

Introduction to Open Source Tools

History, Advantages, Disadvantages, and Licenses

Edition • 2024

Instructor: Adel MOUSSAOUI

Course: Open Source Tools

© 2024

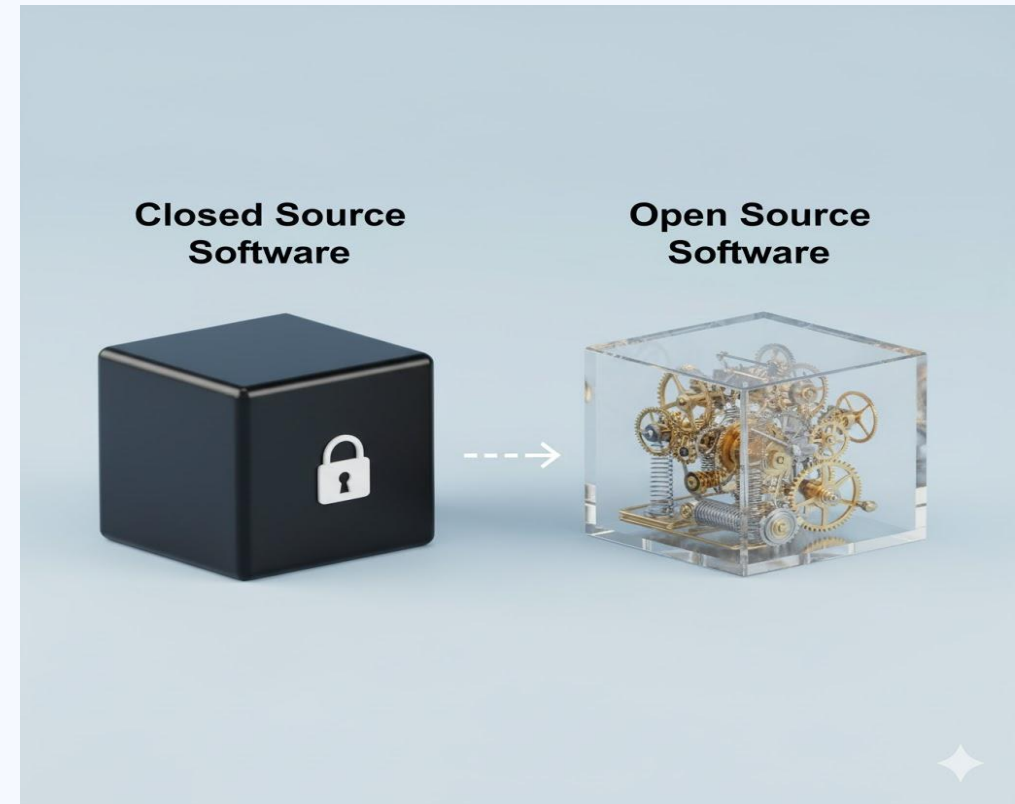
What is "Open Source"?

Open Source Software (OSS) is defined by its **license** and its **collaborative development model**.

- 👁 **Inspect** the code
- ✎ **Modify** it
- 📈 **Enhance** it
- 🔗 **Distribute** original and modified versions

80%

of organizations increased open source usage in 2023



A Brief History: The Early Days (1960s-1980s)

- 1960s

Software **shared freely** among academics and researchers
- 1970s-80s

Culture shifted toward **proprietary, closed-source** software
- 1983

Richard Stallman launched the **GNU Project**
- 1985

Free Software Foundation (FSF) established

The Four Freedoms (FSF)

- 1

Run the program as you wish
- 2

Study and change the source code
- 3

Redistribute copies
- 4

Distribute copies of your modified versions

Category	Open Source Software	Proprietary Software
Definition	It refers to the software that is developed and tested through open collaboration.	It refers to the software that is solely owned by the individual or the organization that developed it.
Source Code Access	Anyone with academic knowledge can access, inspect, modify, and redistribute the source code.	Only the owner or publisher who holds the legal property rights of the source code can access it.
Management	The project is managed by an open-source community of developers and programmers.	The project is managed by a closed group of individuals or team that developed it.
Target Users	They are not aimed at unskilled users outside of the programming community.	They are focused on a limited market of both skilled and unskilled end users.
Flexibility & Innovation	It provides better flexibility, which means more freedom and encourages innovation.	There is a very limited scope of innovation with restrictions and control.
Examples	Android, Firefox, LibreOffice, Ubuntu, FreeBSD, Drupal, GNOME, etc.	Windows, macOS, iTunes, Google Earth, Adobe Flash Player, etc.

A Brief History: The Spark - Linux & The OSI

- 1991** **Linus Torvalds**, a Finnish student, created the Linux kernel
- 1992** Linux kernel combined with GNU tools created a complete **free operating system**: GNU/Linux
- 1998** Term "**Open Source**" coined to make the concept more palatable to the business world
- 1998** **Open Source Initiative (OSI)** founded to formally define and promote the term

The combination of Linux kernel with GNU tools created a powerful, free alternative to proprietary operating systems

"Open source is a development method for software that harnesses the power of distributed peer review and transparency of process."

— The Open Source Initiative



Advantages of Open Source: Why Does It Thrive?



Cost-Effective

No licensing fees. Saves money for individuals and businesses.



Transparency & Security

"Given enough eyeballs, all bugs are shallow."
Code can be audited by anyone.



Freedom & Flexibility

Not locked into a single vendor. Customize software to fit exact needs.



Quality & Collaboration

Global community of developers contributes to innovation and bug fixes.



Longevity

Software isn't tied to a single company's fate. Communities can sustain projects.



92%

of developers are using or experimenting with **AI coding tools** in 2023

Disadvantages & Challenges of Open Source



Learning Curve

Can be less user-friendly than mainstream proprietary alternatives.



Fragmentation

Multiple versions/forks split community effort (e.g., Linux distributions).



Support

Relies on community forums and documentation rather than dedicated phone support.



Compatibility

Can have issues with file formats or hardware drivers designed for proprietary systems.



Abandonment

If a project loses its community, it can become "abandonware."



Top Concern: Security

Security remains the **#1 challenge** for organizations using open source software in 2023

The Heart of It All: Open Source Licenses

The license is the **legal engine** that makes open source work. It legally enforces the freedoms of use, modification, and distribution.

⇒ Copyleft

Requires derivatives to be released under the same license

✔ Permissive

Minimal restrictions on use, modification, and distribution



1.7M

pageviews for **MIT license** in 2024, making it the most popular open source license

	Types of Licenses		Permissions					Conditions					Limitations		
License Name	Permissive	Copyleft	Commercial Use	Distribution	Modification	Private Use	Patent Use	License & Copyright notice	State Changes	Same Changes	Disclosure Source	Network use in Distribution	Liability	Warranty	Trademark Use
Apache License 2.0	✔		✔	✔	✔	✔	✔	✔	✔				✔	✔	✔
MIT License	✔		✔	✔	✔	✔		✔					✔	✔	
Berkeley Software Distribution	✔		✔	✔	✔	✔		✔					✔	✔	
Eclipse Public License	✔		✔	✔	✔	✔	✔	✔	✔				✔	✔	
Microsoft Public Licenses (Ms-PL)	✔		✔	✔	✔	✔		✔					✔	✔	
Mozilla Public Licenses 2.0		✔	✔	✔	✔	✔	✔	✔	✔		✔		✔	✔	✔
GNU GPLv3		✔	✔	✔	✔	✔	✔	✔	✔		✔		✔	✔	
GNU AGPLv3		✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	
GNU AGPLv3		✔	✔	✔	✔	✔	✔	✔	✔	✔	✔		✔	✔	✔

Copyleft Licenses: "Share Alike"

⇒ Philosophy

If you use and modify this code, you must release your modifications under the **same license**

📋 Famous Examples

GNU GPL

Strongest copyleft. Used by Linux kernel, Git, WordPress

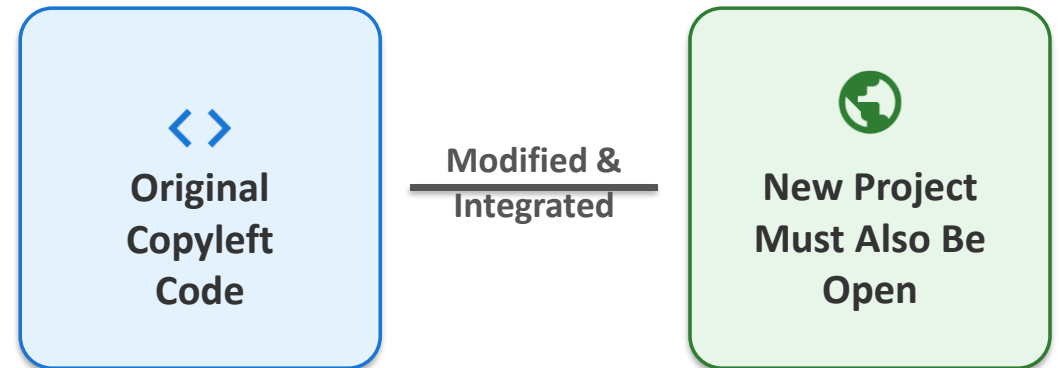
GNU LGPL

Less restrictive. Often used for libraries

👍 Benefits & Considerations

- ✓ Ensures derivatives remain open
- ✓ Protects software freedom
- ✓ Prevents proprietary lock-in
- ⚠️ Less business-friendly

How Copyleft Works



Permissive Licenses: "Do Almost Anything"

✓ Philosophy

Minimal restrictions. You can freely use, modify, and distribute the code, even in **closed-source** software.

📋 Famous Examples

MIT License

Very short and simple. Used by React, .NET Core

Apache 2.0

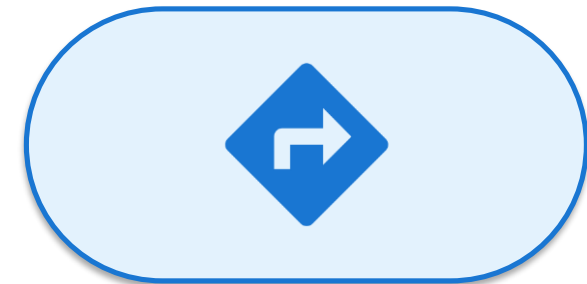
Similar to MIT but with explicit patent grant. Used by Android

BSD License

Family of licenses with similar permissiveness. Used by FreeBSD

👍 Benefits & Considerations

- ✓ Business-friendly
- ✓ Simple requirements
- ✓ Maximizes adoption
- ⚠ Less protection of freedom



"Take my code and do what you want, just give me credit."



78%

of open source components have **permissive licenses**

Other Licenses & The Public Domain



Creative Commons Licenses

Often used for non-software content (docs, images, music). **CC BY-SA** is similar to copyleft.



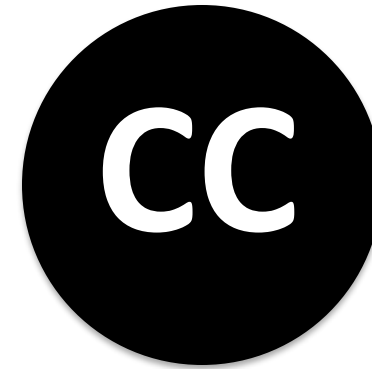
Unlicensed Code

Code without a license is **NOT open source**. All rights are reserved by default. You cannot use it freely.



Latest Trend: Defining Open Source AI

The Open Source Initiative (OSI) is leading global efforts to define open source AI, considering not just code and algorithms but also **data** and model transparency.



Creative Commons Zero (CC0)

A "public domain" dedication. You waive all your rights to the work, allowing anyone to use it for any purpose without attribution.

Open Source in Action: Where Is It?

You use it every day!



Internet

~70% of web servers
run on Apache or
Nginx



Your Phone

Android is built on the
Linux kernel



Development

Git, VSCode, Python,
and countless other
tools



Supercomputers

100% of top 500
supercomputers run
on Linux



2023

Saw the **largest number** of first-time open source contributors



Famous Open Source Success Stories



Firefox

A privacy-focused browser championing an **open web**. Proved that community-driven browsers can compete with commercial giants.



Wikipedia

The free encyclopedia that anyone can edit. A testament to **open collaboration** and the power of collective knowledge.



WordPress

Powers over **40%** of all websites on the internet. Democratized web publishing for millions of users worldwide.



Blender

A full-featured, professional-grade 3D creation suite that **rivals expensive proprietary tools**. Used in major film productions.

Latest Success: OpenTofu



When HashiCorp moved Terraform to a Business Source License, the community created OpenTofu as a truly open-source fork, demonstrating the resilience of the open source ecosystem.

Common Misconceptions



"It's always free."

Not always. Some companies sell support or proprietary versions. But the core is almost always free.



"It's unreliable and low-quality."

Often the opposite. The scrutiny of many experts leads to very stable and secure code.



"It has no support."

Community support is vast. Professional, paid support is also available for enterprise-grade OSS.



"It's only for programmers."

End-users benefit massively from the results (browsers, office suites, media players).

Debunking Open Source Myths

Open source is always free

Open source has no support

Open source is low quality

Open source is only for developers

Why Should You Care?



As a User

Access to powerful, free software. Escape vendor lock-in.



As a Student/Learner

Unprecedented access to read and learn from real-world code.



As a Developer

Build on the shoulders of giants. Contribute to something bigger.



As a Professional

OSS skills are highly valuable in the job market.



As a Human

Support a movement based on openness, collaboration, and shared knowledge.

Open source empowers collaboration across borders and backgrounds

Recap & Next Session

What We Covered

 **Definition:** Source code is open for collaboration

 **History:** From Free Software to Linux and OSI

 **Pros:** Cost, freedom, security

 **Cons:** Learning curve, support

 **Licenses:** Copyleft (GPL) vs. Permissive (MIT/Apache)

Next Session: Development Environment

We'll get hands-on! Introduction to Linux and powerful code editors to set up your own open-source development workstation.

Latest Trends

-  **"Linux Wars"** began when Red Hat restricted access to RHEL source code
-  Shift from open source to **source-available licenses** (e.g., HashiCorp's BUSL)
-  OSI defining **open source AI** standards considering code, algorithms, and data
-  2023 saw the **largest number** of first-time open source contributors

Links for Further Learning

-  opensource.org
-  choosealicense.com
-  [The Cathedral and the Bazaar \(Essay\)](#)