BLOCKCHAIN

SMART CONTRACT SECURİTY AND AUDİTS

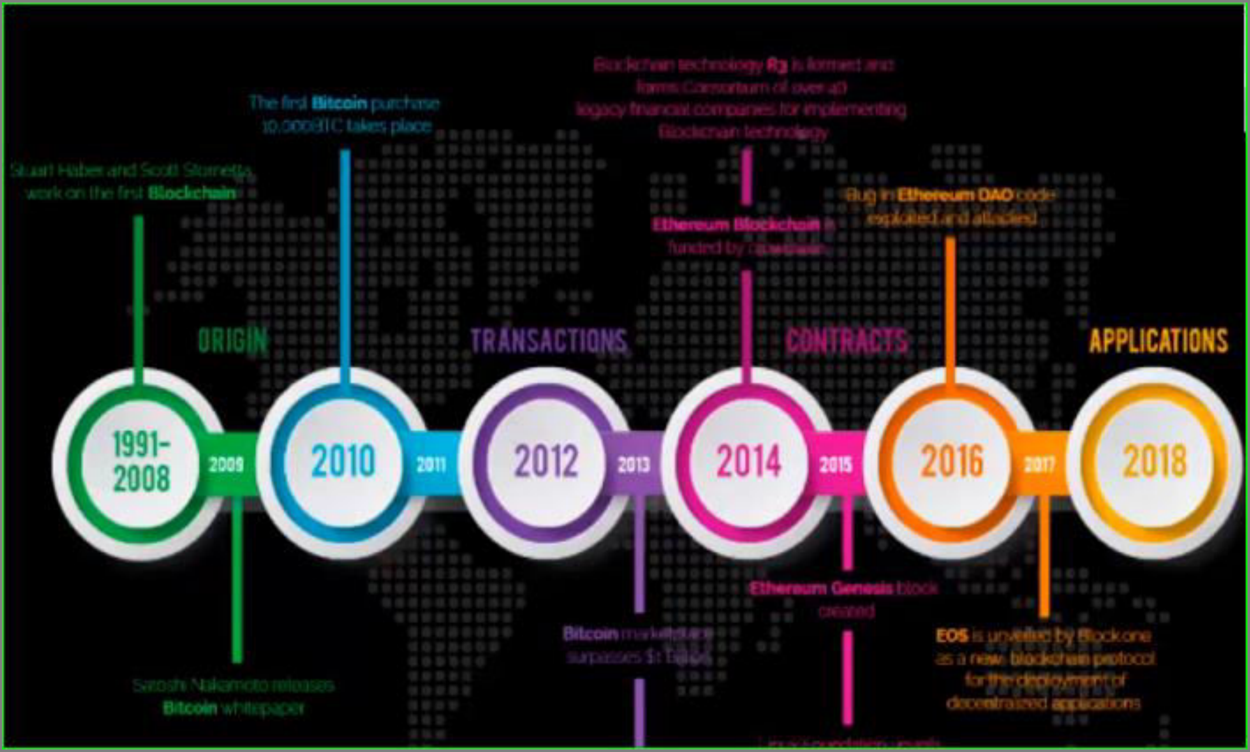
Rüzgar Üren

1107090006

Dr. Enis Karaaslan

Mugla Sitki Kocman University

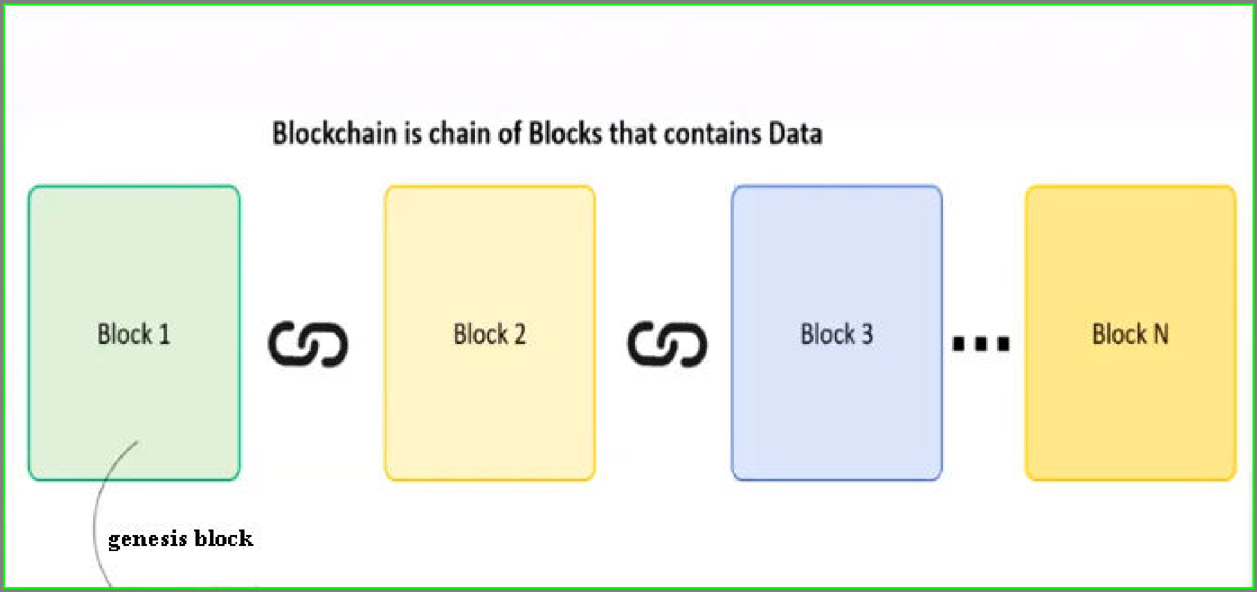
BLOCK CHAIN


Stuart Haber and Scott Stornetta work on the first Blockchain
The first Bitcoin purchase 10,000BTC takes place
Blockchain technology R3 is formed and forms Consortium of over 40 legacy financial companies for implementing Blockchain technology
Ethereum Blockchain is funded by crowdsale
Bug in Ethereum DAO code exploited and attacked
ORIGIN
TRANSACTIONS
CONTRACTS
APPLICATIONS
1991-2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
Satoshi Nakamoto releases Bitcoin whitepaper
Bitcoin marketplace surpasses $1 billion
Ethereum Genesis block created
EOS is unveiled by Blockone as a new blockchain protocol for the deployment of decentralized applications

2009-

Smart Contracts

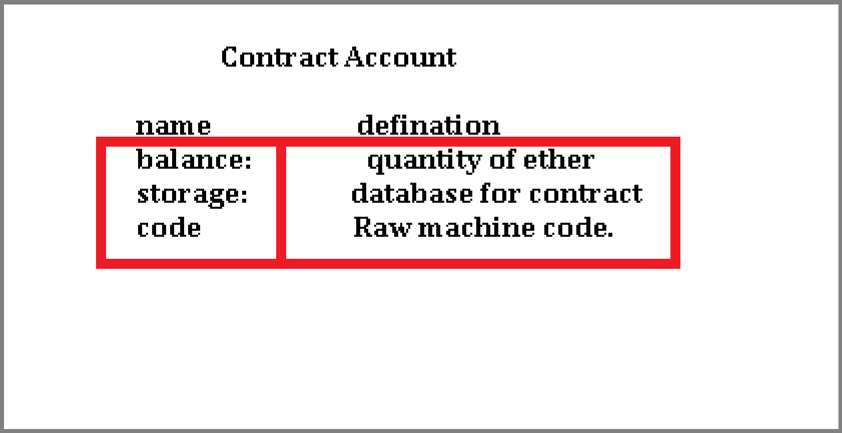
A smart contract is a computer code that can be built into the blockchain to facilitate, verify, or negotiate a contract agreement. Smart contracts operate under a set of conditions that users agree to. When those conditions are met, the terms of the agreement are automatically carried out



What is inside the blocks ?

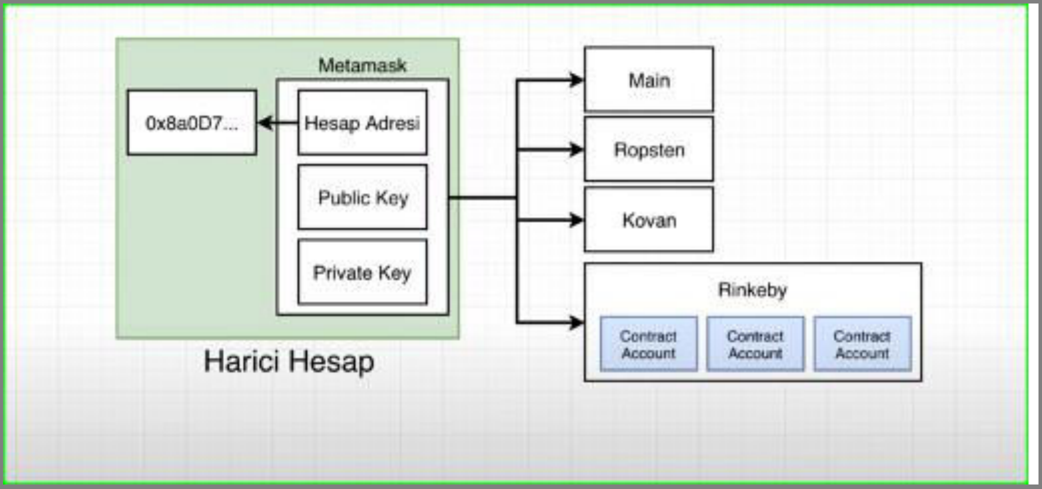
What is an account has : balance, account number, public key …

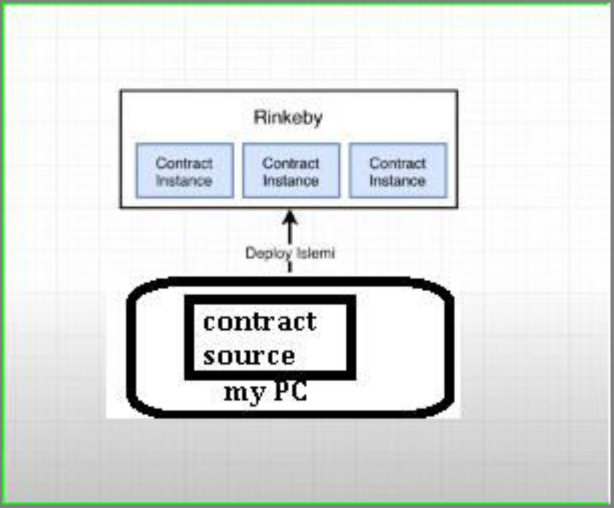
What a smart contract has: …



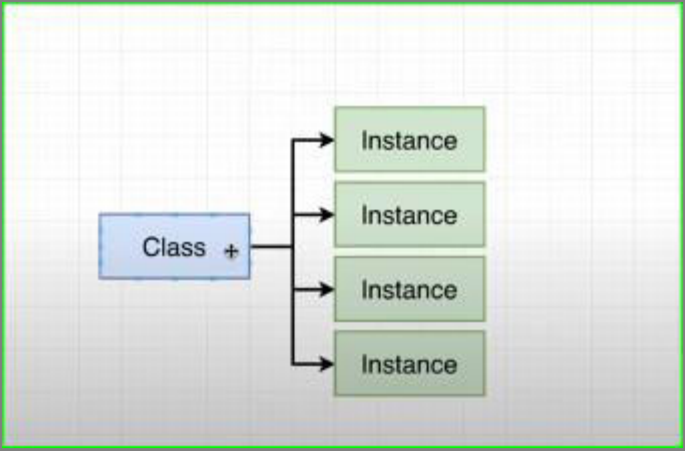
Contract Account

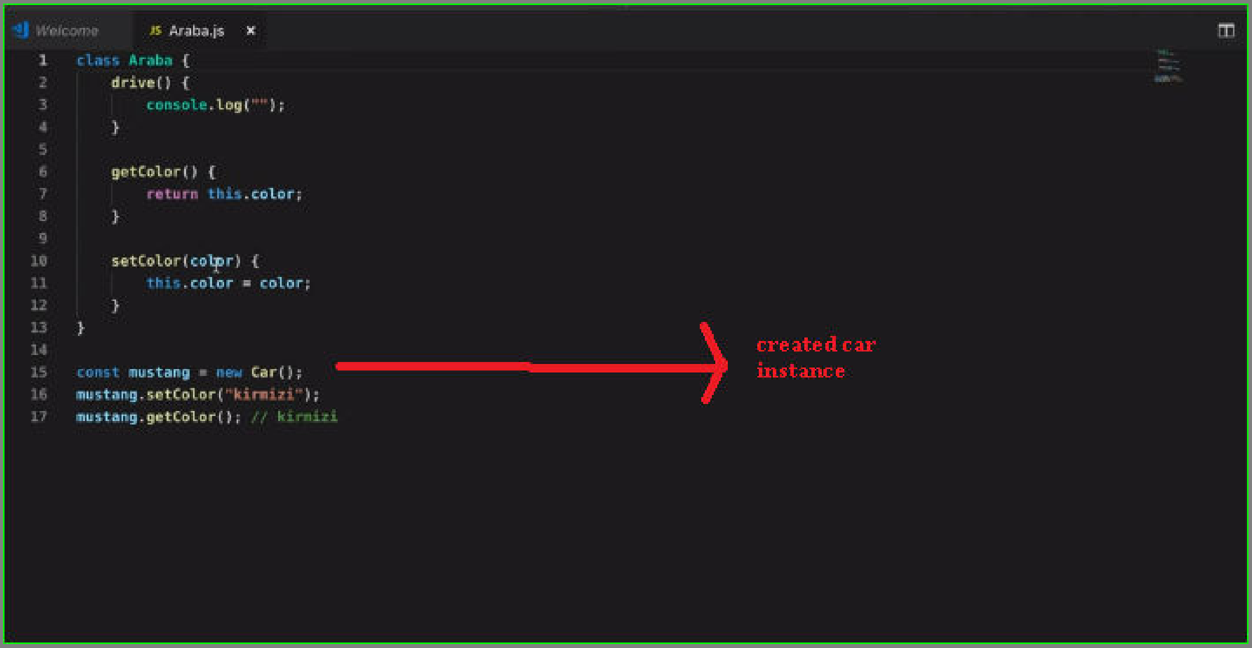
name	defination
balance:	quantity of ether
storage:	database for contract
code	Raw machine code.


Metamask
0x8a0D7...
Hesap Adresi
Public Key
Private Key
Harici Hesap
Main
Ropsten
Kovan
Rinkeby
Contract Account
Contract Account
Contract Account



How many times you deploy the contract sourse - that much times you created instances

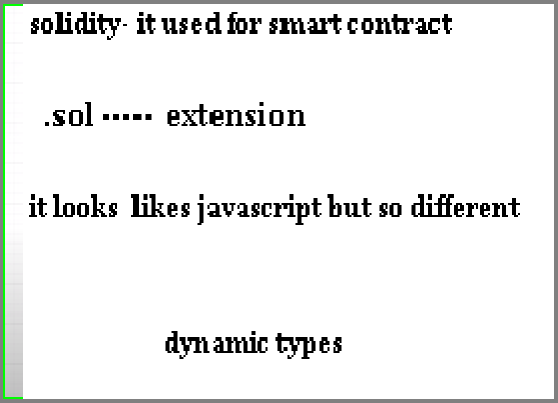




```javascript
class Araba {
    drive() {
        console.log("");
    }

    getColor() {
        return this.color;
    }

    setColor(color) {
        this.color = color;
    }
}

const mustang = new Car();
mustang.setColor("kirmizi");
mustang.getColor(); // kirmizi
```

When we write const mustang = new car() – here is a creation of instance.

Solidity Programing Language

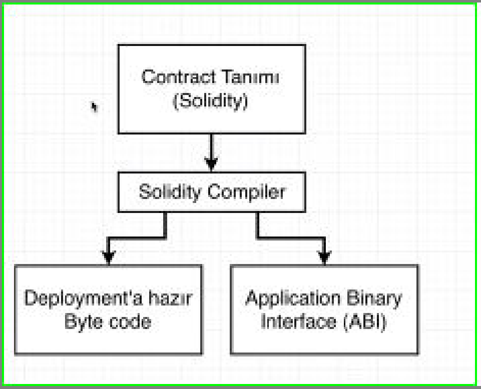

solidity- it used for smart contract

.sol ····· extension

it looks likes javascript but so different

dynamic types

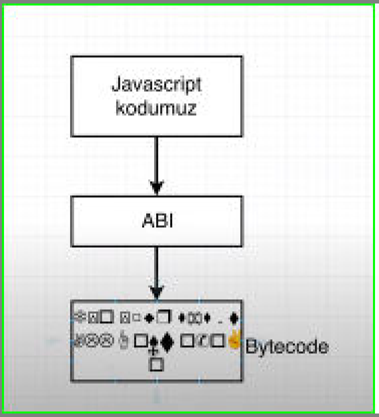
-


Contract Tanımı (Solidity)
Solidity Compiler
Deployment'a hazır Byte code
Application Binary Interface (ABI)

contract defination

when we write any contract definition it is compile by solidity language and we have two output

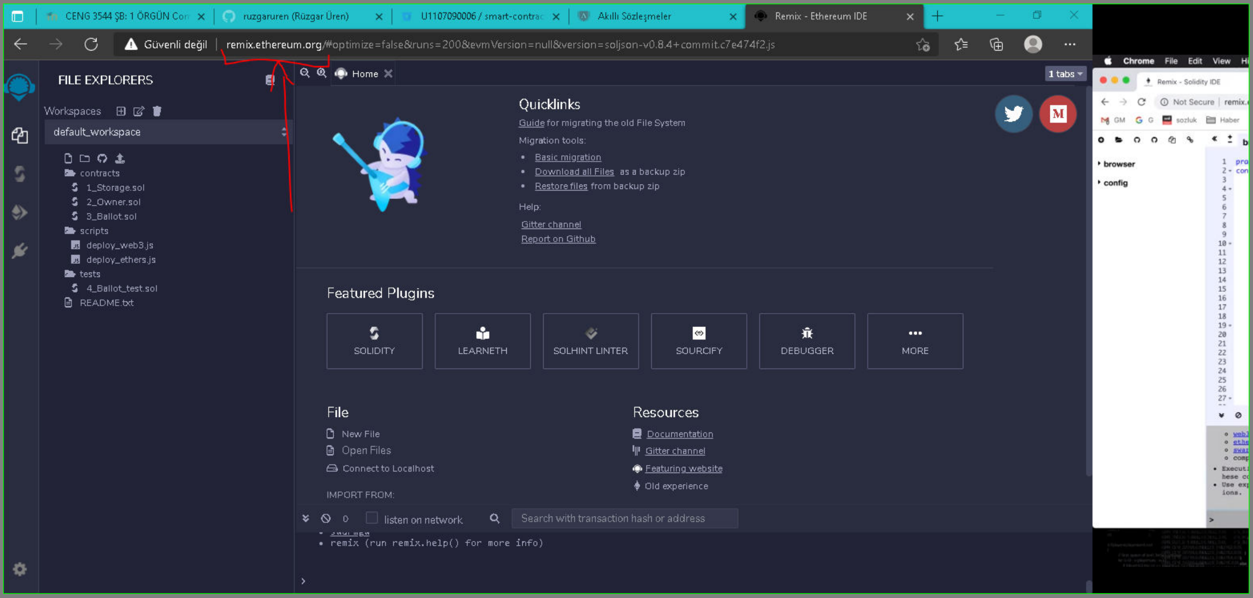
1- byte code- ready for deployment
2-application binary interface" ABİ"



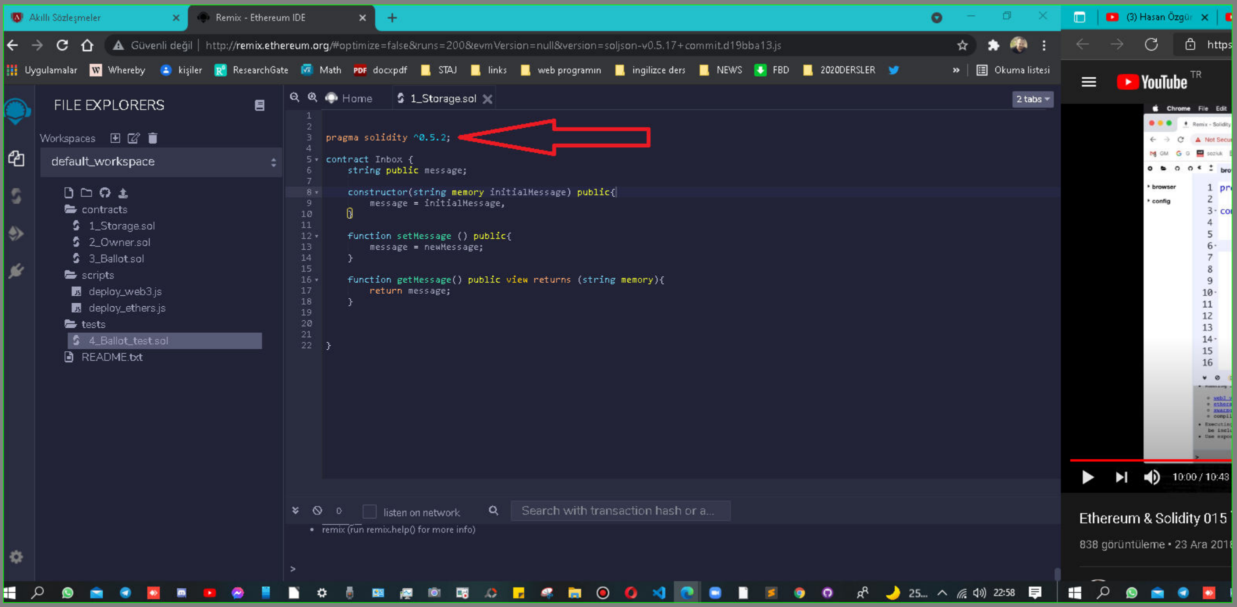
ABI is manage this bytecodes

Link 1:

the editor: http://remix.ethereum.org/

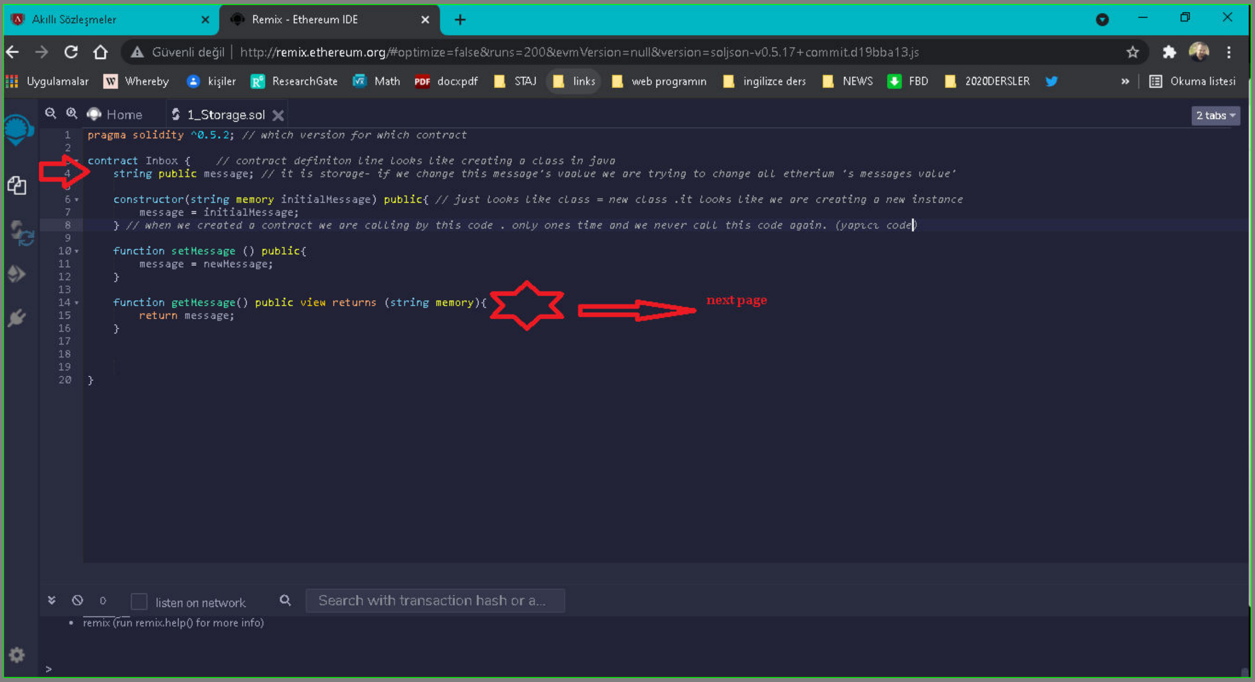


```
remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.4+commit.c7e474f2.js
```



```
remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.5.17+commit.d19bba13.js
```

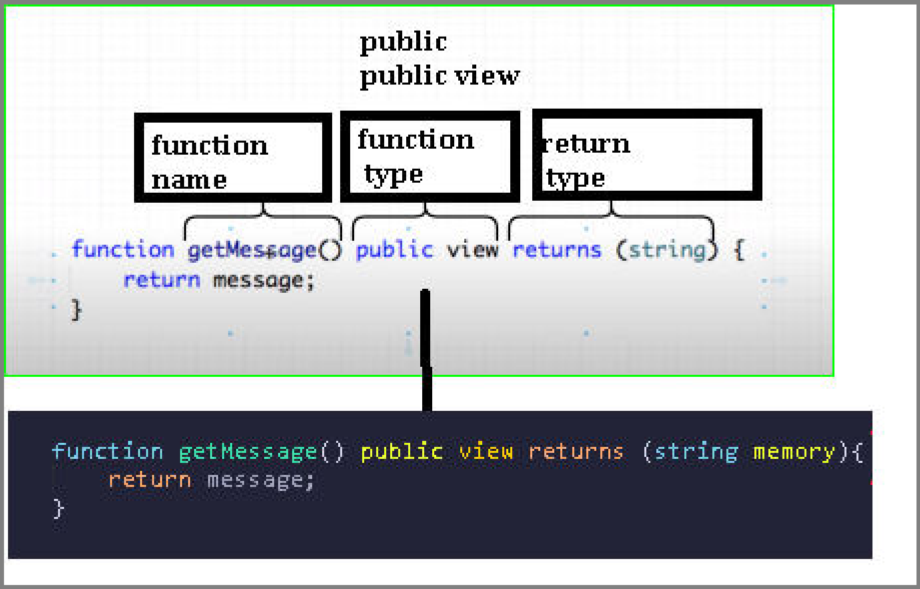
```
pragma solidity ^0.5.2;

contract Inbox {
    string public message;

    constructor(string memory initialMessage) public{
        message = initialMessage,
    }

    function setMessage () public{
        message = newMessage;
    }

    function getMessage() public view returns (string memory){
        return message;
    }



}
```



```
pragma solidity ^0.5.2; // which version for which contract

contract Inbox {    // contract definiton line looks like creating a class in java
    string public message; // it is storage- if we change this message's vaalue we are trying to change all etherium 's messages value'

    constructor(string memory initialMessage) public{ // just looks like class = new class .it looks like we are creating a new instance
        message = initialMessage;
    } // when we created a contract we are calling by this code . only ones time and we never call this code again. (yapıcı code)

    function setMessage () public{
        message = newMessage;
    }

    function getMessage() public view returns (string memory){
        return message;
    }



}
```

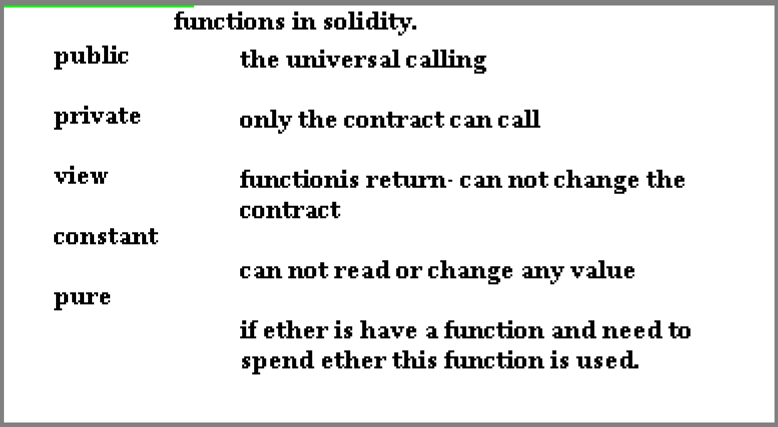
next page

```
remix (run remix.help() for more info)
>
```



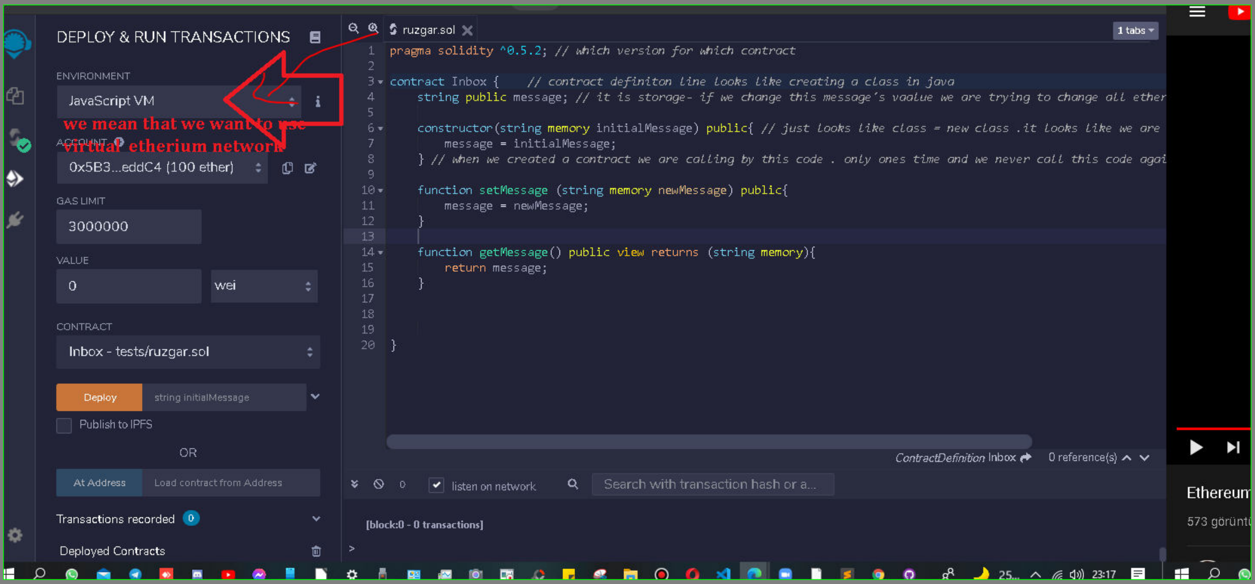
```
function getMessage() public view returns (string) {
    return message;
}
```

```
function getMessage() public view returns (string memory){
    return message;
}
```



functions in solidity.

public	the universal calling
private	only the contract can call
view	functionis return- can not change the contract
constant	can not read or change any value
pure	if ether is have a function and need to spend ether this function is used.



DEPLOY & RUN TRANSACTIONS

ENVIRONMENT: JavaScript VM

we mean that we want to use virtual etherium network

ACCOUNT: 0x5B3...eddC4 (100 ether)

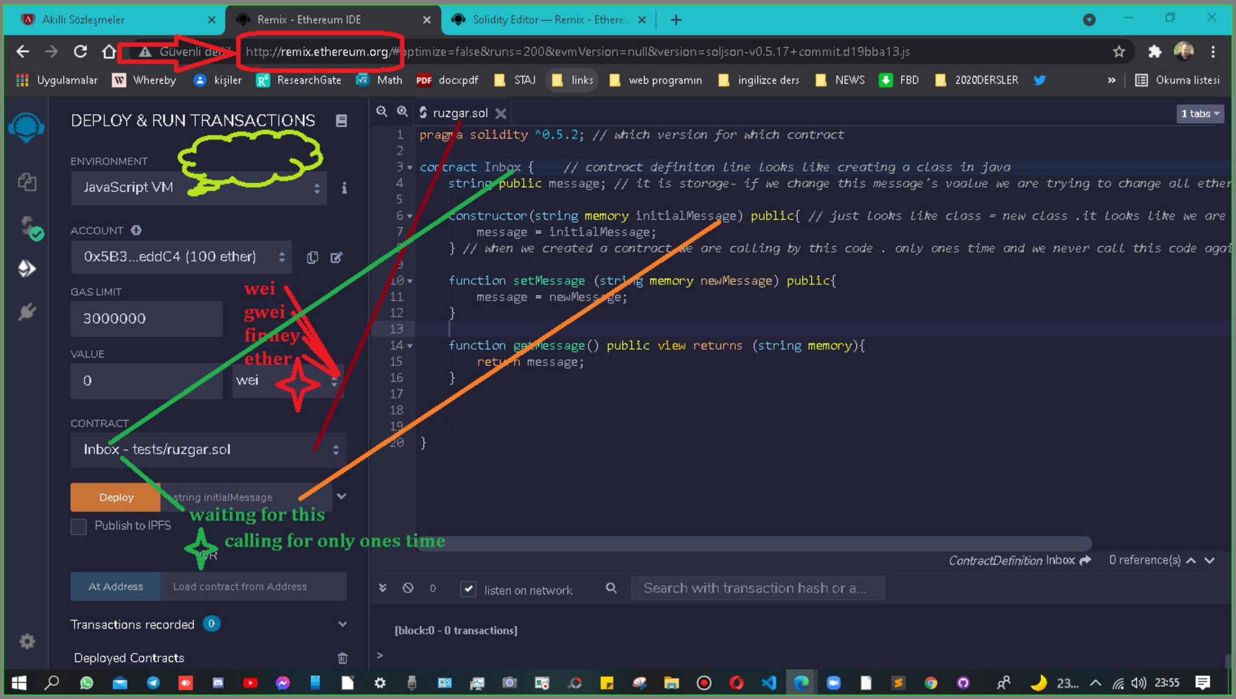
GAS LIMIT: 3000000

VALUE: 0 wei

CONTRACT: Inbox - tests/ruzgar.sol

```
pragma solidity ^0.5.2; // which version for which contract

contract Inbox {    // contract definiton line looks like creating a class in java
    string public message; // it is storage- if we change this message's vaalue we are trying to change all ether

    constructor(string memory initialMessage) public{ // just looks like class = new class .it looks like we are
        message = initialMessage;
    } // when we created a contract we are calling by this code . only ones time and we never call this code agai

    function setMessage (string memory newMessage) public{
        message = newMessage;
    }

    function getMessage() public view returns (string memory){
        return message;
    }



}
```

```
[block:0 - 0 transactions]
>
```



http://remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.5.17+commit.d19bba13.js

DEPLOY & RUN TRANSACTIONS

ENVIRONMENT: JavaScript VM

ACCOUNT: 0x5B3...eddC4 (100 ether)

GAS LIMIT: 3000000

VALUE: 0 wei

CONTRACT: Inbox - tests/ruzgar.sol

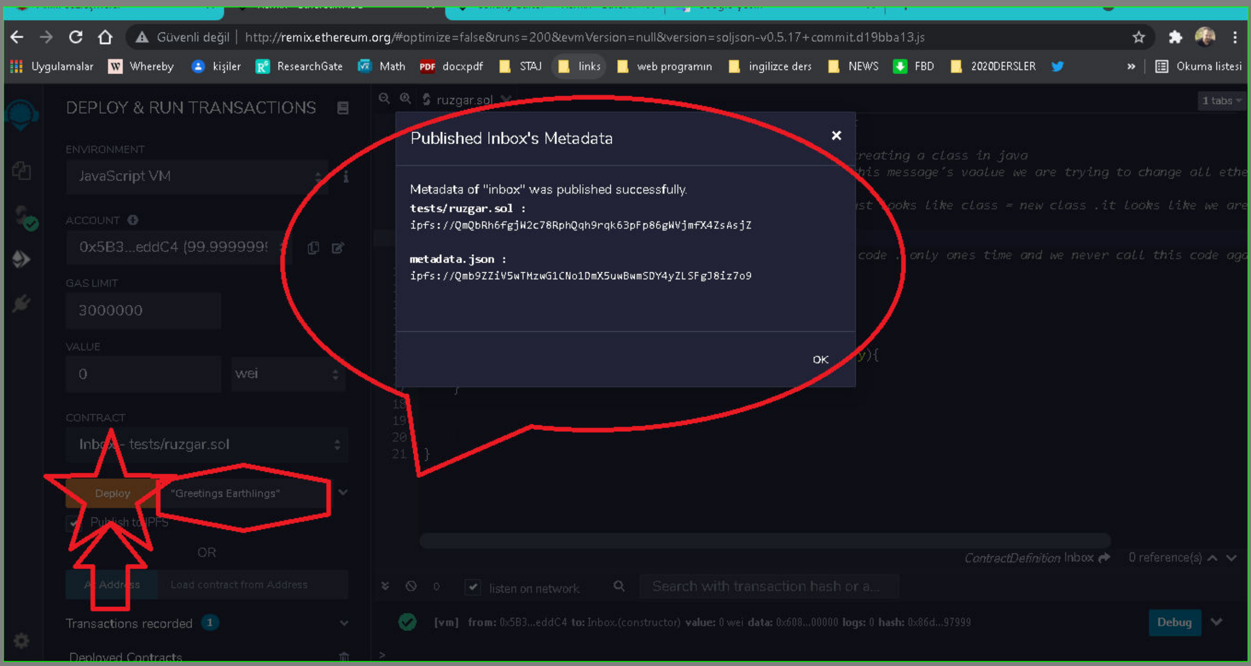
Deploy: string initialMessage

☐ Publish to IPFS

ruzgar.sol

```
pragma solidity ^0.5.2; // which version for which contract

contract Inbox {    // contract definiton line looks like creating a class in java
    string public message; // it is storage- if we change this message's vaalue we are trying to change all ether

    constructor(string memory initialMessage) public{ // just looks like class = new class .it looks like we are
        message = initialMessage;
    } // when we created a contract we are calling by this code . only ones time and we never call this code agai

    function setMessage (string memory newMessage) public{
        message = newMessage;
    }

    function getMessage() public view returns (string memory){
        return message;
    }


}
```

```
[block:0 - 0 transactions]
```



ACCOUNT: 0x5B3...eddC4 (99.999999

Deploy: "Greetings Earthlings"

Published Inbox's Metadata

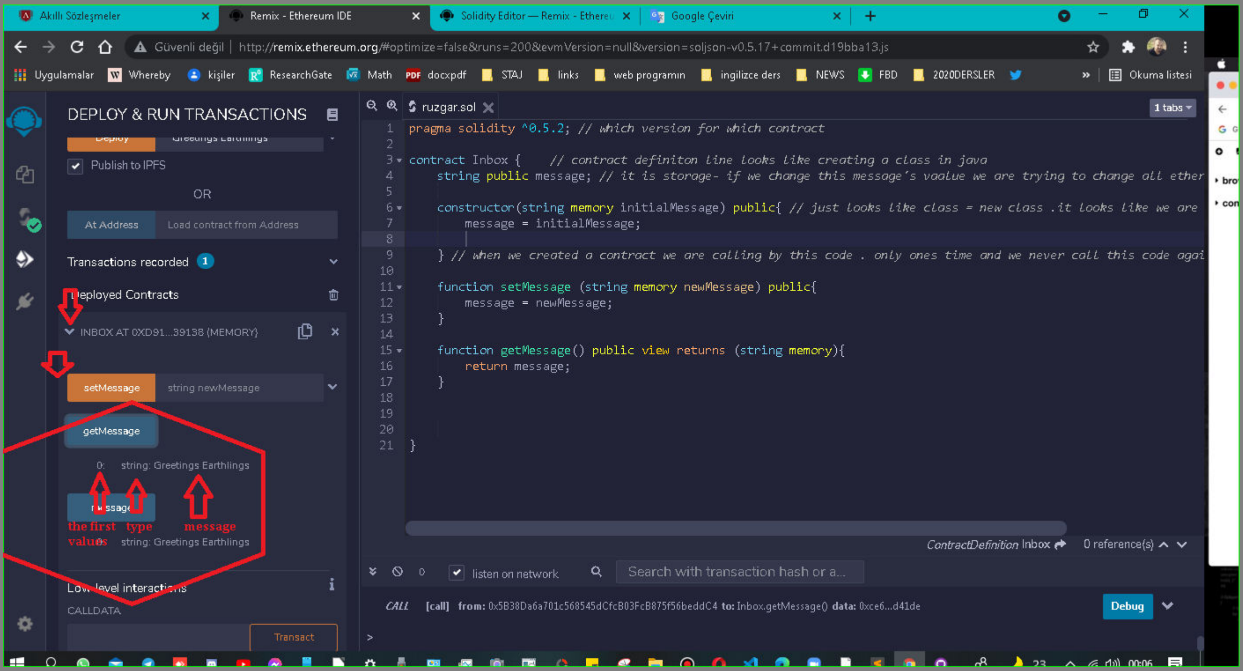
Metadata of "inbox" was published successfully.

```
tests/ruzgar.sol :
ipfs://QmQbRh6fgjW2c78RphQqh9rqk63pFp86gWVjmfX4ZsAsjZ

metadata.json :
ipfs://Qmb9ZZiV5wTMzwG1CNo1DmX5uwBwmSDY4yZLSFgJ8iz7o9
```

```
[vm] from: 0x5B3...eddC4 to: Inbox.(constructor) value: 0 wei data: 0x608...00000 logs: 0 hash: 0x86d...97999
```

Link 2:

source: https://remix-ide.readthedocs.io/en/latest/solidity_editor.html



0: string: Greetings Earthlings

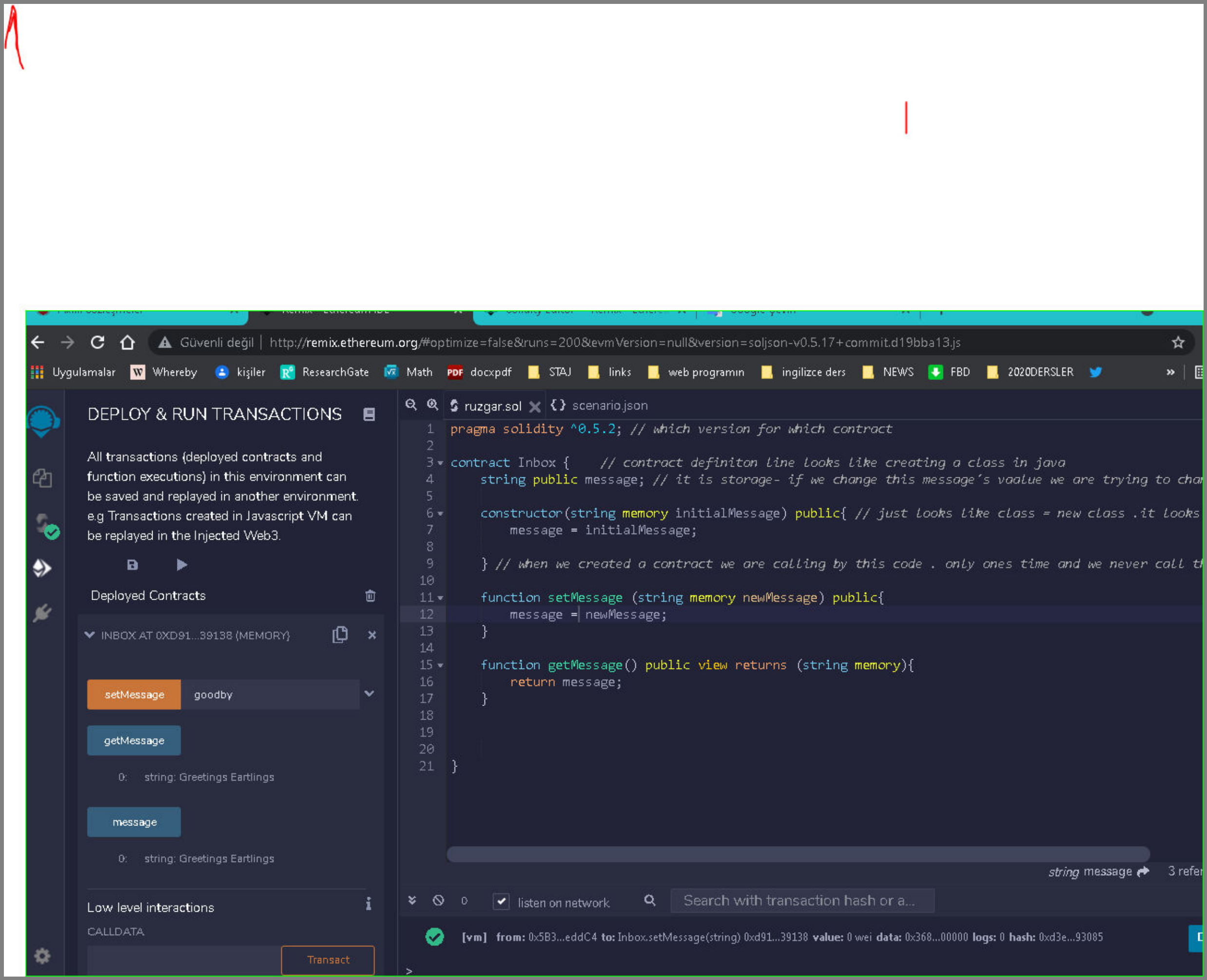
0: string: Greetings Earthlings

```
pragma solidity ^0.5.2; // which version for which contract

contract Inbox {    // contract definiton line looks like creating a class in java
    string public message; // it is storage- if we change this message's vaalue we are trying to change all ether

    constructor(string memory initialMessage) public{ // just looks like class = new class .it looks like we are
        message = initialMessage;

    } // when we created a contract we are calling by this code . only ones time and we never call this code agai

    function setMessage (string memory newMessage) public{
        message = newMessage;
    }

    function getMessage() public view returns (string memory){
        return message;
    }



}
```

```
CALL    [call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: Inbox.getMessage() data: 0xce6...d41de
>
```

when we set a new message we are not changing the get   message or the message itself

– privious messga are atill there



DEPLOY & RUN TRANSACTIONS

All transactions (deployed contracts and function executions) in this environment can be saved and replayed in another environment. e.g Transactions created in Javascript VM can be replayed in the Injected Web3.

Deployed Contracts

INBOX AT 0XD91...39138 (MEMORY)

setMessage: goodby

getMessage
0: string: Greetings Earthlings

message
0: string: Greetings Earthlings

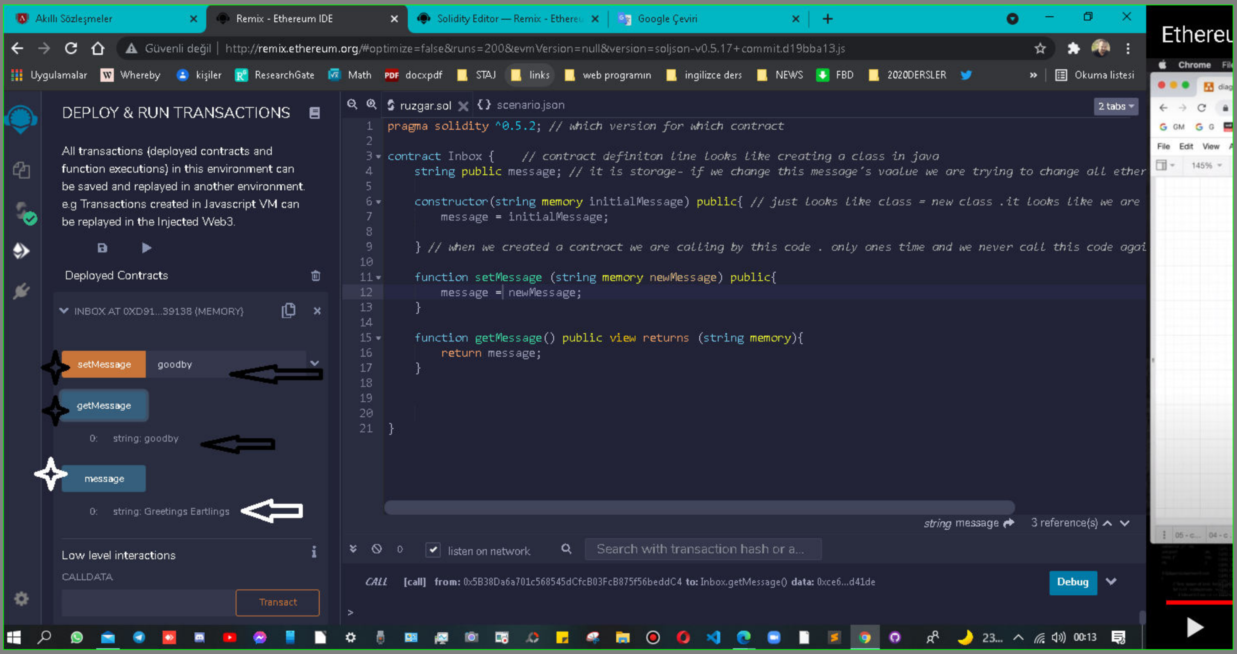
Low level interactions

CALLDATA

```
pragma solidity ^0.5.2; // which version for which contract

contract Inbox {     // contract definiton line looks like creating a class in java
    string public message; // it is storage- if we change this message's vaalue we are trying to cha

    constructor(string memory initialMessage) public{ // just looks like class = new class .it looks
        message = initialMessage;

    } // when we created a contract we are calling by this code . only ones time and we never call t

    function setMessage (string memory newMessage) public{
        message = newMessage;
    }

    function getMessage() public view returns (string memory){
        return message;
    }



}
```

string message   3 refer

```
[vm] from: 0x5B3...eddC4 to: Inbox.setMessage(string) 0xd91...39138 value: 0 wei data: 0x368...00000 logs: 0 hash: 0xd3e...93085
```



DEPLOY & RUN TRANSACTIONS

All transactions (deployed contracts and function executions) in this environment can be saved and replayed in another environment. e.g Transactions created in Javascript VM can be replayed in the Injected Web3.

INBOX AT 0XD91...39138 (MEMORY)

setMessage goodby

getMessage

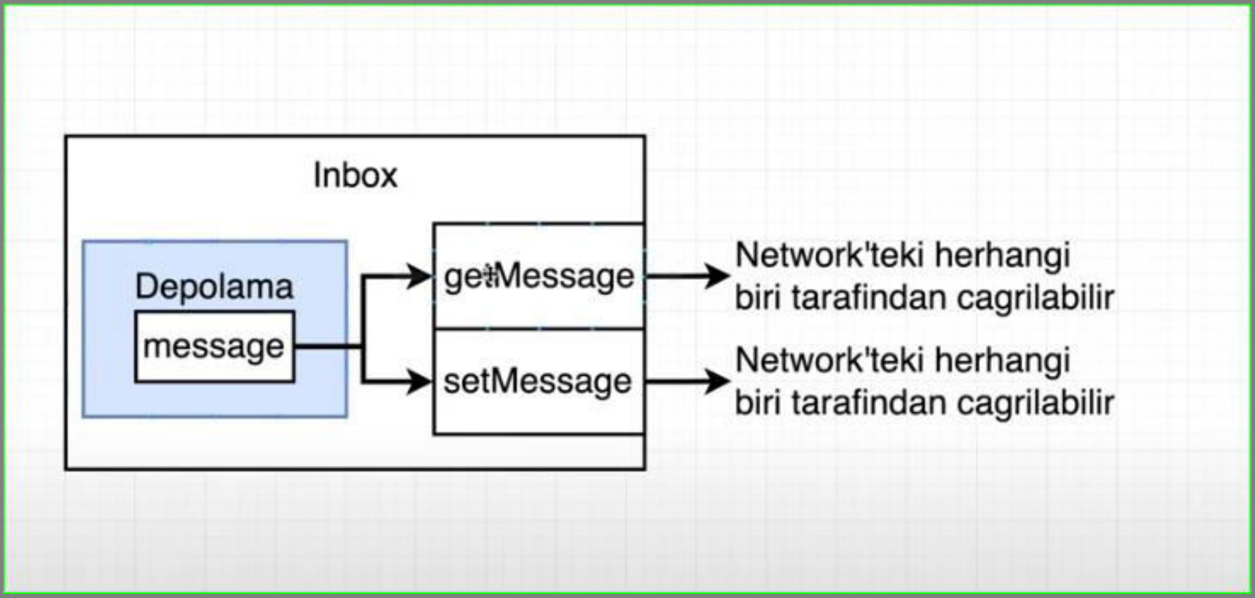
0: string: goodby

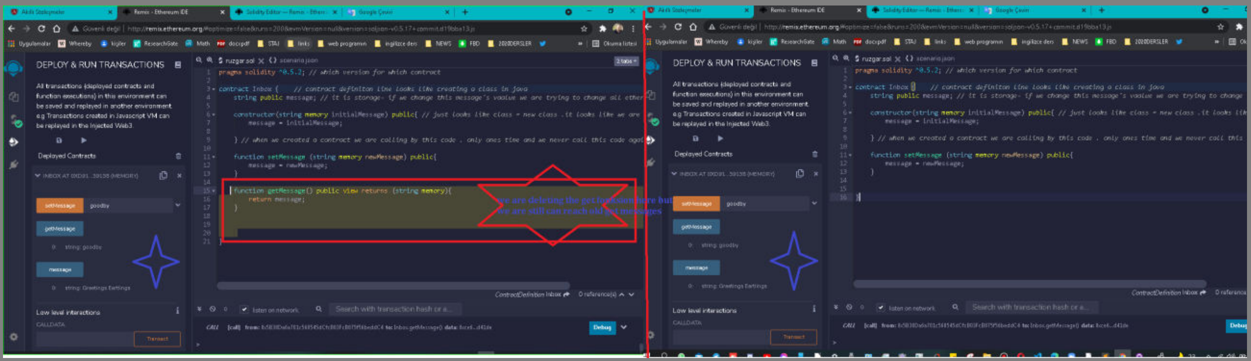
message

0: string: Greetings Eartlings

```
pragma solidity ^0.5.2; // which version for which contract

contract Inbox {    // contract definiton line looks like creating a class in java
    string public message; // it is storage- if we change this message's vaalue we are trying to change all ether

    constructor(string memory initialMessage) public{ // just looks like class = new class .it looks like we are
        message = initialMessage;

    } // when we created a contract we are calling by this code . only ones time and we never call this code agai

    function setMessage (string memory newMessage) public{
        message = newMessage;
    }

    function getMessage() public view returns (string memory){
        return message;
    }



}
```

```
CALL   [call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: Inbox.getMessage() data: 0xce6...d41de
```


Inbox
Depolama
message
getMessage
setMessage
Network'teki herhangi biri tarafindan cagrilabilir
Network'teki herhangi biri tarafindan cagrilabilir



All transactions (deployed contracts and function executions) in this environment can be saved and replayed in another environment. e.g Transactions created in Javascript VM can be replayed in the Injected Web3.

0: string: goodby

0: string: Greetings Earthings

```
pragma solidity ^0.5.2; // which version for which contract

contract Inbox {     // contract definiton line looks like creating a class in java
    string public message; // it is storage- if we change this message's vaalue we are trying to change all ether

    constructor(string memory initialMessage) public{ // just looks like class = new class .it looks like we are
        message = initialMessage;

    } // when we created a contract we are calling by this code . only ones time and we never call this code agai

    function setMessage (string memory newMessage) public{
        message = newMessage;
    }

    function getMessage() public view returns (string memory){
        return message;
    }


}
```

we are deleting the get fonksion here but
we are still can reach old get messages

```
CALL   [call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: Inbox.getMessage() data: 0xce6...d41de
```

```
pragma solidity ^0.5.2; // which version for which contract

contract Inbox {     // contract definiton line looks like creating a class in java
    string public message; // it is storage- if we change this message's vaalue we are trying to change

    constructor(string memory initialMessage) public{ // just looks like class = new class .it looks lik
        message = initialMessage;

    } // when we created a contract we are calling by this code . only ones time and we never call this

    function setMessage (string memory newMessage) public{
        message = newMessage;
    }


}
```

Thanks for listening .

İf you have any question or want to this work results

Contact by E-mail

ruzgaruren@posta.mu.edu.tr