

A gentle introduction: Stylus

In a nutshell:

- Stylus lets you write smart contracts in programming languages that compile to WASM, such as Rust, C, C++, and many others
- , allowing you to tap into their ecosystem of libraries and tools. Rich language and tooling support already exist for Rust. You can try the SDK and CLI with the [quickstart](#)
- .
- Solidity contracts and Stylus contracts are fully interoperable
- . In Solidity, you can call a Rust program and vice versa, thanks to a second, coequal WASM virtual machine.
- Stylus contracts offer significantly faster execution and lower gas fees
- for memory and compute-intensive operations, thanks to the superior efficiency of [WASM](#)
- programs.

What's Stylus?

Stylus is an upgrade to Arbitrum Nitro ([ArbOS 32](#)), the tech stack powering Arbitrum One, Arbitrum Nova, and Arbitrum Orbit chains. This upgrade adds a second, coequal virtual machine to the EVM, where EVM contracts continue to behave exactly as they would in Ethereum. We call this paradigm MultiVM since everything is entirely additive.

This second virtual machine executes WebAssembly (WASM) rather than EVM bytecode. WASM is a modern binary format popularized by its use in major web standards, browsers, and companies to speed up computation. WASM is built to be fast, portable, and human-readable. It has sandboxed execution environments for security and simplicity. Working with WASM is nothing new for Arbitrum chains. Ever since the [Nitro upgrade](#), WASM has been a fundamental component of Arbitrum's fully functioning fraud proofs.

With a WASM VM, any programming language compilable to WASM is within Stylus's scope. While many popular programming languages can compile into WASM, some compilers are more suitable for smart contract development than others, like Rust, C, and C++. Other languages like Go, Sway, Move, and Cairo are also supported. Languages that include their own runtimes, like Python and Javascript, are more complex for Stylus to support, although not impossible. Compared to Solidity, WASM programs are much more efficient for memory-intensive applications. There are many reasons for this, including the decades of compiler development for Rust and C. WASM also has a faster runtime than the EVM, resulting in faster execution. Third-party [contribution](#) in the form of libraries for new and existing languages is welcomed!

Use Cases

While many developers will be drawn to new use cases, rebuilding existing applications in Stylus will also open the door to innovation and optimization. dApps have never been faster, cheaper, or safer. Stylus can integrate easily into existing Solidity projects by calling a Stylus contract to optimize specific parts of your dApp or building the entire dApp with Stylus. It's impossible to list all of the use cases Stylus enables; think about the properties of all WASM-compatible languages! That said, here are some particularly exciting ideas:

- Efficient On-Chain Verification with ZK-Proofs
 - : Enable cost-effective onchain verification
 - using zero-knowledge proving systems for privacy, interoperability, and more (see [case study](#))
- Advanced DeFi Instruments
 - : Power complex financial instruments and processes like custom
 - pricing curves for AMMs, synthetic assets, options, and futures with onchain computation via
 - extending current protocols (ie. Uniswap V4 hooks) or building your own.
- High-Performance On-Chain Logic
 - : Support memory and compute-intensive applications like
 - onchain games and generative art either by writing all of the application in Stylus or enhance
 - performance of existing Solidity contracts by optimizing specific parts.
- Endless Possibilities
 - :
 - Enable innovative use cases such as generative art, compute-heavy
 - AI models, on-chain games, and projects utilizing advanced cryptography, unlocking the full potential
 - of resource-intensive applications on-chain.

Getting Started

1. Utilize our [quickstart](#)
2. , [Rust SDK](#)
3. , to help you start building.
4. Join our Stylus Developer [Telegram](#)

5. group and [Arbitrum Discord](#)
6. for support as well as the official Arbitrum [@Arbitrum](#)
7.) and Arbitrum Developers [@ArbitrumDevs](#)
8.) X accounts for announcements.
9. Check out the [Awesome Stylus](#)
10. repository for various community contributed Stylus projects and tools. If you build something useful, we'd be happy to add it there. [Edit this page](#) Last updated on Jan 27, 2025 [Previous](#) [Write Stylus contracts](#) [Next](#) [Quickstart](#)