
```
log.setLevel(logging.DEBUG)
```

In [7]:
```
log.error('This is an error')
log.warn('This is a warning')
log.info('This is info')
log.debug('This is debug')
```

```
ERROR:root:This is an errorWARNING:root:This is a warningINFO:root:This is infoDEBUG:root:This i
```

In [8]:
```
log.info('This is a message with an argument %r', 'The argument')
```

```
INFO:root:This is a message with an argument 'The argument'
```

Sub-loggers

We can configure "child loggers" of the root logger by passing a name to the `getLogger` function:

In [9]:
```
root = logging.getLogger()
mylogger = logging.getLogger('mylogger')
mylogger.setLevel(logging.INFO)
root.debug('The root logger will print debug information')
mylogger.debug('mylogger will not')
```

```
DEBUG:root:The root logger will print debug information
```

In [10]:
```
mylogger.info('Information will propagate up to other loggers')
mylogger.propagate = 0
mylogger.info('But not if we set propagate to 0')
```

```
INFO:mylogger:Information will propagate up to other loggersNo handlers could be found for logge
```

Handlers and formatters

In [11]:
```
handler = logging.StreamHandler()
mylogger.handlers = [handler]
mylogger.info('Now this is being handled by mylogger')
```

```
Now this is being handled by mylogger
```

In [12]:
```
handler.setLevel(logging.WARN)
mylogger.info('Now this is suppressed by the handler')
```

In [13]:
```
handler.setLevel(logging.INFO)
handler.formatter = logging.Formatter('%(levelname)s:%(message)s')
mylogger.info('This is a message')
```

```
INFO:This is a message
```

If we set propagate back to 1, we'll see "doubled" messages:

In [14]:
```
mylogger.propagate = 1
mylogger.info('Hello, there')
```

```
INFO:Hello, thereINFO:mylogger:Hello, there
```

Logging Configuration

In [27]:
```
import sys
import logging.config


config = {
    'version': 1,
    'loggers': {
        'root': {
            'level': logging.ERROR,
            'handlers': ['stream' ] },
        'mylogger2': {
            'level': logging.INFO,
            'handlers': [ 'stream', 'file'] } },
    'handlers': {
        'stream': {
            'class': 'logging.StreamHandler',
            'formatter': 'basic',
            'stream': sys.stdout },
        'file': {
            'class': 'logging.FileHandler',
            'formatter': 'precise',
            'filename': '/tmp/logfile.log',
            'mode': 'w' } },
    'formatters': {
        'basic': {
            'format': '%(levelname)-8s %(message)s' },
        'precise': {
            'format': '%(asctime)s %(levelname)-8s %(name)-15s %(message)s',
            'datefmt': '%Y-%m-%d %H:%M:%S' } }
}


logging.config.dictConfig(config)

root = logging.getLogger()
mylogger2 = logging.getLogger('mylogger2')

root.error('error from root')
print
mylogger2.error('error from mylogger')
print
mylogger2.info('info from mylogger')
```

```
ERROR:root:error from root
ERROR    error from mylogger
ERROR:mylogger2:error from mylogger
INFO     info from mylogger
INFO:mylogger2:info from mylogger
```

In [28]:
```
print open('/tmp/logfile.log').read()
```

```
2012-10-07 16:47:15 ERROR    mylogger2       error from mylogger
2012-10-07 16:47:15 INFO     mylogger2       info from mylogger
```