Introduction to
open-Source Software (OSS)

Concepts, strategies, and methodologies
related to open-source software development

Week 01 – Lecture 01

- Introduction (Couse Overview)

Jamil Hussain
jamil@sejong.ac.kr
010-6252-8807

Office: 421, Innovation Center
Sejong University

Today, Agenda

- Teaching methodologies
- Discussed the overall learning path
- Course objectives
- Brief introduction of OSS
- History of Free and Open Source
- GNU and the Free Software Foundation (FSF)
- Open-Source Initiative

Welcome to the Course!

- Greetings and Introduction
  - Warm greetings to all participants!
  - An exciting journey into the world of OSS awaits.
- Engage and Interact
  - Your questions are encouraged!
  - Feel free to seek clarity if concepts seem challenging.
- Course Focus: Starting point for beginners to the open-source world
  - Educate you on the objectives of open-source
  - Understand open-source software licensing requirements
  - Get an introduction to the norms followed in the open-source world
  - Join the open-source movement and begin contributing.

Prerequisites

- English Language
- A good understanding of software development and software engineering
- Experience with at least one programming language

Teaching Methods

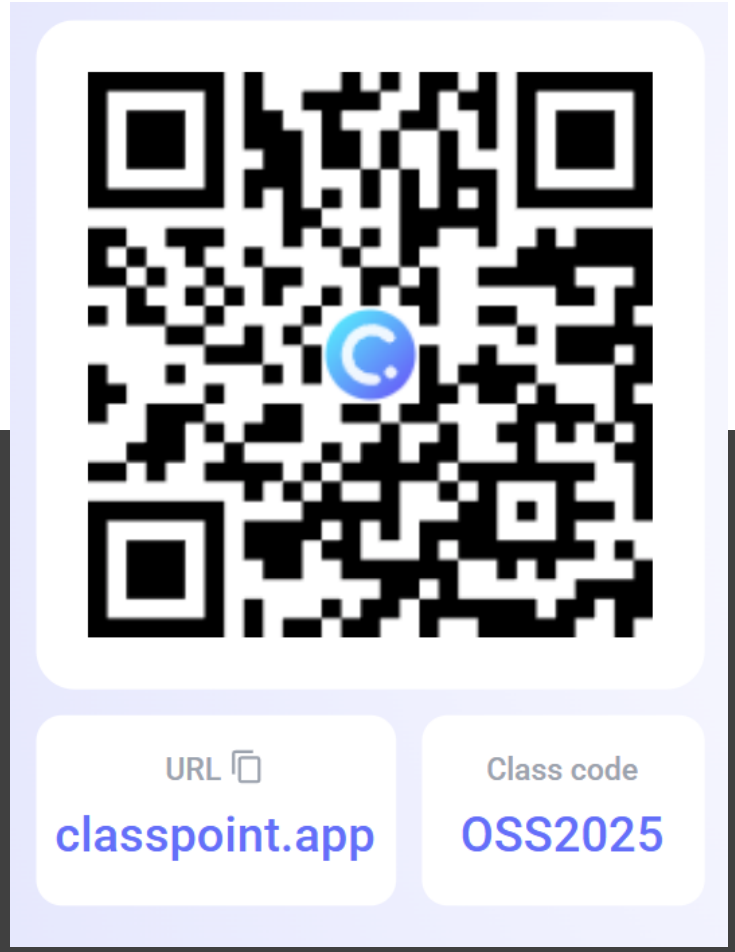
Project Based Learning

Gamification into your classroom

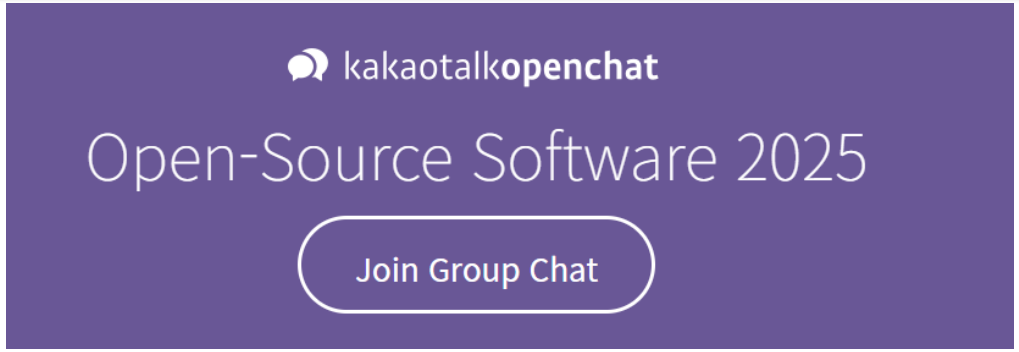
Teaching Methods

Interactive Classroom Quiz in PowerPoint

- Turn slides into quizzes and engage your students



Communication Channel



https://open.kakao.com/o/gDh7VmQh

Grading

- **Mid Exam: 20 %**
  - Mid Exam. Covers material seen during the lectures before the exam.
- **Final Exam: 25 %**
  - Final Exam. Covers material seen after the mid-exam
- **Quizzes: 10 %**
  - There will be popup quizzes during lectures
- **Attendance: 10 %**
- **Assignments & Group Project: 35 %**
  - Assignment will be due within seven (7) days of the announcement
  - No late assignments will be accepted
  - A group of max Three (4) students will be allowed

Assignments & Group Project!

- They will be posted on the website.
- Due 11:59 pm on the due date, submitted online.
- Can use the discussion board and labs to meet with your group members.

How can you succeed?

- Attend all classes
- Be punctual
- Download lecture slides and programming files before the class and follow along.
- Actively read reference materials and practice practice practice OSS tools
- Read course notifications in the portal
- Get help immediately when you need it
- Actively participate in a team project
- Do not cheat or copy assignments from others
- Be responsible for your own learning

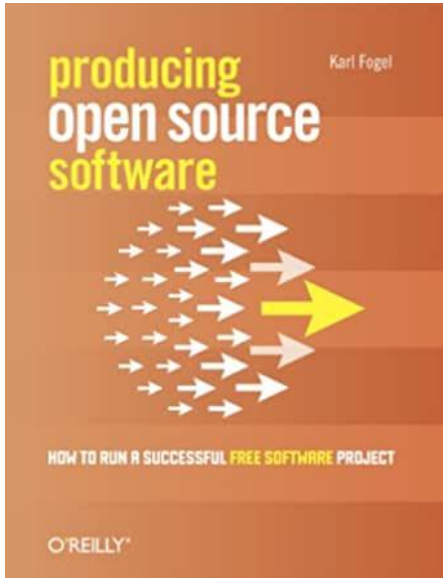
Academic Violations

- You should do all the work that you submit
- Never look at another team's work.
- Never show another team your work.
- Applies to all drafts and partial solutions.
- Discuss how to solve an assignment only with me

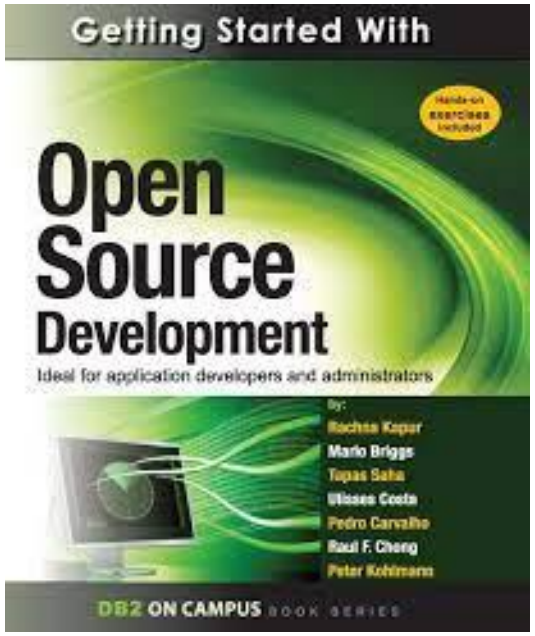
Getting Help

- **Office Hours.**
  - We're deciding on these right now!
- Can ask for help from me during labs.
- Course Discussion board.
- **Monday-Thursday.**
  - Office: Room 421, Innovation Building
  - Email: jamil@sejong.ac.kr
  - Office Hours: Thurs & Fri – 9:00 AM to 6:00PM (or appointment by email)

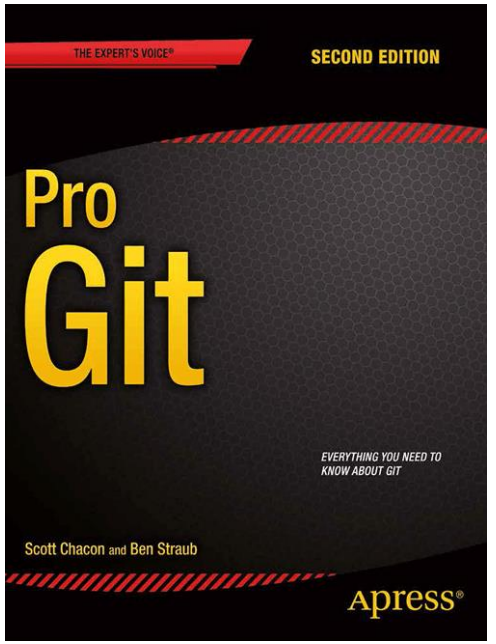
Recommend Books



Producing Open Source Software How to Run a Successful Free Software Project-Karl Fogel

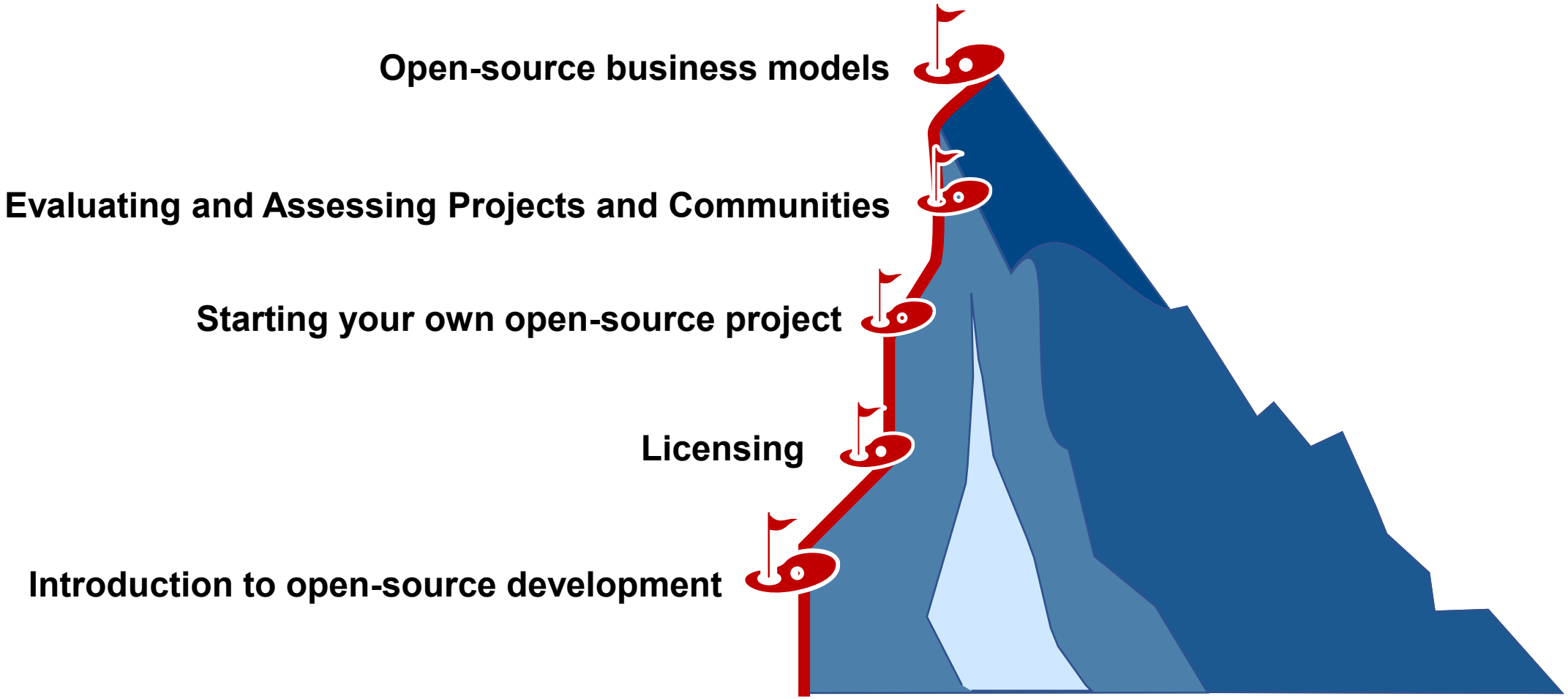


Getting started with open source development



Pro Git

Learning Path



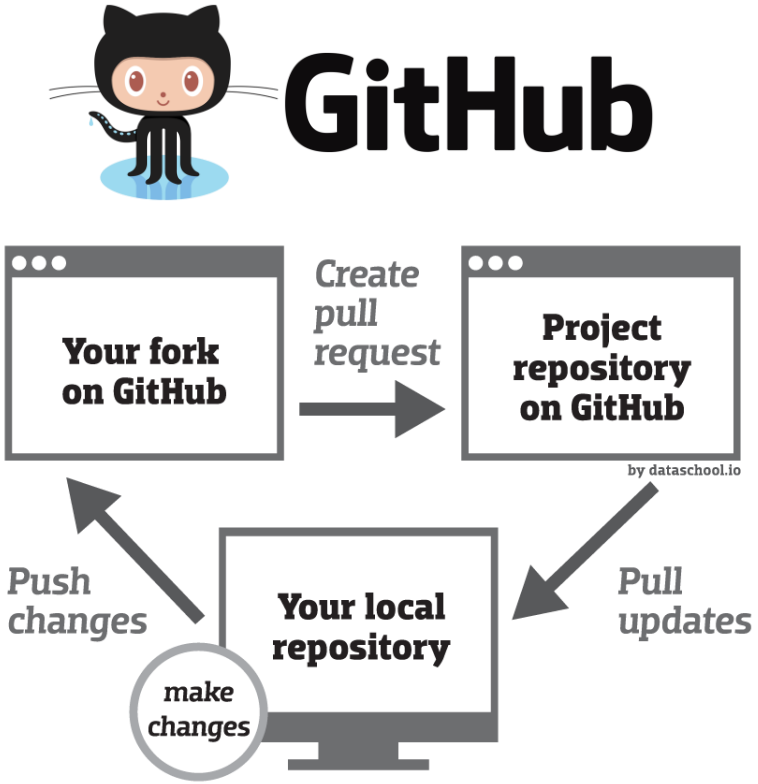
Minimum Objectives [1/4]

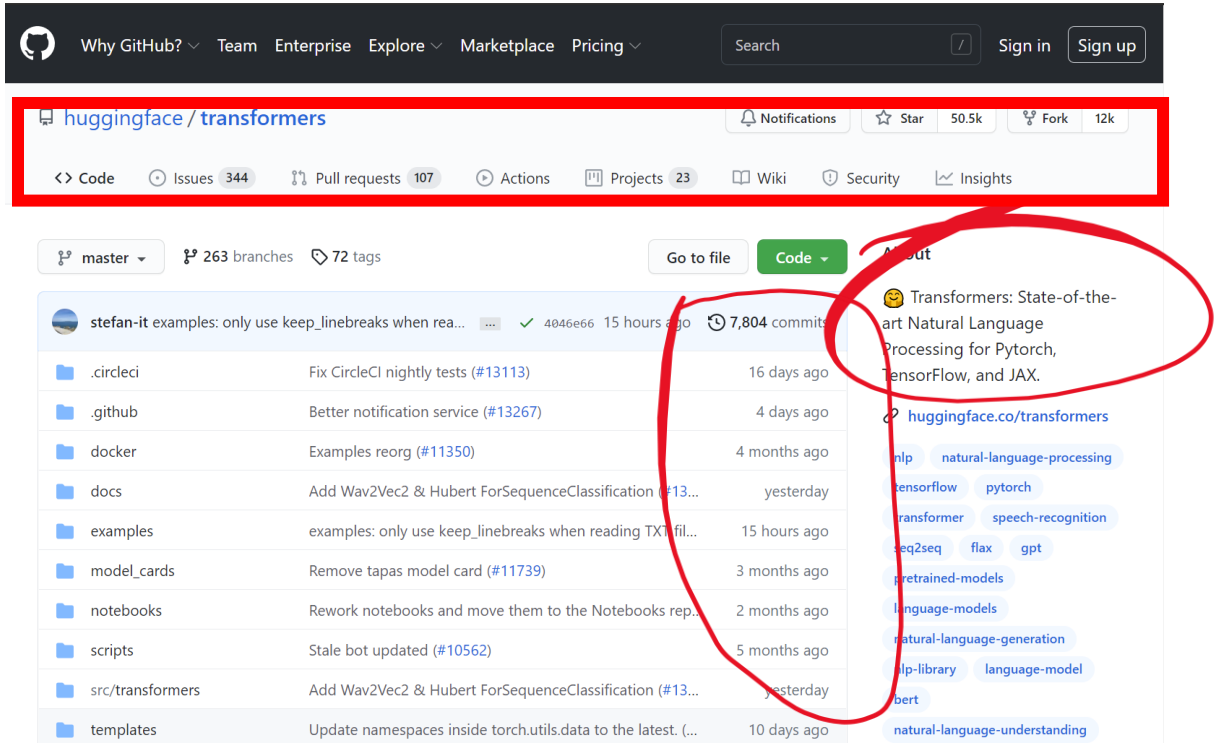
- Understand concepts, strategies, and methodologies related to open source-software development.

- Understand the common open-source licenses and the impact of choosing a license

Minimum Objectives [2/4]

- Understand open-source project structure and how to set up a project successfully





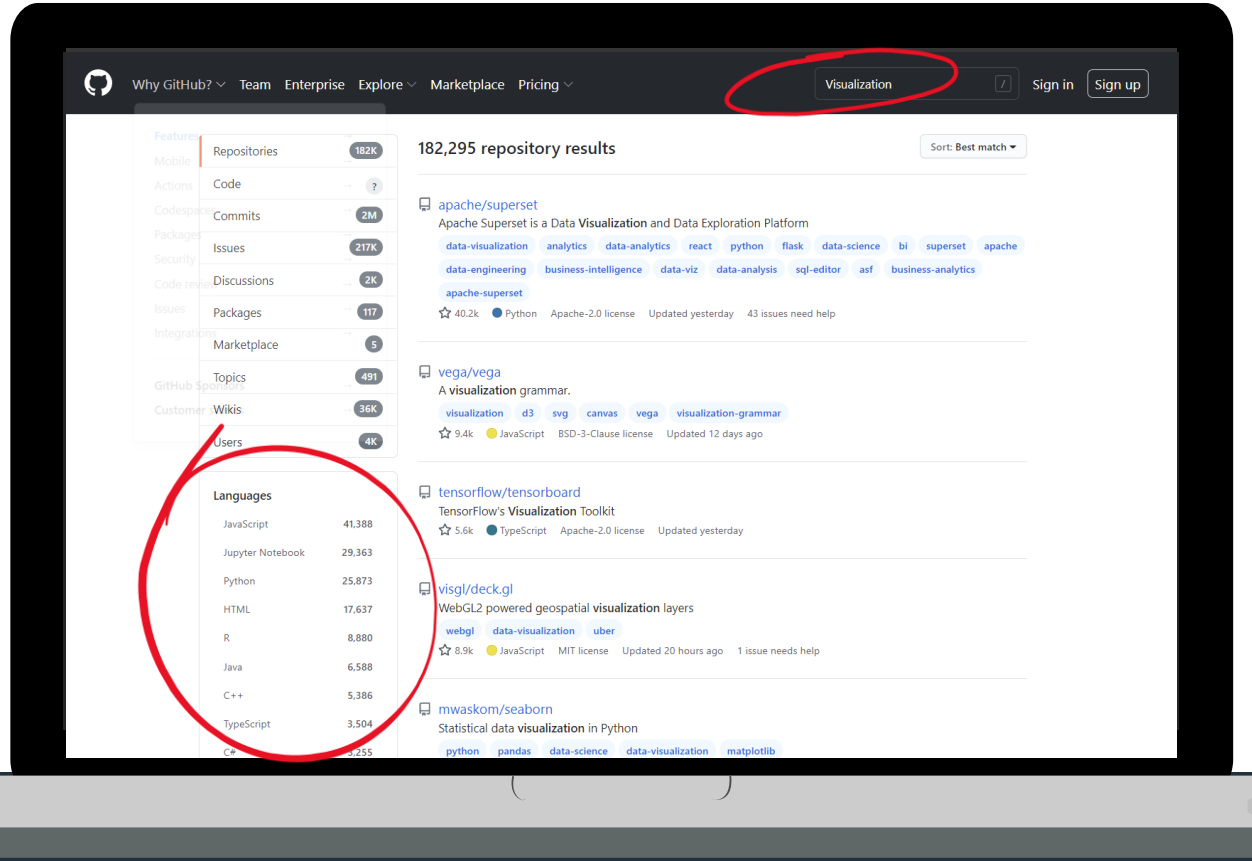
Minimum Objectives [3/4]

- Be familiar with open-source software products and development tools currently available.

Utilize open-source software for developing a variety of software applications.

Minimum Objectives [4/4]

- Be able to find open-source projects related to a given development problem.
- Be able to install from source code an open-source project and start using it.



Assessment about OSS

Assessment about OSS

Q1: Did you ever use any version control software?

A Yes

B No

Multiple Choice

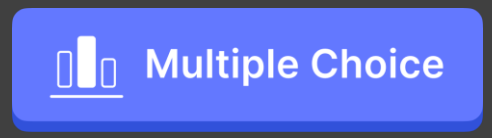
Assessment about OSS

Q2: Rate your skill in software development?

A Absolutely no previous SW development experience

B Developed small scale projects

C Expert in SW development



Assessment about OSS

Q3: Any Knowledge about the software Licensing?

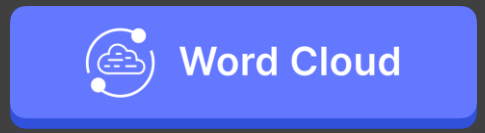
A Yes

B No

Multiple Choice

Assessment about OSS

Q3: Why do you enroll in this course?

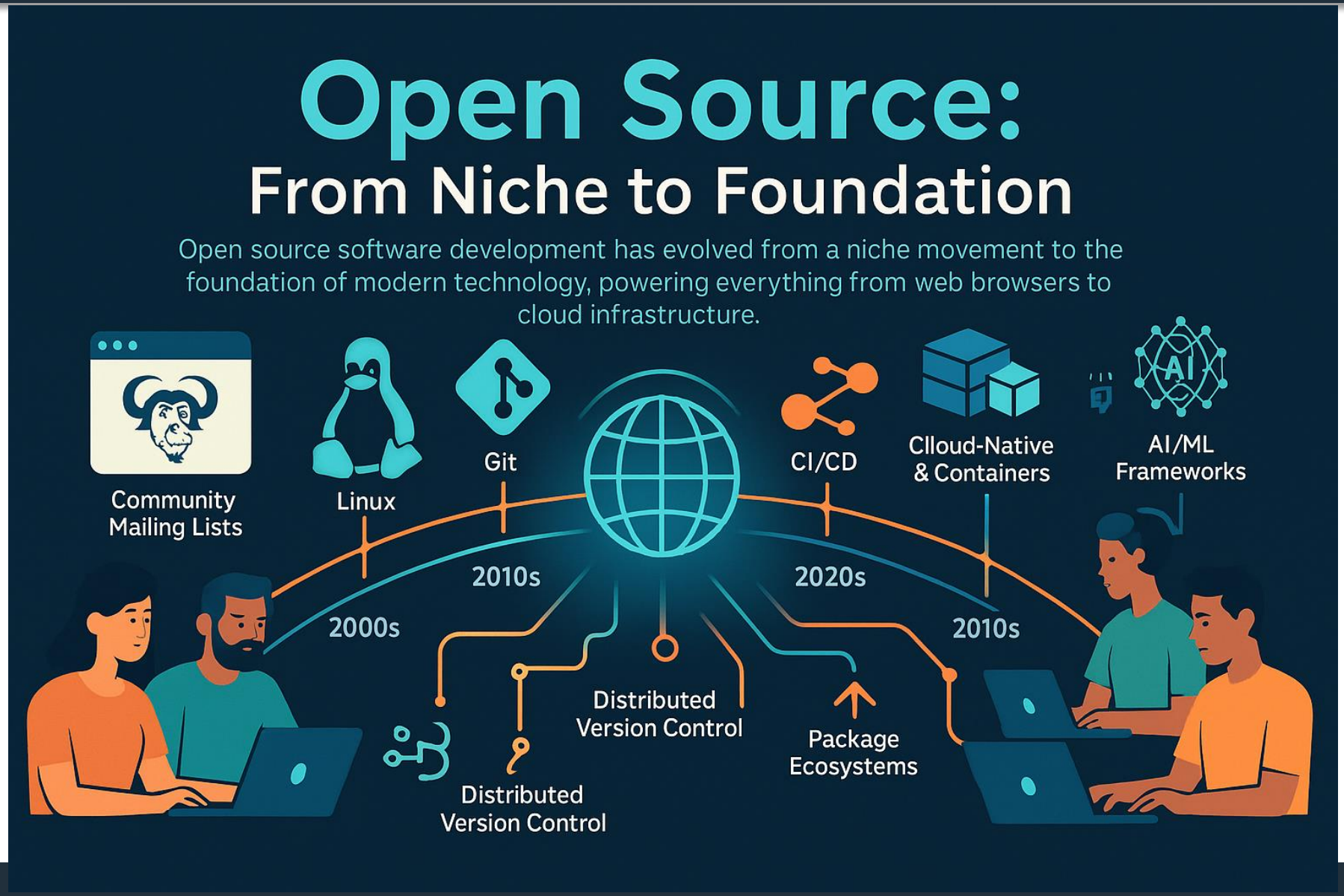


Getting Started with

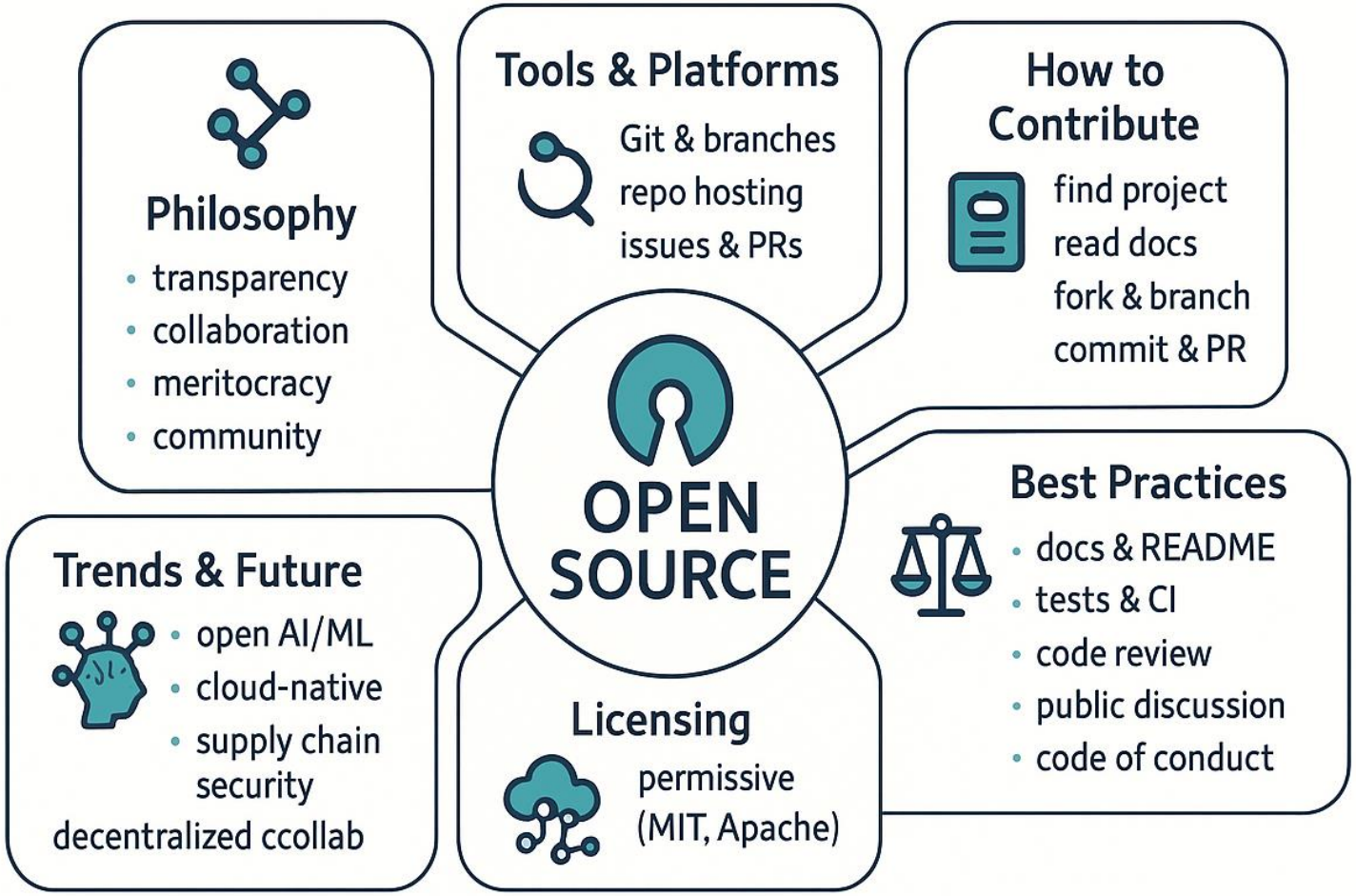
Open-source Software development

From Beginner's to Advanced

Open source software development



Open source software development



Discover → Fork → Branch → Commit → PR→ Review → Merge

Brief Introduction of OSS

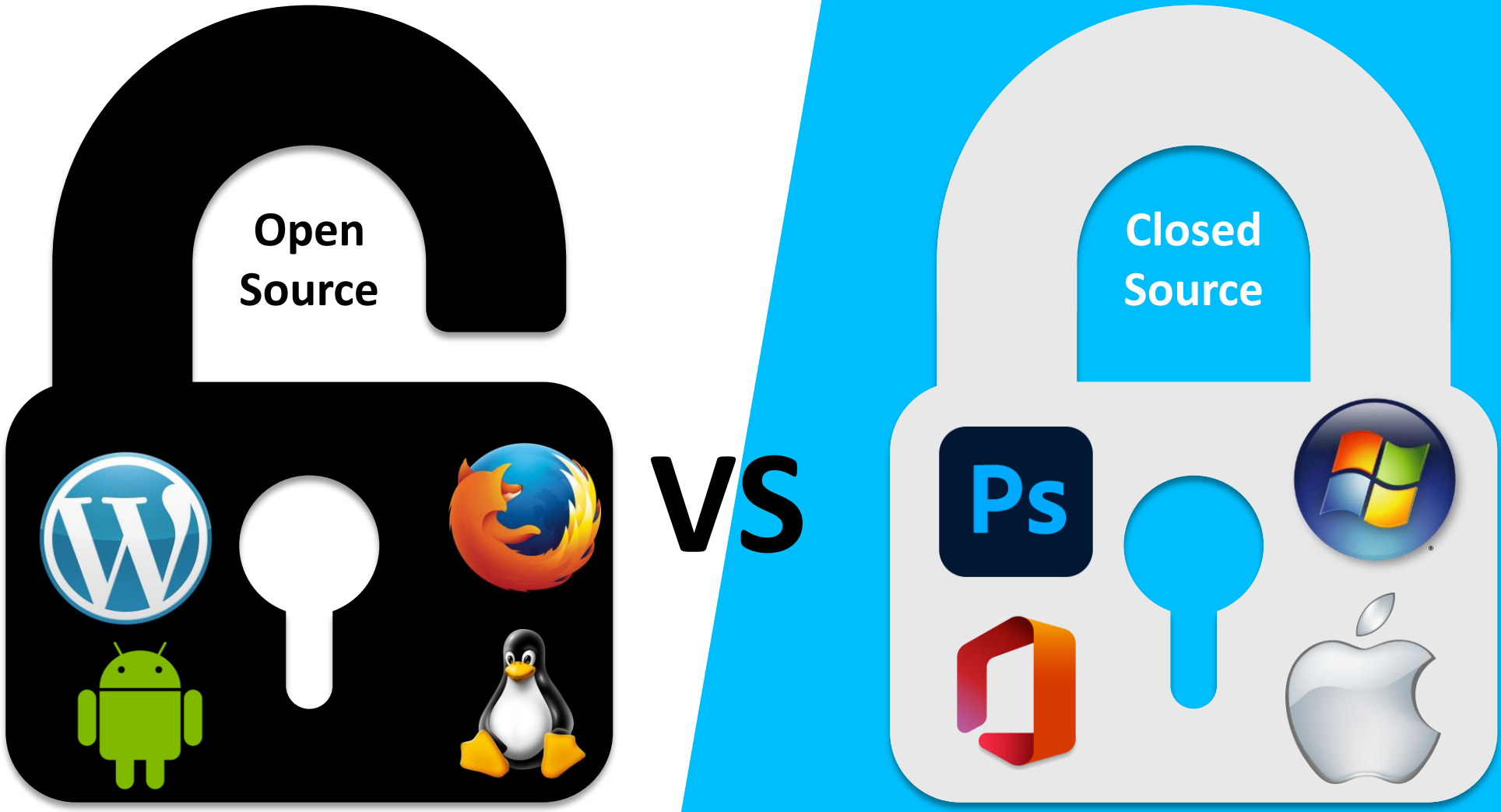
→ **Open Source**

We need to know about the Openness

→ **Software**

We need to know about the software development process

Brief Introduction of OSS

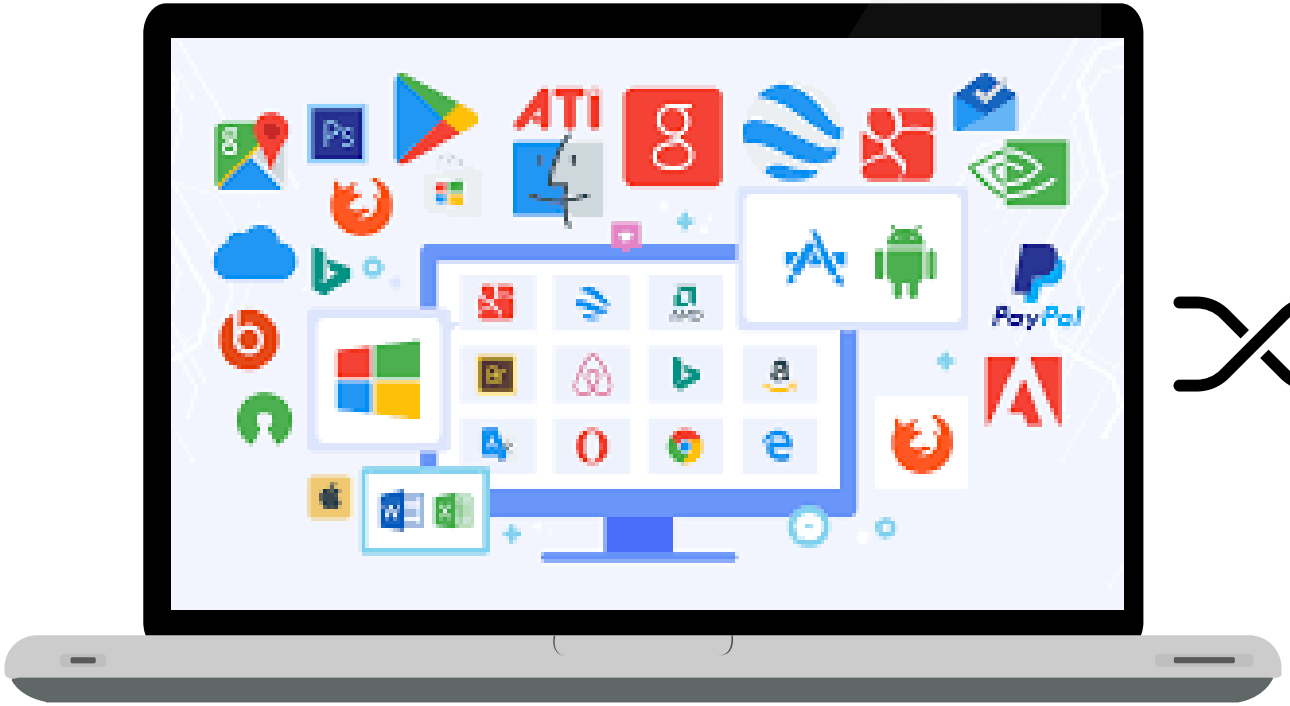


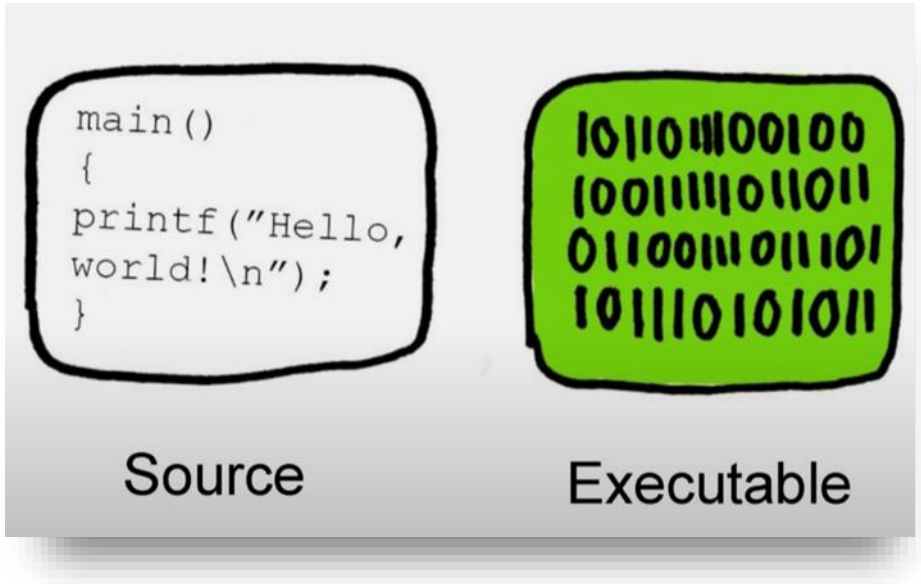
Software by license type

Software type	Free (cost)	Redistri-butable	Unlimited use and users	Source code available	Source code modifiable
Commercial (Close-source)					
Shareware	X	X			
Freeware	X	X	X		
Royalty-free libraries	X	X	X	X	
Open source	X	X	X	X	X

Source Code

The Technical blueprint that tells a program how to function





Software Release

Software Creators

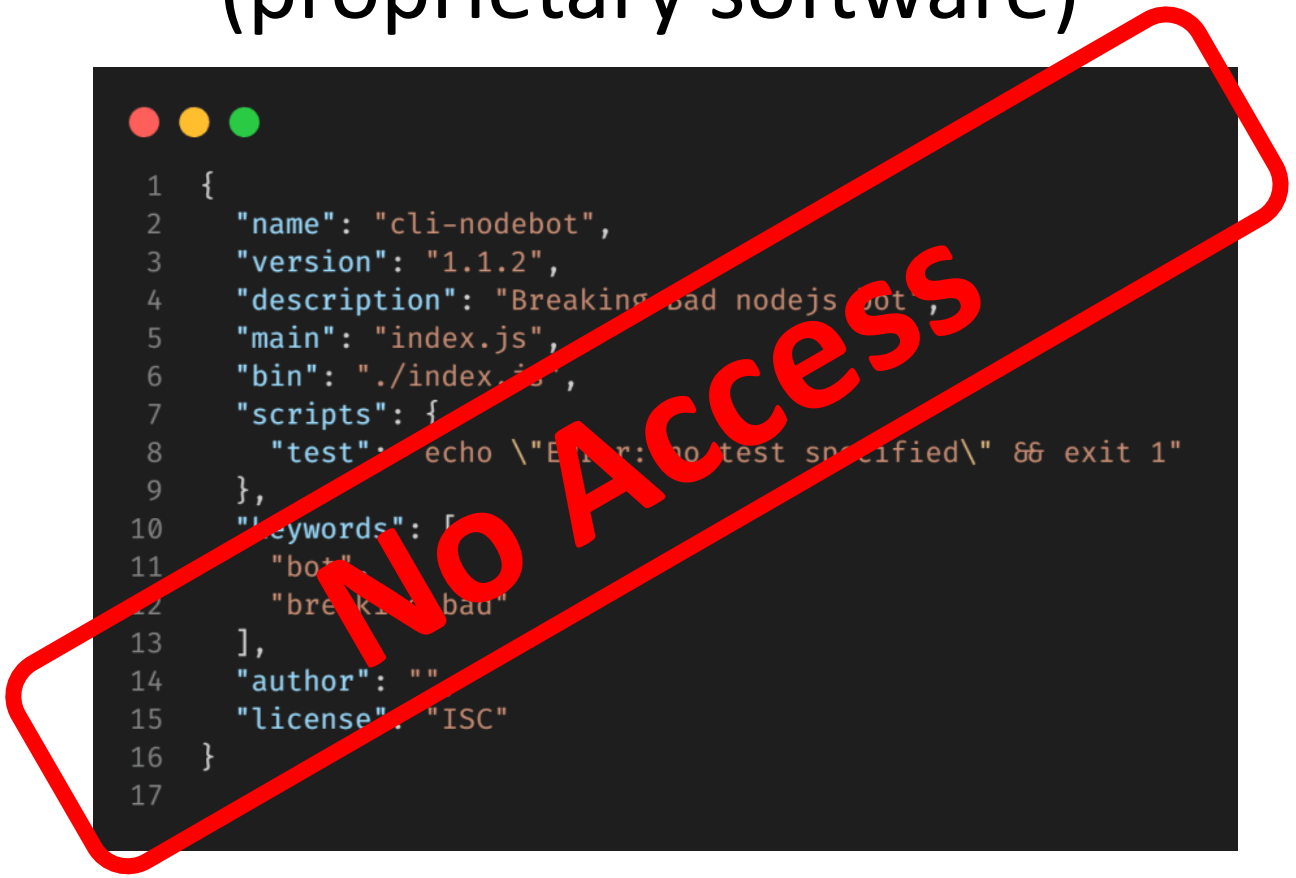
Open Source

Close Source

Close Source Software

- Closed source software, also known as

(proprietary software)

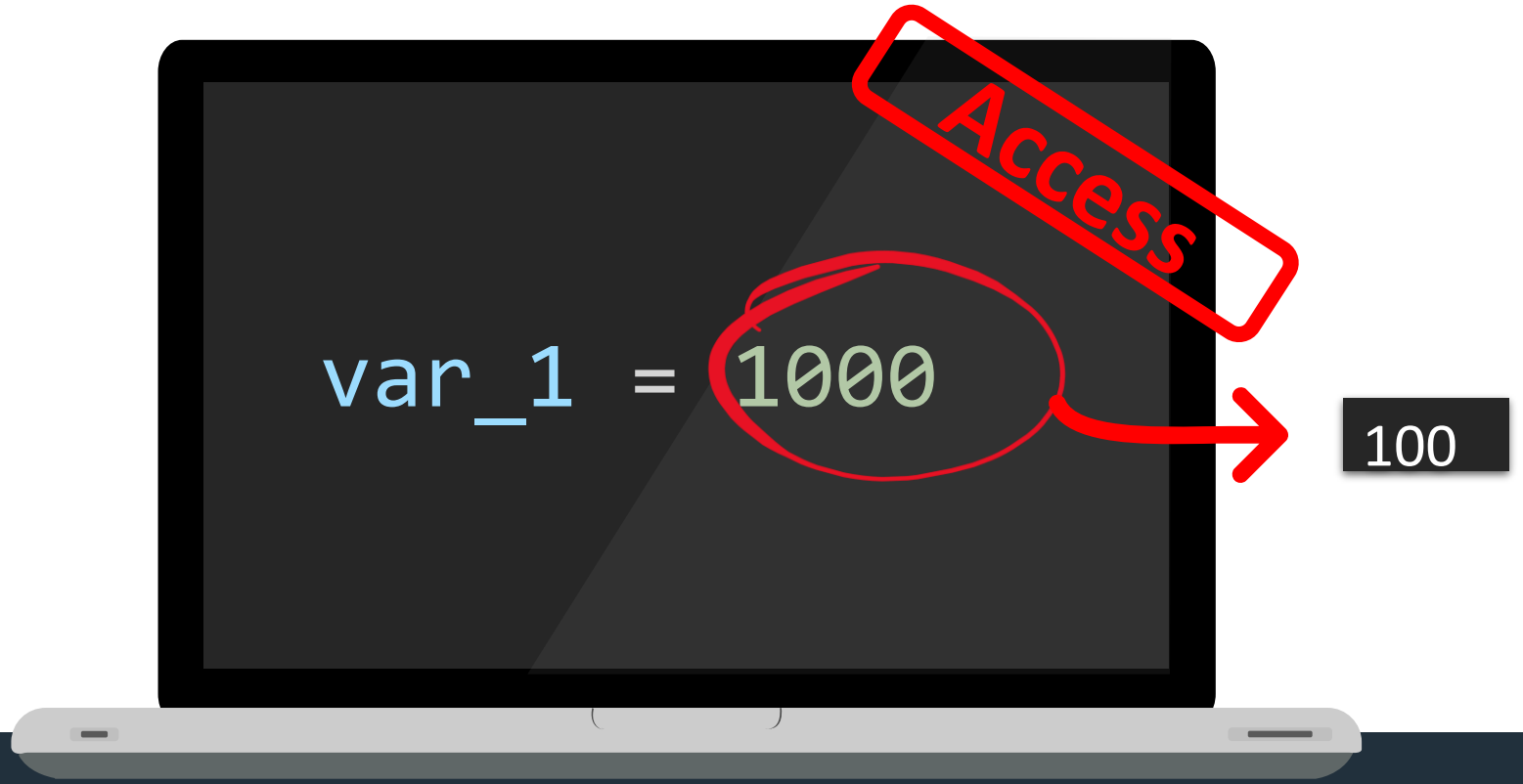


Close Source Software

What is Open Source Software?

Access to source code

Free = freedom to use, modify, copy



Open Source Is Not Just the Code

- **Beyond code availability**: Open-source software is more than just software with visible source code.
- **Community-driven**: It is inseparable from the community of contributors, users, and supporters who actively sustain it.
- **Rooted in philosophy**: It embodies the principles of sharing and freedom that shaped the **Free and Open-Source Software** movements.
- **Philosophy of openness**: At its core, open source is the belief that software should be free and accessible.
- **Collaborative methodology**: It is a way of building software collectively through collaboration.
- **Shared goals**: Contributors work together toward common, shared objectives.
- **Social structure**: Development takes place within communities and structures they create themselves.
- **Success through cooperation**: The approach is guided by the belief that collective effort ensures long-term project success.

Source: Stewart Weiss : http://www.compsci.hunter.cuny.edu/~sweiss/course_materials/csci395.86/slides/introduction.html#1

What is Open-Source Software?

- **So what does that mean?**
- Open-source software is **collectively developed** by a community of technologists who share an interest in a particular application or tool.
- It is then **distributed freely** to the broader community of individuals who can benefit from it.
- It is software that **anyone is licensed to use, copy, study, and modify** in any way.
- The **source code is openly shared**, encouraging people to **voluntarily improve and enhance** its design. *(Wikipedia)*

What is Open-Source Software?

- Free to use
- Free to change
- Free to distribute
- An alternative to commercial software

Open Source Software is Everywhere

- Free and open source software is all around us, more than you probably realize. Some examples:
  - The code that secures Internet transactions, OpenSSL;
  - The Android operating system in many smartphones;
  - The Firefox browser;
  - The Linux kernel and operating system;
  - The code that many web developers use to build web pages, such as Wordpress and Drupal.
- Look at The Octoverse 2024, a report produced by GitHub, to see just how much activity happened in 2024 surrounding free and open source software.

Source: Stewart Weiss : http://www.compsci.hunter.cuny.edu/~sweiss/course_materials/csci395.86/slides/introduction.html#1

Openness In General

- The **Free and Open-Source Software (FOSS) movement** tiled the way for a broader philosophy of **open access**, often called *the Open Source Way* or simply *Openness*.
- This philosophy extends beyond software into many domains, including:
- **Open encyclopedias**: e.g., Wikipedia
- **Open digital libraries**: e.g., Internet Archive
- **Open maps**: e.g., OpenStreetMap
- **Open data**: thousands of datasets are freely available, such as:
  - **Municipal level**: NYC Open Data
  - **State level**: New York State Open Data
  - **Federal level**: United States Open Government
  - And many more across governments, institutions, and communities.

Supporting Institutions

- There are many, many institutions that support free and open source software. Some of the major ones in the U.S. are:

The Free Software Foundation

The Open Source Initiative

The Software Freedom Conservancy

The Mozilla Foundation

The Linux Foundation

The Creative Commons

there are many others around the world

Q1: Android OS is OSS?

A True **B** False

Multiple Choice

Q2: Can you distribute the OSS Software?

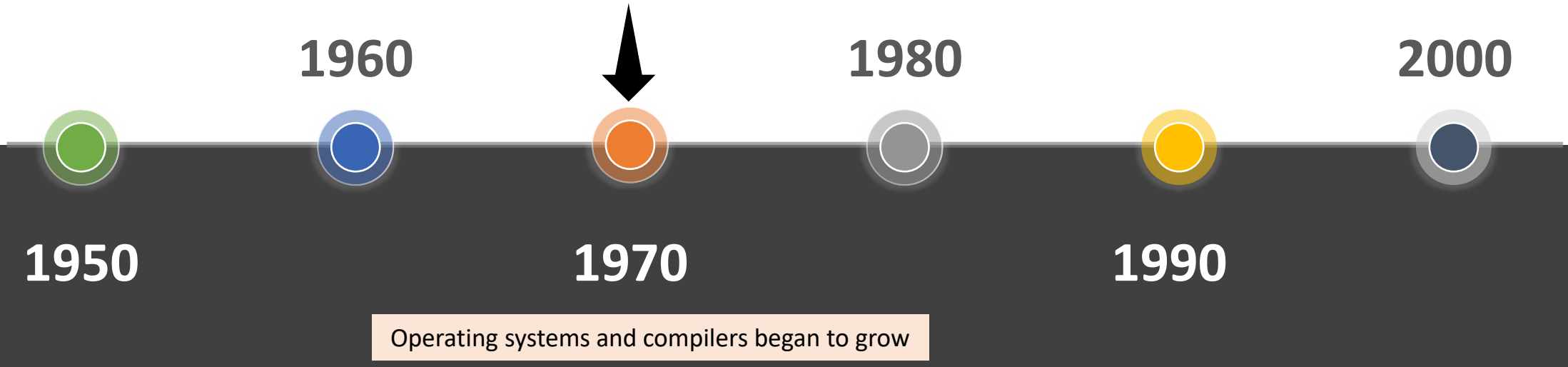
A Yes **B** No

Multiple Choice

History of Open-Source (OSS) Software

The Origins of Open Source

- The story of open-source development began before the Free Software movement.
- In the 50s and 60s, research institutes primarily produced software.



Some Historical Facts

Richard Stallman

GNU Project **1983**

FSF **1985**

Linux Torvalds

Linux (GNU tools) **1991**

GPL **1992**

Eric Raymond

OSI **1998**

1990

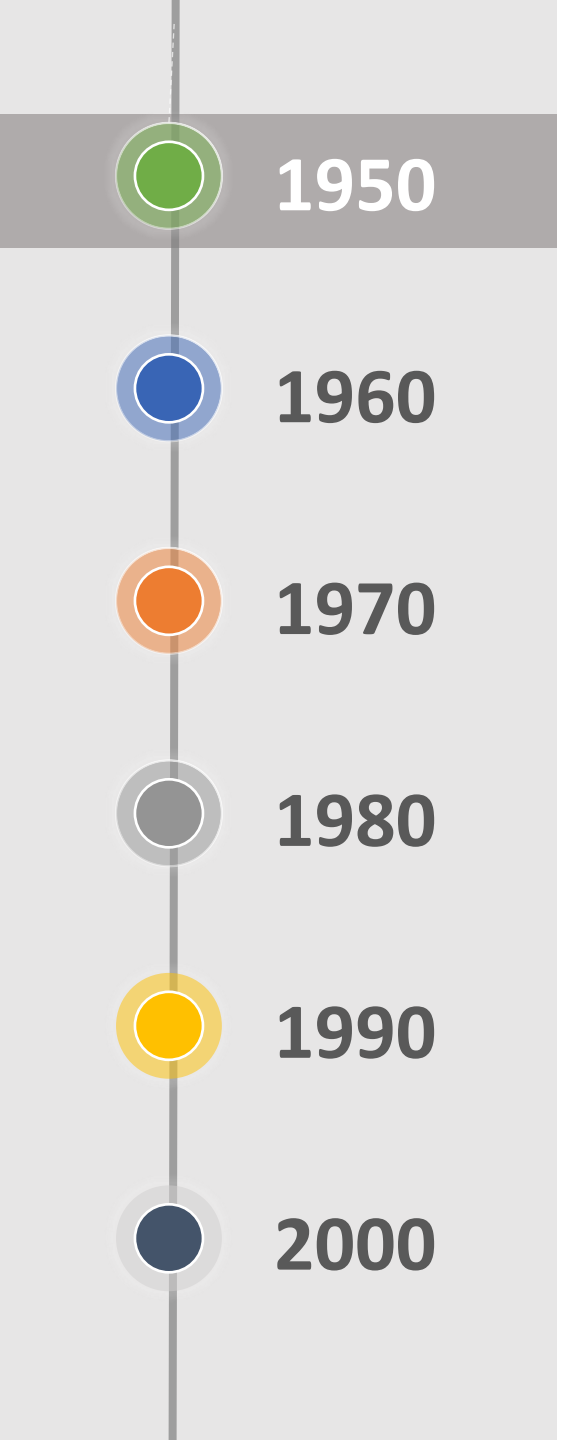
1996

2000

Xerox 9700 printer - AI lab at MIT
No source code for drivers

Linus visits aquarium, gets bitten by a penguin and chooses it as Linux mascot (Tux)

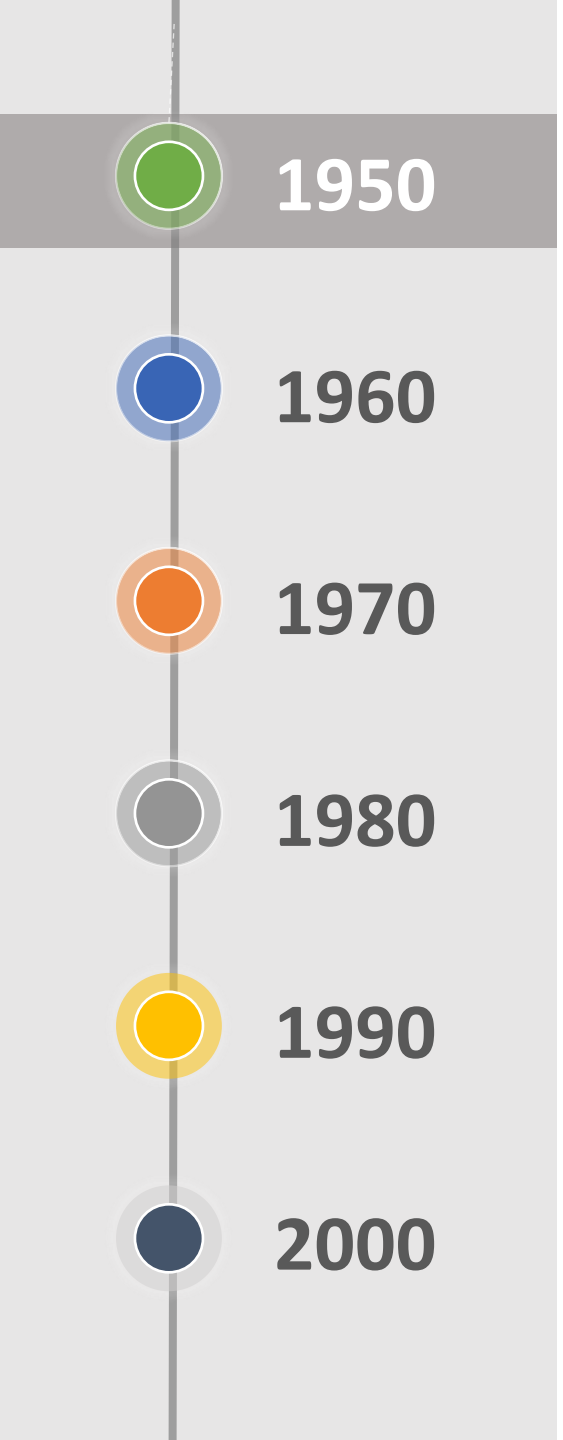
https://www.reddit.com/r/linux/comments/cqf3hf/floss_timeline_1980_2000/



History of Open-Source (OSS) Software

1950s and Earlier

- Software sharing has a long history alongside the software itself.
- Early computer manufacturers primarily emphasized hardware innovation.
- Customers, often scientists or technicians, modified and extended the provided software.
- They shared their patches not only with the manufacturer but also with other machine owners.
- Manufacturers viewed software improvements as enhancing hardware appeal to potential customers.



History of Open-Source (OSS) Software

1950s and Earlier (cont'd)

- The earliest commercial computers were shipped with their own *operating code* so that users could customize the code for their needs.
  - There were no manuals, no instructions, no help from vendors for the shipped software - **users just had to figure it all out on their own**.
  - Code-sharing was common: It was a **culture of sharing**.
- Typical hardware business model:
  - Buyers paid for hardware, the software did not matter much because it was not standardized - it worked only on the purchased hardware.
  - **Software had NO monetary value**.

1950

1960

1970

1980

1990

2000

History of Open-Source (OSS) Software

1950s and Earlier (Cont'd)

- Software arise from researcher, both academic and corporate
- Distributed openly and cooperatively
- Source always distributed, binaries less often
- Software is not seen as a separate commodity; instead bundled for free with hardware
- They didn't care about Licensing

1950

1960

1970

1980

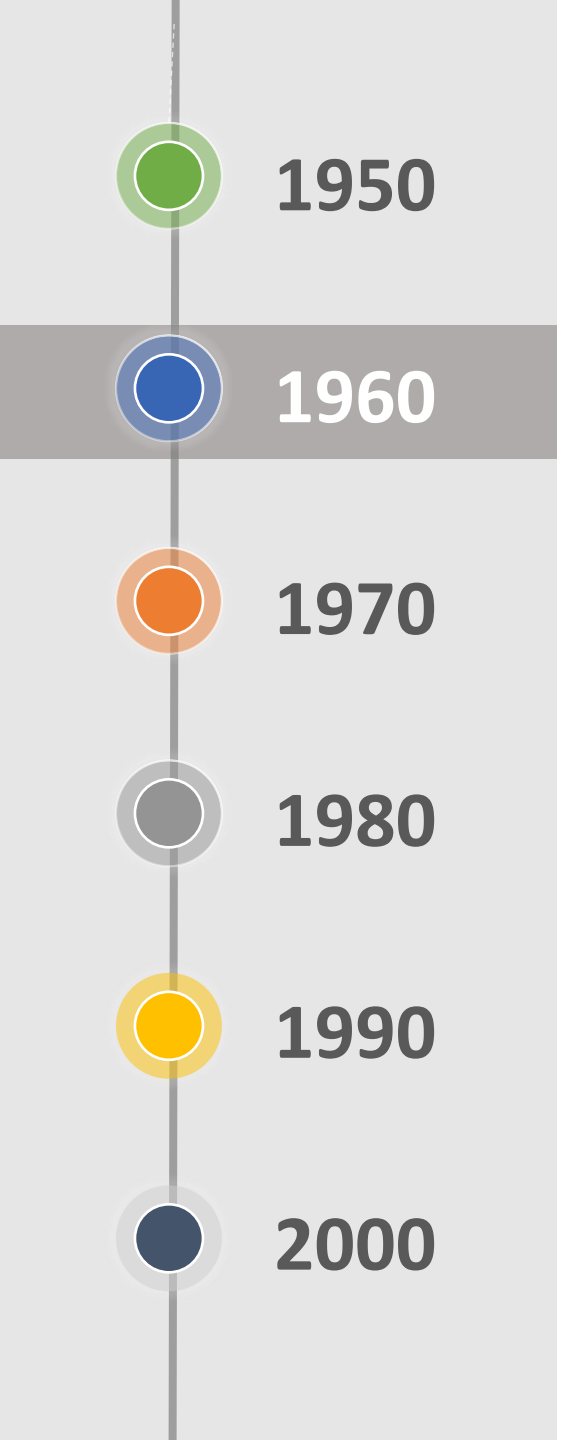
1990

2000

History of Open-Source (OSS) Software

1950s and Earlier (Cont'd)

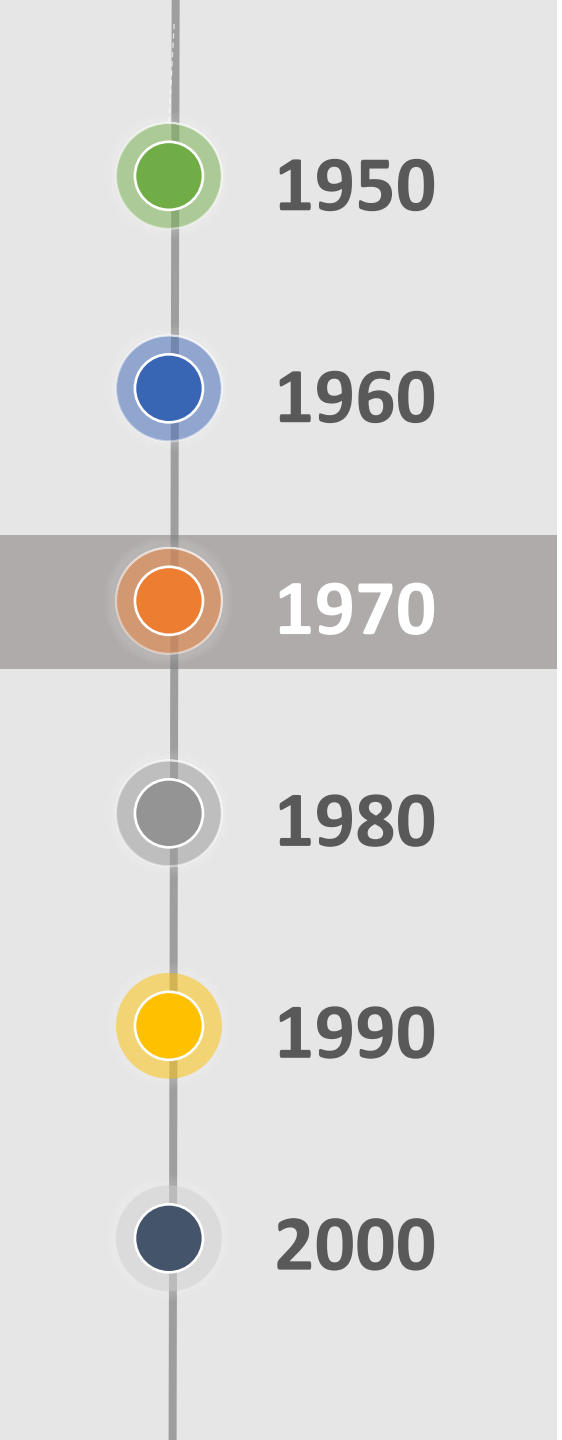
- The early period resembled today's free software culture but with two critical differences.
- Little hardware standardization;
  - diverse and incompatible computing architectures.
  - Expertise was architecture-specific, unlike today's language-oriented expertise.
  - Manufacturers promoted the spread of machine-specific code and knowledge.
- Lack of widespread Internet;
  - fewer legal restrictions but greater technical hurdles.
  - Inconvenient data transmission methods
  - limited sharing possibilities.
  - Overcoming barriers through local networks, mail, and university collaborations.



History of Open-Source (OSS) Software

1960s

- Major aspects of computer science and software rapidly developed both in academia (MIT, UC Berkeley) and industrial research labs (Bell Labs, Xerox)
- 1968: ARPANET emerges
- 1969: UNIX given birth at Bell Labs (AT&T); IBM forced to break software and hardware apart and sell/distribute separately



History of Open-Source (OSS) Software

1970s - Free Software's Origins

- During the 1970's software became a commodity: companies started selling it for profit.
- Code became proprietary and closed.
- The era of personal computers (PCs) arrived. Almost all are shipped with proprietary, closed code.
- 1976: emaces released (GNU emacs not released until 1985)
- 1978: First version of TeX released (still in widespread use, usually in the LaTeX version)

1950

1960

1970

1980

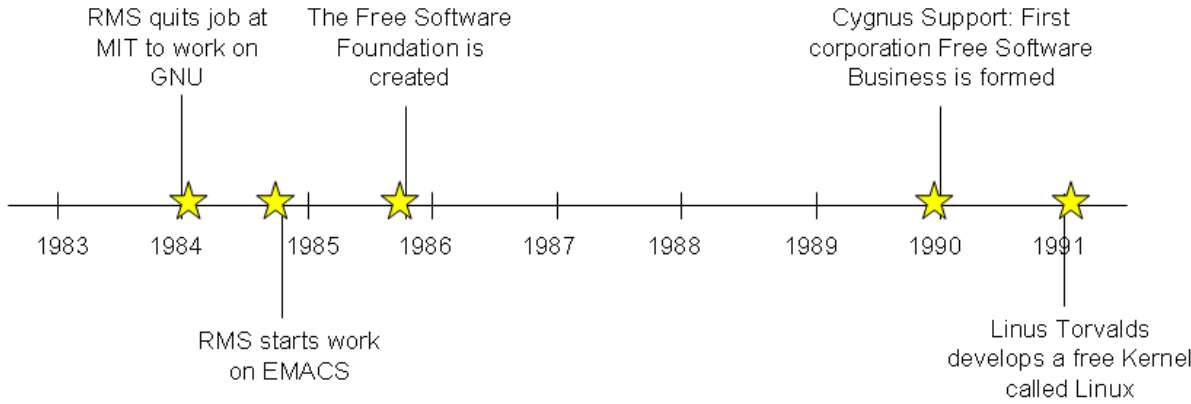
1990

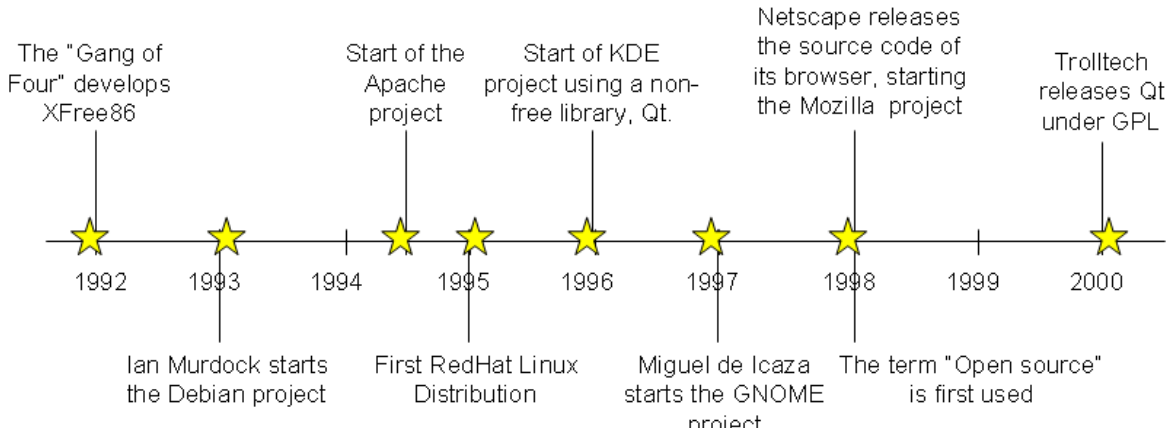
2000

History of Open-Source (OSS) Software

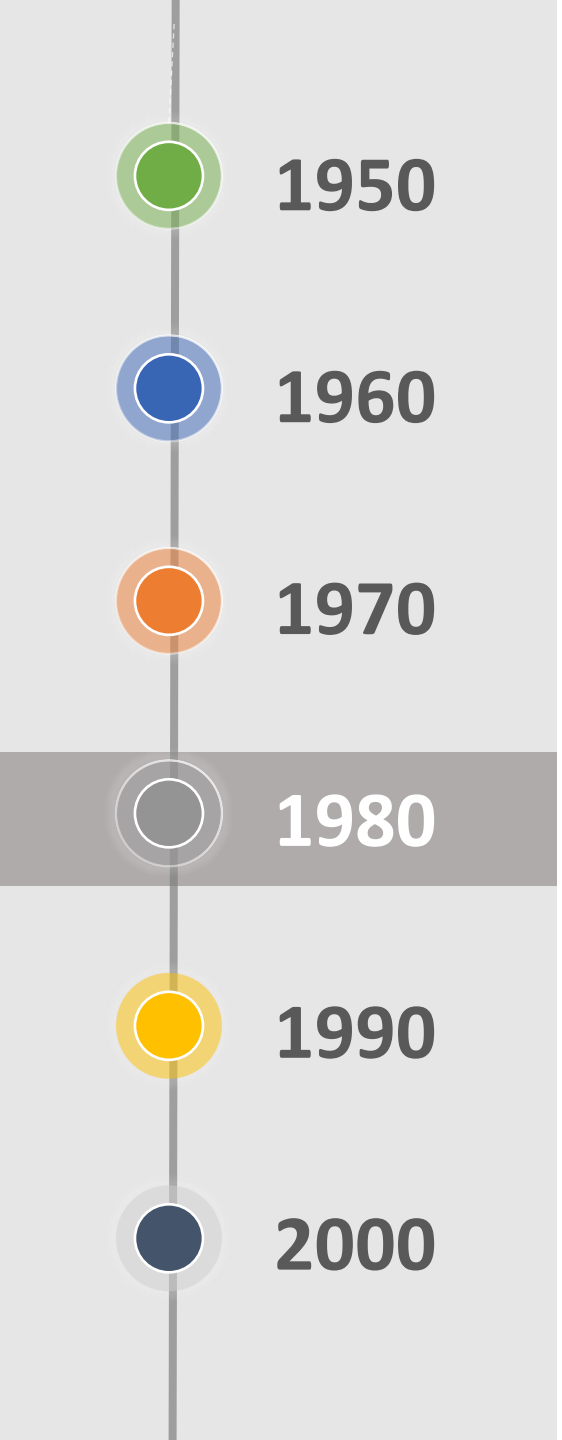
1980s - Evolution of Open-Source development

the evolution of open source development from the creation in 1986 of the Free Software Foundation by Richard Stallman (who likes to use his initials RMS). After this foundation was established, several major open source projects were initiated as shown



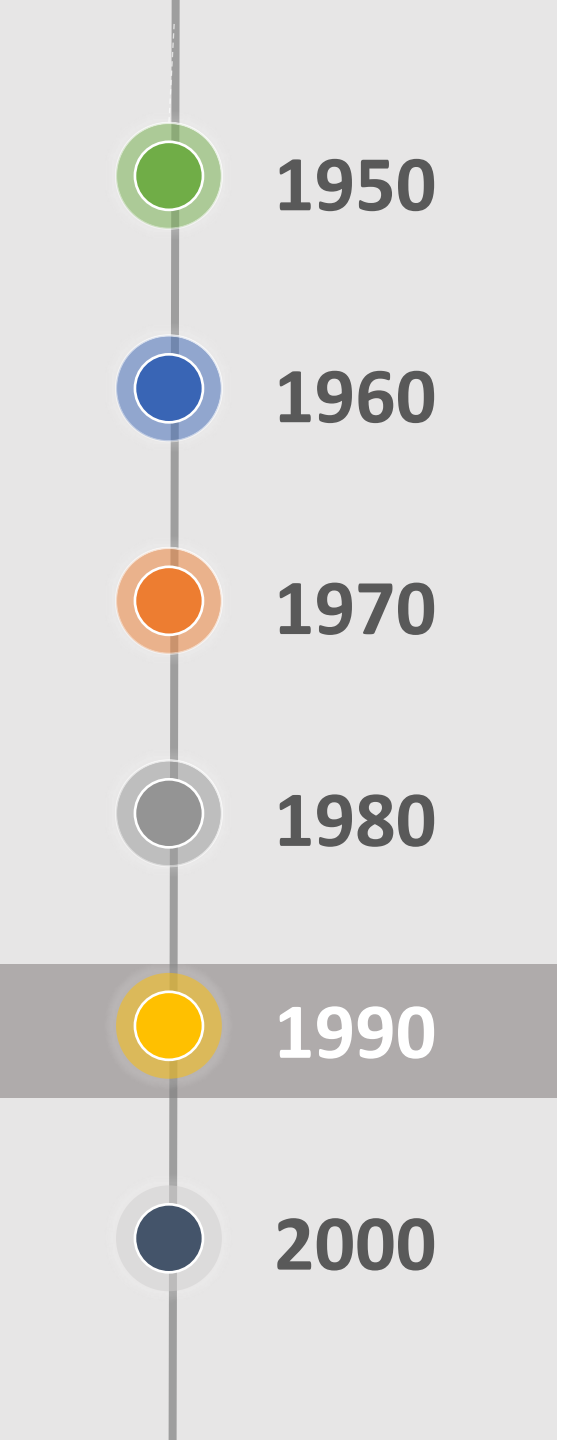


History of Open-Source (OSS) Software



1980s

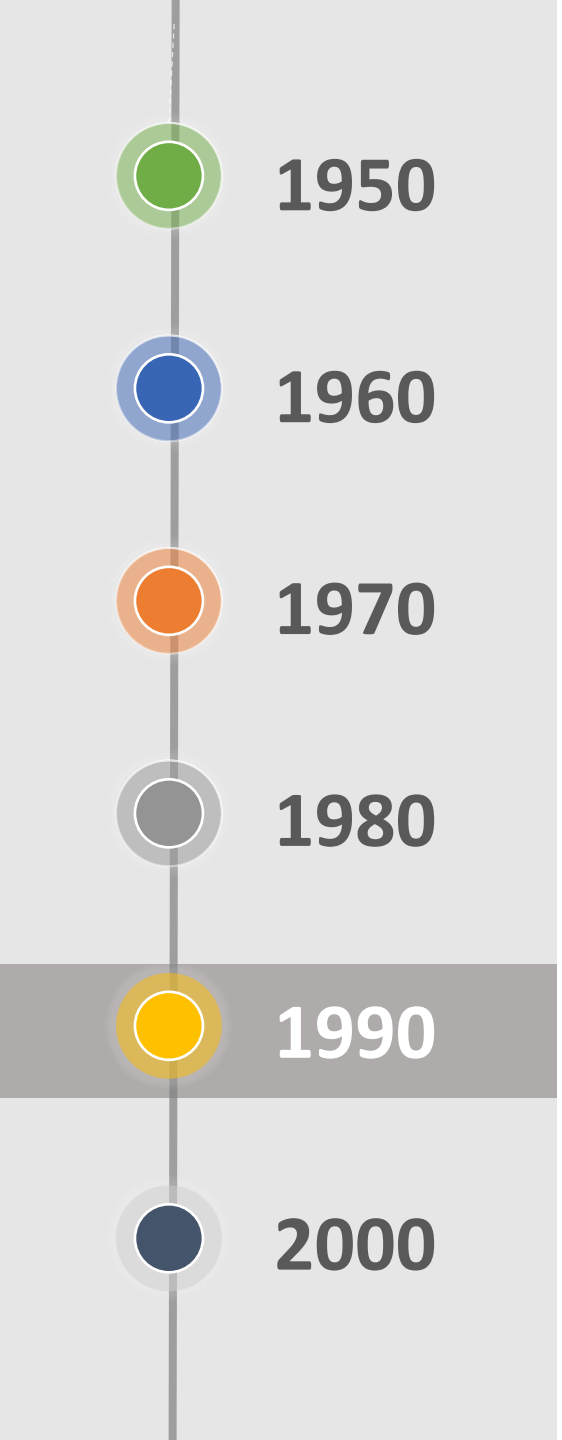
- **1980**: Usenet begins as the ancestor of user forums and the World Wide Web
- **1982:** GNU project announced
- **1984:** X Window System released (X11 protocol released in 1987)
- **1985:** Free Software Foundation (FSF) founded
- **1987: GCC** (now known as GNU C Compiler) and Perl released



History of Open-Source (OSS) Software

1990s

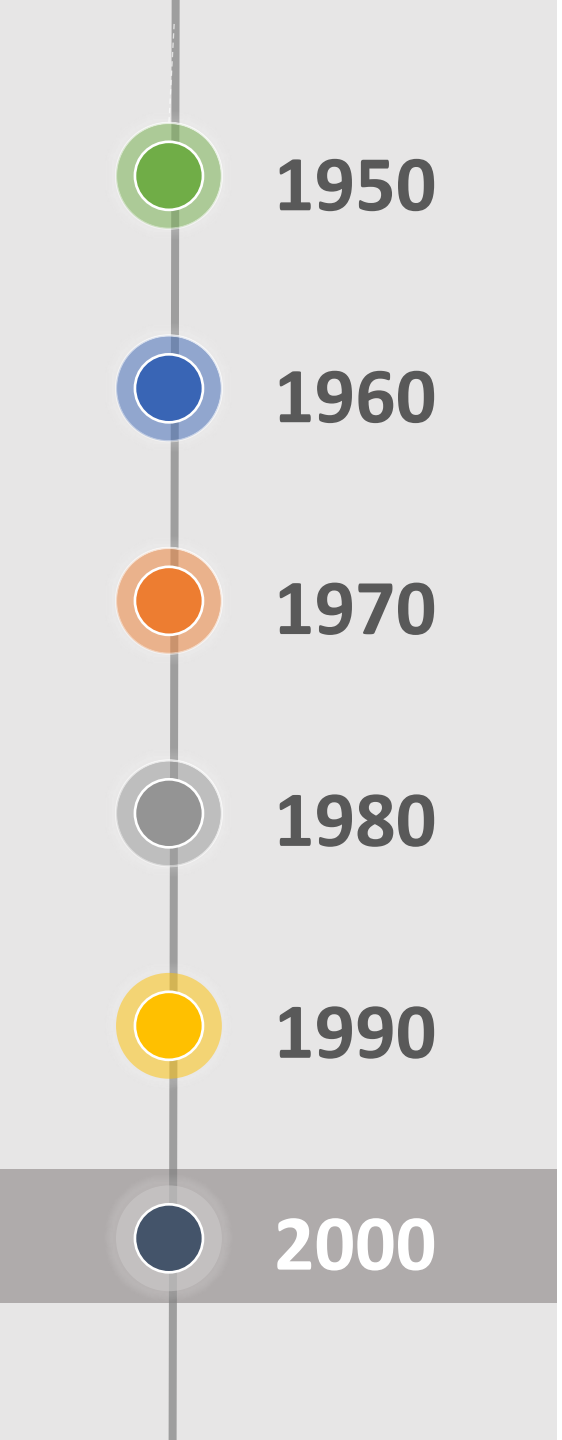
- **1991:** Linux was begun by Linus Torvalds
- **1992:** Python and 386BSD released; Samba developed
- **1993:** Debian (still the largest non-commercial Linux distribution), NetBSD, FreeBSD, and Wine released; Red Hat is founded
- **1994:** MySQL development begins (first release in 1995)



History of Open-Source (OSS) Software

1990s (Cont.)

- **1995:** PHP, GIMP, and Ruby released
- **1996:** Apache web server and KDE released
- **1997:** GNOME released
- **1998**: Netscape open sources its browser (later becomes Firefox)
- **1999:** OpenOffice released (eventually forks into LibreOffice)



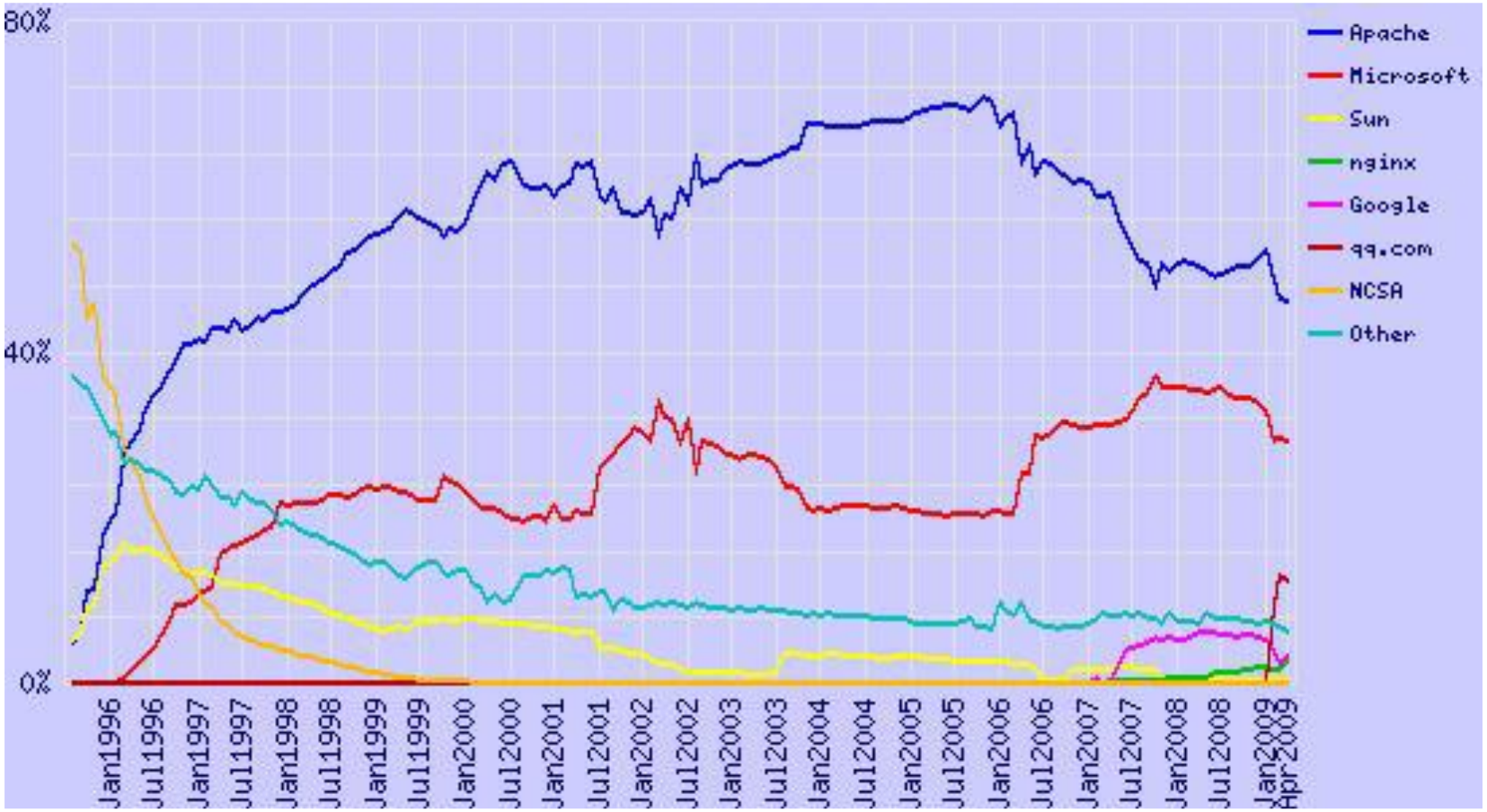
History of Open-Source (OSS) Software

2000s

- **2000:** LLVM compiler project begun
- **2002:** Blender released as an open-source project
- **2003**: Firefox released
- **2004:** Ubuntu released
- **2005:** Git released by Linus Torvalds
- **2007:** Android released, based on Linux kernel; first devices on the market in 2008
- **2008:** Chromium released by Google; the basis of Google Chrome

History of Open-Source (OSS) Software

Market share for top Web servers across all domains according to Netcraft (http://www.netcraft.com)



Clarifying "Open Source" vs. "Free Software"

- Open Source vs. Free Software Debate
  - Richard Stallman and similar thinkers criticize "open source" for not emphasizing user freedom.
  - Stallman's perspective: "Free software" = Freedom, not just price.
  - Confusion: "Free software" is often misunderstood as no-cost software.
- Introducing FLOSS: Free, Libre, Open-Source Software
  - Three terms explain the concept:
    - Free software: Focuses on liberty, not price; can be used, studied, and modified without restrictions.
    - Libre Software: Emphasizes freedom as a fundamental right, not zero-cost
    - Open Source: Emphasizes freely available source code.

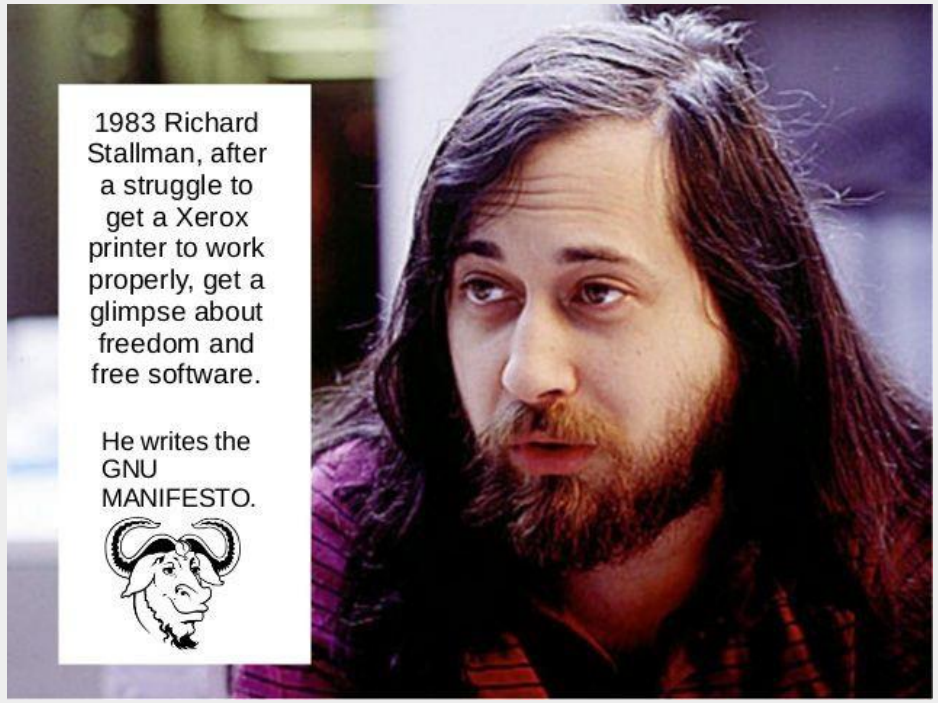
Advantages and disadvantages of open source

- Advantages of Open Source Software
  - **Cost-Free Access:** Open source software is freely available to everyone.
  - **Transparency:** The source code is open and can be viewed by anyone.
  - **Collaboration:** A global community of developers can contribute and improve the software.
  - **Customizability:** Users can modify the software to meet their specific needs.
  - **Security:** The collective effort often results in quicker identification and resolution of vulnerabilities.
- Challenges of Nonprofit Community Development
  - **Lack of Central Authority**: The absence of a single entity can lead to coordination challenges.
  - **Funding Issues**: Relying on donations and volunteers may limit resources for development.
  - **Quality Control**: Maintaining software quality can be a concern without strict oversight.
  - **Fragmentation**: Diverse contributions can result in varying versions and compatibility issues.
  - **Support and Documentation:** This may be less comprehensive compared to commercial software.
  - **Desktop Adoption Challenges:** Open source software has primarily focused on servers, leading to slower desktop adoption.

Open Source Trends and Perspectives

- Increasing Adoption:
  - Companies favor open source for development and testing.
  - Open source gaining ground in production.
- LAMP Stack Dominance:
  - LAMP (Linux, Apache, MySQL, PHP/Perl/Python) is widely used for web applications.
- Sustained Projects:
  - Many '80s-era open source projects thriving.
  - Example: Eclipse IDE.
- Open Source-Centric Companies:
  - Businesses offer integrated solutions based on open source.
  - Reflects open source community strength.

GNU and the Free Software Foundation (FSF)



Gothfather of Free Software Foundation

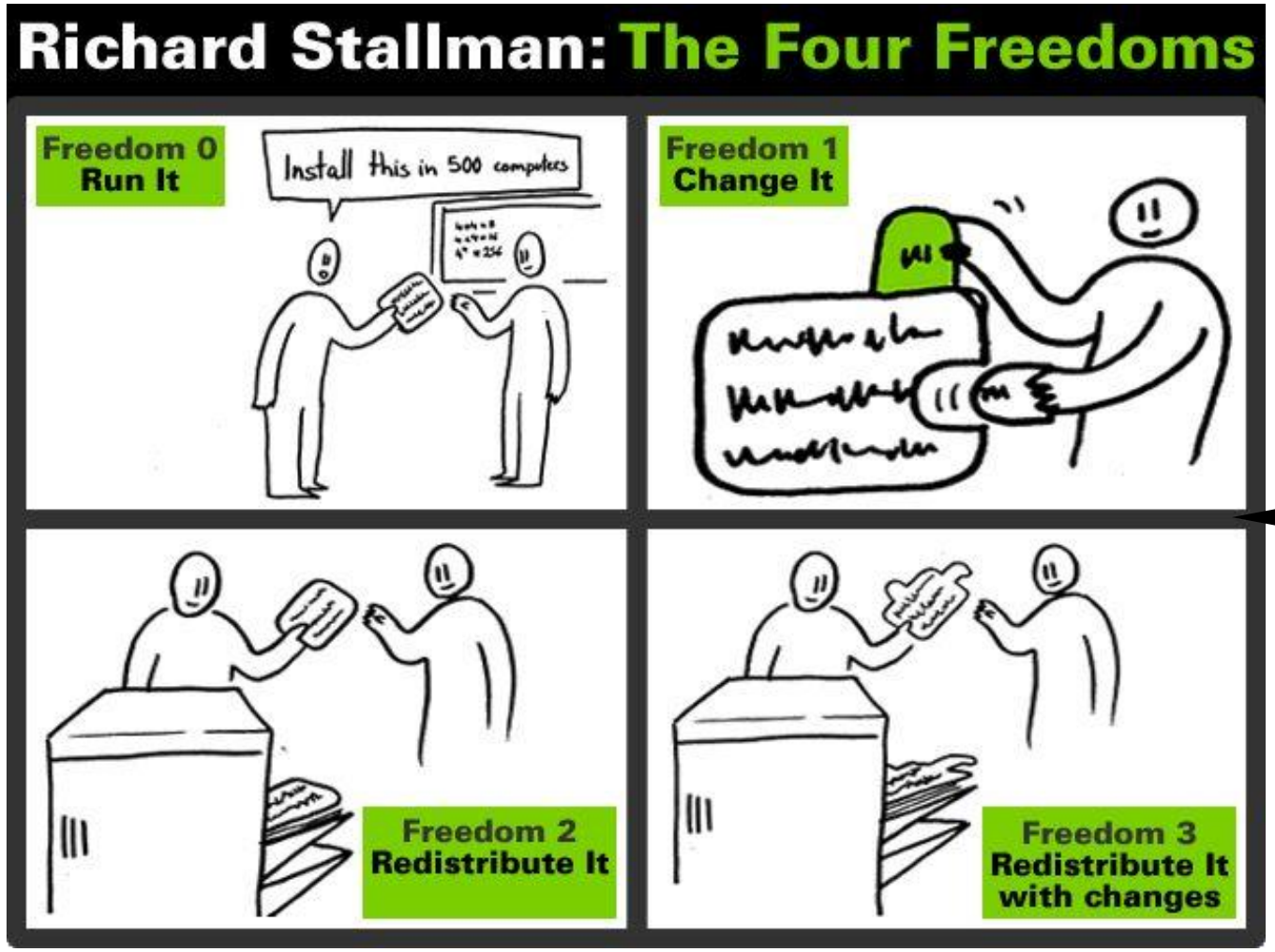
GNU Project in 1983, GNU Manifesto in 1985

Starting this Thanksgiving

"I am going to write a complete Unix-compatible software system called GNU (for Gnu's Not Unix), and give it away free to everyone who can use it"

He defined the Four Freedoms as the core tenets of the Free Software Movement (purposely starting with number zero)

Free Software Foundation - Freedoms explained



Any software that does not guarantee these freedoms to its users cannot be considered “free” because it limits the users’ rights in some way.

Software Licenses and the GPL

- Richard and the FSF invented a class of software licenses that turned copyright on its head, and they named them copyleft licenses.

- Copyleft software licenses ensure that software can never violate the Four Freedoms.

- The invention of these licenses is probably his most significant contribution to software; it really smooth the way for the open-source movement that followed.

The Open Source Movement's Origins

- On January 22, 1998, Netscape released its browser Netscape Navigator together with its source code, drawing attention of businesses.[2]
- Although interested in the concept, businesses were not keen on the politics of the FSF and the Free Software movement.
- In February 1998, Christine Peterson suggested "open source" to replace "free" to make the idea more appealing to businesses.
- Later that month the Open Source Initiative (OSI) was created by a group of advocates (Bruce Perens and Eric S. Raymond), with the mission of explaining and protecting the "open source label".

2 https://blog.lizardwrangler.com/2008/01/22/january-22-1998-the-beginning-of-mozilla/

The Open Source Definition

- OSI created a definition of open source software that was more detailed than Stallman's Free Software Definition, and called it the Open Source Definition (OSD).

- The OSD defines valid software licenses.
- The software license is what determines whether software is open source!

The two families

Free Software Foundation	*Open Source Initiative*
"In 1983, Richard Stallman published the GNU Manifesto and launched the GNU Project to write a complete operating system free from constraints on use of its source code."	*"The organization was founded in February 1998 by Bruce Perens and Eric S. Raymond, part of a group inspired by the Netscape Communications Corporation..."*

Free Software vs Open Source

Free Software Foundation

"In 1983, Richard Stallman published the GNU Manifesto and launched the GNU Project to write a complete operating system free from constraints on use of its source code."

FREE SOFTWARE FOUNDATION
www.fsf.org

Open Source Initiative

"The organization was founded in February 1998 by Bruce Perens and Eric S. Raymond, part of a group inspired by the Netscape Communications Corporation..."

open source initiative

The two families and their own definition of open

1. *The freedom to run the program as you wish, for any purpose .*
2. *The freedom to study how the program works, and change it so it does your computing as you wish.*
3. *The freedom to redistribute copies so you can help your neighbor.*
4. *The freedom to distribute copies of your modified versions to others.*

By doing this you can give the whole community a chance to benefit from your changes. Access to the source code is a precondition for this.

1. *Free Redistribution*
2. *Source Code Availability*
3. *Derived Works*
4. *Integrity of Author's Source Code*
5. *No Discrimination Against Persons or Groups*
6. *No Discrimination against Fields of Endeavor*
7. *Same Distribution License*
8. *License not Specific to a Product*
9. *License Must Not Restrict Other Software*
10. *License Must be Technology Neutral*

Free Software vs Open Source

Free Software Foundation

Free Software Foundation (FSF)

Founded in 1985 by

Richard Stallman

Non-profit organization

Defend the rights of all software users

This is a social movement. They are ”software activists”.

Free software is software that ensures the user’s freedoms

Run, Study, Share, Modify

Open Source Initiative

Open Source Initiative (OSI)

Founded in 1998 by

Eric S. Raymond and Bruce Perens

Non-profit organization

Educates about and defends open source

Promotes this model of collaboration for companies.

Open Source is what complies with the OSD

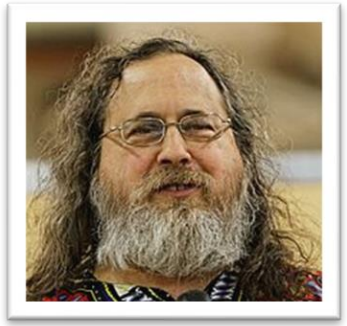
Similar benefits but less restrictions

Free Software vs Open Source

Free Software Foundation

Free Software Foundation (FSF)

Founder: ***Richard Stallman***



Created GNU Project

Activist, software hippie

"A proprietary program puts its developer or owner in a position of power over its users. This power is in itself an injustice."

FSF website

Open Source Initiative

Open Source Initiative (OSI)

Co-founder: ***Eric S. Raymond***

Released Netscape's source code

He also defends proprietary software

"I think that if a programmer wants to write a program and sell it, it's neither my business nor anyone else's but his customer's…"

Essay in 2008

10 criteria for OSS

1. Free redistribution
   - No restriction on redistribution (free or paid) of the software
   - The software can be redistributed alone or as a component of an aggregate software distribution
2. Source Code
   - The software must include the source code, or it must be easily obtainable through well-published means
   - The source code should not be deliberately obfuscated, and intermediate forms (preprocessor/translator output) are not allowed.
3. Derived works
   - The license must allow modification and derived works
   - Derived works must be allowed to be distributed under the same terms as the original

10 criteria for OSS

4. Integrity of the author's source code
   - The license may only restrict source-code from being distributed in modified form if it allows patch-files that can modify it at build time
   - The license might require derived works to use a different name and/or version number
5. No discrimination against persons or groups
6. No discrimination against fields/domain
7. Distribution of license
   - License for software must also apply to those it is redistributed to

10 criteria for OSS

8. License for software must not depend on it being part of a software distribution
   - The same license must apply if the software is extracted from a distribution and distributed separately
9. License must not restrict other software
   - The license must not place restrictions on other software that is distributed alongside the licensed software
10. License must be technology-neutral
   - No provision of the license may be predicated on any individual technology or style of interface

10 criteria for OSS

- If the license does not meet the definition, it is not open source!

- If there is No license, it is not open source!
  - (It is code available, not open source)

Free vs Open Source

- There was and still is much controversy about Free versus Open Source software.
- The difference is mostly philosophical:
  - The Four Freedoms and Free software express an ideology and a moral point.
  - The OSI OSD expresses a practical system for supporting the expansion and growth of open source, acceptable to business interests.
- Many people use the term **Free and Open Source Software** (**FOSS**) to avoid the contention. We do the same.

Reading Materials

- **Books:**
  1. VM Brasseur, *Forge Your Future with Open Source*, The Pragmatic Programmers, LLC. 2018.
  2. Karl Fogel, *Producing Open Source Software: How to Run a Successful Free Software Project*, O'Reilly Media, 2009.
- https://www.gnu.org/philosophy/free-sw.html
- https://blog.lizardwrangler.com/2008/01/22/january-22-1998-the-beginning-of-mozilla
- https://opensource.org/licenses
- https://www.coredna.com/blogs/comparing-open-closed-source-software
- https://linkedretail.com/open-vs-closed-source/
- https://www.assignmentprime.com/blog/open-vs-closed-source-software
- https://www.gnu.org/philosophy/free-sw.en.html
- https://opensource.org/osd
- https://pavanganeshbutha1998.blogspot.com/2019/10/free-software-philosphy.html
- https://www.youtube.com/watch?v=Ag1AKIl_2GM&t=89s

Attribution

- Some Slides are copied from **Prof. Stewart Weiss Lectures :** http://www.compsci.hunter.cuny.edu/~sweiss/course_materials/csci395.86/slides/introduction.html#1

Thanks

Office Time: Monday-Friday (1000 - 1800)

You can send me an email for meeting, or any sort of discussion related to class matters.

jamil@sejong.ac.kr