①

#23 16 oct 2022

__/__/__

Today Agenda

Blockchain
- Etherium account
- Etherium Address
- Message & Transaction
- Code exuction
- Block
- Architectur of Application
-

Etherium Account

EOA → controlled by Anyone with private keys

Contract Account → A smart contract

Etherium Address

→ 42 chas hex-dec address drived from the last 20 bytes of public key.

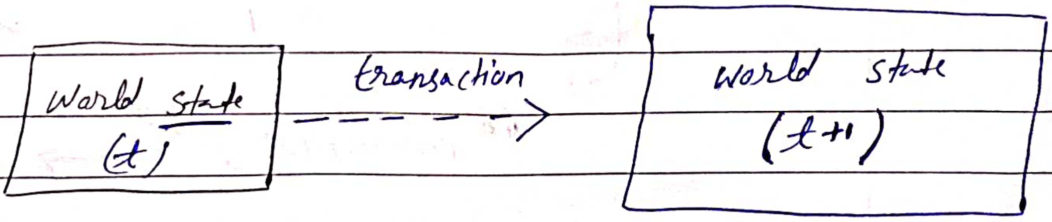
private key generate publick key generate address.

Message and Transaction

✓ Transaction refers to an action initiated by an EOA

Externally own Account EOA

Navin $\xrightarrow{1\ ETH}$ Moteeullah Promi



_/_/_

Transaction Includes

PoW → miners
PoS → Validators
→ Etherium

→ riciepent	(reciever address)
→ signature	(Sender private key)
→ nonce	(counter ++)
→ Value	(amount of ETH in web)
→ data	(optional)
→ gas limit	(max amount of gas)
→ Max Priority fees per gas	(max tip)
→ Max fees per gas	(base fee per Gas + max Priority Fee Gas)

Objects

```
class Abc
  {
     func Send ()
     {        (id, navin)
     }

   func recived ()
      { }
```

Data field

function add (69, 1) → 70

```
0xcd cd 77c 00 0000
0 0 0 00 000 00
0 0 00 45 00000
0 0 0 0 0000 00
0 0 0 0 1
```

Hexa Decimal
0x start is called

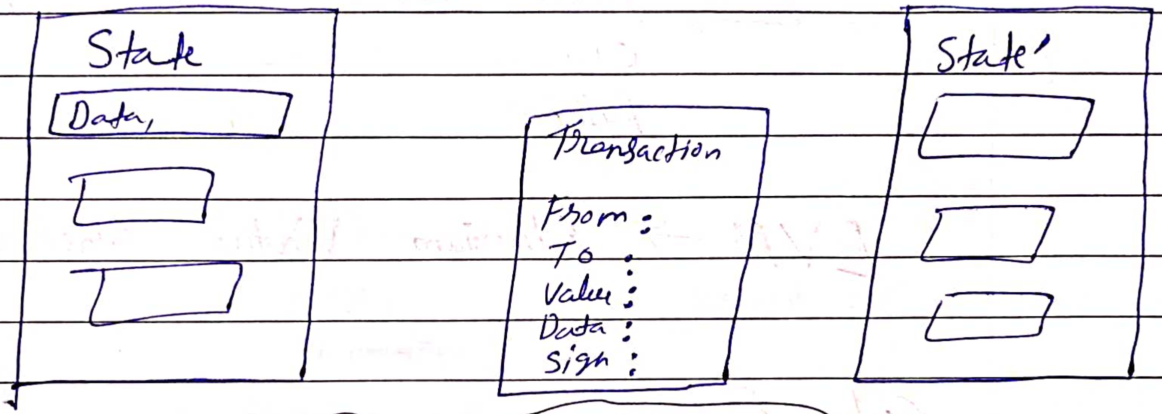
"etherium.org" → documentation

③ _/_/_

Types of Transactors

- Regular Transaction
- Contract Deployment transactions
- Execution of a contract.

Etherium State Transaction Function



Apply (S0, tx) → S'

Etherium State transition function Apply (S, TX) → S'

1) Gas → Computation
1 Gas → 1 computation
transaction → 2000 computation
→ 2000 gas

"5 Gas per byte of data"

- Gas per byte to pay the bytes
- Revert all State changes
- fees to the miner's account.
- Refund the fee all remaining gas to sender, fees paid gas consumed to miner.

④

before solidity used Serpent language

NodeJS - is a runtime of javascript ___/___/___

Example   A $\xrightarrow{10\ ETH}$ B

Gas → 2000   / litre / petrol

Gas price → 0.001 eth   / petrol price

Data → 64 bytes

CPU
- → binary → Machine code
- → Assembly → Low level language
- → Java → High level language
  - C++
  - Python

EVM → Etherium Virtual Machine

Code Execution

→ Ethereum contract code

→ Low level

→ stack based bytecode

LIFO

Last in first out

EVM Code

↓

Etherium Virtual Machine

Type of space
- → Stack
- → Memory —
- → Storage —

Block —10 minute

Block Time : 12 sec (Slots)

Etherium much master bitcoin

Block Size   15 million gas

↓

30 million gas

⑤

React JS by haider sir

⇒ 5 pm to 7pm } Depend on you

★ ⇒ React JS ← skillset trending } (10-15LPA fresher)

⇒ React + Blockchain { Bitcoin, Eth }

⇒ **FOCUS** Newzeland, Australia, UK

⇒ **CONFIDENCE** } - skillset

ACCKnowledgement

HTML + CSS + Javascript + ES6 .....

HTML ⇓ generic

CSS ⇓ generic

Javascript ⇓ Const, let, arrow function etc

html → 2-3 hr
css → 2-3hr
Javascrip - 5-6 hr

Revise then back

React JS ÷

Javascript ⇒ Framework } ⇒ Angular, Nodejs

Library } ⇒ React ⇒ UI

⇑ UI

⇒ Framework Vs Library (no entire setup)

Framework: entire ~~step~~ setstep

Library ⇓ React JS

⑥

JSON - Javascript object Notation

_/_/_

⇒ Facebook (META) ⇒ 2010, 2012, 2015

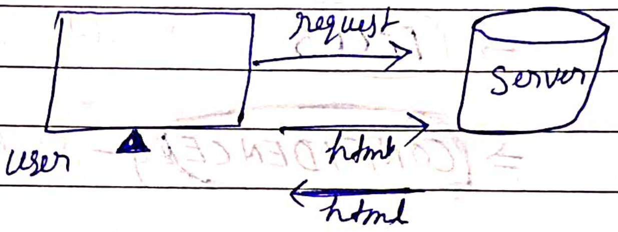
View → UI ⇒ React ← Trending

⇒ Huge Large Community Supports How to google

⇒ lot of updates ⇒

⇒ SPA - Single Page Application

2:35:00



★ Why React fast ⇒ because, SPA → Single Page Applicatio

⇒ React Virtual DOM

exa <UL>

<LI> — <LI>

<LI> — <LI>

Virtual Dom - Specific

Dom - Entire changes

why fast

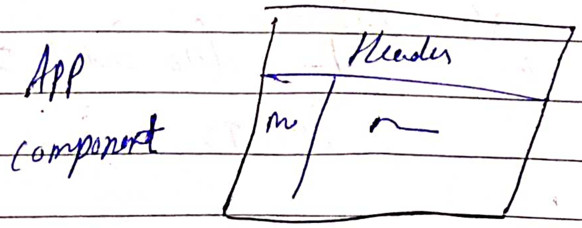
React used Virtual Dom

React fast

⇓

- Virtual DOM
- Single Page Applicatio

⇒ React fallow component based Architichure.

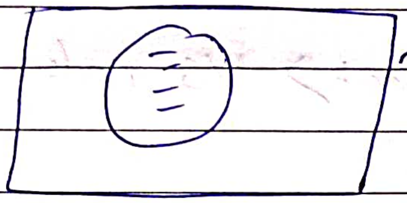


•⇒ Reusable component

⑦ __/__/__

Facebook — ~~32~~ 50K component
Netflix
Whatsweb
iNeuron

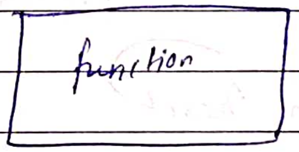
⇒ Component ÷ Javascript code is known as.



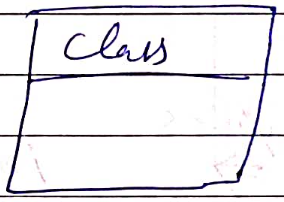
functional Style

⇒ 2 Way → function Based
→ Class Based

Stateless function



Statefull



React → Component Based Architechure
Developing React JS ⇒ Developing component?
How? ⇒ function based or classed Based

Functional is best ✓ | Class type expend component

function Hydes ()
{
  <h1> — —
  <h1>
}

render ()
{
}

⑧ __/__/__

☆ ⇒ Statefull component → function class

⇒ React ⇒ Clear

⇒ node JS + VS Code } neurolab }

newfolder ⇒ React > VS code

Terminal ⇒ npx create-react-app ~~[illegible]~~ ~~azmi~~ hyder

cd hyder

src → All file here

hyder.JS ⇒

Component

library

```
import React from 'react'
function Welcome ()
{ return <h1> React is best </h1>
}
export default Wellome
```

```
import Mycomponent from './component/Hyder'
```

——X——X——X——