

# COMP1531

## 3.3 - SDLC Testing - pylint

What is good style?

Programs must be written for  
people to read, and only  
incidentally for machines to  
execute – *Abelson & Sussman,*  
*"Structure and Interpretation of*  
*Computer Programs"*

# Style

- Ultimately about readability and maintainability
- Style guides give rules of thumb and conventions to follow
- ...but good style is ultimately hard, if not impossible, to measure
- That said, tools can be a lot of help

There are a lot of  
tools in modern  
software  
engineering

# Pylint

- An external tool for *statically* analysing python code
- Can detect errors, warn of potential errors, check against conventions, and give possible refactorings
- By default, it is **very** strict
- Can be configured to be more lenient

# Controlling Messages

- Disable messages via the command line

```
$ pylint --disable=<checks> <files_to_check>
```

```
$ pylint --disable=missing-docstring <files>
```

- Disable messages in code; e.g.

```
if year % 4 != 0: #pylint: disable=no-else-return
```

- Disable messages via a config file

- If a `.pylintrc` file is in the current directory it will be used
- Can generate one with:

```
pylint <options> --generate-rcfile > .pylintrc
```