# GREEN TECHNOLOGY

## About:

Going green is not an option but a necessity for humanity continuous existence. No one ever runs from his/her shadow. However, with careful lighting selection we can eliminate it in a controlled-knowledge environment. Going green is about knowledge-based utilization of God’s nature material gift for our sustainability and preservation using environmentally friendly technology and processes. The environment responds to human ignorance; setting off uncontrollable often-immediate inconsequential processes that eventually changes the course of our existence. Necessity has placed on us the role to promote green education to ensure the survivability of the nation economy and ourselves in particular.

## Definition:

Understanding the knowledge, you have to challenge and change the discomfort you feel, propels good and bad innovation. Hard time requires not only knowledgeable policy but also willing spirit character that is ready to bear pain for new products development.

## Introduction:

It takes more energy to live what you were not design to be. Most of man early technological inventions had low efficiency. They were welcomed since they meet our quest for easier life style. They soon fade as the aesthetics, economic and environmental burden of keeping entirely that of technical and technological necessities but the consequent effect on visibleiog environs’ ecosystem and decision of man for his personal sustenance. Man ultimately decides the fate of what goes on around. Man decides the quality of his existence by the extent he can go in dominating the environment.

The greatest of man is not the materials he has but the grace to rightly divide the knowledge sustaining it. Sustain-ability is navigating using the best of our character and knowledge, life at the mountaintop and valley, good and bad times alike. From the very beginning, man knew he must develop the right attitude to sustain continuous existence on earth. We have nature instinct to identify personal and societal risk. However, we do not dominate and sustain the products without revelation knowledge. We can only rise above the present by developing the right character and technology to make life more meaningful. Failures are rarely acknowledge.

The most important and sustaining part of man is his spirit side. You can do all things when your spirit man puts on the desired character. Knowledge and character are not always compatible. You will hurt others and yourself when you have more knowledge than your character can bear. When we become obsess with unquenchable desire to be rich by all means, we develop foolish character with enabling policy and technology to support it. Today we spend more on defensive devices as evil knowledge and character reign. The spirit is covered by darkness.

## Basic Concept of Green Computing:

The concept of green is in the heart of creation. Everything around us is predominantly green. We depend on the green herbs, grass, trees for life. Ensuring their existence is a divine mandate as the quality of our existence depends on the management of the green around us. In fact, our existence centers on the comfort we can extra from the relationship and the devices applied in cultivating and processing the green.

Green vegetation is an epitome of resilience. They always come back except we determinedly decide to fight them. Their sustainable is in their rapid multiplication and adaptation: Nature instinct to respond with season change. We mimic this when we talk about green technology. Green Technology therefore is knowledge based interactive regulated processing of materials for our continuous comfortable existence. Necessity lays it in our heart to think, conceptualize and develop innovative ideas for new product either for monetary reward through introduction of innovative conservation ideas or selfless service for protection of the environment. Green technology is about man, knowledge, formulated philosophy application interaction with material processing and the consequences of the products in the present and future life.

## Basic Motivating Criteria to Go Green:

• Strong compassion to see humanity and environment mutually sustaining each other.

• Strong knowledge base; multidisciplinary approach to research methodology

• Strong political will to bear the pain of changing from the assumptive idea that the environment can sustain itself. It is interesting to note that world leaders have not agreed on generally acceptable protocol to mitigate the effects of greenhouse gases emission.

• Thing are changing and old unregulated ideas are incapable of supporting rising human population; they must accommodate science-based deductions. Human technology applications effects over the years have change nature ecological equilibrium. There is now the necessity to create new green conscious innovation services and technology to ensure the future we desire.

• Spirally energy cost with the devastating environmental consequences of hydrocarbon fuel use.

# FREE AND OPEN SOURCE SOFTWARE (FOSS)

## About:

FOSS is [software](https://en.wikipedia.org/wiki/Software) that can be classified as both [free software](https://en.wikipedia.org/wiki/Free_software) and [open-source software](https://en.wikipedia.org/wiki/Open-source_software). That is, anyone is [freely licensed](https://en.wikipedia.org/wiki/Free_software_license) to use, copy, study, and change the software in any way, and the [source code](https://en.wikipedia.org/wiki/Source_code) is openly shared so that people are encouraged to voluntarily improve the design of the software. This is in contrast to [proprietary software](https://en.wikipedia.org/wiki/Proprietary_software), where the software is under restrictive [copyright](https://en.wikipedia.org/wiki/Copyright) [licensing](https://en.wikipedia.org/wiki/Licensing) and the source code is usually hidden from the users.

FOSS maintains the software user's civil liberty rights (see the [Four Essential Freedoms](https://en.wikipedia.org/wiki/Free_and_open-source_software#Four_essential_freedoms_of_Free_Software), below). Other benefits of using FOSS can include decreased software costs, increased [security](https://en.wikipedia.org/wiki/Security_(computing)) and stability (especially in regard to [malware](https://en.wikipedia.org/wiki/Malware)), protecting [privacy](https://en.wikipedia.org/wiki/Privacy), education, and giving users more control over their own hardware. Free and open-source operating systems such as [Linux](https://en.wikipedia.org/wiki/Linux) and descendants of [BSD](https://en.wikipedia.org/wiki/BSD) are widely utilized today, powering millions of [servers](https://en.wikipedia.org/wiki/Server_(computing)), [desktops](https://en.wikipedia.org/wiki/Desktop_computer), smartphones (e.g. [Android](https://en.wikipedia.org/wiki/Android_(operating_system))), and other devices. [Free-software licenses](https://en.wikipedia.org/wiki/Free-software_license) and [open-source licenses](https://en.wikipedia.org/wiki/Open-source_license) are used by [many software packages](https://en.wikipedia.org/wiki/List_of_open-source_software_packages). The [free-software movement](https://en.wikipedia.org/wiki/Free-software_movement) and the [open-source software movement](https://en.wikipedia.org/wiki/Open-source_software_movement) are [online social movements](https://en.wikipedia.org/wiki/Online_social_movement) behind widespread production and adoption of FOSS.

## History:

In the 1950s through the 1980s, it was common for computer users to have the source code for all programs they used, and the permission and ability to modify it for their own use. [Software](https://en.wikipedia.org/wiki/Software), including source code, was commonly shared by individuals who used computers, often as [public domain software](https://en.wikipedia.org/wiki/Public_domain_software).[[18]](https://en.wikipedia.org/wiki/Free_and_open-source_software#cite_note-infoworld1983-19) Most companies had a business model based on [hardware](https://en.wikipedia.org/wiki/Computer_hardware) sales, and provided or [bundled software](https://en.wikipedia.org/wiki/Bundled_software) with hardware, free of charge.

By the late 1960s, the prevailing business model around software was changing. A growing and evolving software industry was competing with the hardware manufacturer's bundled software products; rather than funding software development from hardware revenue, these new companies were selling software directly. Leased machines required software support while providing no revenue for software, and some customers who were able to better meet their own needs did not want the costs of software bundled with hardware product costs. In United States vs. [IBM](https://en.wikipedia.org/wiki/IBM), filed January 17, 1969, the government charged that bundled software was anticompetitive. While some software was still being provided without monetary cost and license restriction, there was a growing amount of software that was only at a monetary cost with restricted licensing. In the 1970s and early 1980s, some parts of the [software industry](https://en.wikipedia.org/wiki/Software_industry) began using technical measures (such as distributing only [binary copies](https://en.wikipedia.org/wiki/Executable) of [computer programs](https://en.wikipedia.org/wiki/Computer_programs)) to prevent [computer users](https://en.wikipedia.org/wiki/Computer_users) from being able to use [reverse engineering](https://en.wikipedia.org/wiki/Reverse_engineering) techniques to study and customize software they had paid for. In 1980, the copyright law was extended to computer programs in the [United States](https://en.wikipedia.org/wiki/United_States)—previously, computer programs could be considered ideas, procedures, methods, systems, and processes, which are not copyrightable.

Early on, [closed-source](https://en.wikipedia.org/wiki/Closed-source) software was uncommon until the mid-1970s to the 1980s, when IBM implemented in 1983 an "object code only" policy, no longer distributing source code.

In 1983, [Richard Stallman](https://en.wikipedia.org/wiki/Richard_Stallman), longtime member of the [hacker](https://en.wikipedia.org/wiki/Hacker_(programmer_subculture)) community at the [MIT Artificial Intelligence Laboratory](https://en.wikipedia.org/wiki/MIT_Artificial_Intelligence_Laboratory), announced the [GNU project](https://en.wikipedia.org/wiki/GNU_project), saying that he had become frustrated with the effects of the change in culture of the computer industry and its users. Software development for the [GNU operating system](https://en.wikipedia.org/wiki/GNU_operating_system) began in January 1984, and the [Free Software Foundation](https://en.wikipedia.org/wiki/Free_Software_Foundation) (FSF) was founded in October 1985. An article outlining the project and its goals was published in March 1985 titled the [GNU Manifesto](https://en.wikipedia.org/wiki/GNU_Manifesto). The manifesto included significant explanation of the GNU philosophy, [Free Software Definition](https://en.wikipedia.org/wiki/Free_Software_Definition) and "[copyleft](https://en.wikipedia.org/wiki/Copyleft)" ideas. The FSF takes the position that the fundamental issue [Free software](https://en.wikipedia.org/wiki/Free_software) addresses is an ethical one—to ensure software users can exercise what it calls "[The Four Essential Freedoms](https://en.wikipedia.org/wiki/Free_software#Definition_and_the_Four_Freedoms)".

The [Linux kernel](https://en.wikipedia.org/wiki/Linux_kernel), created by [Linus Torvalds](https://en.wikipedia.org/wiki/Linus_Torvalds), was released as freely modifiable source code in 1991. Initially, Linux was not released under either a Free software or an Open-source software license. However, with version 0.12 in February 1992, he [relicensed](https://en.wikipedia.org/wiki/Software_relicensing) the project under the [GNU General Public License](https://en.wikipedia.org/wiki/GNU_General_Public_License).

[FreeBSD](https://en.wikipedia.org/wiki/FreeBSD) and [NetBSD](https://en.wikipedia.org/wiki/NetBSD) (both derived from [386BSD](https://en.wikipedia.org/wiki/386BSD)) were released as Free software when the [USL v. BSDi](https://en.wikipedia.org/wiki/USL_v._BSDi) lawsuit was settled out of court in 1993. [OpenBSD](https://en.wikipedia.org/wiki/OpenBSD) [forked](https://en.wikipedia.org/wiki/Fork_(software_development)) from NetBSD in 1995. Also in 1995, The [Apache HTTP Server](https://en.wikipedia.org/wiki/Apache_HTTP_Server), commonly referred to as Apache, was released under the [Apache License 1.0](https://en.wikipedia.org/wiki/Apache_License).

In 1997, [Eric Raymond](https://en.wikipedia.org/wiki/Eric_S._Raymond) published [The Cathedral and the Bazaar](https://en.wikipedia.org/wiki/The_Cathedral_and_the_Bazaar), a reflective analysis of the hacker community and Free software principles. The paper received significant attention in early 1998, and was one factor in motivating [Netscape Communications Corporation](https://en.wikipedia.org/wiki/Netscape_Communications_Corporation) to release their popular [Netscape Communicator](https://en.wikipedia.org/wiki/Netscape_Communicator) Internet suite as [Free software](https://en.wikipedia.org/wiki/Free_software). This code is today better known as [Mozilla Firefox](https://en.wikipedia.org/wiki/Mozilla_Firefox) and [Thunderbird](https://en.wikipedia.org/wiki/Mozilla_Thunderbird).

Netscape's act prompted Raymond and others to look into how to bring the FSF's Free software ideas and perceived benefits to the commercial software industry. They concluded that FSF's social activism was not appealing to companies like Netscape, and looked for a way to rebrand the Free software movement to emphasize the business potential of sharing and collaborating on software source code. The new name they chose was "Open-source", and quickly [Bruce Perens](https://en.wikipedia.org/wiki/Bruce_Perens), publisher [Tim O'Reilly](https://en.wikipedia.org/wiki/Tim_O%27Reilly), [Linus Torvalds](https://en.wikipedia.org/wiki/Linus_Torvalds), and others signed on to the rebranding. The [Open Source Initiative](https://en.wikipedia.org/wiki/Open_Source_Initiative) was founded in February 1998 to encourage the use of the new term and evangelize open-source principles.

While the Open Source Initiative sought to encourage the use of the new term and evangelize the principles it adhered to, commercial software vendors found themselves increasingly threatened by the concept of freely distributed software and universal access to an application's [source code](https://en.wikipedia.org/wiki/Source_code). A [Microsoft](https://en.wikipedia.org/wiki/Microsoft) executive publicly stated in 2001 that "Open-source is an intellectual property destroyer. I can't imagine something that could be worse than this for the software business and the intellectual-property business." This view perfectly summarizes the initial response to FOSS by some software corporations.[[citation needed](https://en.wikipedia.org/wiki/Wikipedia:Citation_needed" \o "Wikipedia:Citation needed)] For many years FOSS played a niche role outside of the mainstream of private software development. However the success of FOSS Operating Systems such as Linux, BSD and the companies based on FOSS such as [Red Hat](https://en.wikipedia.org/wiki/Red_Hat), has changed the software industry's attitude and there has been a dramatic shift in the corporate philosophy concerning the development of Free and Open-source software (FOSS)

## Further information:

"Free and open-source software" (FOSS) is an umbrella term for software that is simultaneously considered both [Free software](https://en.wikipedia.org/wiki/Free_software) and [open-source software](https://en.wikipedia.org/wiki/Open-source_software). FOSS (free and open-source software) allows the user to inspect the source code and provides a high level of control of the software's functions compared to [proprietary software](https://en.wikipedia.org/wiki/Proprietary_software). The term "free software" does not refer to the monetary cost of the software at all, but rather whether the license maintains the software user's civil liberties ("free” as in “free speech,” not as in “free beer”).[[3]](https://en.wikipedia.org/wiki/Free_and_open-source_software#cite_note-:1-4) There are a number of related terms and abbreviations for free and open-source software (FOSS or F/OSS), or free/libre and open-source software (FLOSS or F/LOSS—FLOSS is the FSF-preferred term).

Although there is almost a complete overlap between [free-software](https://en.wikipedia.org/wiki/Free_software) licenses and [open-source-software](https://en.wikipedia.org/wiki/Open-source_software) licenses, there is a strong philosophical disagreement between the advocates of these two positions. The terminology of FOSS or "Free and Open-source software" was created to be a neutral on these philosophical disagreements between the FSF and OSI and have a single unified term that could refer to both concepts.

### Free software:

Richard Stallman's [Free Software Definition](https://en.wikipedia.org/wiki/Free_Software_Definition), adopted by the [Free Software Foundation](https://en.wikipedia.org/wiki/Free_Software_Foundation) (FSF), defines [free software](https://en.wikipedia.org/wiki/Free_software) as a matter of liberty not price, and it upholds the Four Essential Freedoms. The earliest-known publication of the definition of his free-software idea was in the February 1986 edition of the FSF's now-discontinued GNU's Bulletin publication. The canonical source for the document is in the philosophy section of the [GNU Project](https://en.wikipedia.org/wiki/GNU_Project) website. As of August 2017, it is published there in 40 languages.

### Four essential freedoms of Free Software:

The freedom to run the program as you wish, for any purpose (freedom 0).

The freedom to study how the program works, and change it so it does your computing as you wish (freedom 1). Access to the source code is a precondition for this.

The freedom to redistribute copies so you can help others (freedom 2).

The freedom to distribute copies of your modified versions to others (freedom 3). 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.

### Open source:

The [open-source-software definition](https://en.wikipedia.org/wiki/Open_Source_Definition) is used by the [Open Source Initiative](https://en.wikipedia.org/wiki/Open_Source_Initiative) (OSI) to determine whether a [software](https://en.wikipedia.org/wiki/Computer_software) license qualifies for the organization's insignia for [Open-source software](https://en.wikipedia.org/wiki/Open-source_software). The definition was based on the [Debian Free Software Guidelines](https://en.wikipedia.org/wiki/Debian_Free_Software_Guidelines), written and adapted primarily by [Bruce Perens](https://en.wikipedia.org/wiki/Bruce_Perens). Perens did not base his writing on the Four Essential Freedoms of free software from the [Free Software Foundation](https://en.wikipedia.org/wiki/Free_Software_Foundation), which were only later available on the web. Perens subsequently stated that he felt [Eric Raymond](https://en.wikipedia.org/wiki/Eric_S._Raymond)'s promotion of Open-source unfairly overshadowed the Free Software Foundation's efforts and reaffirmed his support for Free software. In the following 2000s, he spoke about open source again.

## **Usage**:

**FOSS benefits over proprietary software**

### Personal control, customizability and freedom:

Users of FOSS benefit from the [Four Essential Freedoms](https://en.wikipedia.org/wiki/Free_software#Definition_and_the_Four_Freedoms) to make unrestricted use of, and to study, copy, modify, and redistribute such software with or without modification. If they would like to change the functionality of software they can bring about changes to the code and, if they wish, distribute such modified versions of the software or often − depending on the software's [decision making model](https://en.wikipedia.org/wiki/Group_decision-making) and its other users − even push or request such changes to be made via updates to the original software.

### Privacy and security:

See also: [Open-source software security](https://en.wikipedia.org/wiki/Open-source_software_security), [Surveillance capitalism](https://en.wikipedia.org/wiki/Surveillance_capitalism), [Global surveillance disclosures (2013–present)](https://en.wikipedia.org/wiki/Global_surveillance_disclosures_(2013%E2%80%93present)), and [Software update system](https://en.wikipedia.org/wiki/Software_update_system)

Manufacturers of proprietary, closed-source software are sometimes pressured to building in [backdoors](https://en.wikipedia.org/wiki/Backdoor_(computing)) or other covert, undesired features into their software. Instead of having to trust software vendors, users of FOSS can inspect and verify the source code themselves and can put trust on a community of volunteers and users.[[36]](https://en.wikipedia.org/wiki/Free_and_open-source_software#cite_note-pcw1-37) As proprietary code is typically hidden from public view, only the vendors themselves and hackers may be aware of any [vulnerabilities](https://en.wikipedia.org/wiki/Vulnerability_(computing)) in them while FOSS involves as many people as possible for exposing bugs quickly.

### Low costs or no costs:

FOSS is often free of charge although donations are often encouraged. This also allows users to better test and compare software.

### Quality, collaboration and efficiency:

See also: [§ Bugs and missing features](https://en.wikipedia.org/wiki/Free_and_open-source_software#Bugs_and_missing_features)

FOSS allows for better collaboration among various parties and individuals with the goal of developing the most efficient software for its users or use-cases while proprietary software is typically [meant to generate profits](https://en.wikipedia.org/wiki/Profit_motive). Furthermore, in many cases more organizations and individuals contribute to such projects than to proprietary software. It has been shown that technical superiority is typically the primary reason why companies choose open source software.

**Drawbacks compared to proprietary software:**

### Security and user-support:

See also: [Common good](https://en.wikipedia.org/wiki/Common_good), [Public participation](https://en.wikipedia.org/wiki/Public_participation), and [Proactive cyber defence § Measures](https://en.wikipedia.org/wiki/Proactive_cyber_defence#Measures)

According to [Linus's law](https://en.wikipedia.org/wiki/Linus%27s_law) the more people who can see and test a set of code, the more likely any flaws will be caught and fixed quickly. However, this does not guarantee a high level of participation. Having a grouping of full-time professionals behind a commercial product can in some cases be superior to FOSS.

Furthermore, publicized source code might make it easier for hackers to find vulnerabilities in it and write exploits. This however assumes that such malicious hackers are more effective than [white hat hackers](https://en.wikipedia.org/wiki/White_hat_(computing)) which [responsibly disclose](https://en.wikipedia.org/wiki/Responsible_disclosure) or help fix the vulnerabilities, that no code leaks or [exfiltrations](https://en.wikipedia.org/wiki/Data_breach) occur and that [reverse engineering](https://en.wikipedia.org/wiki/Reverse_engineering) of proprietary code is a hindrance of significance for malicious hackers.[[41]](https://en.wikipedia.org/wiki/Free_and_open-source_software#cite_note-wash1-42)

### Hardware and software compatibility:

Further information: [Software incompatibility](https://en.wikipedia.org/wiki/Software_incompatibility) and [System requirements](https://en.wikipedia.org/wiki/System_requirements)

Sometimes, FOSS is not compatible with proprietary hardware or specific software. This is often due to manufacturers obstructing FOSS such as by not disclosing the [interfaces](https://en.wikipedia.org/wiki/Interface_(computing)) or other specifications needed for members of the FOSS movement to write [drivers](https://en.wikipedia.org/wiki/Device_driver) for their hardware − for instance as they wish customers to run only their own proprietary software or as they might benefit from partnerships. [[additional citation(s) needed](https://en.wikipedia.org/wiki/Wikipedia:Citation_needed)]

### Bugs and missing features:

See also: [§ Quality, collaboration and efficiency](https://en.wikipedia.org/wiki/Free_and_open-source_software#Quality,_collaboration_and_efficiency)

While FOSS can be superior to proprietary equivalents in terms of software features and stability, in many cases FOSS has more unfixed bugs and missing features when compared to similar commercial software. [[additional citation(s) needed](https://en.wikipedia.org/wiki/Wikipedia:Citation_needed)] This varies per case and usually depends on the level of interest and participation in a FOSS project. Furthermore, unlike with typical commercial software missing features and bugfixes can be implemented by any party that has the relevant motivation, time and skill to do so. [[additional citation(s) needed](https://en.wikipedia.org/wiki/Wikipedia:Citation_needed)]

### Less guarantees of development:

There is often less certainty in FOSS projects gaining the required resources / participation for continued development than commercial software backed by companies.[[52]](https://en.wikipedia.org/wiki/Free_and_open-source_software#cite_note-arthur-53)[[additional citation(s) needed](https://en.wikipedia.org/wiki/Wikipedia:Citation_needed)] However companies also often abolish projects for being unprofitable and often large companies rely on and hence co-develop open source software.[[42]](https://en.wikipedia.org/wiki/Free_and_open-source_software#cite_note-scmagazine-43)

Missing applications:

As the FOSS operating system distributions of [Linux](https://en.wikipedia.org/wiki/Linux) has a lower [market share](https://en.wikipedia.org/wiki/Market_share) of end users there are also fewer applications available.[[53]](https://en.wikipedia.org/wiki/Free_and_open-source_software#cite_note-Kenneth-54)[[54]](https://en.wikipedia.org/wiki/Free_and_open-source_software#cite_note-55)

Adoption by governments[[edit](https://en.wikipedia.org/w/index.php?title=Free_and_open-source_software&action=edit&section=18)]

Main article: [Adoption of free and open-source software by public institutions](https://en.wikipedia.org/wiki/Adoption_of_free_and_open-source_software_by_public_institutions)

See also: [Sovereignty](https://en.wikipedia.org/wiki/Sovereignty), [National security](https://en.wikipedia.org/wiki/National_security), [Cyber emergency response team](https://en.wikipedia.org/wiki/Cyber_emergency_response_team), and [Global public good](https://en.wikipedia.org/wiki/Global_public_good)

This list is [incomplete](https://en.wikipedia.org/wiki/Wikipedia:WikiProject_Lists#Incomplete_lists); you can help by [expanding it](https://en.wikipedia.org/w/index.php?title=Free_and_open-source_software&action=edit)