

Solidity Programmeren

Doelgroep Cursus Solidity Programmeren

De cursus Solidity Programmeren is bedoeld voor developers die willen leren hoe je applicaties voor de Ethereum blockchain kunt ontwikkelen met de taal Solidity.

Voorkennis Cursus Solidity Programmeren

Om aan de cursus Solidity Programmeren te kunnen deelnemen is basiskennis van blockchain technology, cryptocurrencies en programmeren vereist.

Uitvoering Training Solidity Programmeren

De theorie wordt behandeld aan de hand van presentatie slides. De theorie wordt nader uitgelegd middels demo's. Na iedere module is er de gelegenheid tot oefenen.

Certificaat Cursus Solidity Programmeren

De deelnemers krijgen na het goed doorlopen van de cursus een certificaat Solidity Programmeren.

Duur: 2 dagen

Prijs: € 1499

[Open Rooster](#)



Inhoud Solidity Programmeren

In de cursus Solidity Programming leren de deelnemers de Solidity Language voor het schrijven van smart contracts die draaien op de Ethereum Virtual Machine (EVM). Solidity wordt niet alleen gebruikt voor het schrijven van smart contracts, maar ook om Decentralized Apps (DApps) te bouwen die op een decentralized platform zoals Ethereum draaien.

Intro Solidity

De cursus Solidity Programming begint met een uitleg over de Solidity syntax en de rol van de Ethereum Virtual Machine (EVM). Ook wordt aandacht besteed aan het opzetten van een ontwikkelomgeving voor Solidity.

Smart Contracts

Vervolgens wordt de structuur en werking van Smart Contracts besproken. Eveneens wordt de rol van state variables, functies en events uitgelegd. En er worden verschillende veelvoorkomende use-cases voor smart contracts gedemonstreerd.

Gegevenstypen

De verschillende data types die beschikbaar zijn in Solidity maken ook deel uit van de cursus Solidity Programming. Behandeld worden basis data types zoals integers, booleans, adressen en strings. En ook complexere data types en user defined data types zoals arrays, structs en enums komen aan bod.

Creating Contracts

Dan wordt het schrijven van een smart contract in Solidity behandeld. Aandacht wordt besteed aan constructors, functies en het afhandelen van events. Ook het logging in Solidity contracts en het testen en debuggen van contracts komt aan bod.

Contract Inheritance

Vervolgens wordt uitgelegd wat contract inheritance is en waarom het nuttig is. Function visibility, function modifiers, immutable en constant state variables komen in dit verband aan de orde. En het gebruik van polymorfisme in Solidity komt ook aan bod.

Decentralized apps (DApps)

Tot slot wordt aandacht besteed aan het creëren van decentralized apps (DApps) in Solidity. Gedemonstreerd wordt hoe je een DApp met behulp van Solidity en een front-end framework zoals React kunt bouwen. Interactie met een DApp met behulp van web3.js, een lokale blockchain zoals Ganache en een wallet zoals MetaMask wordt ook behandeld.

Modules Solidity Programmeren

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