Solidity

Solidity is a contract-oriented, high-level language whose syntax is similar to that of JavaScript and it is designed to target the Ethereum Virtual Machine (EVM).

Solidity is statically typed, supports inheritance, libraries and complex user-defined types among other features.

As you will see, it is possible to create contracts for voting, crowdfunding, blind auctions, multi-signature wallets and more.

> **Note**
>
> The best way to try out Solidity right now is using Remix (it can take a while to load, please be patient).

Useful links

- Ethereum
- Changelog
- Story Backlog
- Source Code
- Ethereum Stackexchange
- Gitter Chat

Available Solidity Integrations

- **Remix**
  Browser-based IDE with integrated compiler and Solidity runtime environment without server-side components.

- **Ethereum Studio**
  Specialized web IDE that also provides shell access to a complete Ethereum environment.

- **IntelliJ IDEA plugin**
  Solidity plugin for IntelliJ IDEA (and all other JetBrains IDEs)

- **Visual Studio Extension**
  Solidity plugin for Microsoft Visual Studio that includes the Solidity compiler.

- **Package for SublimeText — Solidity language syntax**
  Solidity syntax highlighting for SublimeText editor.

- **Etheratom**
  Plugin for the Atom editor that features syntax highlighting, compilation and a runtime environment (Backend node & VM compatible).

- **Atom Solidity Linter**
  Plugin for the Atom editor that provides Solidity linting.

- **Atom Solium Linter**
  Configurable Solidty linter for Atom using Solium as a base.

- **Solium**
  A commandline linter for Solidity which strictly follows the rules prescribed by the Solidity Style Guide.

- **Visual Studio Code extension**
  Solidity plugin for Microsoft Visual Studio Code that includes syntax highlighting and the Solidity compiler.

- **Emacs Solidity**
  Plugin for the Emacs editor providing syntax highlighting and compilation error reporting.

- **Vim Solidity**
  Plugin for the Vim editor providing syntax highlighting.

- **Vim Syntastic**
  Plugin for the Vim editor providing compile checking.

Discontinued:

- **Mix IDE**
  Qt based IDE for designing, debugging and testing solidity smart contracts.

Solidity Tools

- **Dapp**
  Build tool, package manager, and deployment assistant for Solidity.

- **Solidity REPL**
  Try Solidity instantly with a command-line Solidity console.

- **solgraph**
  Visualize Solidity control flow and highlight potential security vulnerabilities.

- **evmdis**
  EVM Disassembler that performs static analysis on the bytecode to provide a higher level of abstraction than raw EVM operations.

- **Doxity**
  Documentation Generator for Solidity.

Third-Party Solidity Parsers and Grammars

- **solidity-parser**
  Solidity parser for JavaScript

- **Solidity Grammar for ANTLR 4**
  Solidity grammar for the ANTLR 4 parser generator

Language Documentation

On the next pages, we will first see a simple smart contract written in Solidity followed by the basics about blockchains and the Ethereum Virtual Machine.

The next section will explain several features of Solidity by giving useful example contracts Remember that you can always try out the contracts in your browser!

The last and most extensive section will cover all aspects of Solidity in depth.

If you still have questions, you can try searching or asking on the Ethereum Stackexchange site, or come to our gitter channel. Ideas for improving Solidity or this documentation are always welcome!

See also Russian version (русский перевод).

Contents