

Solidity support

Arbitrum chains are Ethereum compatible, and therefore allow you to trustlessly deploy Solidity smart contracts, as well as contracts written in Vyper or any other language that compiles to EVM bytecode. However, when calling certain properties and functions on a Solidity smart contract, there might be some differences between the result you'd obtain if that contract was on Ethereum, and the result on Arbitrum.

This page compiles a list of functions and properties that return a different result when called in Arbitrum.

Differences from Solidity on Ethereum

Although Arbitrum supports Solidity code, there are differences in the effects of a few operations, including language features that don't make much sense in the Layer 2 context.

Operation Description `blockhash(x)` Returns a cryptographically insecure, pseudo-random hash for `x` within the range `block.number - 256 <= x < block.number`. If `x` is outside of this range, `blockhash(x)` will return 0. This includes `blockhash(block.number)`, which always returns 0 just like on Ethereum. The hashes returned do not come from L1.
⚠ Arbitrum's L2 block hashes should not be relied on as a secure source of randomness. `block.coinbase` Returns the designated internal address `0xA4b000` if the message was posted by a sequencer. If it's a delayed message, it returns the address of the delayed message's poster (Note: the handling of delayed message's `block.coinbase` will likely be changed in a future ArbOS version). `block.difficulty` Returns the constant 1. `block.prevrandao` Returns the constant 1. `block.number` Returns an "estimate" of the L1 block number at which the sequencer received the transaction. For more information, see [Block numbers and time](#). `msg.sender` Works the same way it does on Ethereum for regular L2 to L2 transactions. For transactions submitted via the delayed inbox, it will return the L2 address alias of the L1 contract that triggered the message. For more information, see [address aliasing](#). `OPCODEPUSH0` This OPCODE was added as part of ArbOS 11 and is now supported. [Edit this page](#) Last updated on Mar 22, 2024 [Previous](#) [RPC methods](#) [Next](#) [Oracles overview](#)