

# Exceptions & Logging

- Why exceptions over error codes
- Granular logging for network automation
- Patterns for retries/timeouts

# Exception Hierarchy

- Built-ins: `ValueError`, `TypeError`, `OSError`
- Network: `socket.timeout`, `requests.Timeout`
- Custom exceptions (optional)

# Try / Except Patterns

```
try:
    resp = client.get(url, timeout=5)
    resp.raise_for_status()
except TimeoutError:
    backoff()
except HTTPError as e:
    log.error("HTTP %s", e)
```

# Logging Setup

- `setup_logging(name)` from `common/lib/logging_setup.py`
- INFO → file, WARNING/ERROR → console
- Include device name and request IDs

# Structured Messages

- Avoid printing raw dicts; include keys of interest
- Use `%s` formatting in logger for performance

# Timeouts & Retries

- Always set timeouts for network calls
- Exponential backoff for transient failures
- Cap total retry time

# Don't Crash the Classroom

- Catch, log, continue on a per-device basis
- Summaries at the end: success/fail counts

# Exercise

- Wrap a Netmiko call with try/except
- Log failures and continue to next host