

# Multisig Wallet

Step 1: pragma

Step 2: contract and struct

Step 3: Array and mapping formation

Step 4: Constructor

Step 5: Functions

## 1. Pragma -

```
pragma solidity ^0.8.20;
```

## 2. Contract and struct :-

```
Contract MultisigWallet  
    owners;  
    minNumOfConfirmations;
```

```
struct Transactions {  
    to, value, data, executed,  
    (address)(uint) (byte) Cbool  
    numofConfirmations  
    (uint)  
}
```



### 3. Array and Mapping Formation :-

Array  $\Rightarrow$  Transactions[] public transactions;

Map  $\Rightarrow$  mapping (uint  $\Rightarrow$  mapping (address  $\Rightarrow$  bool))  
public  
isConfirming;

### 4. Constructor :-

2 requires (i) - owners.length > 0  
(ii) - owners.length  $\geq$  \_minNumOfConfirmations

push \_owners to main owners array of address.

### 5. Functions :-

(1) submitTransaction (\_to, \_value, \_data) public  
{  
  to: \_to, value: \_value, data: \_data,  
  executed: false, num of Confirmations: 0  
}



(2) ConfirmTransaction (-txIndex) public &  
require (-txIndex > transactions.length)  
Transactions storage txn = transactions  
[-txIndex]

```
require(!txn_executed);
```

```
require(!isConfirmed);
```

$$tx_n, \text{ num of Confirmations} + 1;$$

```
psConfirmed [_txIndex] [msg.sender] = true;
```

```
(3) executeTransaction (-txIndex) public {
    (bool success, ) = txn.to.call
    {value: txn.value}
```

receive() external payable { }

for receiving to our account.

