

 **Read the Chain – Web3.js Basics**  
**Objective/Aim:**

To learn how to create, compile, and deploy an ERC-20 token smart contract on a local blockchain environment for development and testing purposes before deploying to live networks.

**Apparatus/Software Used:**

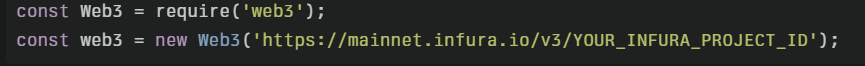
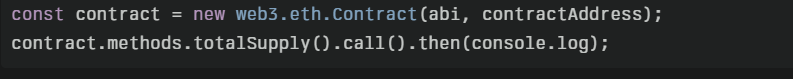
* Web3.js JavaScript library
* MetaMask
* Laptop (Brave)
* Node.js or browser enviroment

**Theory/Concept:**

Web3.js is a collection of JavaScript libraries allowing web clients or servers to communicate with Ethereum nodes via JSON-RPC protocol. It enables reading blockchain data and invoking smart contract functions by creating contract instances with ABI and contract addresses. Functions in Web3.js include getting balances, transactions, blocks, and calling smart contract methods.



**Procedure:**

* Install Web3.js via npm: npm install web3.
* Create a Web3 instance and set a provider such as Infura or local node URL.
* Read blockchain data, for example, get account balance.
* Interact with smart contracts by creating a contract instance with ABI and address:
* Handle asynchronous calls with Promises or async/await for efficient data fetching.

**Observation:**

* Web3.js allows seamless blockchain data retrieval and interaction from JavaScript.
* Asynchronous API calls provide real-time blockchain status including balances, transactions, and contract states.
* Developers can build responsive decentralized applications (dApps) with frontends connected to blockchain data through Web3.js.