Software license terms

A license says what you can and can't do with a piece of code, such as:

- Run it
- Share it
- Re-sell it
- Tinker with how it works
- Modify it
- Write new software that uses it

Closed-source software

Two sub-categories:

- Proprietary / Commercial Software
- Shareware and Freeware

Open-source software

The source code is available, but what can you do with it?

- Permissive licenses (e.g. BSD, MIT):
  Do what you want, just don't sue us
- **Copyleft** licenses (e.g. GPL2, GPL3):
  If you use this code, you have to open-source your project too

Controversy: Security

Is open-source software more or less secure than proprietary?

Proprietary software benefits

- Hackers can’t see the source code to find vulnerabilities
  (unless they compromise the parent company)
- Company that makes the software can (maybe) be held responsible

Open-source software benefits

- Vulnerabilities can be spotted and patched more rapidly;
  lots of eyes on the code (in principle)
- Users (or agencies) can make their own informed security decisions

Some case studies

- Companies that use GPL (copyleft) software but don’t want to open-source their own code
  (Amazon Kindle, John Deere, Vizio)
- In-house fighting over which version of open source is best
  (GPL2, LGPL, GPL3, MIT, BSD k-clause, etc)
- Open-source licenses in the 2020s
  (Hashicorp and ”source-available”; lawsuits against AI coding tools)

Class outcomes

You should know:

- What software licenses can allow and restrict
- The meaning of proprietary and shareware programs
- The two main types of open-source licenses
- Examples and recent developments in the meaning and use of open-source software