

# Logging Mode Control

# basiclog.py

Remember this?

```
# Create a new file name `basiclog.py`. Type in this program:
import logging
mode = "development" # or "production"
log_file = "mylog.txt"
if mode == "development":
    log_level = logging.DEBUG
    log_mode = "w"
else:
    log_level = logging.WARNING
    log_mode = "a"
logging.basicConfig(level=log_level, filename=log_file, filemode=log_mode)
logging.debug("debug message")
logging.warning("look out!")
logging.critical("we have a problem here")
```

Modifying mode in a real program isn't going to cut it. What are your options?

# Environment Switch

```
import os
mode = os.environ.get('MODE', 'production')
print('Running in mode: ' + mode)
if mode == 'development':
    # ...
```

```
$ python3 myprogram.py
Running in mode: production
```

```
$ export MODE=development
$ python3 myprogram.py
Running in mode: development
```

# Config files

There are MANY ways to do this. I'll show you one way.

Start with this structure:

```
# config.py - in version control
import logging
LOGLEVEL = logging.WARNING
LOGMODE = "a"
LOGFILE="/var/log/myapp.log"

def init_logging():
    logging.basicConfig(level=LOGLEVEL, filename=LOGFILE, filemode=LOGMODE)
```

```
# In your main application:
import logging
from config import init_logging

init_logging()
logging.info("And we're live!")
```

# Adding an override

Next step: ADD the override block to the end of `config.py`.

```
# config.py - in version control
import logging
LOGLEVEL = logging.WARNING
LOGMODE = "a"
LOGFILE="/var/log/myapp.log"

def init_logging():
    logging.basicConfig(level=LOGLEVEL, filename=LOGFILE, filemode=LOGMODE)

# overrides
try:
    from localconfig import *
except ImportError:
    pass
```

# Local config override

This `localconfig.py` file is NEVER put in version control. Mark it in `.gitignore`, for example.

Any overrides to the safe, default config settings - for logging, or anything else - go in this file.

```
# localconfig.py
LOGLEVEL = "DEBUG"
LOGMODE = "w"
LOGFILE="/path/to/myapp-dev.log"
```

# Other Options

Maybe you'd like to structure this differently. Or you've seen it done differently in a project you've worked on.

THAT'S FINE. Infinite variations are possible, and valid.

Use these as a starting point, and vary it based on what works best for the project.