# Software overtakes aviation as a major carbon emitter, pushing green coding to the forefront



The tech industry is at a crossroads, faced with an urgent need to address its environmental impact. Software now accounts for 4% of global greenhouse gas emissions, an increase that has surpassed the emissions of the aviation sector. As developers wield significant influence over this trajectory, adopting sustainable software development practices—often referred to as "green coding"—has become not only an environmental necessity but a business imperative. Companies that have embraced such practices frequently report reductions in energy costs by as much as 30% and improvements in performance metrics, underscoring the dual benefits of environmental responsibility and economic efficiency.

Every facet of software development contributes to this hidden carbon footprint, particularly as applications operate across various servers and data centres that consume enormous amounts of electricity. In fact, data centres alone account for approximately 1% of total global electricity consumption, amounting to a staggering 200 terawatt-hours annually. Alarmingly, software inefficiency is responsible for wasting 30-40% of computing resources. This inefficiency can lead to increased energy consumption by as much as 300% due to poorly optimised code. For instance, simple database queries, if left inefficient, can multiply energy consumption significantly, translating into substantial financial costs.

To counteract this trend, several principles of sustainable software development have emerged. An energy-first design philosophy advocates for energy consumption to be considered from the earliest stages of design, rather than merely optimising for speed or feature set. This introduces key strategies such as minimising computational complexity in algorithms, reducing the need for excessive data transfer and storage, and optimising for idle states. Moreover, the Green Software Foundation has established a framework composed of three pillars: energy efficiency, hardware efficiency, and carbon awareness—each a vital component in creating software that respects environmental limits.

Specific programming techniques can considerably enhance energy efficiency. These include algorithm optimisation, smart caching strategies, and efficient database query designs. A shift towards more energy-efficient data structures and algorithms not only benefits the environment; it enhances application performance as well. For example, selecting efficient lookup methods over less efficient alternatives improves both energy use and execution speed, demonstrating that environmentally responsible choices can yield practical benefits.

The solutions extend to cloud computing as well, where companies are increasingly implementing carbon-aware computing strategies. By scheduling workloads during periods when cleaner energy sources are available, developers can leverage renewable energy for their operations, further reducing the carbon footprint of their software. In 2025, notable players in the cloud space are setting ambitious goals related to sustainability. Google Cloud has achieved carbon neutrality since 2007, while Microsoft Azure aims to be carbon-negative by 2030. Data from companies such as Spotify further illustrates the potential for energy savings, having optimised streaming processes to achieve a 40% reduction in energy per stream.

Despite these developments, challenges persist. Recent reports highlight that while some tech giants have made significant strides in renewable energy usage, their greenhouse gas emissions continue to rise due to the expansion of data centres supporting artificial intelligence systems. According to a Financial Times analysis, Google’s emissions have surged by 48% over the past five years as it strives to enhance its AI capabilities, indicating a complex interplay between technological advancement and environmental sustainability.

As the demand for energy-intensive solutions grows, the industry must remain vigilant. The International Monetary Fund predicts that while economic benefits from advances in AI could outweigh emissions costs, it is crucial to mitigate the environmental impacts associated with these expanding infrastructures. In turn, industry-wide efforts are emerging for better carbon accounting standards and certification programs, intended to push forward both innovation and sustainability hand in hand.

Ultimately, the pathway to sustainable software development requires a holistic approach that combines technical knowledge, environmental awareness, and a commitment to continual improvement. Developers are encouraged to audit their current coding practices, implement quick wins in optimization, and measure the energy consumption of their applications. As individual initiatives accumulate, the positive effects on energy efficiency and global carbon emissions can be profoundly transformative. By adopting green coding practices today, the tech world not only positions itself as a leader in sustainability but also helps ensure a viable future for the planet.

Indeed, every line of code holds the potential to contribute to a greener, more sustainable world; the question remains—how will developers leverage that power in their own practices?

## Reference Map:

* Paragraph 1 – [[1]](https://www.besttechie.com/green-coding-how-to-write-energy-efficient-software-in-2025/), [[2]](https://www.besttechie.com/green-coding-how-to-write-energy-efficient-software-in-2025/)
* Paragraph 2 – [[1]](https://www.besttechie.com/green-coding-how-to-write-energy-efficient-software-in-2025/), [[3]](https://www.reuters.com/sustainability/climate-energy/ai-economic-gains-likely-outweigh-emissions-cost-says-imf-2025-04-22/), [[4]](https://www.ft.com/content/383719aa-df38-4ae3-ab0e-6279a897915e)
* Paragraph 3 – [[5]](https://devtechinsights.com/green-coding-guide/), [[6]](https://www.csm.tech/blog-details/embracing-green-coding-best-practices-for-sustainable-software-development), [[7]](https://green-it.ma/en/sustainable-software-development-best-practices)
* Paragraph 4 – [[6]](https://www.csm.tech/blog-details/embracing-green-coding-best-practices-for-sustainable-software-development), [[7]](https://green-it.ma/en/sustainable-software-development-best-practices)
* Paragraph 5 – [[3]](https://www.reuters.com/sustainability/climate-energy/ai-economic-gains-likely-outweigh-emissions-cost-says-imf-2025-04-22/), [[4]](https://www.ft.com/content/383719aa-df38-4ae3-ab0e-6279a897915e)

Source: [Noah Wire Services](https://www.noahwire.com)

## Bibliography

1. <https://www.besttechie.com/green-coding-how-to-write-energy-efficient-software-in-2025/> - Please view link - unable to able to access data
2. <https://www.besttechie.com/green-coding-how-to-write-energy-efficient-software-in-2025/> - This article discusses the environmental impact of software development, highlighting that software accounts for 4% of global greenhouse gas emissions, surpassing the aviation industry. It emphasizes the importance of green coding practices, noting that companies implementing sustainable development report a 30% reduction in energy costs and improved performance metrics. The piece provides a comprehensive guide on writing energy-efficient code, covering topics such as energy-first design philosophy, the Green Software Foundation framework, and various energy-efficient programming techniques.
3. <https://www.reuters.com/sustainability/climate-energy/ai-economic-gains-likely-outweigh-emissions-cost-says-imf-2025-04-22/> - A Reuters article reports on an International Monetary Fund (IMF) study indicating that global economic output is expected to rise by about 0.5% annually from 2025 to 2030 due to advances in artificial intelligence (AI). Despite the environmental costs from increased carbon emissions due to energy-demanding data centers, the IMF suggests that these economic gains will outweigh the emissions costs. The report also highlights the need for policymakers and businesses to manage societal impacts responsibly.
4. <https://www.ft.com/content/383719aa-df38-4ae3-ab0e-6279a897915e> - An article from the Financial Times discusses Google's greenhouse gas emissions, which have increased by 48% over the past five years, reaching 14.3 million tonnes of carbon equivalent in 2023. This rise is attributed to the expansion of its data centers to support artificial intelligence systems. The company's goal to achieve 'net zero' emissions by 2030 is now uncertain, highlighting the challenges of reducing emissions while developing AI infrastructure.
5. <https://devtechinsights.com/green-coding-guide/> - This guide provides best practices for sustainable software development, emphasizing the importance of energy-efficient coding. It outlines five principles of green coding, including using energy-efficient algorithms, optimizing APIs and microservices, reducing computational waste, choosing efficient programming languages, and leveraging hardware efficiency. The article also introduces tools for sustainable development, such as Scaphandre for measuring energy usage and GreenFrame for tracking CO₂ impact of cloud apps.
6. <https://www.csm.tech/blog-details/embracing-green-coding-best-practices-for-sustainable-software-development> - This blog post explores best practices for sustainable software development, focusing on green coding principles. It discusses adopting lean coding principles to eliminate waste and build quality, switching to microservices architecture for independent scalability and improved resource utilization, leveraging renewable energy for hosting to reduce carbon footprint, and prioritizing automation to optimize processes and reduce energy consumption. The article also emphasizes the importance of efficient algorithms, eliminating redundant code, implementing caching techniques, choosing green programming languages, and reducing unnecessary dependencies.
7. <https://green-it.ma/en/sustainable-software-development-best-practices> - This article outlines best practices for sustainable software development, aiming to minimize environmental impact while maintaining high-quality code. It emphasizes optimizing code efficiency by using appropriate data structures and algorithms, implementing energy-aware design patterns like lazy loading and event-driven architecture, optimizing database queries through proper indexing and caching, embracing cloud-native technologies such as containerization and serverless computing, and implementing efficient data storage and transfer methods like compression algorithms and content delivery networks.