

Open source software



Computer science department third year cs

Lecturer : yasmin maki

What is open source?

- ▶ The term "**open source**" refers to something people can modify and share because its design is publicly accessible.
- ▶ The term originated in the context of **software development** to designate a **specific approach to creating computer programs**.
- ▶ Today, however, "**open source**" designates a broader set of values—what we call "**the open source way**."
- ▶ Open source projects, products, or initiatives hold and celebrate principles of :
 - ▶ **open exchange,**
 - ▶ **collaborative participation ,**
 - ▶ **rapid prototyping,**
 - ▶ **and community-oriented development.**

What is open source software?

3

- **Open source software :**

is software with source code that anyone can review, modify, and enhance.

- **"Source code"** is the part of software that most computer users don't ever see; it's the code computer programmers can manipulate to change how a piece of software—a "program" or "application" works.

Programmers who have access to a computer program's source code can improve that program by:

- adding features to it
- or fixing parts that don't always work correctly.

What's the difference between open source software and other types of software?

4

- **"proprietary" or "closed source"** software is a software has **source code** that **only** the **person, team, or organization** who created it—and maintains exclusive **control over** it—can modify. Only the **original authors** of proprietary software can **legally copy, inspect, and alter** that software.

- **to use proprietary software:** **computer users** must **agree** (usually by **signing a license displayed the first time they run this software**) that they will **not do** anything with the software that the **software's authors** have not **expressly permitted**.

Microsoft Office and Adobe Photoshop are **examples** of proprietary software.

- **Open source software is different.**

Its **authors** make its source code available to others who would like to (**view** that code, **copy** it, **learn** from it, **alter** it, or **share** it).

LibreOffice and the GNU Image Manipulation Program are examples of open source software.

As they do with proprietary software, **users** must accept the **terms of a license** when they use open source software—but the **legal terms** of open source licenses **differ dramatically** from those of **proprietary licenses**.

- **Open source licenses** affect the **way** people can use, study, modify, and distribute software.

In general, open source **licenses grant** computer users permission to use open source software for any purpose they wish.

"copyleft": Some open source licenses—what some people call **"copyleft" licenses**—stipulate that anyone who releases a modified open source program must also release the source code for that program alongside it.

some open source licenses stipulate that anyone who **alters and shares** a program with others must also share that **program's source code** without charging a **licensing fee** for it.

By **design**, open source **software licenses** promote **collaboration** and **sharing** because they **permit other people to make modifications** to source code and incorporate those changes into their own projects.

They encourage computer programmers to **access, view, and modify** open source software whenever they like, as long as they let others do the same when they share their work.

Is open source software only important to computer programmers?

No. Open source technology and open source thinking both benefit **programmers** and **non-programmers**.

Because early inventors built much of the Internet itself on open source technologies—like the Linux operating system and the Apache Web server application—anyone using the Internet today benefits from open source software.

Examples :

Every time computer **users view web pages, check email, chat with friends, stream music online, or play multiplayer video games**, their computers, mobile phones, or gaming consoles connect to a global network of computers **using open source** software to route and transmit their data to the "local" devices they have in front of them.

The computers that do all this important work are typically located in faraway places that users don't actually see or can't physically access—which is why some people call these computers "remote computers."

More and more, people rely on remote computers when performing tasks they might otherwise perform on their local devices.

For example, they may use online word processing, email management, and image editing software that they **don't install and run on their personal computers**. Instead, they simply access these programs on remote computers by using a Web browser or mobile phone application. When they do this, they're engaged in **"remote computing."**

Some people **call remote computing "cloud computing,"** because it involves activities (like **storing files, sharing photos, or watching videos**) that incorporate not only local devices but also a global network of remote computers that form an "atmosphere" around them.

Cloud computing is an increasingly important aspect of everyday life with Internet-connected devices.

Some cloud computing applications, like **Google Apps**, are proprietary.

Others, like ownCloud and Nextcloud, are open source.

Cloud computing applications run "on top" of additional software that helps them operate smoothly and efficiently, so people will often say that software running "underneath" cloud computing applications acts as a "platform" for those applications.

Cloud computing **platforms** can be **open source or closed source**.

OpenStack is an example of an open source cloud computing platform.

Why do people prefer using open source software?

13

People prefer open source software to proprietary software for a number of reasons, including:

Control. Many people prefer open source software because they have more control over that kind of software.

They can examine the code to **make sure it's not doing anything they don't want it to do**, and **they can change parts of it they don't like**.

Users who **aren't programmers** also benefit from open source software, because they can **use this software for any purpose** they wish—not merely the way someone else thinks they should.

Training. Other people like open source software because it helps them become better programmers.

Because open source code is **publicly accessible, students** can easily study it as they learn to make better software.

Students can also share their work with others, inviting comment and analysis, as they develop their skills.

When people discover mistakes in programs' source code, they **can share** those **mistakes** with others to help them avoid making those same mistakes themselves.

Security. Some people prefer open source software because they consider it **more secure** and stable than proprietary software.

Because anyone can view and modify open source software, someone might spot and correct errors or omissions that a program's original authors might have missed.

And **because so many programmers** can work on a piece of open source software without asking for permission from original authors, they can fix, update, and upgrade open source software more **quickly** than they can proprietary software.

Stability. Many users prefer open source software to proprietary software **for important, long-term projects.**

Because programmers publicly distribute the source code for open source software, users relying on that software for critical tasks can be sure their tools won't disappear or fall into disrepair if their original **creators stop working on them.**

Additionally, open source software tends to both incorporate and operate according to open standards.

Doesn't "open source" just mean something is free of charge?

17

No. This is a common misconception about what "open source" implies, and the concept's implications are not only economic.

Open source software programmers can **charge money** for the open source software they create or to which they contribute.

But in some cases, because an **open source license** might require them to release their source code when they sell software to others, **some programmers** find that **charging users money** for *software services* and *support* (rather than for the software itself) is more profitable.

This way, their software **remains free of charge**, and they make money helping others install, use, and troubleshoot it.

While some open source software may **be free of charge**, skill in programming and troubleshooting open source software can be quite valuable. Many **employers** specifically seek to hire programmers with experience working on open source software.

Examples Of Open Source Software

1. Apache HTTP Server [<http://httpd.apache.org/>] (web server)
2. Blender [<http://www.blender.org>] (3D graphics and animation package)
3. DSpace [<http://www.dspace.org/>] (digital repository)
4. EPrints [<http://www.eprints.org>] (digital repository)
5. The GIMP [<http://www.gimp.org/>] (image editor),
6. GNOME [<http://www.gnome.org/>] (Linux desktop environment)
7. GNU Compiler Collection
[<http://www.gnu.org/software/gcc/gcc.html>] (GCC, a suite of compilation tools for C, C++, etc)
8. KDE [<http://www.kde.org/>] (Linux desktop environment)
9. LORLS [<http://bookworm.lboro.ac.uk/distribution.html>] (reading lists management system)
10. Mailman [<http://sourceforge.net/projects/mailman>] (mailing list manager)
11. Moodle [<http://www.moodle.org/>] (virtual learning system)

1. Firefox [<http://www.mozilla.com/en-US/firefox/>] (web browser based on Mozilla)
- 2.9 Thunderbird [<http://www.mozilla.com/en-US/thunderbird/>] (mail client based on Mozilla code)
3. MySQL [<http://www.mysql.com/>] (database)
4. OpenOffice.org [<http://www.openoffice.org/>] (office suite, including word processor, spreadsheet, and presentation software)
5. PHP [<http://www.php.net/>] (web development)
6. Perl [<http://www.perl.org/>] (programming/scripting language)
7. Plone [<http://plone.org/>] (content management system)
8. PostgreSQL [<http://www.postgresql.org/>] (database)
9. Python [<http://www.python.org/>] (programming/scripting language)
10. Sakai [<http://sakaiproject.org/>] (learning management system)
11. Samba [<http://www.samba.org/>] (file and print server)
12. SSL-Explorer: Community Edition [<http://sourceforge.net/projects/sslexplorer/>] (browser-based SSL VPN solution)
13. TeX [<http://www.tug.org/>] (typesetting language)
14. WUBS [<http://sourceforge.net/projects/wubs/>] (resource booking system)
15. Zope [<http://www.zope.org/>] (web application server)