In [28]:

```
import json
```

In [29]:

```
from web3 import Web3
```

In [3]:

```
ganache_url = "HTTP://127.0.0.1:7545"
```

In [4]:

```
web3 = Web3(Web3.HTTPProvider(ganache_url))
```

In [5]:

```
web3.isConnected()
```

Out[5]:

```
True
```

In [6]:

```
web3.eth.defaultAccount = web3.eth.accounts[0]
```

In [7]:

```
abi = json.loads('[{"inputs":[],"payable":false,"stateMutability":"nonpayable","type
byteCode= '608060405260043610610055576000356e01c806312065fe01461005a578063481c6a751
```

In [8]:

```
Lottery = web3.eth.contract(abi=abi, bytecode = byteCode)
```

In [9]:

```
contract = web3.eth.contract(
   address = '0x6D85923e12d04a54Db0e30d87F6d9792fdc8d053',
   abi = abi)
```

In [10]:

```
print(contract.functions.getPlayers().call())
```

```
['0x5F85E0929DEecC48782A08c3b44ec2918F0346FA', '0xEedb5de916B6C252568737F36Ac1461C0c
6576ea']
```

In [19]:

```
tx_hash = contract.functions.enter().transact(
  {'from' : web3.eth.accounts[2], 'value' : 1000000000000000000 }
)
```

In [21]:

```
tx_receipt = web3.eth.waitForTransactionReceipt(tx_hash)
```

In [11]:

```
balance = web3.eth.getBalance('0x6D85923e12d04a54Db0e30d87F6d9792fdc8d053')
```

In [12]:

```
balance
```

Out[12]:

```
2000000000000000000
```

In [30]:

```
infura_url = 'https://ropsten.infura.io/v3/085c4f80a1514422994bfe8a15fb5b70'
```

In [31]:

```
web3 = Web3(Web3.HTTPProvider(infura_url))
```

```
In [32]: web3.isConnected()
```

```
Out[32]: True
```

```
In [33]: web3.eth.blockNumber
```

```
Out[33]: 10205997
```

```
In [34]: balance = web3.eth.getBalance('0xe770909707aF1285b111C8d2a9a9434D0BfDA825')
```

```
In [35]: balance
```

```
Out[35]: 499006238000000000
```

```
In [36]: print(web3.fromWei(balance, "ether"))
```

```
0.499006238
```

```
In [37]: abi = json.loads('[{"constant":false,"inputs":[],"name":"enter","outputs":[],"payabl
         address = '0x251791e2BB0382Af9647E07D509D78E14E878940'
```

```
In [38]: contract = web3.eth.contract(address=address, abi = abi)
```

```
In [41]: print(contract.functions.getPlayers().call())
```

```
['0x49354A8E856CBD5E8e68aD8d7876b663BbA73d22']
```

```
In [42]: print(contract.functions.getBalance().call())
```

```
50000000000000000
```

```
In [ ]: # так не будет работать смотрите решение  в презентации.
        #tx_hash = contract.functions.enter().transact(
        #  {'from' : '0x49354A8E856CBD5E8e68aD8d7876b663BbA73d22', 'value' : 500000000000000
        #)
```

```
In [1]: # перевод денег
        account1 = "0xe770909707aF1285b111C8d2a9a9434D0BfDA825"
        account2 = "0x49354A8E856CBD5E8e68aD8d7876b663BbA73d22"
        private_key = "0x1d919f04fdbf76e2f455bfc19ea396c025e5c23976806d78dce23eabfc0faf30"
        #Получите количество транзакций, отправленных с этого адреса.
        nonce = web3.eth.getTransactionCount(account1)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-1-3b1e71f85b21> in <module>
      4 private_key = "0x1d919f04fdbf76e2f455bfc19ea396c025e5c23976806d78dce23eabfc0
faf30"
      5 #Получите количество транзакций, отправленных с этого адреса.
----> 6 nonce = web3.eth.getTransactionCount(account1)
```

```
NameError: name 'web3' is not defined
```

In [44]:

```
nonce
```

Out[44]:

```
3
```

In [48]:

```
tx = {
    'nonce':nonce,
    'to':account2,
    'value':web3.toWei('0.01', 'ether'),
    'gas':2000000,
    'gasPrice':web3.toWei('50', 'gwei')
}
signed_tx = web3.eth.account.signTransaction(tx, private_key)
```

In [49]:

```
tx_hash = web3.eth.sendRawTransaction(signed_tx.rawTransaction)
```

In [50]:

```
print(web3.toHex(tx_hash))
```

```
0x5f992ca7a0732cc9ae16c2aca1aaf0822ba9bd9c9f90bb8e390221737eff701d
```

In [ ]: