

Updated on 11/08/2025

Sign up

## Vyper training

2 days (14 hours)

### Presentation

Vyper is a smart contract language for Ethereum, designed for security, readability and reliability. Inspired by Python, it deliberately limits certain functionalities in order to reduce the attack surface and design errors.

This 2-day training course covers the creation, testing and deployment of Vyper contracts: basic structures, state management, good security practices, integration with web dApps and the production release process.

Guided workshops enable you to quickly acquire operational reflexes.

At the end of the course, you'll be able to write robust contracts, audit them, and orchestrate their deployment on testnet/mainnet by integrating them into a modern web application.

Like all our training courses, this one uses the latest stable version [v0.4.3 of Vyper](#).

### Objectives

- Understand Vyper philosophy and syntax
- Write and deploy secure smart contracts
- Integrate Vyper into a web dApp
- Apply good security practices
- Test, debug and prepare for production

### Target audience

- Blockchain developers
- Web developers
- Technical architects
- Project managers

## Prerequisites

- Programming basics (Python recommended)
- Basic knowledge of blockchain and smart contracts
- Familiarity with web development tools

## Our Vyper training program

### Introduction to Vyper and the Ethereum ecosystem

- Presentation of Vyper and comparison with Solidity
- Vyper's strengths: simplicity, readability, security
- Use cases for smart contracts on Ethereum
- Setting up the environment and tools (compiler, CLI)
- First compilation and basic deployment
- Workshop: creating and deploying a first simple contract

### Syntax and basic structures

- Data types and variables
- Functions, visibility and decorators
- Control structures: conditions, loops
- Event management
- Specific features vs. Solidity
- Workshop: implementing a storage contract

### State management and storage

- State variables and persistent memory
- Structures: arrays, mappings
- Constants and immutables
- Read/write in storage
- Optimizing gas (cost)
- Workshop: on-chain managed list with search

### Security and best practices

- Security principles integrated into Vyper
- Preventing common vulnerabilities

- Intentional language limitations
- Audit and formal verification
- Secure development patterns
- Workshop: auditing and correcting a vulnerable contract

## Interaction with blockchain and dApps

- Deployment on testnet / mainnet
- Interaction via Web3.py and scripts
- Connection to a web interface
- Transaction and event management on the frontend
- Integration with web3 frameworks
- Workshop: mini dApp talking to a Vyper contract

## Testing, debugging and going live

- Testing tools: Brownie, Hardhat
- Unit and scenario testing
- Debugging and error analysis
- Pre-production checklist
- Migrations and updates
- Workshop: testing and final deployment of a contract

## Companies involved

This course is aimed at both individuals and companies, large or small, wishing to train their teams in a new advanced computer technology, or to acquire specific business knowledge or modern methods.

## Positioning on entry to training

Positioning at the start of training complies with Qualiopi quality criteria. As soon as registration is finalized, the learner receives a self-assessment questionnaire which enables us to assess his or her estimated level of proficiency in different types of technology, as well as his or her expectations and personal objectives for the forthcoming training course, within the limits imposed by the selected format. This questionnaire also enables us to anticipate any connection or security difficulties within the company (intra-company or virtual classroom) which could be problematic for the follow-up and smooth running of the training session.

## Teaching methods

Practical training: 60% hands-on, 40% theory. Training material distributed in digital format to all participants.

## Organization

The course alternates theoretical input from the trainer, supported by examples, with brainstorming sessions and group work.

## Validation

At the end of the session, a multiple-choice questionnaire is used to check that skills have been correctly acquired.

## Certification

A certificate will be awarded to each trainee who completes the training course.