AMBIENT-IT

+33 (0)1 48 10 09 46
formation@ambient-it.net
140 Avenue Jean Lolive 93500 Pantin - France

Updated on 11/08/2025

Sign up

Move training

4 days (28 hours)

Overview

Move is a resource-oriented smart contract language designed for secure and predictable applications on next-generation blockchains such as Aptos and Sui. Its resource model avoids common error classes and strengthens the protection of digital assets.

Our Move training course teaches you how to structure your modules and scripts, manage resources, publish contracts, interact with the chain (transactions, events, storage) and connect your on?chain code to a modern web dApp. The focus is on best practices, security and code quality.

Through a series of progressive workshops, you'll go from getting to grips with the language to putting it into production: writing, testing, verification, deployment on testnet/mainnet, JavaScript/TypeScript integration and advanced patterns (permissions, DAO, upgradability).

At the end of the course, you'll be able to design robust contracts, industrialize your deployments and integrate Move into ready-to-use web applications.

As with all our training courses, this one uses the latest Move update.

Objectives

- Master the syntax and structure of a Move project
- Use the resource-oriented model
- Develop secure modules and scripts
- Deploy and interact with testnet / mainnet
- Connect Move to a web dApp

Target audience

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

Prerequisites

- Programming skills
- Notions of blockchain and smart contracts
- Familiarity with web development tools

Move training program

Introduction to Move and its ecosystem

- Move's origins and link with Libra/Diem
- Compatible blockchains: Aptos, Sui
- Strengths vs. other smart contract languages
- Key concepts: resources, modules, scripts
- Installation and configuration of the environment
- Workshop: Move toolchain + first project

Move structure and syntax

- File and project organization
- Basic syntax and language rules
- Primitive and compound types
- Functions, visibility, scope
- Compilation and execution
- Workshop: creating a simple module

Resource management

- Resource?oriented programming: principles
- Defining, instantiating and moving resources
- Copying, borrowing and ownership
- Move security guarantees
- Resource release / destruction
- Workshop: implementing a simple token

Modules and scripts

- Role and interaction of modules vs. scripts
- Importing and reusing code

- Publishing/updating modules
- Parameters and script feedback
- Business logic organization
- Workshop: module + interaction script

Security and formal verification

- Built-in security properties
- Move bytecode verifier
- Tests, assertions and invariants
- Preventing common vulnerabilities
- Static analysis
- Workshop: auditing and securing a contract

Events and storage

- Transmitting/reading events
- Persistent storage structures
- On-chain data access
- Space and cost optimization
- Good structuring practices
- Workshop: contract with event tracking

Interaction with blockchain

- Deployment on testnet / mainnet
- Aptos / Sui CLIs
- Transaction submission
- Accounts, keys, permissions
- On-chain read/write
- Workshop: deploying and interacting on testnet

Web and dApp integration

- Architecture of a Move dApp
- Using APIs and SDKs
- Connection via JavaScript/TypeScript
- Auth and account management
- Front ? blockchain: best practices
- Workshop: web dApp connected to Move

Advanced design patterns

- Permissions and roles
- Reusable libraries
- Upgradeable contracts

- Modeling complex applications
- Performance optimization
- Workshop: implementing a DAO model

Testing and debugging

- Unit and integration testing
- Test scenarios for smart contracts
- Debugging with CLI and logs
- Transaction simulation
- Analysis of common errors
- Workshop: complete test suite

Production and maintenance

- Preparing a mainnet contract
- Best deployment practices
- Migration and upgrade strategies
- Monitoring and performance
- Incident and patch management
- Workshop: publishing & monitoring a Move project

Case study and final project

- Analysis of an existing Move project
- Identification of best practices
- Refactoring/improving code
- Documentation and transfer
- Presentation of final project
- Workshop: complete dApp in Move

Companies concerned

This course is aimed at both individuals and companies, large or small, wishing to train their teams in a new, advanced IT 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 with regard to the training to come, 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 verifies the correct acquisition of skills.

Certification

A certificate will be awarded to each trainee who has completed the entire course.