Introduction to Blockchain Technology

edunet foundation

Introduction

- Most of the traditional system like Bank or insurance working on a centralized database.
- If the centralized database crashes, then the entire system will crash in the traditional centralized System.
- Blockchain works on decentralized approach.

How Traditional Banking System Works?

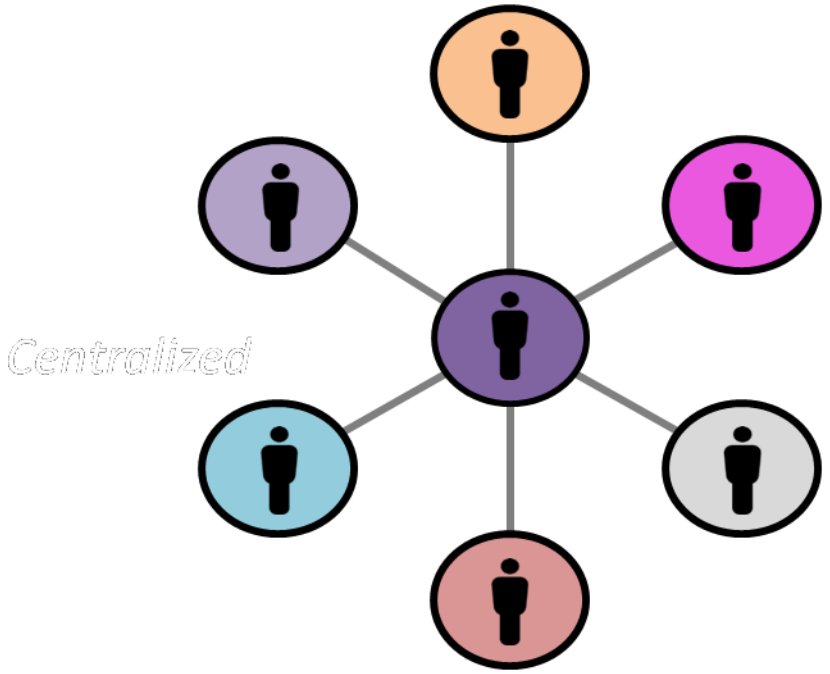
- A banker or bank is a financial institution whose primary activity is to act as a payment agent for Customers to borrow ad Lend money

- It is an institution for receiving, keeping, and lending money.

edunet foundation

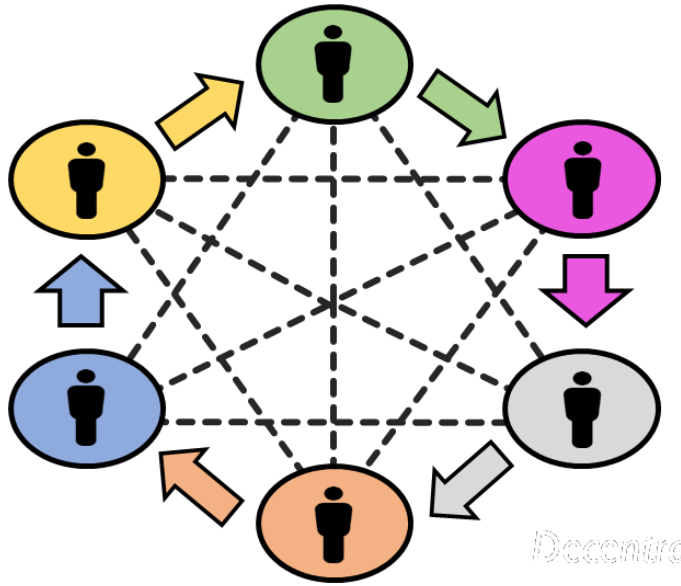
Centralized System

- Centralized banking operation is that system of banking, where processing of all transactions has to be carried out from a central location.
- This is a Centralized System.
- Every one connected to that centralized Database.



Decentralized System-Blockchain

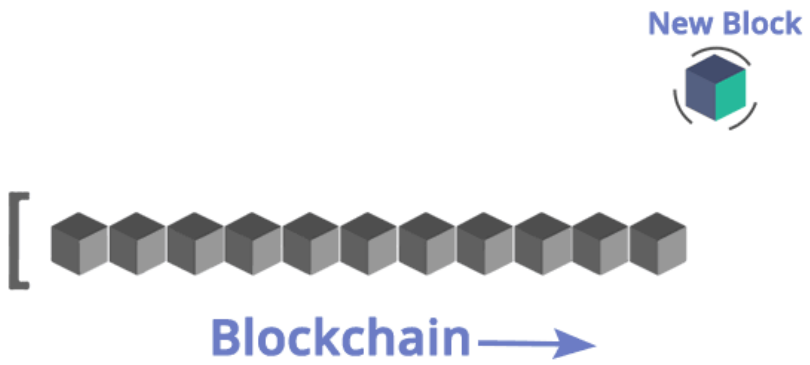
- If the centralized server crashes ,the entire System will crash that leads to loss of access or information from the centralized system.
- Blockchain works on decentralized fashion.
- Blockchain eliminates all intermediaries.
- Blockchain is decentralized Distributed ledger.
- In Blockchain data is stored in a public ledger, so that anybody can view the transactions



A distributed ledger that allows users to verify transactions without an intermediary

What is Blockchain

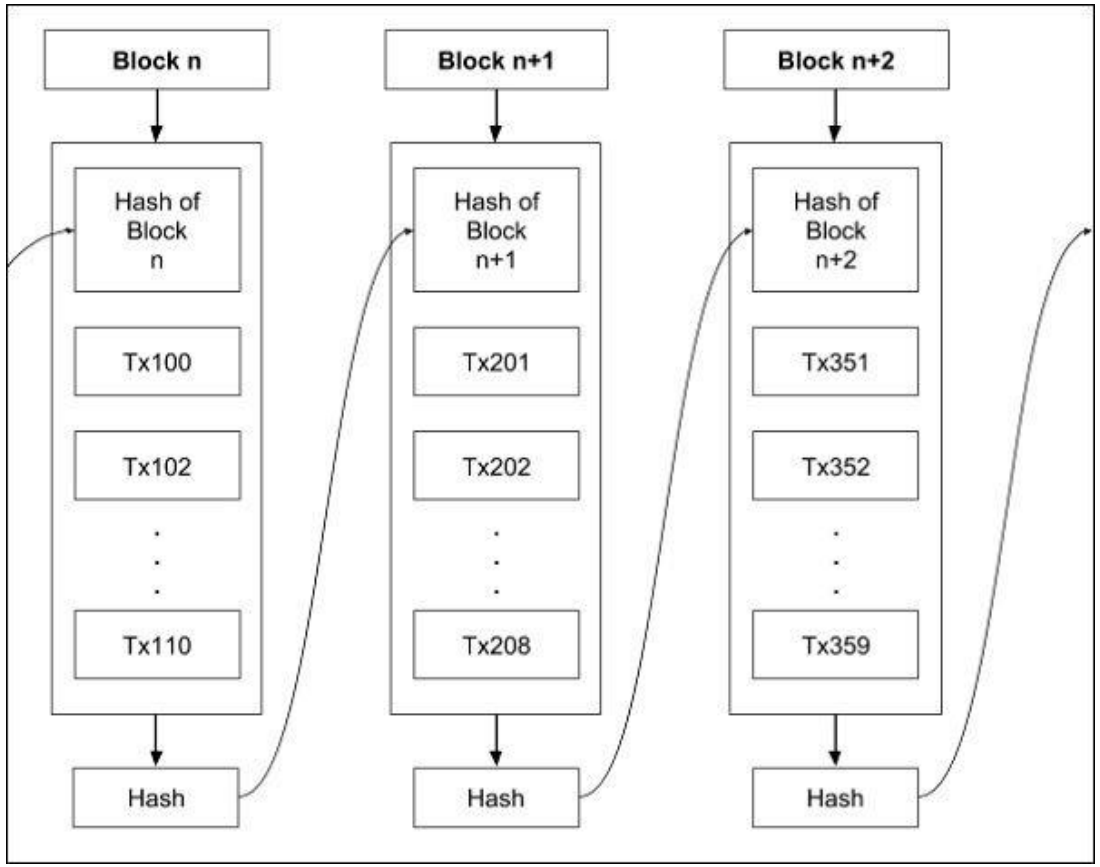
- Blockchain was invented by Satoshi Nakamoto.
- Remember that Blockchain and Bitcoin are different. Blockchain is a Technology and Bitcoin is a Crypto currency .
- Blockchain is a chain of blocks that contains transactions and data.
- Block contains information and transactions.
- When transaction happens , new block is created, and it is appended to the blockchain



How does a Blockchain look like?

- All transactions are recorded in the block. Each block in the blockchain has a header and Transaction. Header contains information about the block and Transactions are the data stored in side the block.

- These blocks are chained together to form a blockchain. Whenever a new block is crated this newly created block is also added to the blockchain.

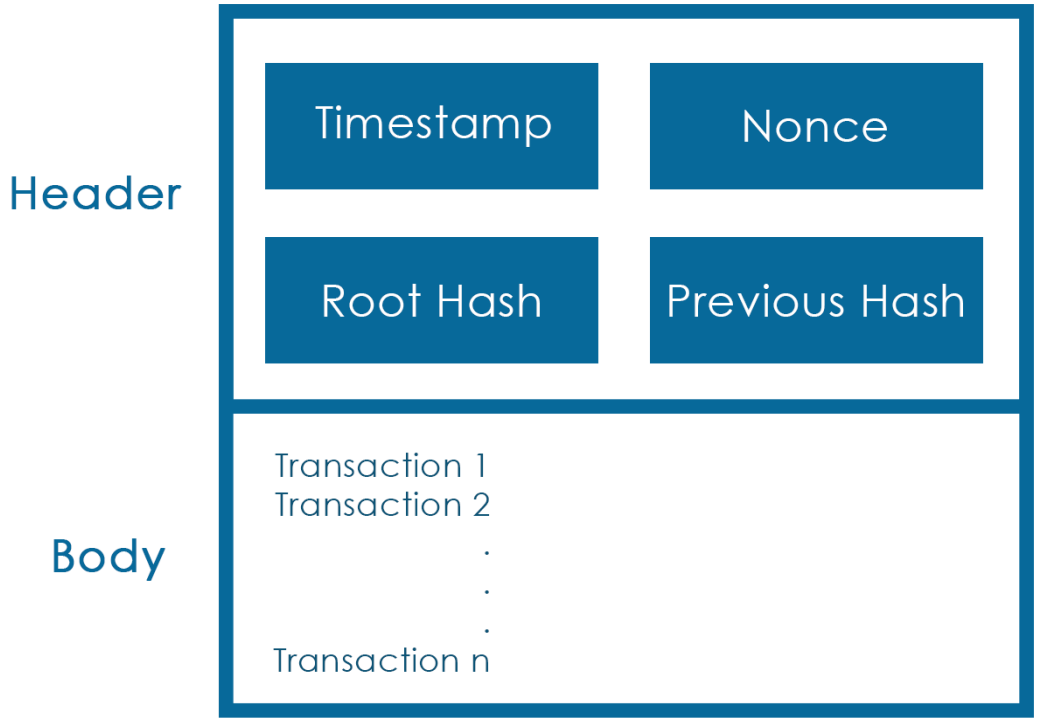


edunet foundation

Structure of a Block

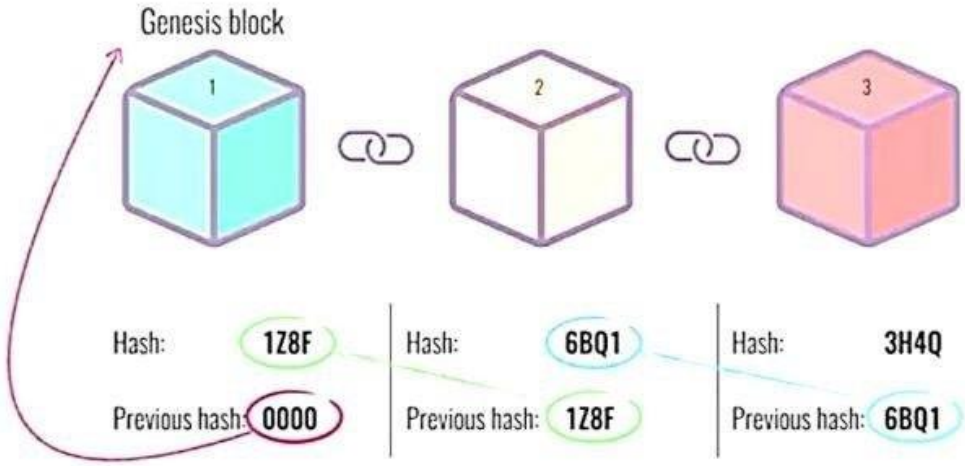
- This is how a block in the blockchain looks like. It has two parts Header and Body.

  - **Timestamp**: The time at which the block was created.
  - **Nonce**: This is a random number. It is used by the miner to mine the block.
  - **Root Hash**: This is the hash value of all transactions inside the block generated by merkle tree algorithm.
  - **Previous Hash**: It is the hash value of the previous block.



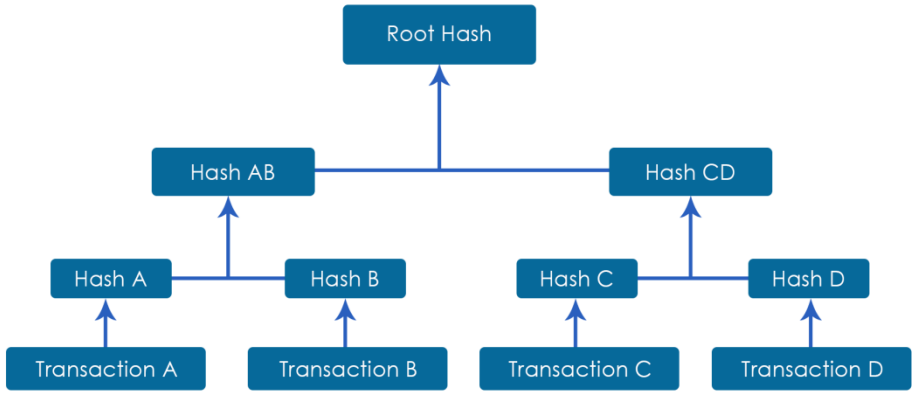
Genesis Block

- The Genesis Block refers to the first block of a blockchain, specifically the Bitcoin blockchain.
- It is the foundation upon which all subsequent blocks are built.
- The Genesis Block is unique because it has no predecessor and contains a hardcoded reward for its miner.



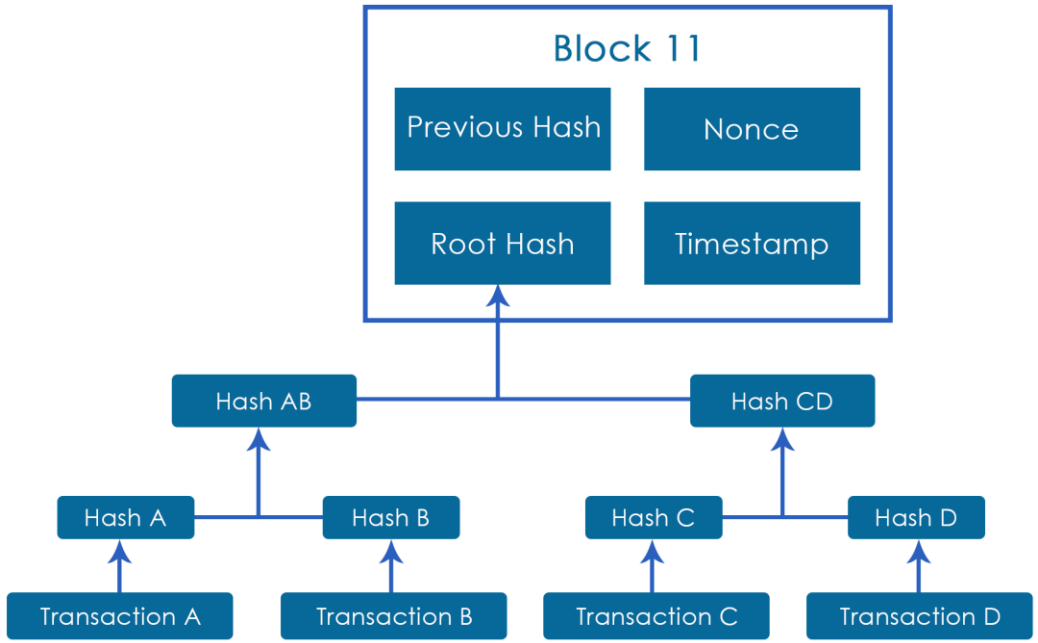
How to create root hash in a Block

- Merkle tree is used to find root hash of the block.
- Merkle tree is a binary tree , generated from hash value of all transaction in the block



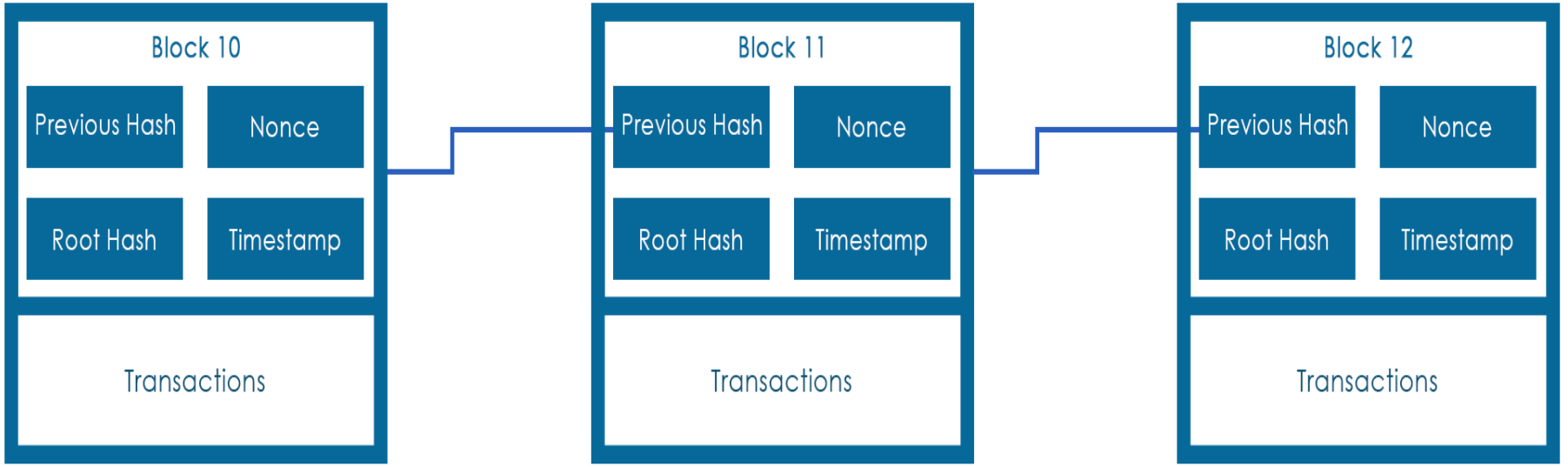
Header of a Block in Blockchain

- This is the complete structure of Block no 11



How blocks form a chain?

- **Previous hash is another component in block, which contains the hash value of previous block**

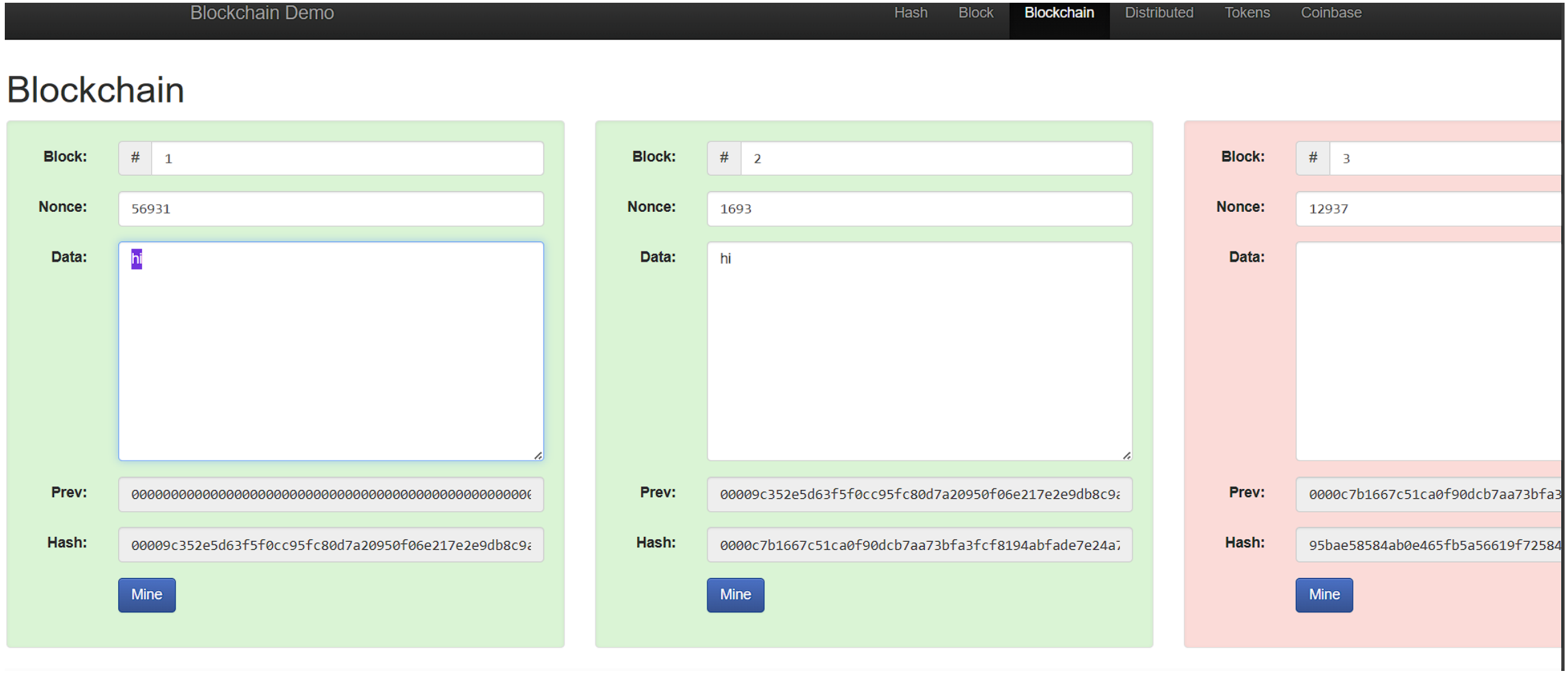


• Hash value of block 10 is stored in Block 11. Hash value of Block 11 is stored in Block 12 and so on. This is how blocks are chained together in a blockchain.

How blockchain works

edunet
foundation

Blockchain Demo

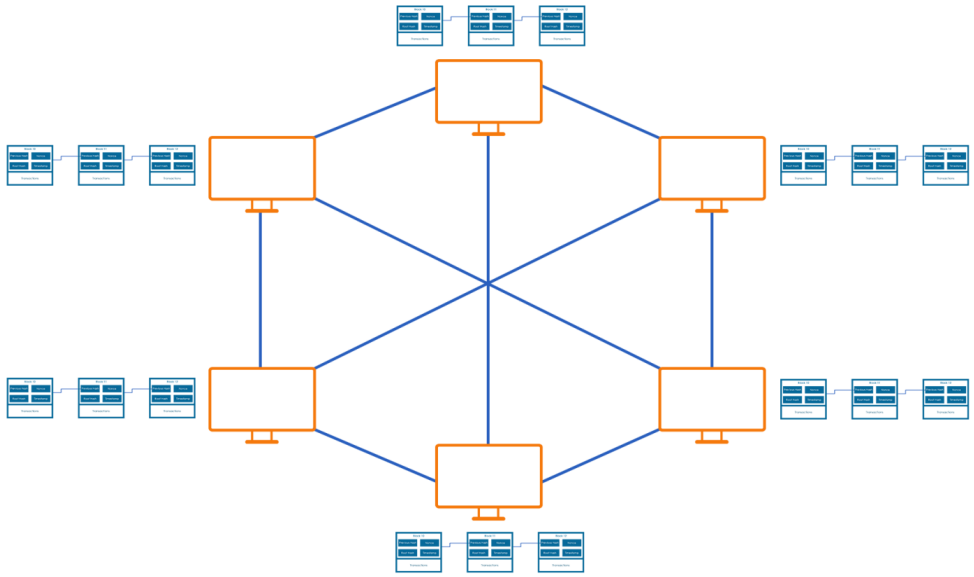


https://andersbrownworth.com/blockchain/blockchain

edunet foundation

Blockchain Network

- A blockchain network is a decentralized, distributed digital ledger that records transactions across multiple computers.
- It ensures transparency, security, and immutability by using cryptographic techniques. Each transaction is grouped into a "block," which is linked to previous blocks, forming a chain.
- This structure prevents data manipulation and eliminates the need for intermediaries, making it widely used in cryptocurrencies, supply chains, and various other applications.



Real world application

Blockchain Vs Database

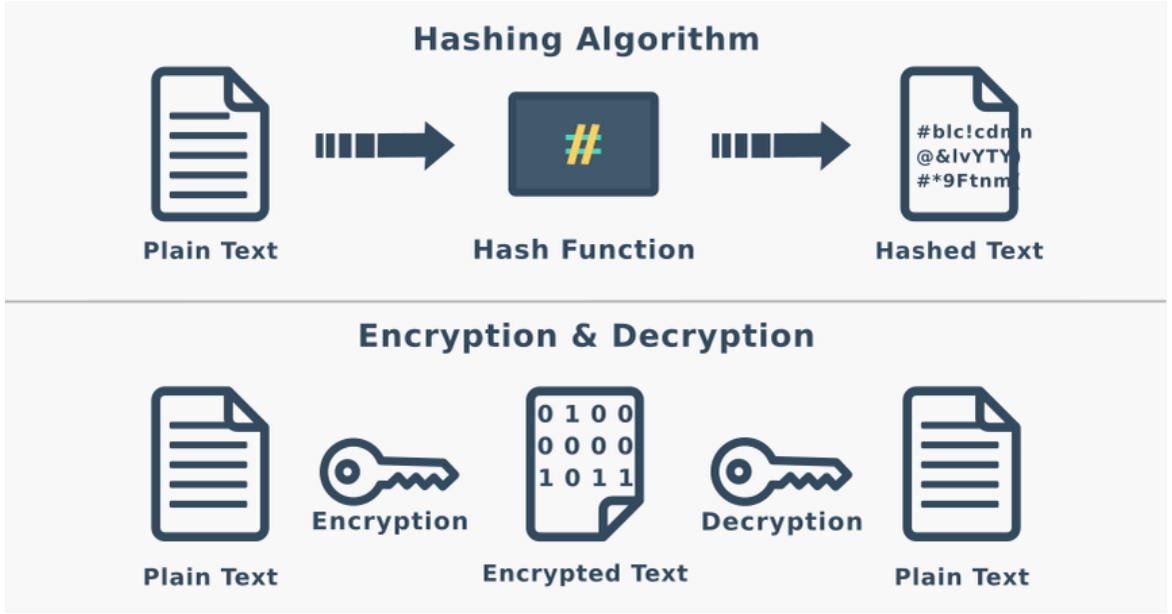
Sl	Blockchain	Database
1.	Blockchain is decentralized	The database is centralized
2.	Blockchain is permissionless.	The database required permission
3.	Blockchains are slow.	Databases are fast.

Encryption, Decryption and Hashing

- **Encryption** is the process of converting data into a secure format to prevent unauthorized access.
- **Decryption** is the process of converting encrypted data back into its original, readable format.
- **Hashing** is the process of converting data into a fixed-size string of characters, typically for data integrity verification.
- **Two Types of Encryption**

  Symmetric Encryption

  Asymmetric Encryption

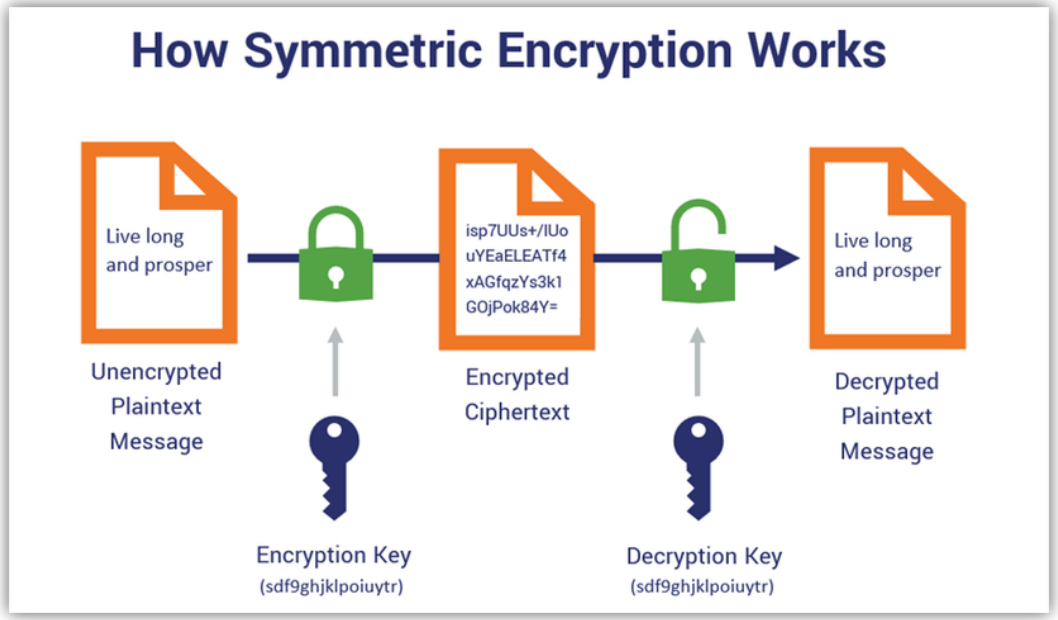


edunet foundation

Encryption, Decryption

- **Symmetric Encryption**

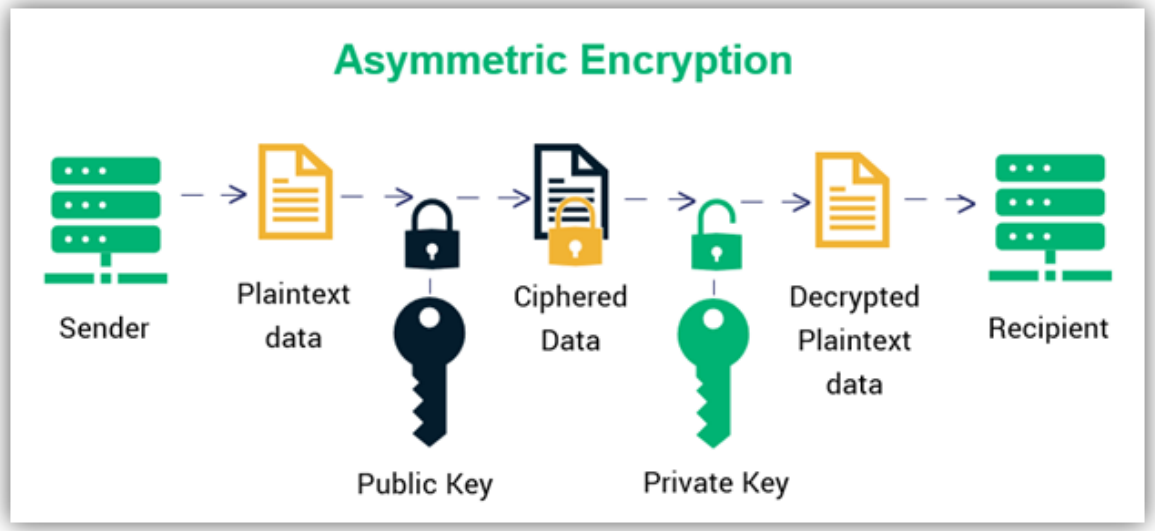
  Symmetric encryption is a type of encryption where the same key is used for both encrypting and decrypting data. It is fast and efficient but requires secure key distribution to ensure confidentiality.



Encryption, Decryption

Asymmetric Encryption

Asymmetric encryption uses a pair of keys: a public key for encryption and a private key for decryption. It enables secure communication without the need to share the private key, ensuring confidentiality and authentication.



AES Encryption

Enter Plain Text to Encrypt

hello

Select Cipher Mode of Encryption

CBC

Select Padding

PKCS5Padding

Enter IV (Optional)

Enter initialization vector

Key Size in Bits

128

Enter Secret Key

wertyuioopasdfgh

Output Text Format (•) Base64 ( ) Hex

Encrypt

AES Encrypted Output

zewMFdtu4VGzU/8mg+1Gqg==

AES Decryption

AES Encrypted Text

zewMFdtu4VGzU/8mg+1Gqg==

Select Cipher Mode of Decryption

CBC

Select Padding

PKCS5Padding

Enter IV Used During Encryption(Optional)

Enter initialization vector

Key Size in Bits

128

Enter Secret Key used for Encryption

wertyuioopasdfgh

Output Text Format (•) Plain-Text ( ) Base64

Decrypt

AES Decrypted Output

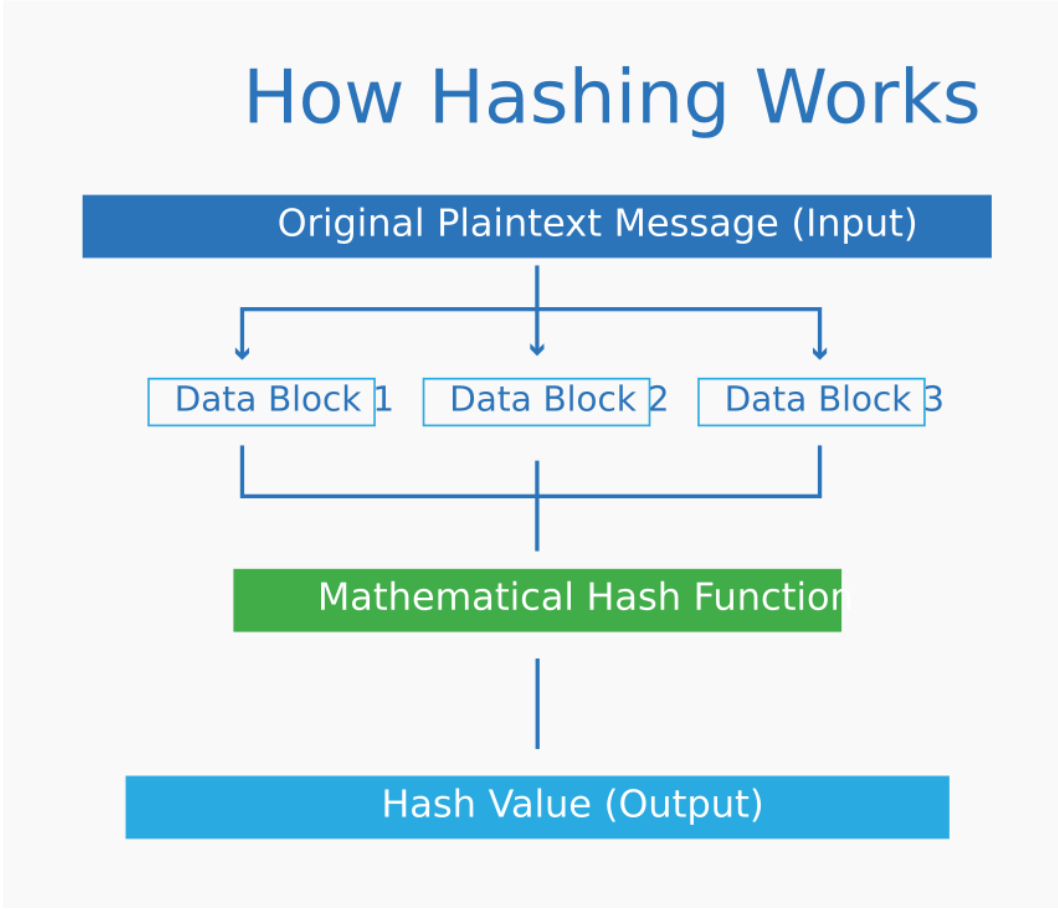
hello

https://www.devglan.com/online-tools/aes-encryption-decryption

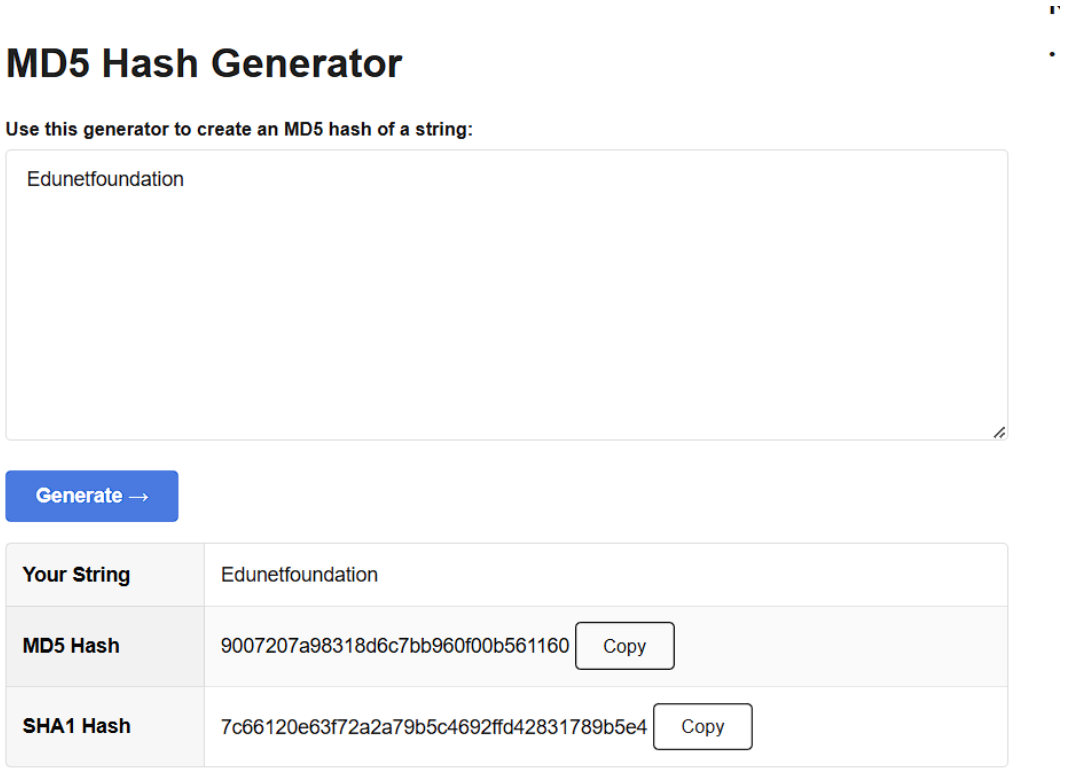
edunet foundation

Hashing

- Hashing is the process of converting input data into a fixed-length string of characters, typically for data integrity verification.
- It is a one-way function, meaning the original data cannot be easily retrieved from the hash.



Hashing

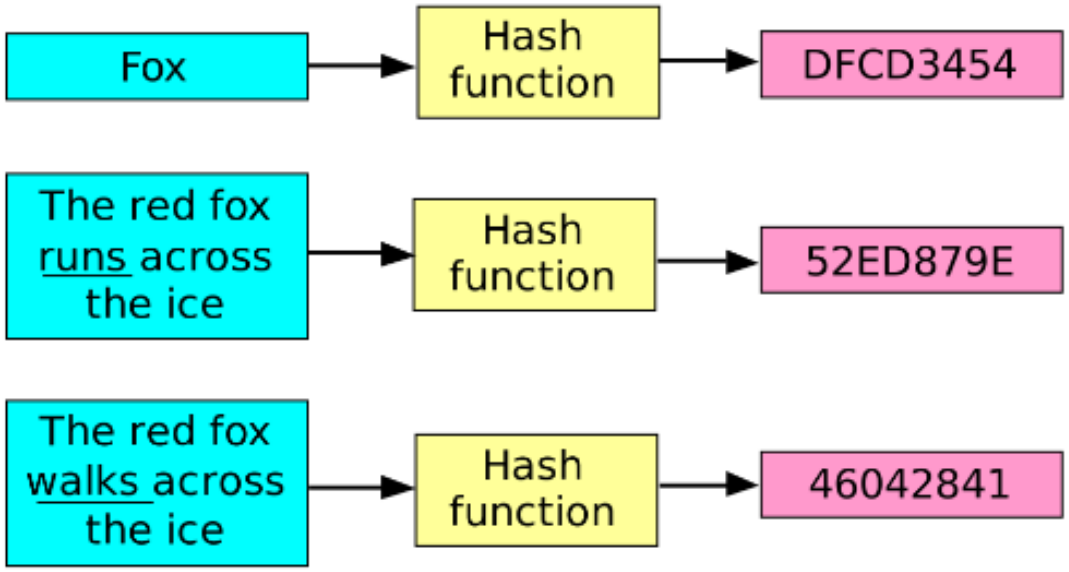


https://www.md5hashgenerator.com/

edunet foundation

Features of Hashing Algorithm

- **Irreversible :** From the hashed output, It is impossible to produce the original message
- **Deterministic:** Same hashing algorithm produce same output for given input.
- **Avalanche Effect:** Any small change in the input data would produce an entirely different hash as the output.
- **Fast**: Hashing is a very fast operation.



What is Ethereum?

- Ethereum is a permissionless blockchain
- In Ethereum blockchain we can implement our business logic
- This Business logic is called smart contract .
- Since Ethereum is decentralized system, there should be consensus algorithm.
- Ethereum using Proof of Work Consensus algorithm.

Ethereum & Bitcoin-Comparison

Ethereum	Bitcoin
Permissionless& Public	Permissionless& Public
Crypto currency is ETH(Ethereum)	Crypto currency is BTC(Bitcoin Crypto currency)
Support smart contract	Not supporting smart contract
Block creation Time is 15 sec	Block creation Time is 10 min
Proof of Work (PoW) consensus algorithm	Proof of Work (PoW) consensus algorithm
Can issue new tokens	Cannot issue new tokens

Smart Contract

What is Smart Contract

A **smart contract** is a program that run on blockchain.
Since smart contract is deployed on Blockchain, anybody can execute the smart contract.

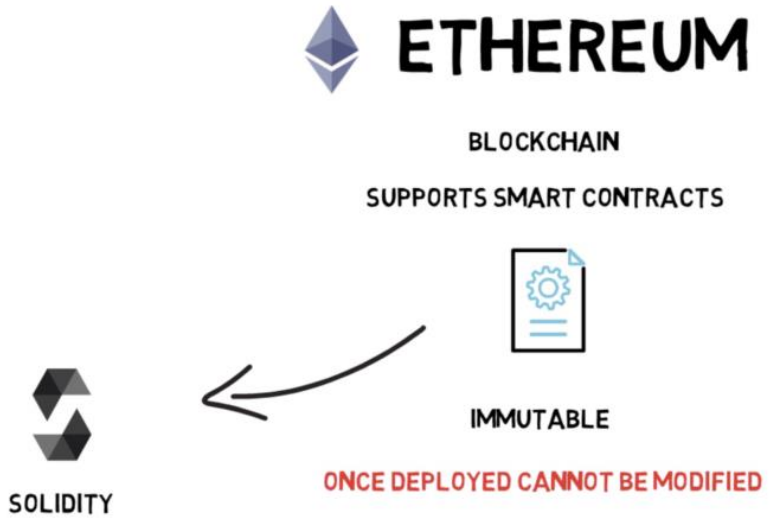
https://academy.shrimpy.io/post/the-best-smart-contract-platforms

In Ethereum Blockchain we can write smart contract in **Solidity** Language

edunet foundation

Benefits - Smart Contract

- Works on Blockchain
- No intermediaries.
- Immutability.
- Transparent
- Reduced Fraud and Manipulation



Solidity

edunet
foundation

Smart Contract with Solidity

Solidity is most popular language for writing smart contract in Ethereum Blockchain

- Gavin Wood created solidity language.
- Solidity is Object Oriented Language
- It is a high level language

Features of Solidity

- Object Oriented
- Influenced by **C++, Python and JavaScript**.
- Extension is **.sol**
- Supports:
  - Inheritance
  - Libraries
  - Struct,mapping

Sample Solidity Program

```
// SPDX-License-Identifier: GPL-3.0
pragma solidity 0.7.6;
contract Sample{
    string public name="Hello
   World";
}
```

//Comments

//Pragma

//Contracts

//Visibility of variable

Comments in Solidity

Comments is used to describe the code or functions. It will not execute

- // SPDX-License-Identifier: GPL-3.0 is comment

Solidity supports two types of comments

1. Single-line comments (//)
2. Multi-line comments (/*...*/)

Eg :

```
// This is a single-line comment.

/*
This is a
multi-line comment.
*/
```

Pragma statement Solidity

- Usually this is the first line of the solidity file.
- pragma is used in every solidity file to mention the compiler version.

Syntax: `pragma solidity <<version number>>;`

Eg: `pragma Solidity 0.7.6;`

Contract in Solidity

- Every solidity file should have at least one contract.
- Contract is defined with key word contract

Syntax:

```
contract <<name of the contract>>{
}
```

Eg:

```
contract Sample{
}
```

```
// SPDX-License-Identifier: GPL-3.0
pragma solidity 0.7.6;
contract Sample{
   string public name="Hello World";
}
```

Visibility mode in Solidity

- Visibility mode can be applied to a variable in the contract.
- It can be public or private

Syntax:

```
<<Datatype>> visibility mode variable_name;
```

Eg:

string public name="Hello World";

```
// SPDX-License-Identifier: GPL-3.0
pragma solidity 0.7.6;
contract  Sample{
   string public name="Hello World";
}
```

Visibility mode - Public

- If any variable is declared as public , then this variable can be accessed from anywhere.

Eg:

```
string public name="Hello World“;
```

```
// SPDX-License-Identifier: GPL-3.0
pragma solidity 0.7.6;
contract Sample{
   string public name="Hello World";
}
```

Visibility mode - Private

- If any variable is declared as private, then this variable can access only inside that smart contract.

Eg:

string private name="Hello World“;

```
// SPDX-License-Identifier: GPL-3.0
pragma solidity 0.7.6;
contract  Sample{
    string private name="Hello
     World";
}
```

edunet foundation

Compiling and Deploying Smart contract

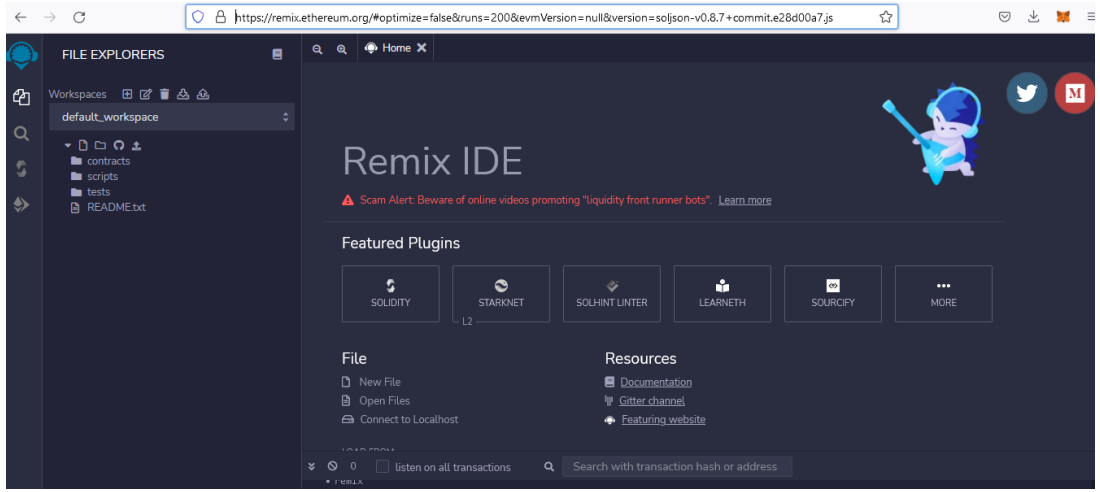
Compiling smart contract

To compile and deploy smart contract we can use Remix IDE

Remix IDE is a web based open source platform to compile and deploy the smart contracts

No need to install any thing our computer

Visit https://remix.ethereum.org

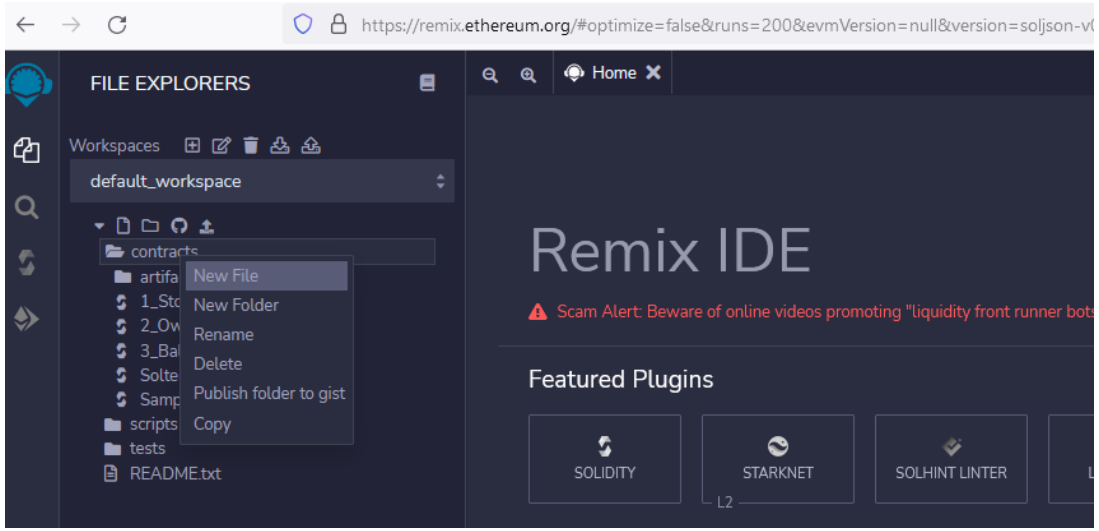


edunet foundation

Compiling smart contract

Create a new file in contracts folder(right click contract folder and click new file)

Give any name to the file. But remember extension should be **.sol**

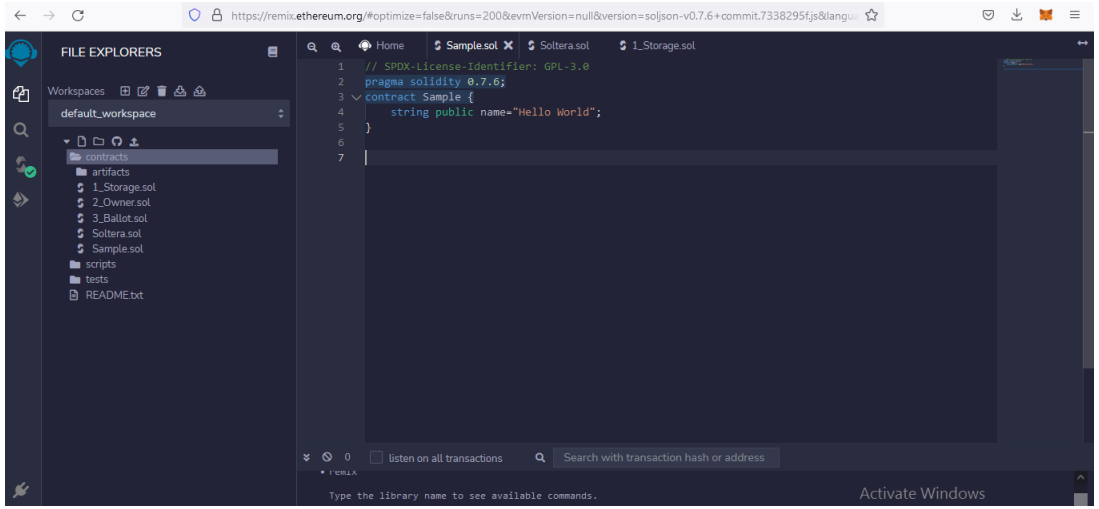


edunet foundation

Compiling smart contract

Open Sample.sol and write the program

```
// SPDX-License-Identifier: GPL-3.0
pragma solidity 0.7.6;
contract Sample{
  string public name="Hello World";
}
```

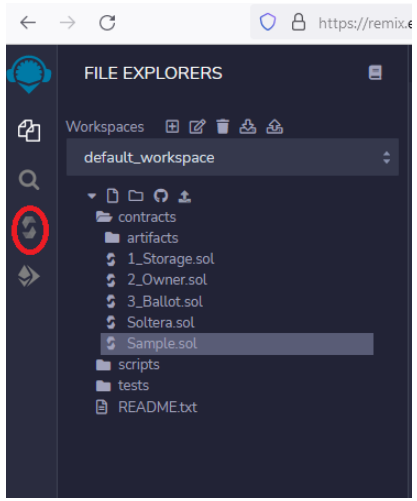


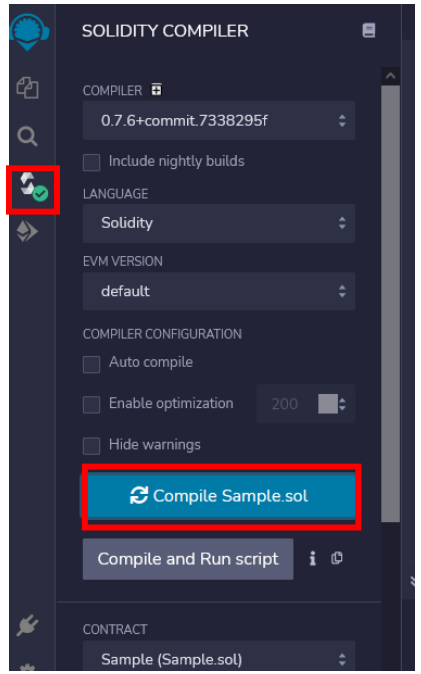
edunet foundation

Compiling smart contract

To compile the smart contract click on the compiler on remix IDE

After selecting the solidity compiler icon, you can see the screen like this. And you can compile the smart contract by clicking compile



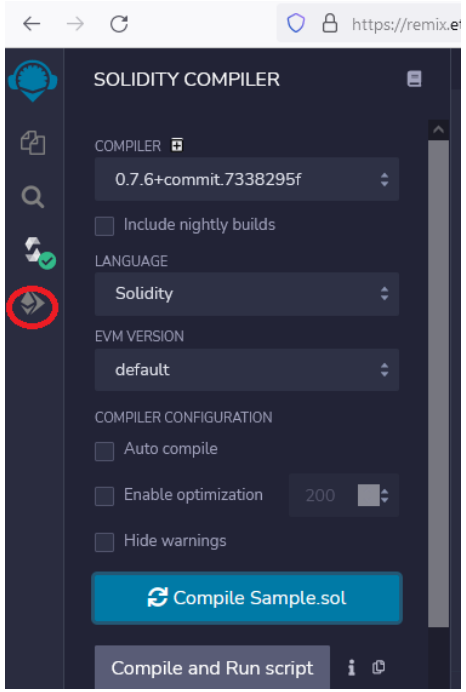


edunet foundation

Deploying smart contract

After successful compilation , we can deploy the smart contract to the blockchain

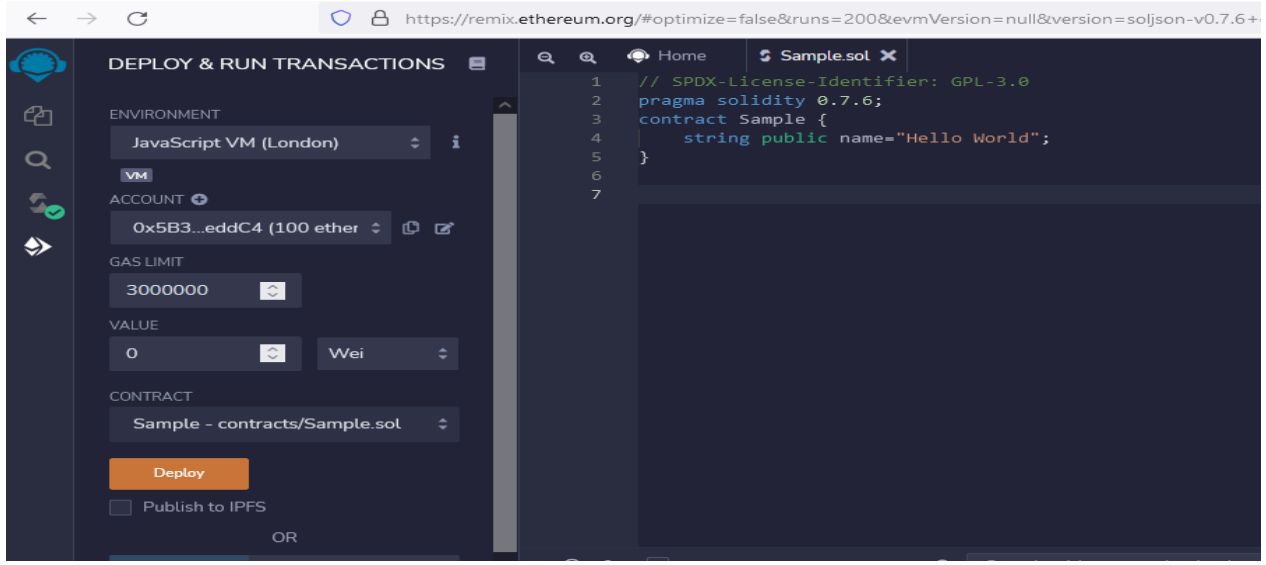
Select the Deploy icon on the Remix IDE. Its shown in red color



edunet foundation

Deploying smart contract

When we select deploy button screen looks like this



Remix IDE provide different types of deployment

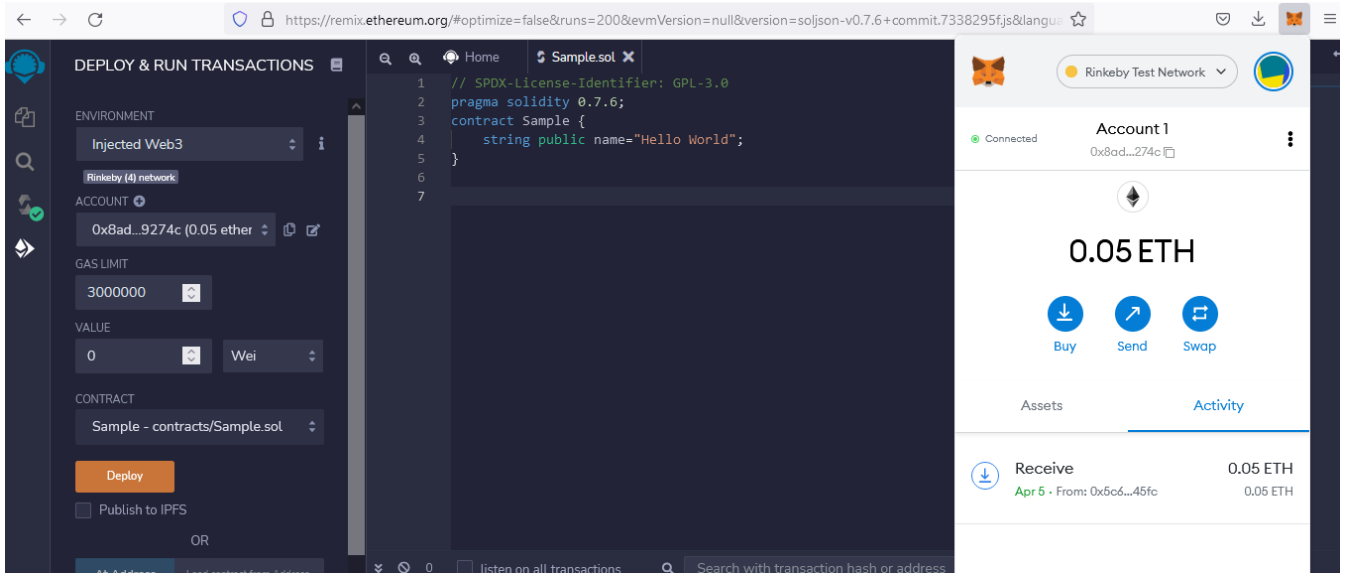
1. Javascript VM
2. Injected Web3

edunet foundation

Deploying smart contract

- Select injected Web3
- It will pop up our wallet .If wallet is not installed on browser install from here https://metamask.io/

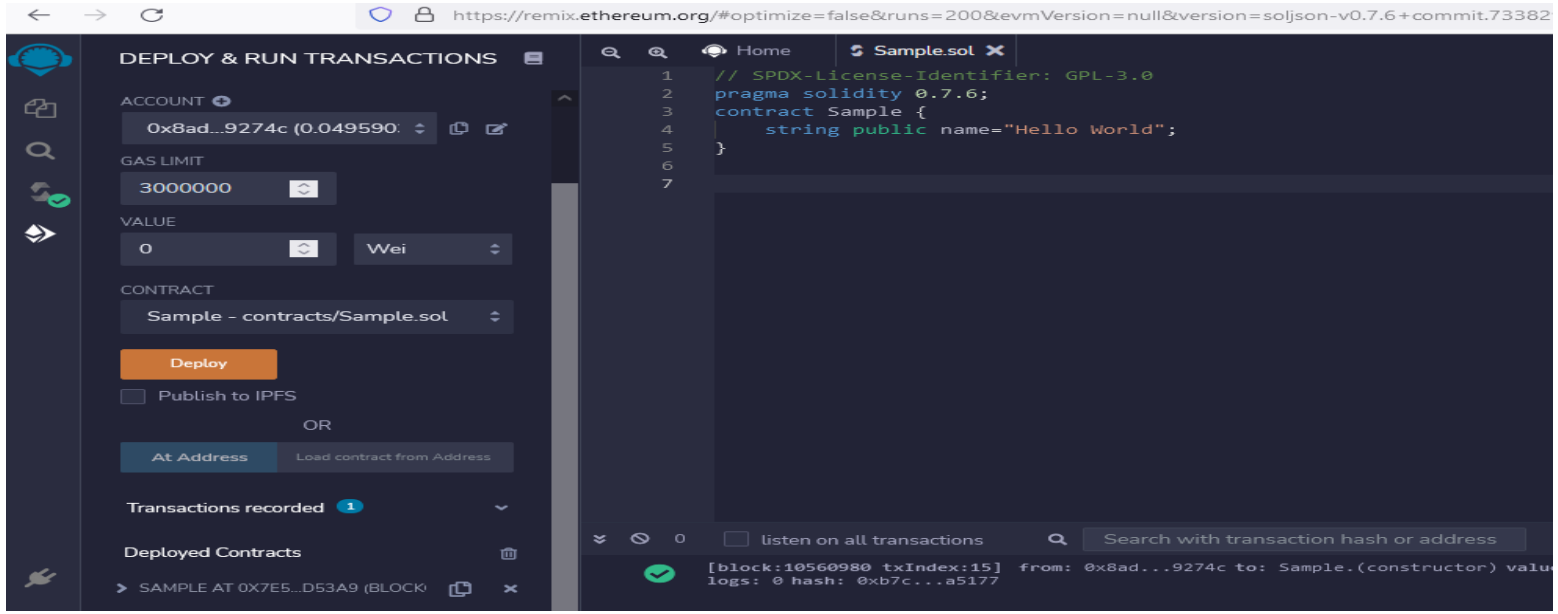
  (See the appendix for more about metamask installation and getting free ETH)
- Connect the wallet in any blockchain network



edunet foundation

Deploying smart contract

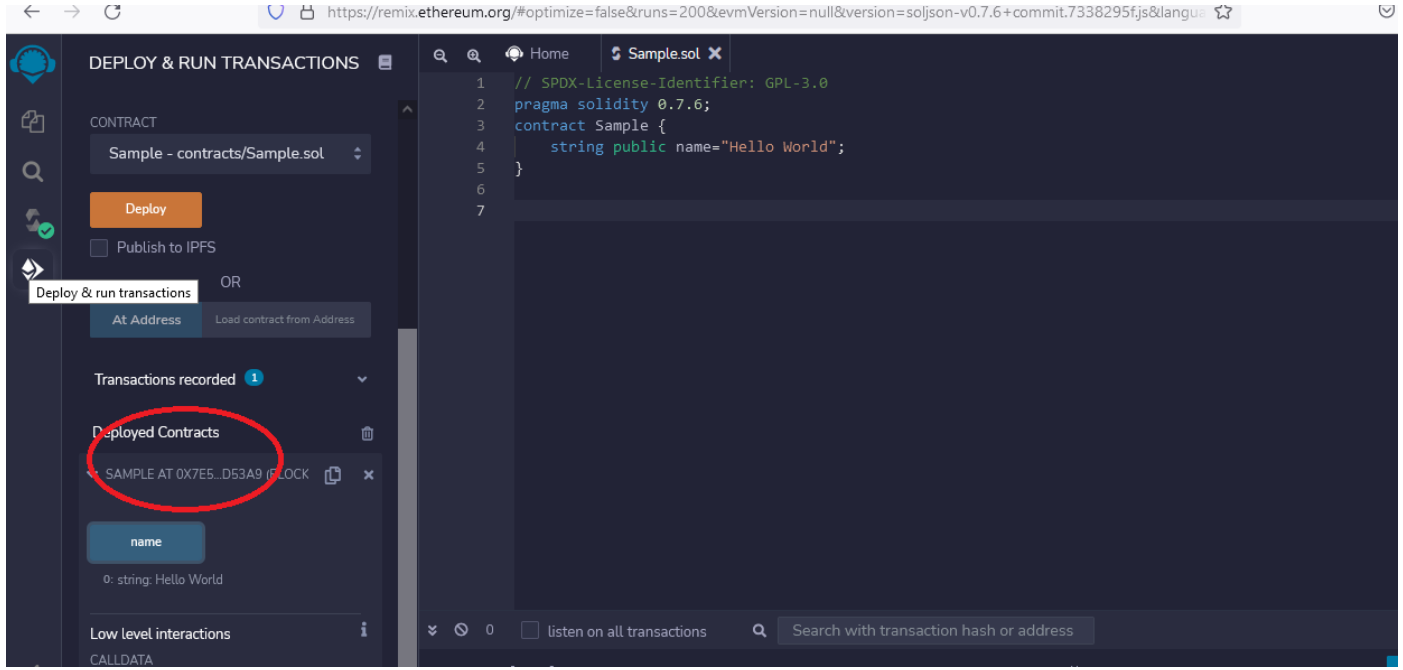
- Now click on the Deploy Button on the Remix IDE



edunet foundation

Deploying smart contract

- We already deployed smart contract on blockchain
- Now we can read the values from the blockchain
- Click on the deployed contracts
- Click on name
- It will display Hello World from the blockchain



edunet foundation

Advanced Solidity Concepts

Different Data types

Booleans – bool

Example

bool public flag=true;

Integer - int, int8, int16, int32, .. int256

Example

int public sum=0;;

Address- address

Example

address public owner=0x8adba3Eef0cf16E3F58741447932f23a6719274c;

Unsigned Ineger - uint, uint8, uint16, uint32, .. uint256

Example

uint public age=25;

edunet foundation

Different Data types

Structure –struct

```
struct Student{
    string name;
    uint age;
    int mark;
  }
  Student s1=Student("Adarsh",25,50);
  Student s2=Student("Remya",20,45);
```

Different Data types

- **mapping(address => uint256) balances**

Associates each address with a uint256 value representing its balance.

- **Functions:**
- **setBalance(address, uint256)**: Assigns a balance to an address.
- **getBalance(address)**: Retrieves the balance of an address (can also use the b
- **resetBalance(address)**: Resets an address's balance to zero.
- **Example Usage:**

1. setBalance: Store a balance for an address.
2. getBalance: Retrieve the balance of an address.
3. resetBalance: Reset the balance of an address to zero.

Functions in Solidity

Functions are the executable units of code within a contract. It should have a set of steps for a particular task

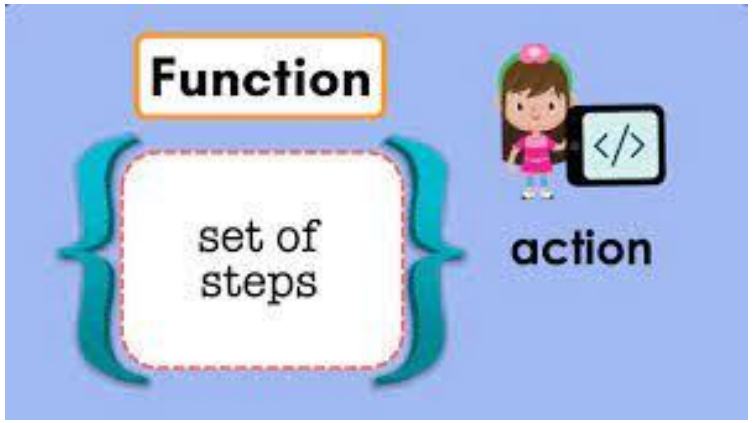
Syntax:

function function_name(params)visibility{

}

Example:

```
Function updateName(string memory _n) public {
        name=_n;
}
```



edunet foundation

Functions in Solidity

Smart contract for storing a string and update the value using function

```
// SPDX-License-Identifier: GPL-3.0
pragma solidity 0.7.6;
contract Sample {
  string public name="Hello World";
  function updateName(string memory _n) public{
    name=_n;
  }
}
```

Array's in Solidity

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.17;

contract arrays{
    /* Array */
    /* uint[]  public arrayname = [1,3,344]*/
       uint[] public array;

       function addElements(uint _element)public {
          array.push(_element);
       }

  function getArray()public {
    array.pop();
  }
```

Branching Statements - Solidity

IF Statement in solidity

Syntax :

```
if(condition) {
        //statements
}
```

Example:

```
if(a==2){
        //statements
}
```

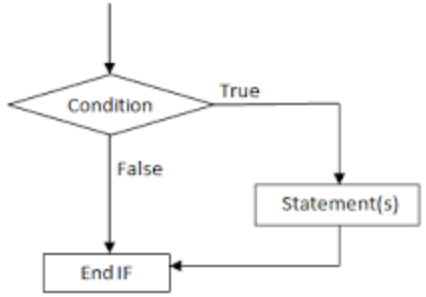


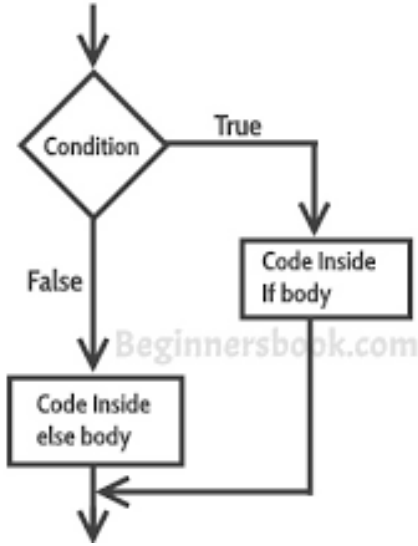
fig: Flowchart for if statement

Branching Statements - Solidity

IF ELSE Statement

Syntax :

```
if(condition) {
        //statements1
}
else {
        //statement2
}
```



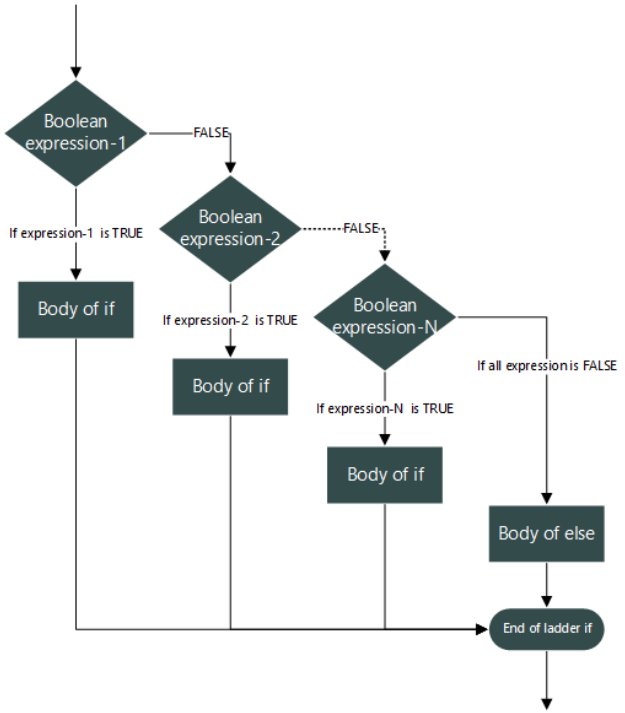
edunet foundation

Branching Statements - Solidity

IF ELSE IF.. ELSE Statement

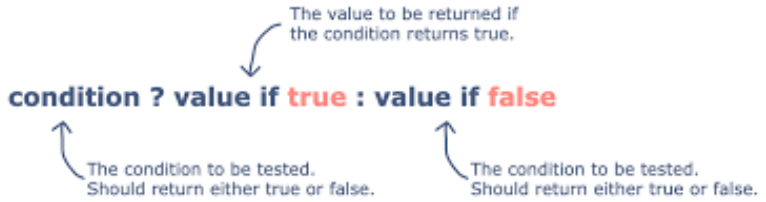
Syntax :

```
    if(condition1) {
            //condition1 body
    }
    else if(condition2) {
            //condition2 body
    }
    ..
    ..
    else{
            //else body
    }
```



Ternary Operator in Solidity

- Exp1?exp2:exp3



Example

A=b>2?5:0;

Looping Statements - Solidity

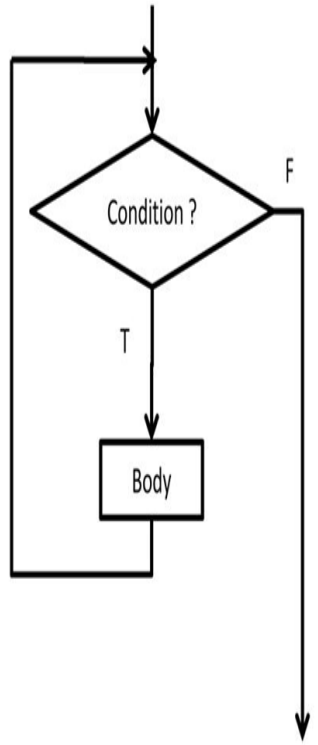
WHILE Loop

Syntax:

```
while(condition){
        //statements
}
```

Example:

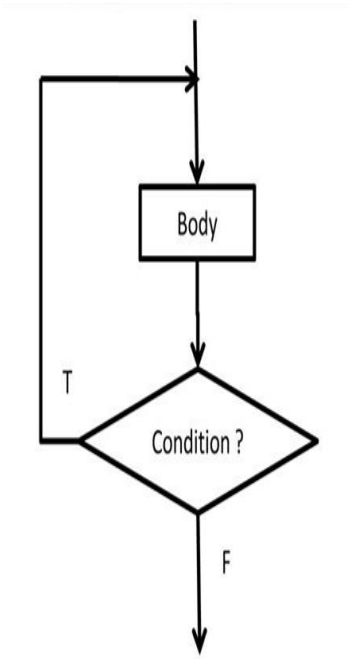
```
While(a<5){
    a=a+1;
}
```



edunet foundation

Looping Statements - Solidity

DO WHILE Loop



Note : semi colon after while

Syntax:

```
do {
        //statement
}
while(condition);
```

Example:

```
do{
    a=a+1;
}
While(a<5);
```

Looping Statements - Solidity

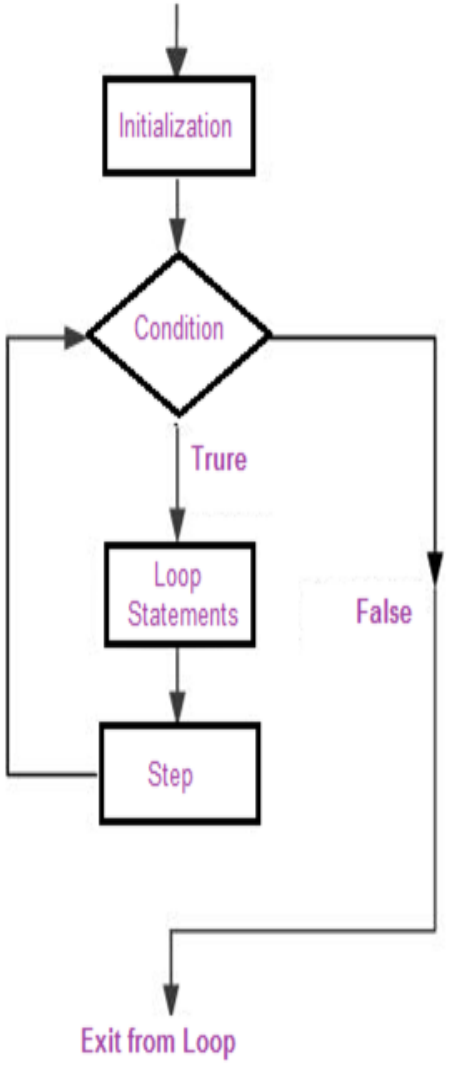
FOR Loop

Syntax:

```
for(initialization; condition; updation){
        //body of loop
}
```

Example:

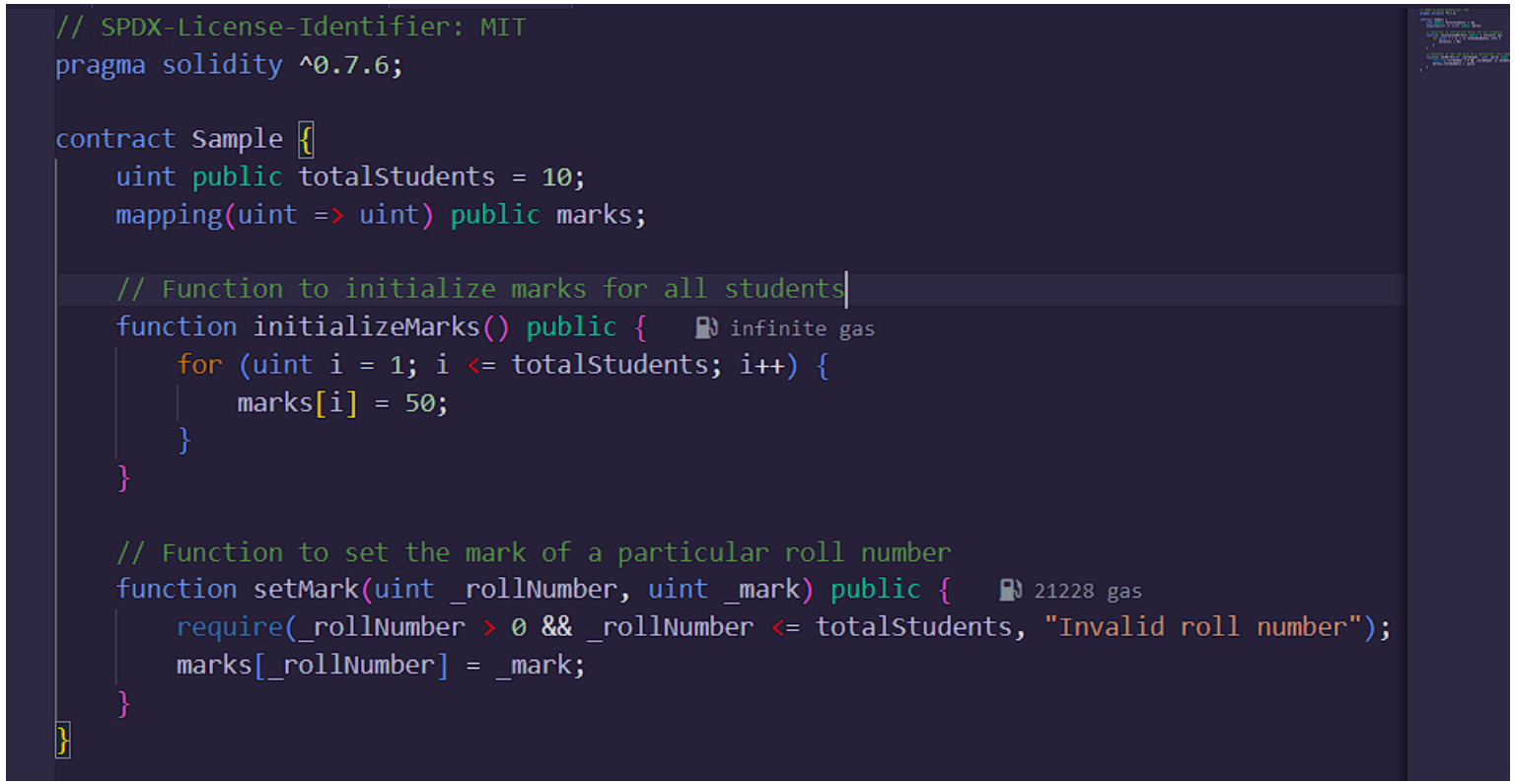
```
for(uint i=1;i<=10;i++){
        marks[i]=0;
}
```



edunet foundation

Sample smart contract

Smart contract for storing students mark



```
// SPDX-License-Identifier: MIT
pragma solidity ^0.7.6;

contract Sample {
    uint public totalStudents = 10;
    mapping(uint => uint) public marks;

    // Function to initialize marks for all students
    function initializeMarks() public {    infinite gas
        for (uint i = 1; i <= totalStudents; i++) {
            marks[i] = 50;
        }
    }

    // Function to set the mark of a particular roll number
    function setMark(uint _rollNumber, uint _mark) public {    21228 gas
        require(_rollNumber > 0 && _rollNumber <= totalStudents, "Invalid roll number");
        marks[_rollNumber] = _mark;
    }
}
```

edunet foundation

Block chain Applications & Use Cases

edunet foundation

Societal impact of blockchain

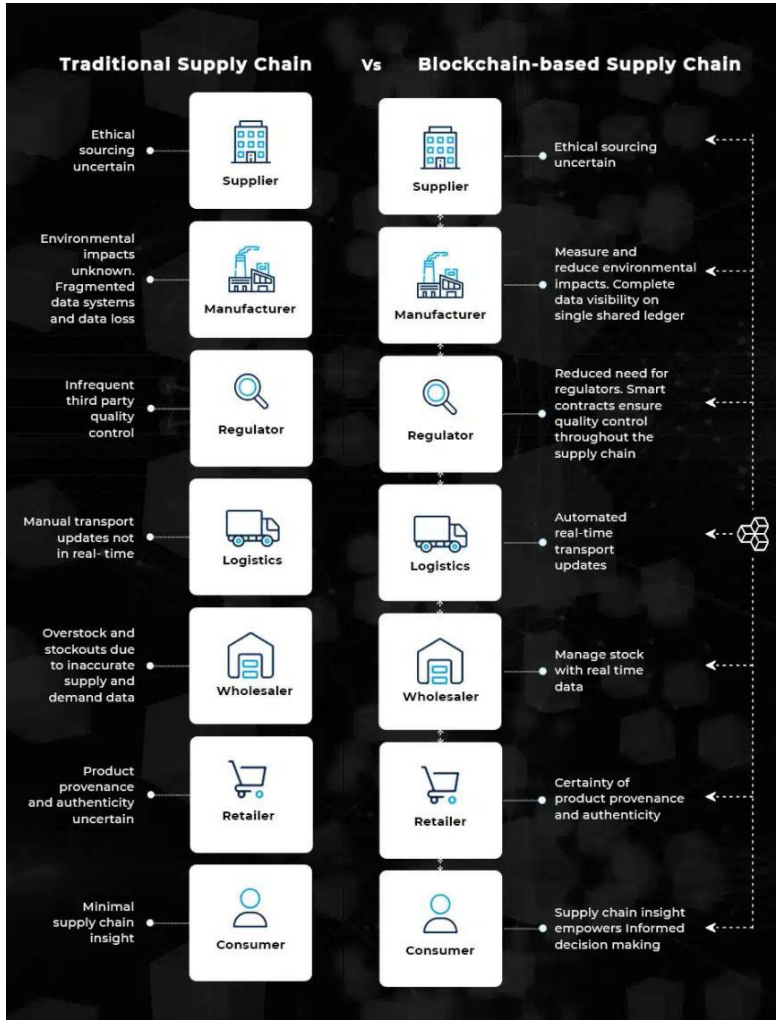
edunet foundation

Major benefits of Blockchain in Banking

- **Improved Efficiency:** Blockchain streamlines processes, speeding up transactions and reducing delays.
- **Reduced Transaction Costs:** Lower fees due to direct peer-to-peer transactions and fewer intermediaries.
- **Elimination of Intermediaries:** Blockchain removes the need for third parties like banks or clearinghouses in transactions.
- **Transparency:** Blockchain records all transactions on an immutable ledger, making them visible and auditable for all participants.

Blockchain in Supply chain

- A blockchain can assist participants better manage the supply chain by recording price, date, location, quality, certification, and other pertinent information.
- Since all the data are stored in blockchain, it is easy for tracking a product.
- It is easy to verify the products, since data in the blockchain are transparent.



edunet foundation

Blockchain in Government Sector

Public notary

Challenges

- **Security :** Because documents are on paper, fraud and errors are common.
- **High Cost:** High cost for the licensing
- **Intermediaries:** Currently, data must be reconciled and cross-checked by several mediators and parties

Benefits of using blockchain

- For the notary, blockchain provides an immutable and tamper-proof data
- Blockchain eliminates the need to go via government agencies or notary writing professionals
- Without the use of third parties, blockchain-based certifications may be checked and confirmed by anyone.
- Blockchain allows for proof-of-ownership for documents

edunet foundation

Land registry

Challenges in Traditional Land registry

- **Paperwork**: There have been numerous reports of malpractices in the paperwork for purchasing a land.
- **Signature forging**: One of the most common tactics for the land mafia to steal land is Signature forging
- **Improper documentation**: To avoid future ownership disputes, the categories of land, all document should properly documented.
- **Multiple middlemen** :Brokers are taking commissions from buyer and seller. It results in higher costs for both parties

Benefits of using blockchain

- **Lower Cost:** Reduces cost because it avoids all the middle man.
- **Security:** All the document will be secure and immutable.
- **Identity Verification:** All the information's are verified by the network itself.

Voting

Challenges

- **Time-consuming:** Constant ballot aggregation and reconciliation, takes a long time.
- **Higher cost:** The cost of the procedure can be increased by factors such as ballot printing, electronic voting equipment and maintenance.
- **Lack of transparency:** Because voting is a centralized process, transparency is limited

Benefits of using blockchain

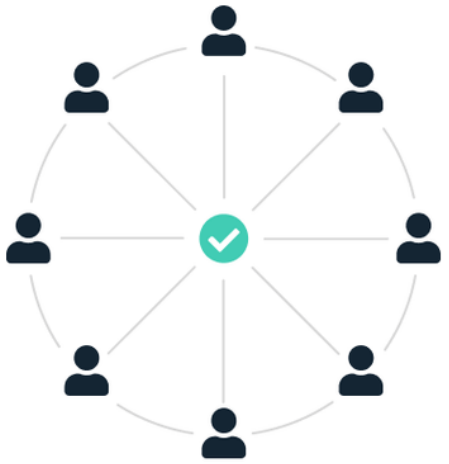
- Every ballot would be final and guaranteed, because blockchain is immutable
- Voters can use an encrypted key to cast ballots.

Consensus mechanism in Blockchain

To make all transactions in Blockchain secure, we can use consensus algorithm. Different blockchain network uses different consensus mechanism

- A **consensus mechanism** in blockchain is a protocol used to achieve agreement on a single data value or state among distributed systems, ensuring that all participants (nodes) in the network are in sync.

- It validates and confirms transactions, maintaining the integrity and security of the blockchain.



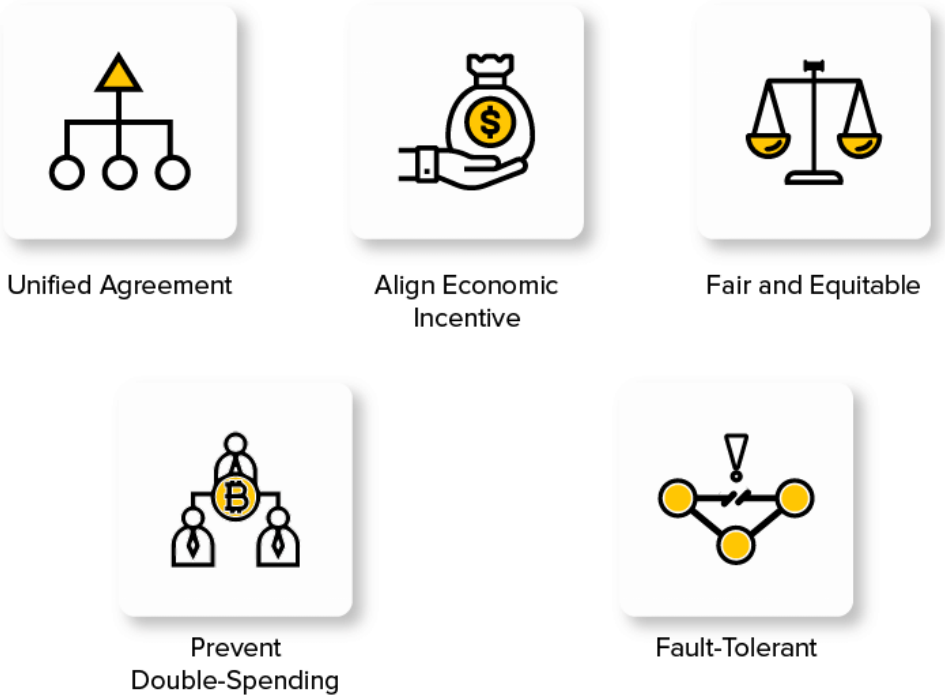
DECENTRALIZED CONSENSUS

CENTRALIZED CONSENSUS

edunet foundation

Purpose of Consensus

The purpose of **consensus** in blockchain is to ensure that all participants agree on the validity of transactions and the state of the distributed ledger, maintaining trust, security, and consistency without the need for a central authority.



Different Consensus Mechanisms

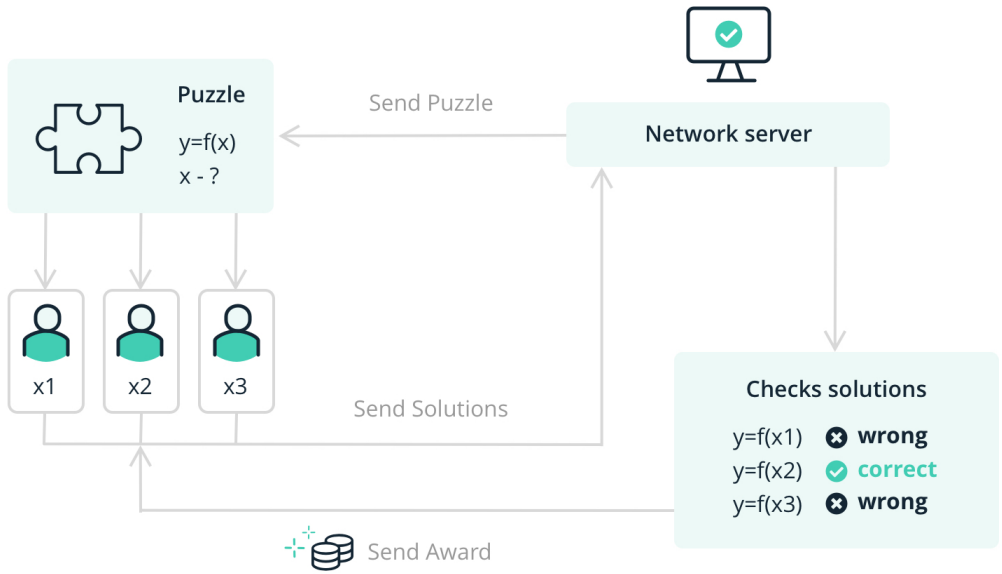
- Proof of Work (PoW): Bitcoin, Ethereum
- Proof of Stake (PoS): Peercoin
- Proof of Burn (PoB): Slimcoin, Counterparty
- Proof of Elapsed Time (PoET): Hyperledger Sawtooth

edunet foundation

Proof of Work(PoW)

Proof of Work

- Bitcoin Uses ‘proof of work ‘Consensus mechanism.
- It is based on leader election. A leader is selected from all the nodes and this node taking decision like adding new block
- In our case, the ‘proof’ is a cryptographic puzzle, which requires lot of ‘work’ to solve.
- This Consensus mechanism is Hard to produce but Easy to verify



edunet foundation

Proof of Work : Pros and Cons

Pros	Cons
Provides a decentralized method of verifying transactions.	High energy usage.
Allows miners to earn crypto rewards.	Mining often requires expensive equipment.
High level of security.	Inefficient with slow transaction speeds and expensive fees

Blockchain Network

Different Blockchain Networks

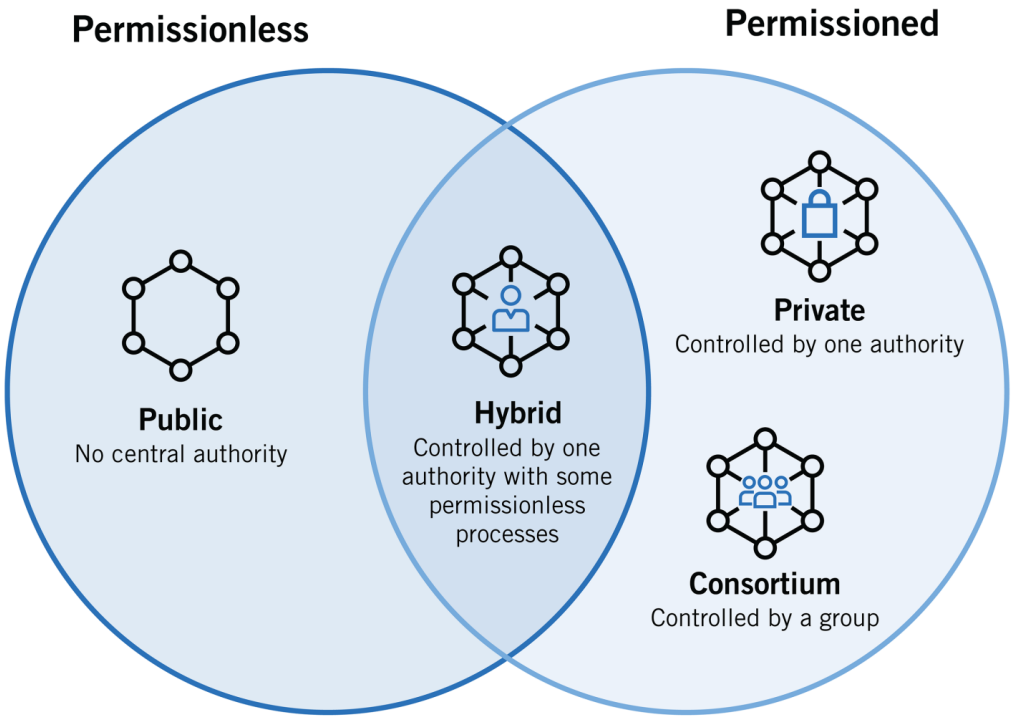
Blockchain network can be broadly classified in to two

1. Permissionless Blockchain

- Public Blockchain

2. Permissioned Blockchain

- Private Blockchain
- Consortium Blockchain



edunet foundation

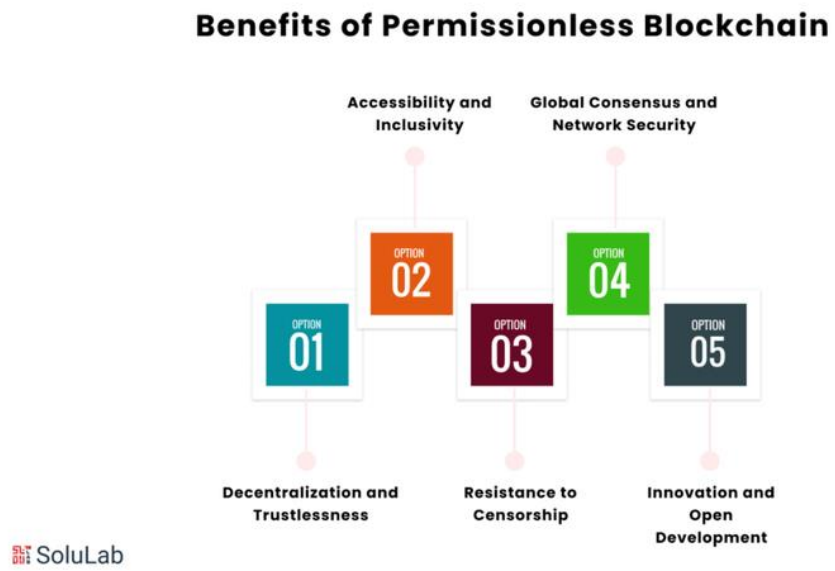
Public / Permission less Blockchain

Features of Public Blockchain:

- Transactions are transparent
- Completely decentralized
- All the nodes have equal rights in blockchain
- Examples are Bitcoin, Ethereum, Litecoin etc

Disadvantage:

- If hackers gain 51% or more of the computing power of a public blockchain network, they can unilaterally alter it

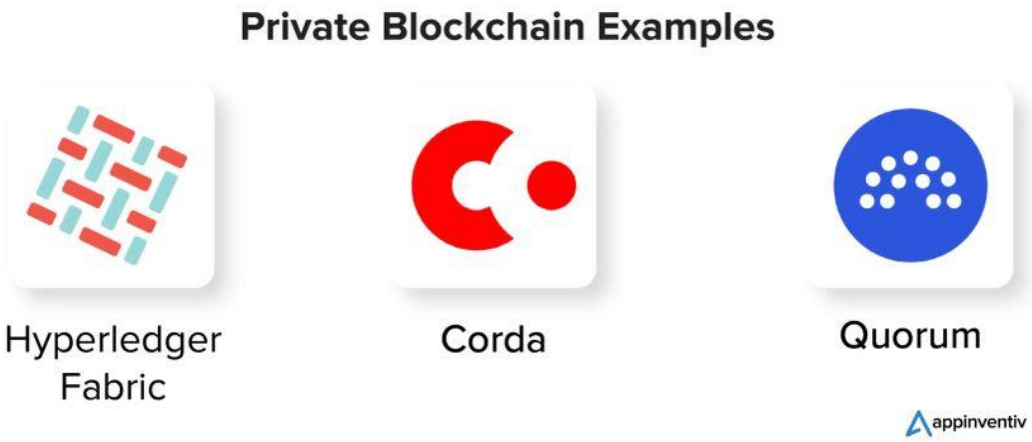


edunet foundation

Private Blockchain

Features of Private Blockchain:

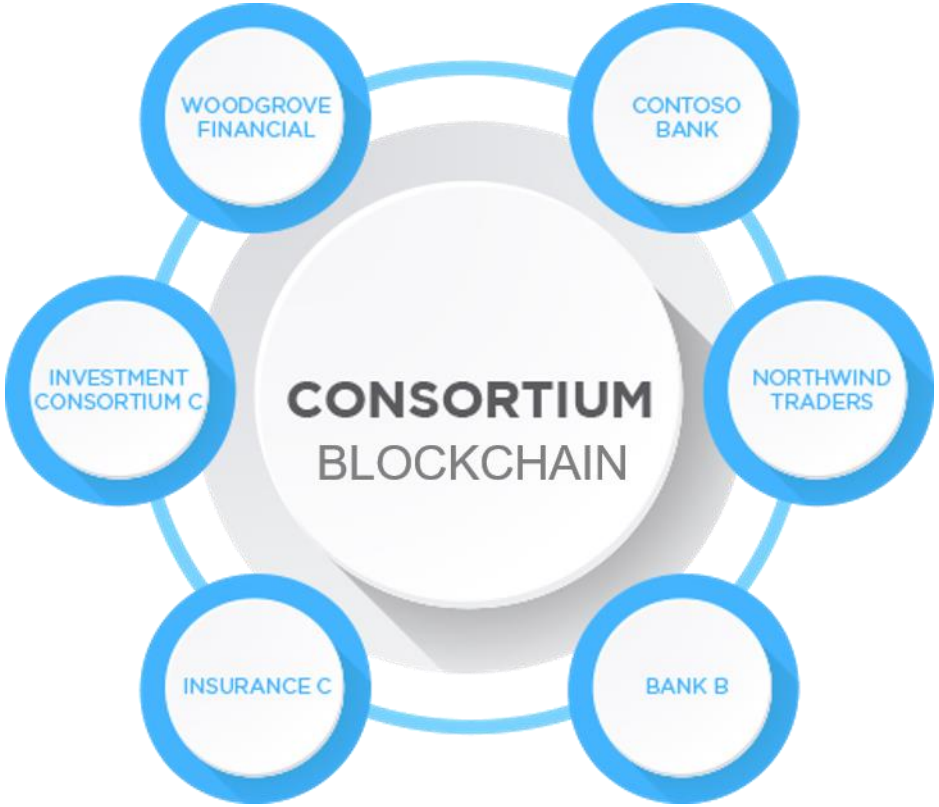
- It is permissioned blockchain
- Data will not be publicly available
- Controlled by single organization or a closed network
- The central authority determines who can be a node
- Partially decentralized
- All the nodes may NOT have equal rights in blockchain



Consortium Blockchain

Features of Consortium Blockchain:

- Consortium blockchain is a permissioned block chain which is controlled by a group of people.
- It provides decentralization, since it is controlled by a group of organizations
- Transactions are not transparent.



edunet foundation

Quiz

1. What is the main difference between a centralized system and a decentralized system?

A) Centralized systems have one central authority; decentralized systems do not.

B) Decentralized systems use a single database.

C) Centralized systems eliminate intermediaries.

D) Decentralized systems are slower than centralized systems.

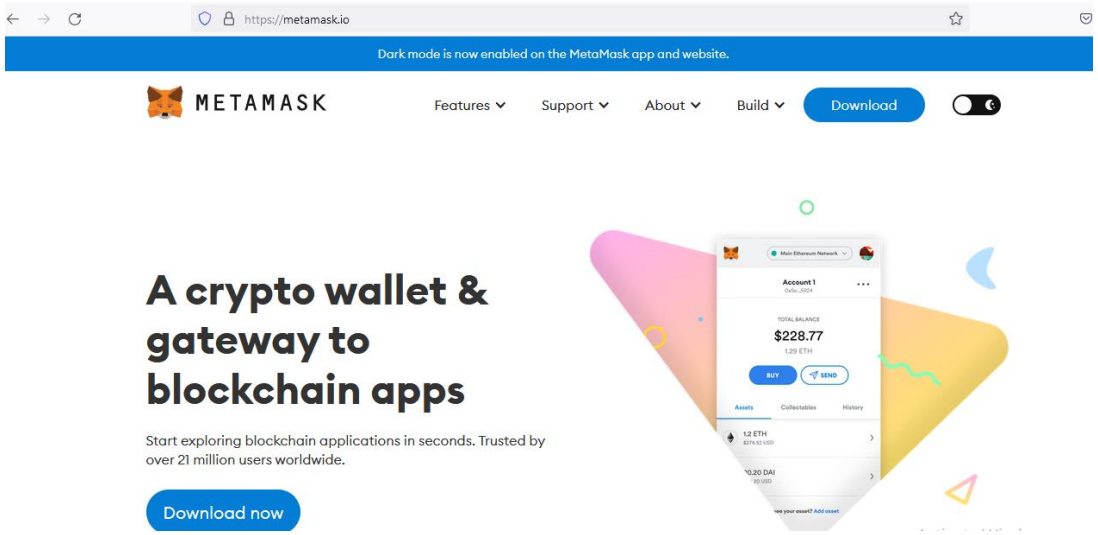
Install Metamask Wallet

Metamask is an Ethereum wallet.

We can install on browser

Visit https://metamask.io/

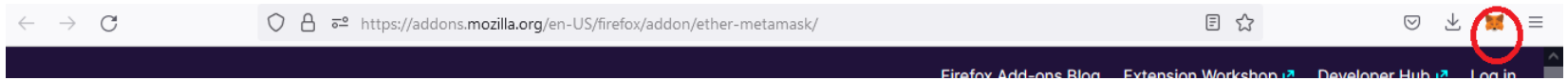
Click down and then install



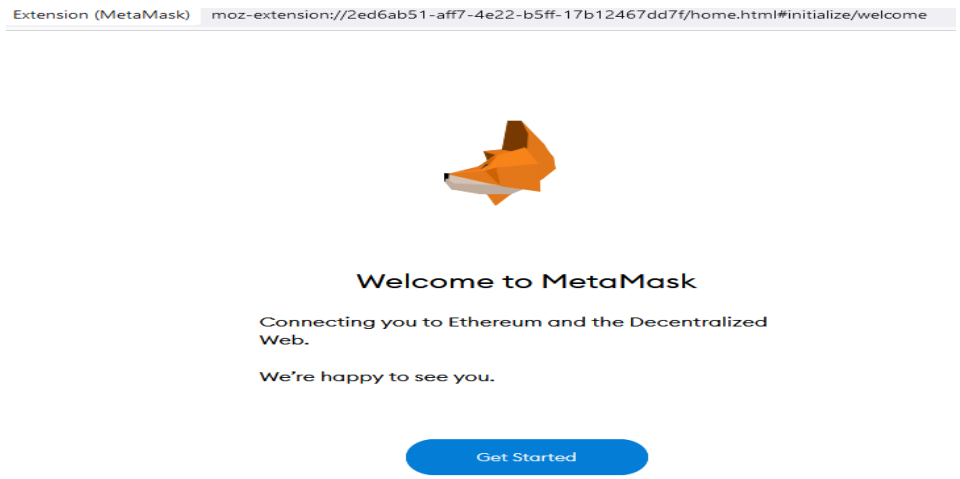
edunet foundation

Install Metamask Wallet

After successful installation, a fox icon can be seen on your browser as shown

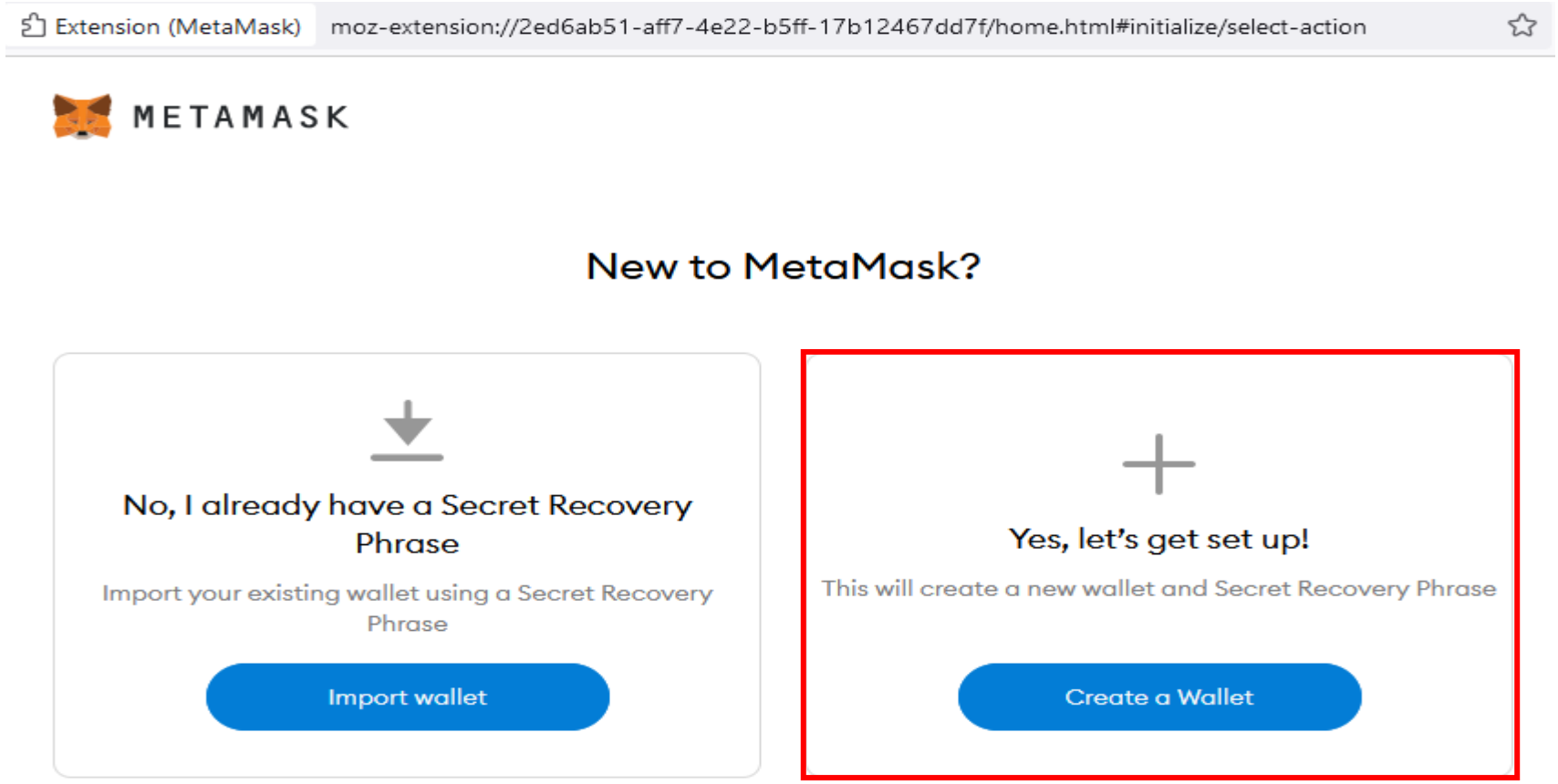


Click on the metamask icon.



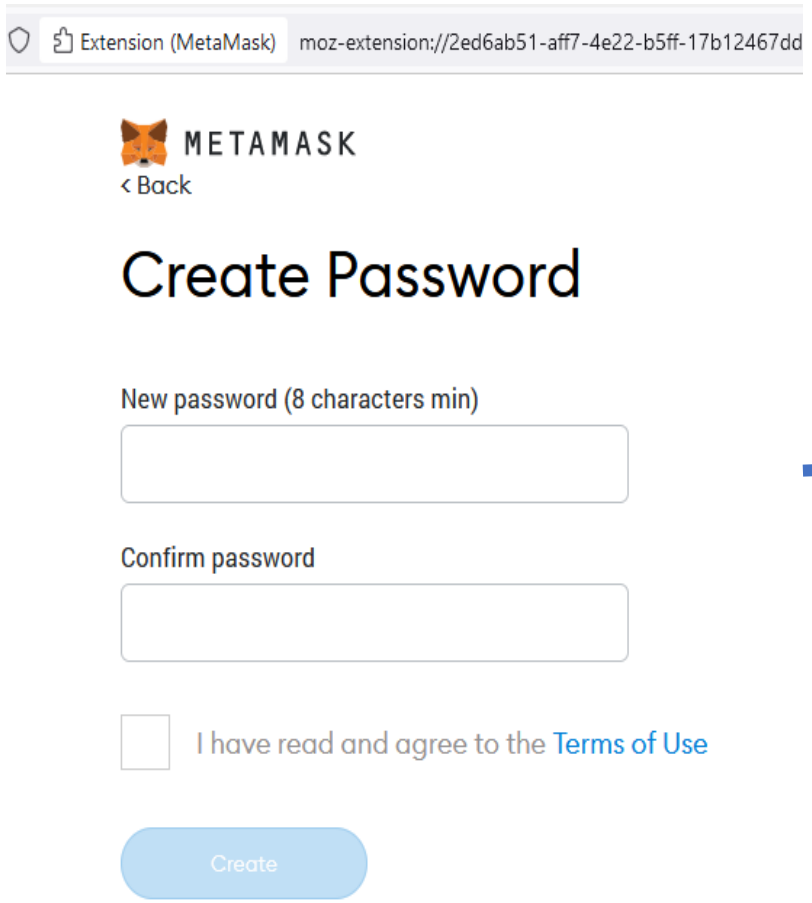
edunet foundation

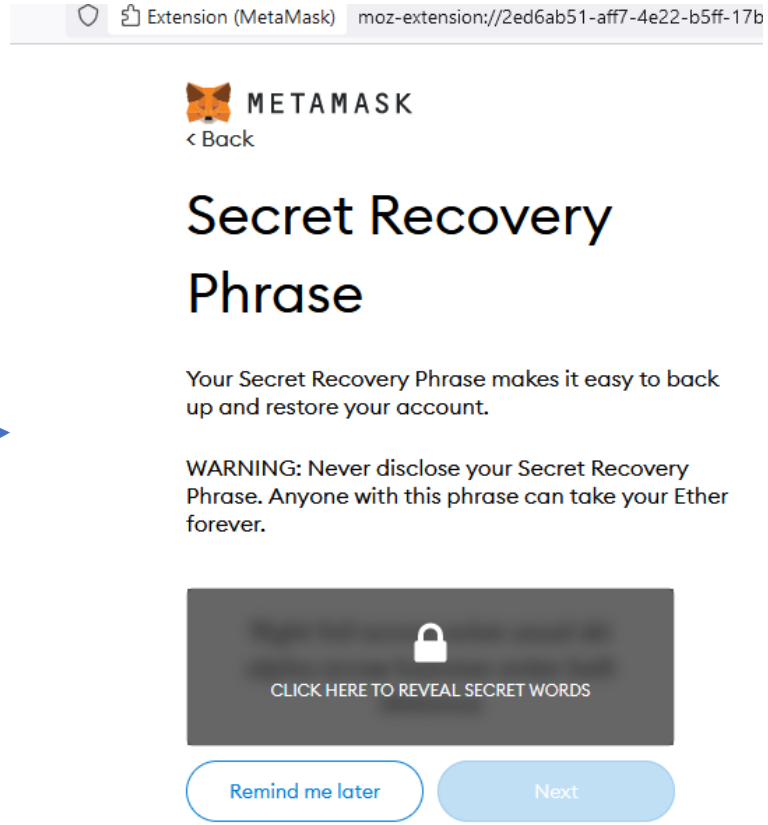
Install Metamask Wallet



edunet foundation

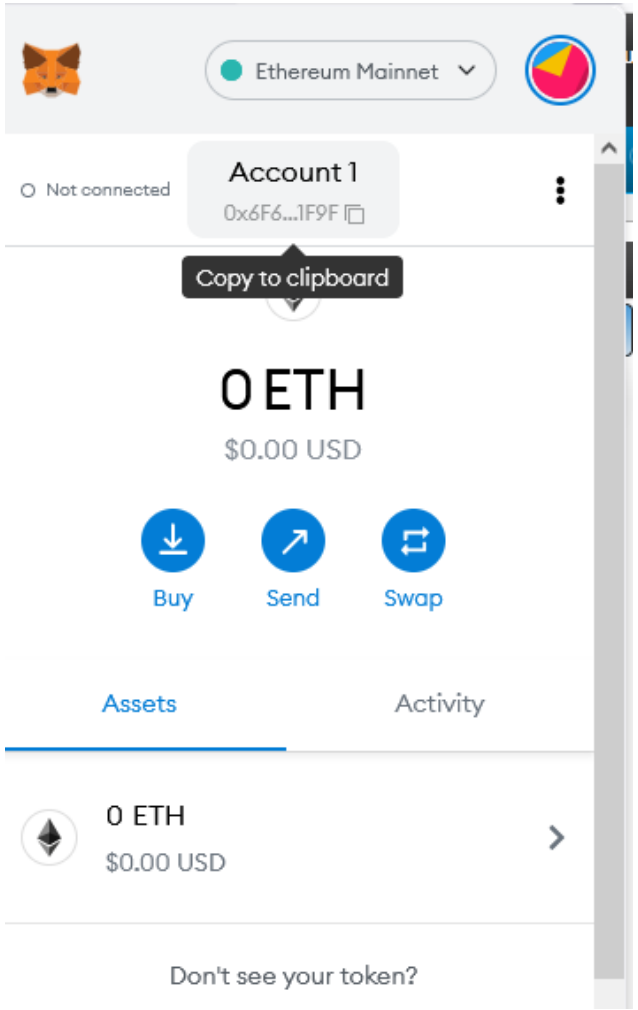
Install Metamask Wallet

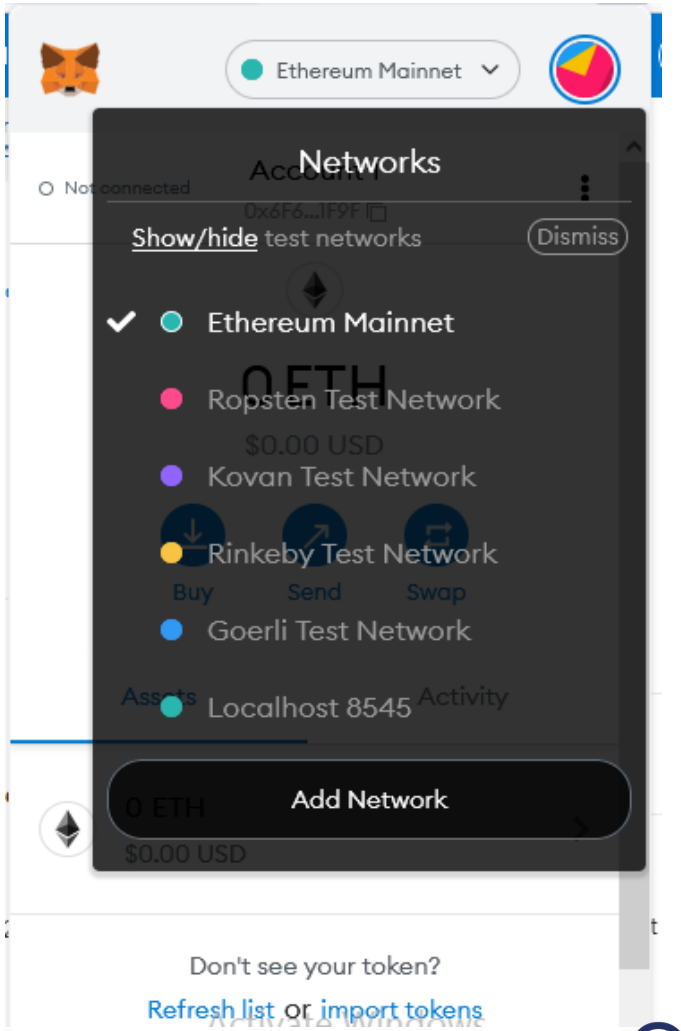




edunet foundation

Getting ETH on Testnet work





edunet foundation

Quiz

2. Who invented Blockchain technology?

A) Vitalik Buterin

B) Satoshi Nakamoto

C) Elon Musk

D) Mark Zuckerberg

Quiz

3. What is stored in a blockchain block?

A) Images and videos

B) Transactions and data

C) Passwords

D) Usernames

Quiz

4. What is the first block of a blockchain called?

A) Header Block

B) Transaction Block

C) Genesis Block

D) Starting Block

Quiz

5. What is the purpose of a “hash” in blockchain?

A) To encrypt and decrypt data

B) To uniquely identify data and ensure integrity

C) To store user details

D) To make the block larger

Quiz

6. What type of encryption uses a pair of public and private keys?

A) Symmetric Encryption

B) Asymmetric Encryption

C) Blockchain Encryption

D) Sequential Encryption

Quiz

7. Which of these is a feature of blockchain?

A) Centralized ledger

B) Data stored on paper

C) Immutable and transparent transactions

D) Requires intermediaries

Quiz

8. What is the role of a “Nonce” in a blockchain block?

A) It stores transaction details

B) It acts as a random number for mining

C) It secures the private key

D) It links two blocks

Quiz

9. What is the purpose of the consensus mechanism in blockchain?

A) To create blocks faster

B) To agree on the validity of transactions

C) To connect nodes

D) To encrypt data

Quiz

10. Which consensus mechanism does Bitcoin use?

A) Proof of Stake (PoS)

B) Proof of Burn (PoB)

C) Proof of Work (PoW)

D) Proof of Elapsed Time (PoET)

Thankyou

edunet
foundation