wallet 생성 실습

1. Dependency install

```
pip install web3
```

`web3` 라이브러리는 블록체인과 상호작용하는 애플리케이션(예: 스마트 컨트랙트, 트랜잭션, 이벤트)을 개발하기 위한 Python 또는 JavaScript 라이브러리입니다. Ethereum 네트워크와 상호작용할 수 있도록 설계되었으며, 블록체인의 다양한 작업을 쉽게 구현할 수 있도록 도와줍니다.

```
pip install py-solc-x
```

`py-solc-x`는 Python에서 Solidity 스마트 컨트랙트를 컴파일할 수 있도록 도와주는 라이브러리입니다. Solidity 컴파일러(`solc`)와 통신하여 스마트 컨트랙트를 Python 코드 내에서 컴파일하고 ABI, 바이트코드 등을 생성할 수 있습니다.

이 라이브러리는 다양한 Solidity 버전을 유연하게 관리하고 사용할 수 있도록 설계되었습니다.

2. Create wallet

1. 기능
    a. 개인키 생성
    b. 주소 생성

```
from web3 import Web3
import json

w3 = Web3()

acc = w3.eth.account.create('Input Any String You Like')
private_key = w3.to_hex(acc.key)
account = acc.address
print(private_key)
print(account)
```

```
wallet = {
    'private_key' : private_key,
    'account' : account
}

wallet_file = open('wallet_file.json', 'w')
j = json.dumps(wallet)
wallet_file.write(j)
```

- **지갑 생성 후 https://wallet.test.wemix.com/faucet 에서 10twemix받기 진행**

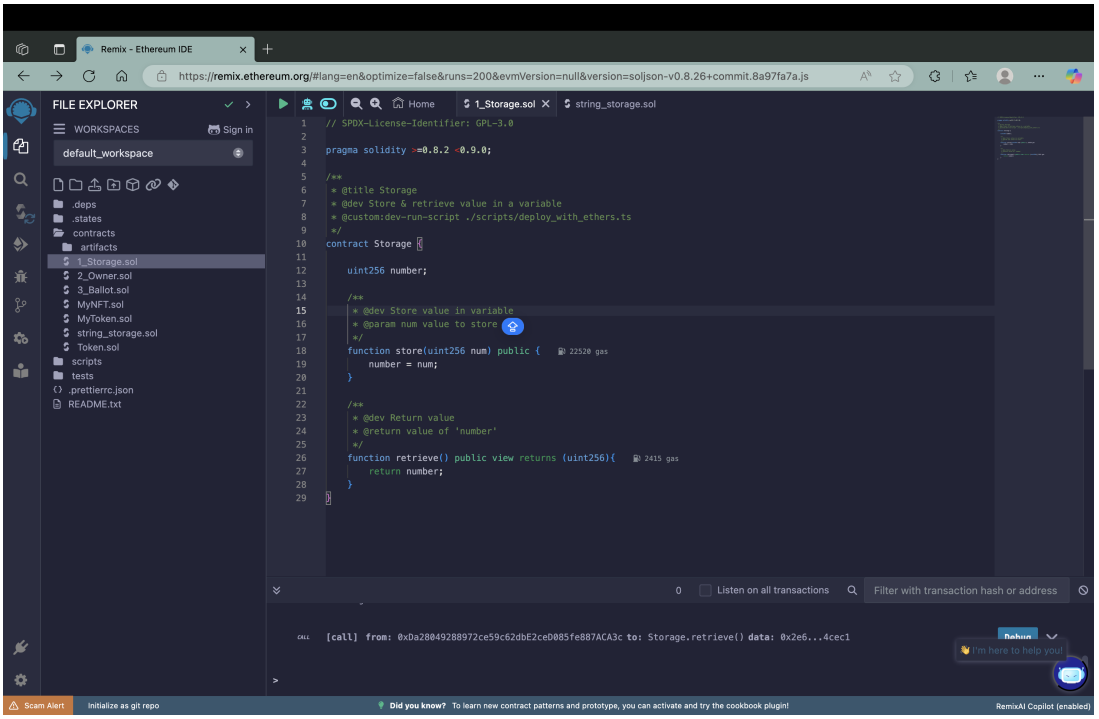
2. contract 사용

   a. 네트워크 연결

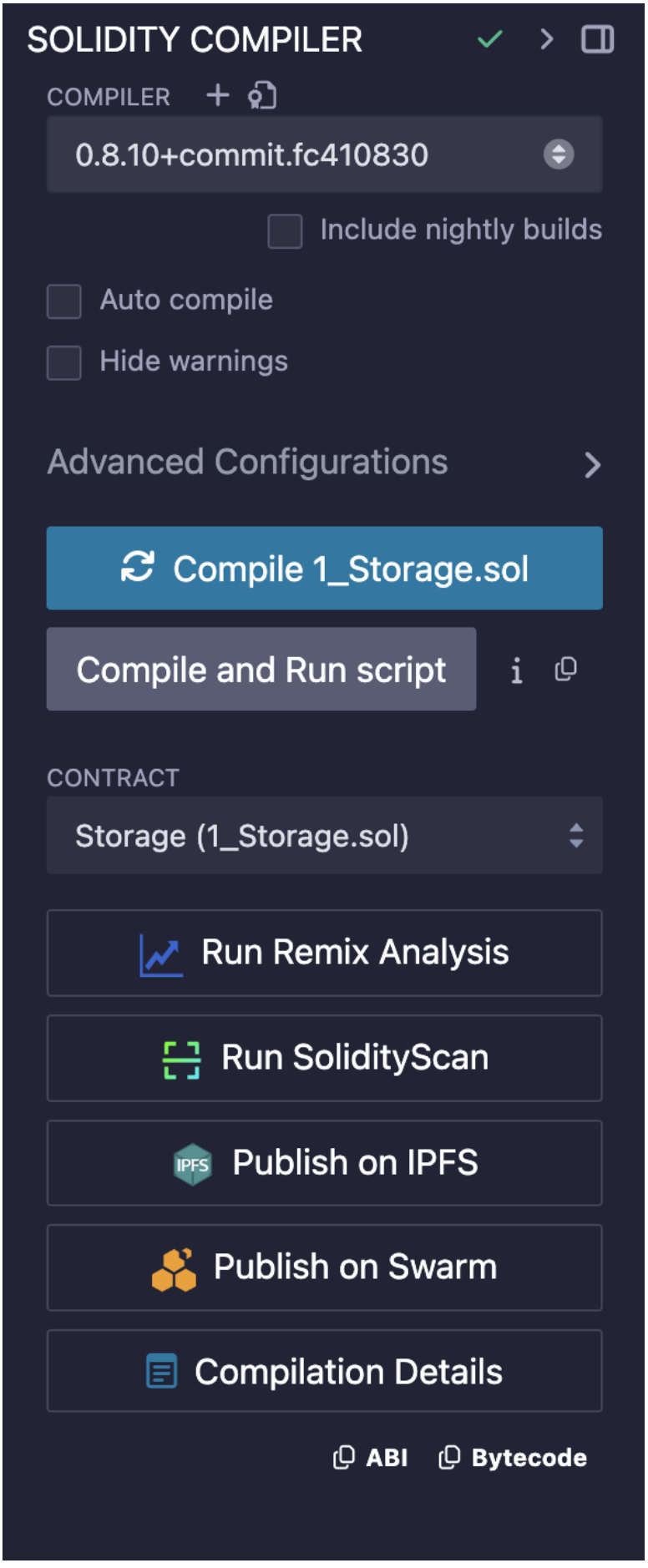
```
from web3 import Web3
import json

wallet_file = open('wallet_file.json', 'r')
wallet = json.load(wallet_file)

private_key = wallet['private_key']
account = wallet['account']

w3 = Web3(Web3.HTTPProvider('https://api.test.wemix.co
m'))
print(w3.is_connected())
```

   b. Remix에서 1_Storage.sol Deploy

   1. Wemix testnet에 연결된 metamask 필요

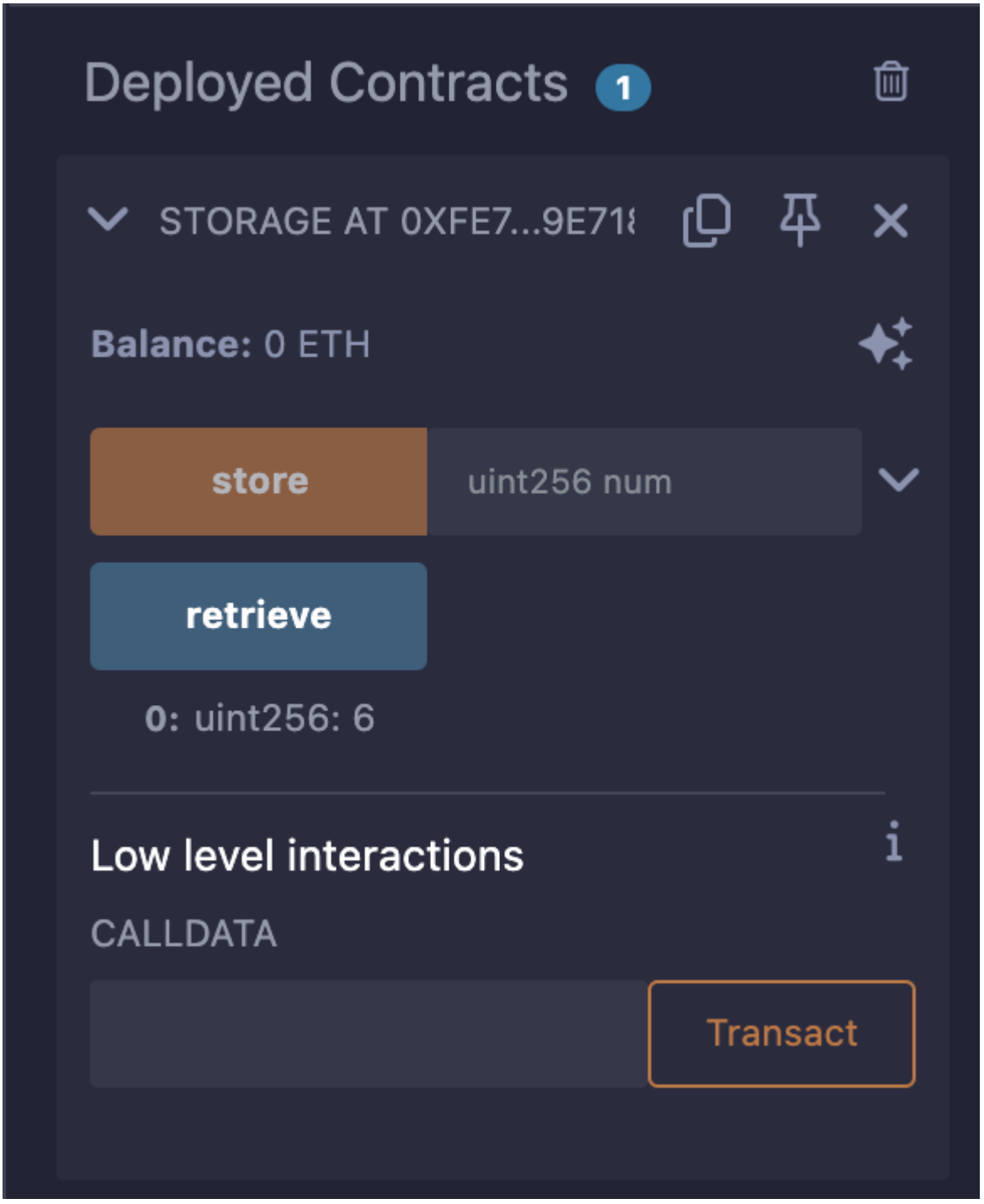


2. 컴파일후 abi 추출 필요


SOLIDITY COMPILER
COMPILER
0.8.10+commit.fc410830
Include nightly builds
Auto compile
Hide warnings
Advanced Configurations
Compile 1_Storage.sol
Compile and Run script
CONTRACT
Storage (1_Storage.sol)
Run Remix Analysis
Run SolidityScan
Publish on IPFS
Publish on Swarm
Compilation Details
ABI
Bytecode

```
[
    {
        "inputs": [
            {
                "internalType": "uint256",
                "name": "num",
                "type": "uint256"
            }
        ],
        "name": "store",
        "outputs": [],
        "stateMutability": "nonpayable",
        "type": "function"
    },
    {
        "inputs": [],
        "name": "retrieve",
        "outputs": [
            {
                "internalType": "uint256",
                "name": "",
                "type": "uint256"
            }
        ],
        "stateMutability": "view",
        "type": "function"
    }
]
```

3. 컨트랙트 주소 가져오기



4. code에 주소와 abi 설정

```
contract_addr = '0xFE72Bf1D03eD53228ee6072a4780a4226009E71
8'
abi = [
    {
        "inputs": [
            {
```

```
                "internalType": "uint256",
                "name": "num",
                "type": "uint256"
            }
        ],
        "name": "store",
        "outputs": [],
        "stateMutability": "nonpayable",
        "type": "function"
    },
    {
        "inputs": [],
        "name": "retrieve",
        "outputs": [
            {
                "internalType": "uint256",
                "name": "",
                "type": "uint256"
            }
        ],
        "stateMutability": "view",
        "type": "function"
    }
]
```

5. 컨트랙트에 변수 저장

```
storage_contract = w3.eth.contract(address=contract_addr, a
bi=abi)
estimated_gas = storage_contract.functions.store(6).estimat
e_gas({'from':account})
gas_price = w3.eth.gas_price
nonce = w3.eth.get_transaction_count(account)
print(nonce)

storage_contract_tx = storage_contract.functions.store(6).b
uild_transaction({
    'from' : account,
```

```
    'gas' : estimated_gas,
    'gasPrice' : gas_price,
    'nonce' : nonce
})

signed_tx = w3.eth.account.sign_transaction(storage_contrac
t_tx, private_key)

tx_hash = w3.eth.send_raw_transaction(signed_tx.raw_transac
tion)
print(f"Transaction sent! Hash: {w3.to_hex(tx_hash)}")
receipt = w3.eth.wait_for_transaction_receipt(tx_hash)
print(f"Transaction mined! Receipt: {receipt}")
```

3. Contract compile & deploy

1. string을 저장할 수 있는 simple_storage.sol

```
// SPDX-License-Identifier: GPL-3.0

pragma solidity ^0.8.10;

contract Storage {

    string private data;


    function store(string memory str) public {
        data = str;
    }

    function retrieve() public view returns (string memory)
{
        return data;
    }
}
```

2. compile.py

```
from solcx import compile_standard
import solcx
import json

solcx.install_solc('0.8.10')
with open("./simple_storage.sol","r") as file:
    simple_storage_file = file.read()

compiled_sol = compile_standard({
                  "language":"Solidity",
                  "sources":{"simple_storage.sol":{"conten
t": simple_storage_file}},
                  "settings":{
                          "outputSelection":{
                       "*": {"*": ["abi","metadata","evm.b
ytecode","evm.sourceMap"]}
                          },
                  },
},
                   solc_version="0.8.10",)

with open("compiled_code.json","w") as file:
    json.dump(compiled_sol,file)
```

3. result

```
{"contracts": {"simple_storage.sol": {"Storage": {"abi":
[{"inputs": [], "name": "retrieve", "outputs": [{"internalT
ype": "string", "name": "", "type": "string"}], "stateMutab
ility": "view", "type": "function"}, {"inputs": [{"internal
Type": "string", "name": "str", "type": "string"}], "name":
"store", "outputs": [], "stateMutability": "nonpayable", "t
ype": "function"}], "evm": {"bytecode": {"functionDebugDat
a": {}, "generatedSources": [], "linkReferences": {}, "obje
ct": "608060405234801561001057600080fd5b506104a880610020600
0396000f3fe608060405234801561001057600080fd5b50600436106100
365760003560e01c8063131a06801461003b5780632e64cec1146100575
75b600080fd5b610055600480360381019061005091906103e565b6100
```

```
75565b005b61005f61008f565b60405161006c91906103ef565b6040518
0910390f35b806000908051906020019061008b929190610121565b5050
565b606060008054610009e90610440565b80601f0160208091040260200
160405190810160405280929190818152602001828054610100ca90610440
565b8015610117578060 1f106100ec576101008083540402835291602 00
191610117565b820191906000526020600020905b815481529060010190
602001808311610fa57829003601f168201915b5050505050905090565
b82805461012d90610440565b9060005260206000209060 1f0160209004
8101928261014f57600085556101965 65b82601f106101685780516 0ff1
916838001178555610196565b828001600101855582156101965791820 1
5b8281115610195578251825591602001919060010196101 7a565b5b5
090506101a391906101a7565b5090565b5b808211156101c05760008160
0090555060010161 01a8565b5090565b60006040519050906 5b600080f
d5b600080fd5b600080fd5b600080fd5b6000601f19601f830116905091
9050565b7f4e487b7100000000000000000000000000000000000000000000
0000000000000000600052604160045260246000fd5b61022b826101e256
5b810181811067ffffffffffffffff8211171561024a576102496101f35
65b5b80604052505050565b600061025d6101c4565b9050610269828261
0222565b919050565b600067ffffffffffffffff8211156102895761028
86101f3565b5b610292826101e2565b905060208101905091905056 5b82
818337600083830152505050565b60006102c16102bc8461026e565b610
253565b90508281526020810184848401111561 02dd576102dc6101dd56
5b5b6102e884828561029f565b5093925050565b600082601f8301126
103055761030461 01d8565b5b81356103158482602086016102ae565b91
505092915050565b6000602082840312156103345761033361 01ce565b5
b600082013567ffffffffffffffff8111156103525761035161 01d3565b
5b61035e848285016102f0565b91505092915050565b600081519050919
050565b600082825260208201905092915050565b60005b838110156103
a1578082015181840152602081019050610386565b838111156103b0576
000848401525b50505050565b60006103c182610367565b6103cb818561
0372565b93506103db818560208601610383565b6103e4816101e2565b8
401915050929150505 65b60006020820190508181036000830152610409
81846103b6565b905092915050565b7f4e487b7100000000000000000000
0000000000000000000000000000000000000000600052602260452602460
00fd5b6000600282049050600182168061045857607f821691505b60208
2108114156104 6c5761046b610411565b5b5091905056fea26469706673
58221220847c35d51f1b5ba7ba6a66b67f04fceeef8941cccee49ac5551
401a68c36286a64736f6c634300080a0033", "opcodes": "PUSH1 0x8
```

```
0 PUSH1 0x40 MSTORE CALLVALUE DUP1 ISZERO PUSH2 0x10 JUMPI
PUSH1 0x0 DUP1 REVERT JUMPDEST POP PUSH2 0x4A8 DUP1 PUSH2 0
x20 PUSH1 0x0 CODECOPY PUSH1 0x0 RETURN INVALID PUSH1 0x80
PUSH1 0x40 MSTORE CALLVALUE DUP1 ISZERO PUSH2 0x10 JUMPI PU
SH1 0x0 DUP1 REVERT JUMPDEST POP PUSH1 0x4 CALLDATASIZE LT
PUSH2 0x36 JUMPI PUSH1 0x0 CALLDATALOAD PUSH1 0xE0 SHR DUP1
PUSH4 0x131A0680 EQ PUSH2 0x3B JUMPI DUP1 PUSH4 0x2E64CEC1
EQ PUSH2 0x57 JUMPI JUMPDEST PUSH1 0x0 DUP1 REVERT JUMPDEST
PUSH2 0x55 PUSH1 0x4 DUP1 CALLDATASIZE SUB DUP2 ADD SWAP1 P
USH2 0x50 SWAP2 SWAP1 PUSH2 0x31E JUMP JUMPDEST PUSH2 0x75
JUMP JUMPDEST STOP JUMPDEST PUSH2 0x5F PUSH2 0x8F JUMP JUMP
DEST PUSH1 0x40 MLOAD PUSH2 0x6C SWAP2 SWAP1 PUSH2 0x3EF JU
MP JUMPDEST PUSH1 0x40 MLOAD DUP1 SWAP2 SUB SWAP1 RETURN JU
MPDEST DUP1 PUSH1 0x0 SWAP1 DUP1 MLOAD SWAP1 PUSH1 0x20 ADD
SWAP1 PUSH2 0x8B SWAP3 SWAP2 SWAP1 PUSH2 0x121 JUMP JUMPDES
T POP POP JUMP JUMPDEST PUSH1 0x60 PUSH1 0x0 DUP1 SLOAD PUS
H2 0x9E SWAP1 PUSH2 0x440 JUMP JUMPDEST DUP1 PUSH1 0x1F ADD
PUSH1 0x20 DUP1 SWAP2 DIV MUL PUSH1 0x20 ADD PUSH1 0x40 MLO
AD SWAP1 DUP2 ADD PUSH1 0x40 MSTORE DUP1 SWAP3 SWAP2 SWAP1
DUP2 DUP2 MSTORE PUSH1 0x20 ADD DUP3 DUP1 SLOAD PUSH2 0xCA
SWAP1 PUSH2 0x440 JUMP JUMPDEST DUP1 ISZERO PUSH2 0x117 JUM
PI DUP1 PUSH1 0x1F LT PUSH2 0xEC JUMPI PUSH2 0x100 DUP1 DUP
4 SLOAD DIV MUL DUP4 MSTORE SWAP2 PUSH1 0x20 ADD SWAP2 PUSH
2 0x117 JUMP JUMPDEST DUP3 ADD SWAP2 SWAP1 PUSH1 0x0 MSTORE
PUSH1 0x20 PUSH1 0x0 KECCAK256 SWAP1 JUMPDEST DUP2 SLOAD DU
P2 MSTORE SWAP1 PUSH1 0x1 ADD SWAP1 PUSH1 0x20 ADD DUP1 DUP
4 GT PUSH2 0xFA JUMPI DUP3 SWAP1 SUB PUSH1 0x1F AND DUP3 AD
D SWAP2 JUMPDEST POP POP POP POP POP SWAP1 POP SWAP1 JUMP J
UMPDEST DUP3 DUP1 SLOAD PUSH2 0x12D SWAP1 PUSH2 0x440 JUMP
JUMPDEST SWAP1 PUSH1 0x0 MSTORE PUSH1 0x20 PUSH1 0x0 KECCAK
256 SWAP1 PUSH1 0x1F ADD PUSH1 0x20 SWAP1 DIV DUP2 ADD SWAP
3 DUP3 PUSH2 0x14F JUMPI PUSH1 0x0 DUP6 SSTORE PUSH2 0x196
JUMP JUMPDEST DUP3 PUSH1 0x1F LT PUSH2 0x168 JUMPI DUP1 MLO
AD PUSH1 0xFF NOT AND DUP4 DUP1 ADD OR DUP6 SSTORE PUSH2 0x
196 JUMP JUMPDEST DUP3 DUP1 ADD PUSH1 0x1 ADD DUP6 SSTORE D
UP3 ISZERO PUSH2 0x196 JUMPI SWAP2 DUP3 ADD JUMPDEST DUP3 D
UP2 GT ISZERO PUSH2 0x195 JUMPI DUP3 MLOAD DUP3 SSTORE SWAP
2 PUSH1 0x20 ADD SWAP2 SWAP1 PUSH1 0x1 ADD SWAP1 PUSH2 0x17
```

```
A JUMP JUMPDEST JUMPDEST POP SWAP1 POP PUSH2 0x1A3 SWAP2 SW
AP1 PUSH2 0x1A7 JUMP JUMPDEST POP SWAP1 JUMP JUMPDEST JUMPD
EST DUP1 DUP3 GT ISZERO PUSH2 0x1C0 JUMPI PUSH1 0x0 DUP2 PU
SH1 0x0 SWAP1 SSTORE POP PUSH1 0x1 ADD PUSH2 0x1A8 JUMP JUM
PDEST POP SWAP1 JUMP JUMPDEST PUSH1 0x0 PUSH1 0x40 MLOAD SW
AP1 POP SWAP1 JUMP JUMPDEST PUSH1 0x0 DUP1 REVERT JUMPDEST
PUSH1 0x0 DUP1 REVERT JUMPDEST PUSH1 0x0 DUP1 REVERT JUMPDE
ST PUSH1 0x0 DUP1 REVERT JUMPDEST PUSH1 0x0 PUSH1 0x1F NOT
PUSH1 0x1F DUP4 ADD AND SWAP1 POP SWAP2 SWAP1 POP JUMP JUMP
DEST PUSH32 0x4E487B7100000000000000000000000000000000000000
0000000000000000 PUSH1 0x0 MSTORE PUSH1 0x41 PUSH1 0x4 M
STORE PUSH1 0x24 PUSH1 0x0 REVERT JUMPDEST PUSH2 0x22B DUP3
PUSH2 0x1E2 JUMP JUMPDEST DUP2 ADD DUP2 DUP2 LT PUSH8 0xFFF
FFFFFFFFFFFF DUP3 GT OR ISZERO PUSH2 0x24A JUMPI PUSH2 0x2
49 PUSH2 0x1F3 JUMP JUMPDEST JUMPDEST DUP1 PUSH1 0x40 MSTOR
E POP POP POP JUMP JUMPDEST PUSH1 0x0 PUSH2 0x25D PUSH2 0x1
C4 JUMP JUMPDEST SWAP1 POP PUSH2 0x269 DUP3 DUP3 PUSH2 0x22
2 JUMP JUMPDEST SWAP2 SWAP1 POP JUMP JUMPDEST PUSH1 0x0 PUS
H8 0xFFFFFFFFFFFFFFFF DUP3 GT ISZERO PUSH2 0x289 JUMPI PUSH
2 0x288 PUSH2 0x1F3 JUMP JUMPDEST JUMPDEST PUSH2 0x292 DUP3
PUSH2 0x1E2 JUMP JUMPDEST SWAP1 POP PUSH1 0x20 DUP2 ADD SWA
P1 POP SWAP2 SWAP1 POP JUMP JUMPDEST DUP3 DUP2 DUP4 CALLDAT
ACOPY PUSH1 0x0 DUP4 DUP4 ADD MSTORE POP POP POP JUMP JUMPD
EST PUSH1 0x0 PUSH2 0x2C1 PUSH2 0x2BC DUP5 PUSH2 0x26E JUMP
JUMPDEST PUSH2 0x253 JUMP JUMPDEST SWAP1 POP DUP3 DUP2 MSTO
RE PUSH1 0x20 DUP2 ADD DUP5 DUP5 DUP5 ADD GT ISZERO PUSH2 0
x2DD JUMPI PUSH2 0x2DC PUSH2 0x1DD JUMP JUMPDEST JUMPDEST P
USH2 0x2E8 DUP5 DUP3 DUP6 PUSH2 0x29F JUMP JUMPDEST POP SWA
P4 SWAP3 POP POP POP JUMP JUMPDEST PUSH1 0x0 DUP3 PUSH1 0x1
F DUP4 ADD SLT PUSH2 0x305 JUMPI PUSH2 0x304 PUSH2 0x1D8 JU
MP JUMPDEST JUMPDEST DUP2 CALLDATALOAD PUSH2 0x315 DUP5 DUP
3 PUSH1 0x20 DUP7 ADD PUSH2 0x2AE JUMP JUMPDEST SWAP2 POP P
OP SWAP3 SWAP2 POP POP JUMP JUMPDEST PUSH1 0x0 PUSH1 0x20 D
UP3 DUP5 SUB SLT ISZERO PUSH2 0x334 JUMPI PUSH2 0x333 PUSH2
0x1CE JUMP JUMPDEST JUMPDEST PUSH1 0x0 DUP3 ADD CALLDATALOA
D PUSH8 0xFFFFFFFFFFFFFFFF DUP2 GT ISZERO PUSH2 0x352 JUMPI
PUSH2 0x351 PUSH2 0x1D3 JUMP JUMPDEST JUMPDEST PUSH2 0x35E
DUP5 DUP3 DUP6 ADD PUSH2 0x2F0 JUMP JUMPDEST SWAP2 POP POP
```

SWAP3 SWAP2 POP POP JUMP JUMPDEST PUSH1 0x0 DUP2 MLOAD SWAP
1 POP SWAP2 SWAP1 POP JUMP JUMPDEST PUSH1 0x0 DUP3 DUP3 MST
ORE PUSH1 0x20 DUP3 ADD SWAP1 POP SWAP3 SWAP2 POP POP JUMP
JUMPDEST PUSH1 0x0 JUMPDEST DUP4 DUP2 LT ISZERO PUSH2 0x3A1
JUMPI DUP1 DUP3 ADD MLOAD DUP2 DUP5 ADD MSTORE PUSH1 0x20 D
UP2 ADD SWAP1 POP PUSH2 0x386 JUMP JUMPDEST DUP4 DUP2 GT IS
ZERO PUSH2 0x3B0 JUMPI PUSH1 0x0 DUP5 DUP5 ADD MSTORE JUMPD
EST POP POP POP POP JUMP JUMPDEST PUSH1 0x0 PUSH2 0x3C1 DUP
3 PUSH2 0x367 JUMP JUMPDEST PUSH2 0x3CB DUP2 DUP6 PUSH2 0x3
72 JUMP JUMPDEST SWAP4 POP PUSH2 0x3DB DUP2 DUP6 PUSH1 0x20
DUP7 ADD PUSH2 0x383 JUMP JUMPDEST PUSH2 0x3E4 DUP2 PUSH2 0
x1E2 JUMP JUMPDEST DUP5 ADD SWAP2 POP POP SWAP3 SWAP2 POP P
OP JUMP JUMPDEST PUSH1 0x0 PUSH1 0x20 DUP3 ADD SWAP1 POP DU
P2 DUP2 SUB PUSH1 0x0 DUP4 ADD MSTORE PUSH2 0x409 DUP2 DUP5
PUSH2 0x3B6 JUMP JUMPDEST SWAP1 POP SWAP3 SWAP2 POP POP JUM
P JUMPDEST PUSH32 0x4E487B7100000000000000000000000000000000
0000000000000000000000 PUSH1 0x0 MSTORE PUSH1 0x22 PUSH1
0x4 MSTORE PUSH1 0x24 PUSH1 0x0 REVERT JUMPDEST PUSH1 0x0 P
USH1 0x2 DUP3 DIV SWAP1 POP PUSH1 0x1 DUP3 AND DUP1 PUSH2 0
x458 JUMPI PUSH1 0x7F DUP3 AND SWAP2 POP JUMPDEST PUSH1 0x2
0 DUP3 LT DUP2 EQ ISZERO PUSH2 0x46C JUMPI PUSH2 0x46B PUSH
2 0x411 JUMP JUMPDEST JUMPDEST POP SWAP2 SWAP1 POP JUMP INV
ALID LOG2 PUSH5 0x6970667358 0x22 SLT KECCAK256 DUP5 PUSH29
0x35D51F1B5BA7BA6A66B67F04FCEEEF8941CCCEE49AC5551401A68C362
8 PUSH11 0x64736F6C634300080A0033 ", "sourceMap": "199:285:
0:-:0;;;;;;;;;;;;;;;;;;;;;"}}, "metadata": "{\"compiler\":
{\"version\":\"0.8.10+commit.fc410830\"},\"language\":\"Sol
idity\",\"output\":{\"abi\":[{\"inputs\":[],\"name\":\"retr
ieve\",\"outputs\":[{\"internalType\":\"string\",\"name
\":\"\",\"type\":\"string\"}],\"stateMutability\":\"view
\",\"type\":\"function\"},{\"inputs\":[{\"internalType
\":\"string\",\"name\":\"str\",\"type\":\"string\"}],\"name
\":\"store\",\"outputs\":[],\"stateMutability\":\"nonpayabl
e\",\"type\":\"function\"}],\"devdoc\":{\"custom:dev-run-sc
ript\":\"./scripts/deploy_with_ethers.ts\",\"details\":\"St
ore & retrieve value in a variable\",\"kind\":\"dev\",\"met
hods\":{\"retrieve()\":{\"details\":\"Return value \",\"ret
urns\":{\"_0\":\"value of 'number'\"}}},\"title\":\"Storage

```
\",\"version\":1},\"userdoc\":{\"kind\":\"user\",\"methods
\":{},\"version\":1}},\"settings\":{\"compilationTarget\":
{\"simple_storage.sol\":\"Storage\"},\"evmVersion\":\"londo
n\",\"libraries\":{},\"metadata\":{\"bytecodeHash\":\"ipfs
\"},\"optimizer\":{\"enabled\":false,\"runs\":200},\"remapp
ings\":[]},\"sources\":{\"simple_storage.sol\":{\"keccak256
\":\"0x9c44660c47d104f65777cd5c93f84e3444ccfe9ce093dac7414f
933fcd2980a9\",\"license\":\"GPL-3.0\",\"urls\":[\"bzz-ra
w://b37748a91b781e02103dc4b114295ec2a587737407fd0c4f2164b75
352d1918c\",\"dweb:/ipfs/QmRfXxvUoyZcqVfkhVuUTvqbxYbkSpsCPN
7f2i1yHRDoin\"]}},\"version\":1}"}}}, "sources": {"simple_s
torage.sol": {"id": 0}}}
```

4. contract deploy

```
from web3 import Web3
import json

wallet_file = open('wallet_file.json', 'r')
wallet = json.load(wallet_file)

private_key = wallet['private_key']
account = wallet['account']

contract_file = open('compiled_code.json', 'r')
contract = json.load(contract_file)

byte_code = contract['contracts']['simple_storage.sol']['St
orage']['evm']['bytecode']['object']
abi = contract['contracts']['simple_storage.sol']['Storag
e']['abi']

w3 = Web3(Web3.HTTPProvider('https://api.test.wemix.com'))
print(w3.is_connected())

storage_contract = w3.eth.contract(abi=abi, bytecode=byte_c
ode)
gas_price = w3.eth.gas_price
```

```
nonce = w3.eth.get_transaction_count(account)

tx = storage_contract.constructor().build_transaction({
    'from' : account,
    'gasPrice' : gas_price,
    'nonce' : nonce
})

signed_tx = w3.eth.account.sign_transaction(tx, private_ke
y)

tx_hash = w3.eth.send_raw_transaction(signed_tx.raw_transac
tion)
print(f"Transaction sent! Hash: {w3.to_hex(tx_hash)}")
receipt = w3.eth.wait_for_transaction_receipt(tx_hash)
print(f"Transaction mined! Receipt: {receipt}")

#0xf96014d896C7cC4Bd2d6D10311638C263899536a
```

5. use contract

```
from web3 import Web3
import json

wallet_file = open('wallet_file.json', 'r')
wallet = json.load(wallet_file)

private_key = wallet['private_key']
account = wallet['account']

contract_file = open('compiled_code.json', 'r')
contract = json.load(contract_file)

w3 = Web3(Web3.HTTPProvider('https://api.test.wemix.com'))
print(w3.is_connected())

contract_addr = '0xf96014d896C7cC4Bd2d6D10311638C263899536
a'
```

```
abi = contract['contracts']['simple_storage.sol']['Storag
e']['abi']

storage_contract = w3.eth.contract(address=contract_addr, a
bi=abi)
estimated_gas = storage_contract.functions.store('Hello').e
stimate_gas({'from':account})
gas_price = w3.eth.gas_price
nonce = w3.eth.get_transaction_count(account)
print(nonce)

storage_contract_tx = storage_contract.functions.store('Hel
lo').build_transaction({
    'from' : account,
    'gas' : estimated_gas,
    'gasPrice' : gas_price,
    'nonce' : nonce
})

signed_tx = w3.eth.account.sign_transaction(storage_contrac
t_tx, private_key)

tx_hash = w3.eth.send_raw_transaction(signed_tx.raw_transac
tion)
print(f"Transaction sent! Hash: {w3.to_hex(tx_hash)}")
receipt = w3.eth.wait_for_transaction_receipt(tx_hash)
print(f"Transaction mined! Receipt: {receipt}")
```

6. call storage retrieve

```
from web3 import Web3
import json

wallet_file = open('wallet_file.json', 'r')
wallet = json.load(wallet_file)

private_key = wallet['private_key']
account = wallet['account']
```

```
contract_file = open('compiled_code.json', 'r')
contract = json.load(contract_file)

w3 = Web3(Web3.HTTPProvider('https://api.test.wemix.com'))
print(w3.is_connected())

contract_addr = '0xf96014d896C7cC4Bd2d6D10311638C263899536
a'
abi = contract['contracts']['simple_storage.sol']['Storag
e']['abi']

storage_contract = w3.eth.contract(address=contract_addr, a
bi=abi)

retrive = storage_contract.functions.retrieve().call()
print(retrive)
```