Solidity Programming

Duration: 2 days Price: € 1499

Open Schedule

Audience Course Solidity Programming

The course Solidity Programming is intended for developers who want to learn how to develop applications for the Ethereum blockchain with the Solidity programming language.

Prerequisites Course Solidity Programming

Basic knowledge of blockchain technology, cryptocurrencies and programming is required to participate in the course Solidity Programming.

Realization Training Solidity Programming

The theory is treated on the basis of presentation slides. The theory is explained in more detail by means of demos. There is an opportunity to practice after each module.

Certificate Course Solidity Programming

After successful completion of the course participants receive a certificate Solidity Programming.

Content Solidity Programming

In the course Solidity Programming the participant learn the Solidity Language for writing smart contracts that run on the Ethereum Virtual Machine (EVM). In addition to being used for writing smart contracts, Solidity is also used to build decentralized applications (DApps) that run on a decentralized platform like Ethereum.

Solidity Intro

The course Solidity Programming starts by explaining the basics of the Solidity language syntax and role of the Ethereum Virtual Machine (EVM). Also attention is paid to setting up a development environment for Solidity.

Smart Contracts

Next the structure of smart contracts and how smart contracts work is discussed. The role of state variables, functions and events is explained as well. Several common use cases for smart contracts are demonstrated.

Data Types

The different data types that are available in the Solidity language are part of the course Solidity Programming as well. Covered are basic data types like integers, booleans, addresses and strings. And also more complex data types and user defined data types like arrays, structs and enums are treated.

Creating Contracts

Subsequently the writing of a basic contract in Solidity is treated. Attention is paid to contract constructors, functions and the handling of events. Also logging in contracts and testing and debugging Solidity contracts is covered.

Contract Inheritance

Next it is explained what contract inheritance is and and why is it useful. Function visibility, function modifiers, immutable and constant state variables are part of subject matter in this respect. And the use of polymorphism in Solidity is treated as well.

Decentralized Apps (DApps)

Finally attention is paid to the creation of decentralized apps (DApps) in Solidity.
It is demonstrated how to build a DApp using Solidity and a front-end framework like React. Interacting with a DApp using web3.js, a local blockchain like Ganache and a wallet like MetaMask is also shown.

Modules Solidity Programming

Module 1 : Solidity Intro	Module 2 : Smart Contracts	Module 3 : Data Types
What is Solidity?	Structure of a Contract	Value Types
Blockchains	State Variables	Booleans
Cryptocurrencies	Functions	Integers
Smart Contracts	Function Modifiers	Strings
Ethereum Virtual Machine	Events	Fixed Point Numbers
Curly Bracket Language	Errors	Address
Remix IDE	EVM Logging	Address Members
Compiler Usage	Revert Statements	Fixed Size Byte Arrays
Source File Layout	Struct Types	Address Literals
Pragma's	Enum Types	User Defined Types
Import Paths	Inheritance	Arrays and Structs
Module 4 : Creating Contracts	**Module 5 : Contract Inheritance**	**Module 6 : Decentralized Apps**
Constructors	Inheritance	What are DApps?
State Variable Visibility	Function Overloading	Benefits of DApps
Function Visibility	Function Overriding	Self Executing Contracts
Immutable State Variables	Modifier Overriding	Decentralization
Constant State Variables	Constructors Calling	Open Source
Getter Functions	Base Constructors	Cryptography
Function Modifiers	Multiple Inheritance	Token System
Parameters and Return Variables	Linearization	Decentralized Finance
Ether Units	Abstract Contracts	Ganache
Time Units	Interfaces	Metamask
Transaction Properties	Libraries	web3.js

SpiralTrain BV
Standerdmolen 10, 2e verdieping
3995 AA Houten

info@spiraltrain.nl
www.spiraltrain.nl
Tel.: +31 (0) 30 – 737 0661

Locations
Houten, Amsterdam, Rotterdam, Eindhoven, Zwolle, Online