Blockchain Developer Ethereum

CODICE	**DT0237**
DURATA	**3 gg**
PREZZO	**1.800,00 €**
EXAM	

DESCRIZIONE

Formazione per sviluppatori che desiderano imparare a creare smart contract e applicazioni decentralizzate sulla blockchain di Ethereum. Ideale per aziende che vogliono esplorare nuove opportunità nel campo delle tecnologie decentralizzate.

OBIETTIVI RAGGIUNTI

Al termine del corso i partecipanti saranno in grado di:

- Approfondire i benefici e svantaggi delle tecnologie Blockchain
- Scrivere e leggere Smart Contract sicuri utilizzando il linguaggio Solidity
- Comprendere il concetto di Gas e come viene calcolato il suo costo
- Comprendere come sono strutturate le transazioni
- Comprendere ed evitare bug costosi ed errori di programmazione
- Creare applicazioni che interagiscono con gli Smart Contract
- Web3.js
- Installare, configurare e usare il framework Truffle
- Scrivere test con il framework Truffle
- Comprendere il workflow di sviluppo e deployment
- Utilizzare e creare private network, test network and the public main network

TARGET

Professionisti sia in ambito aziendale che IT che mirano ad avere un ruolo professionale come architetti/sviluppatori blockchain su Ethereum

PREREQUISTI

- È necessario avere una conoscenza di base di JavaScript e HTML.
- È consigliato avere una conoscenza di base di C ++ o Java, data types, git repositories.

CONTENUTI

Blockchain and smart contract basic

- What is Blockchain and how does it work?
- Centralized vs. Decentralized vs. Distributed
- Blockchain vs. Databases
- Bitcoin vs Ethereum
- What are Smart Contracts?
- How are Smart Contracts used?

Smart contract programming basics

- Advantages and Drawbacks of Smart Contracts
- High-Level Language vs. Low-Level
- Smart Contracts with Solidity
- Types of Variables in Solidity
- Function/Variable Visibility
- Smart Contract Constructors
- Setter- and Getter-Functions

Understanding decentralized information and Web3

- Blockchain Access structures and Architectures
- Blockchain Access vs. centralized RESTful API
- Understanding Web3.js API
- Understanding Transactions and Consensus
- Private Keys, Public Keys and Signatures
- Understanding privacy on pubic Blockchains
- Understanding the architecture of KeyStore’s such as MetaMask or MIST

Basics of Ethereum and the EVM

- Ethereum Denominations
- Understanding EVM and the ABI Interface
- Calls vs. Transactions
- Concurrency and Events
- Use cases of Events
- LAB
    - Install and Use Ganache
    - Work with Web3.js
    - Define Events
    - Listen and React to Events

Solidity Advanced: Modifiers, Mappings, Structs and Inheritance

- Understanding Functions, Mappings and Structs

- When to use Modifiers
- Libraries vs. Inheritance
- Understand and use Modifiers
- Add Mappings and Structs
- Use Inheritance to increase auditability

Understanding deployment and costs

- Understand Development and Deployment Cycles
- Understanding Solidity Compilation and Deployment
- Gas and Gas-Costs
- Upgradeability and Data Migration Techniques
- Understand the moving Parts: Compiler, Blockchain, API, KeyStore

Mining, proof of work vs. Proof of authority

- What is Mining in PoW?
  - How blocks are generated
  - PoW vs. PoA (vs. PoS)
- Understanding Go-Ethereum or Ganache/TestRPC for local development
- Understanding Private Blockchains vs. Public Blockchains
- LAB
  - Installing and using Ganache
  - Installing and using Go-Ethereum
  - Connecting to Ganache/Go-Ethereum from Remix and Web3.js
  - Interact with the Blockchain from HTML/JS

Working in teams, testing and versioning

- Understanding what Truffle is
- How Manage Code for Teams
- Understanding Migrations
- Understanding Unit-Testing with Truffle
- LAB
  - Download and Setup Truffle
  - Adapt the standard Truffle-Project
  - Write A Unit Test