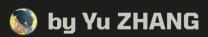
# Foundry: A Smart Contract Development Framework

Foundry is a powerful and comprehensive tool for building, testing, and deploying smart contracts. It's a popular choice for developers of decentralized applications.



## Installation

The installation process is simple and straightforward. Use the provided curl command to download and execute the Foundry installer script.

#### Command

curl -L https://foundry.paradigm.xyz | bash

## **Project Initialization**

Foundry provides a simple command to initialize a new project with basic configuration and folder structure.

forge init

Initializes a new Foundry project in the current directory.

forge init --force

Overwrites an existing project if one already exists in the current directory.

# Configuration

Foundry uses a configuration file, foundry.toml, to customize the project's behavior, such as setting the Solidity compiler version and specifying formatting rules.

Solidity Version	VS Code Formatting	Forge fmt
\[profile.default]	Install Solidity extension	Run command to automatically
solc_version = "0.8.17"	Add settings to VS Code	format code

# Contract Deployment

Forge provides a convenient script for deploying contracts to a blockchain network. This approach simplifies the deployment process and ensures consistent execution.

forge create

Deploy a contract using a forge script

2 --rpc-url

Specifies the URL of the blockchain network to connect to

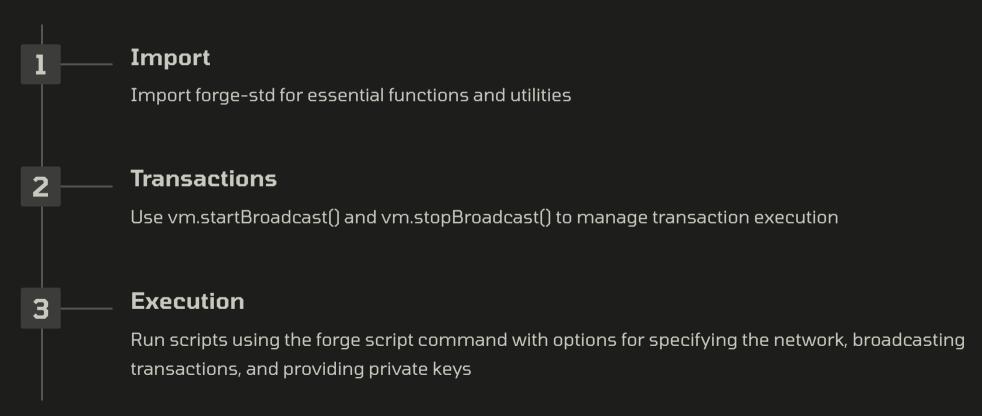
--interactive

3

Enables interactive mode for confirmations and inputs during deployment

## Foundry Scripts

Foundry scripts are powerful tools for automating interactions with smart contracts, enabling tasks like deployment, testing, and data manipulation.



## **Command Line Transaction Calls**

Foundry offers a command-line interface (CLI) for interacting with deployed contracts, allowing for sending transactions and retrieving data.

ı send

Sends a transaction to a contract function

call

Executes a contract function without sending a transaction

cast

3

Provides tools for data conversion and manipulation

# Compilation

Foundry automates the compilation process, ensuring that the Solidity code is compiled into bytecode ready for deployment on the blockchain.

forge build

Compiles all Solidity contracts in the project

# Testing

Foundry provides a comprehensive testing framework that simplifies the process of writing and executing tests for smart contracts, enhancing code quality and reliability.

1

#### **Test Contracts**

Create test files in the test directory, inheriting from the Test contract

2

### **Testing Structure**

Use setup, success, and fail tests, as well as event checks

3

#### **Helper Methods**

Utilize functions like vm.prank(), vm.deal(), and vm.warp() for advanced testing scenarios

4

## **Running Tests**

Use the forge test command with various options for filtering tests, controlling verbosity, and enabling advanced features

# Third-Party Libraries

Foundry integrates well with popular third-party libraries, simplifying development and enabling developers to leverage pre-built components for common functionalities.

forge install	Installs a library from a package manager
forge update	Updates installed libraries to newer versions
forge remove	Removes installed libraries
remapping.txt	Configures paths for libraries, such as OpenZeppelin