

Smart Contract Lab

Task 1.a: Connecting Remix to the SEED Emulator

The image displays the Remix IDE interface with the 'Seed Emulator' configuration window open. The window shows fields for Network name (Seed Emulator), Default RPC URL (10.160.0.71:8545), Chain ID (1337), Currency symbol (ETH), and Block explorer URL. A 'Save' button is at the bottom.

Overlaid on the right is the 'remix.ethereum.org' MetaMask connection modal, showing a list of accounts and a 'Connect' button.

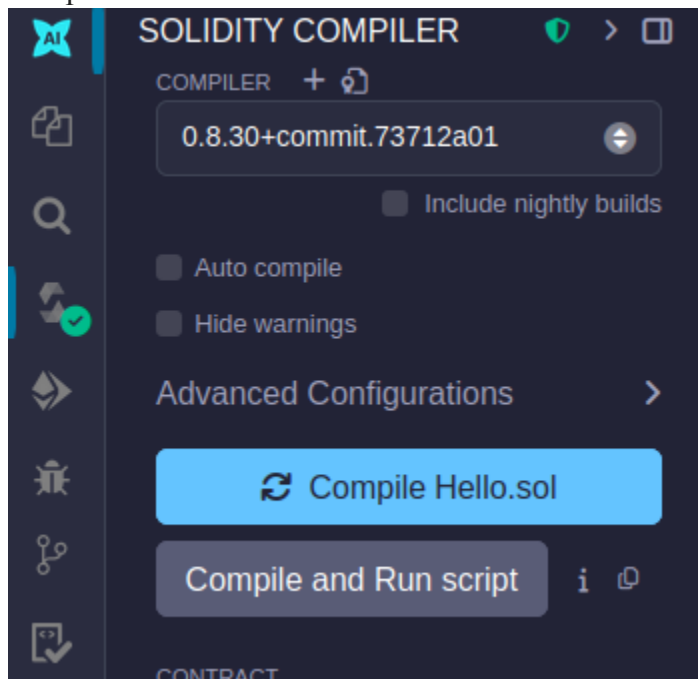
Below the configuration window is a list of Ethereum addresses (0x addresses) with 0 ETH balance, including:

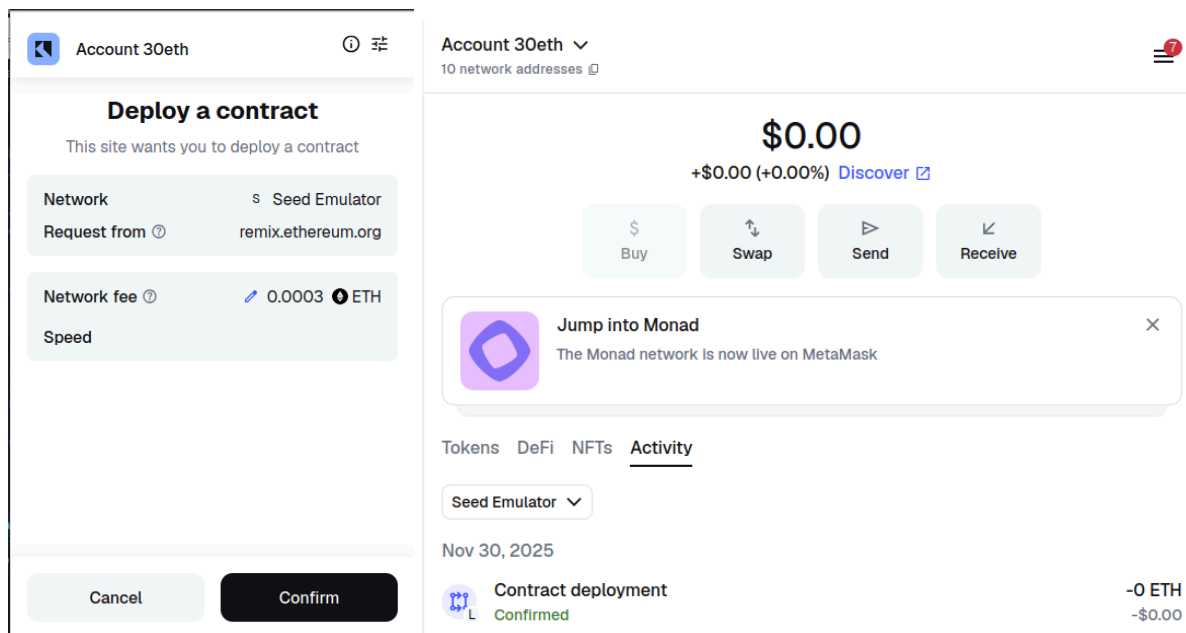
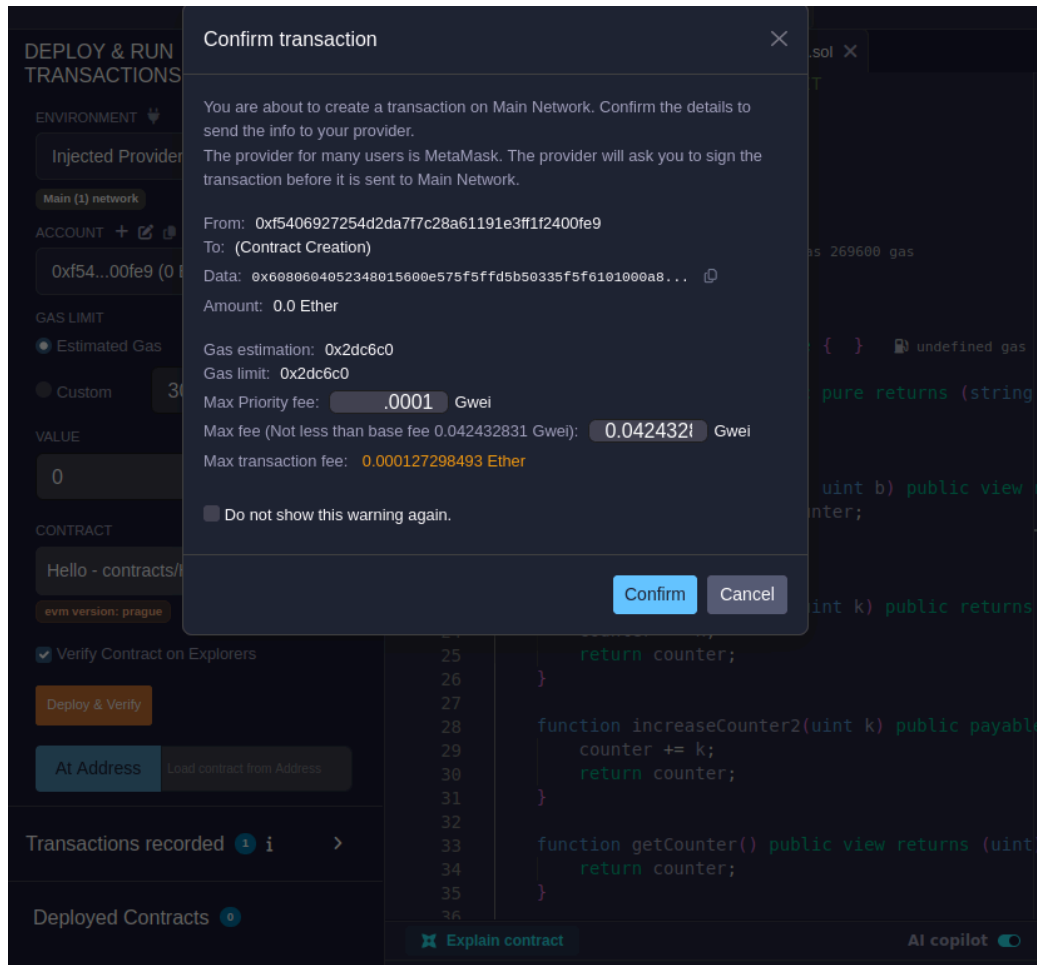
- 0xf54...00fe9 (0 ETH)
- 0x2e2...a88e9 (0 ETH)
- 0xcbf...02e24 (0 ETH)
- 0xa08...26af8 (0 ETH)
- 0x52b...bb3bb (0 ETH)
- 0x3a1...4ba77 (0 ETH)
- 0x127...a1341 (0 ETH)
- 0xab5...54890 (0 ETH)
- 0xeb4...0e2c3 (0 ETH)
- 0x5c5...47c1a (0 ETH)
- 0x963...4c747 (0 ETH)
- 0xf23...c853a (0 ETH)
- 0x5a7...62222 (0 ETH)
- 0xaa8...f045e (0 ETH)
- 0x99a...a48c6 (0 ETH)
- 0x939...b12e6 (0 ETH)
- 0x4e5...df459 (0 ETH)
- 0x4f3...88247 (0 ETH)
- 0x1d3...2d6cb (0 ETH)
- 0x697...af4e0 (0 ETH)
- 0x8b8...85c52 (0 ETH)

Task 1.b: Write, Compile, and Deploy Smart Contract

```
1 // SPDX-License-Identifier: MIT
2 pragma solidity ^0.8.15;
3
4 contract Hello{
5     address public owner;
6     uint256 counter;
7
8     constructor(){ 294174 gas 269600 gas
9         owner = msg.sender;
10    }
11
12    receive() external payable { } undefined gas
13
14    function sayHello() public pure returns (string memory) { infinite gas
15        return "Hello World";
16    }
17
18    function getResult(uint a, uint b) public view returns (uint) { infinite gas
19        uint sum = a + b + counter;
20        return sum;
21    }
22
23    function increaseCounter(uint k) public returns (uint) { infinite gas
24        counter += k;
25        return counter;
26    }
27
28    function increaseCounter2(uint k) public payable returns (uint) { infinite gas
29        counter += k;
30        return counter;
31    }
32
33    function getCounter() public view returns (uint) { 2455 gas
34        return counter;
35    }
36}
```

Compile Hello.sol





Recent Transactions

535: [0x2f08...3532](#) (0xF54...fe --> null)

480: [0x229e...8759](#) (0xF54...fe --> null)

467: [0xd951...c3aa](#) (0xF54...fe --> null)

397: [0x3579...9f21](#) (0x2e2...8e --> null)

394: [0xef84...5e0b](#) (0x2e2...8e --> null)

366: [0x3022...6221](#) (0xF54...fe --> null)

330: [0x87f3...4415](#) (0x2e2...8e --> null)

Task 1.c: Under the Hood

Transaction	Receipt
hash	0x2F083af1a033f2bff689d44b3b2710cf87293ebe3fb48a0eebc67efd25a35321
type	2
accessList	
blockHash	0x1753480bed3708006e0731cc3084679229a37ab653e9909e084f5328ffb593f9
blockNumber	535
transactionIndex	0
confirmations	24
from	0xF5406927254d2dA7F7c28A61191e3FF1f2400fe9
gasPrice	100000007
maxPriorityFeePerGas	100000007
maxFeePerGas	100000007
gasLimit	290883
to	null
value	0
nonce	5
data	0x608060405234801561001057600080fd5b50336000806101000a81548173ffffffffffffffff
r	0x564e01345905f71363f07d7972fd942d978c8138af28348eb49611cf5d535e81
s	0x63c03e04a6bf69f50afbc7c5d2d47126f4341697e8ed4eea43cdf2dac2152cd5
v	1
creates	0x4e6A5dbdf46270331aC1618F1497BEFbFee00b47
chainId	1337
wait	(omitted)

We can see that the main differences between contract deployment transaction and normal fund transfer is that deployment has no “to” address, deployment stores bytecode in the data field and gas usage is higher

Task 2: Invoke Contract Functions

Task 2.a: Invoke a function via local call

[illegible]

Deployed Contracts 1

▼ HELLO AT 0x56B...04EAA (BLU)

Balance: 0 ETH

getCounter

0: uint256: 0

getResultuint256 a, uint256 b

increaseCounteruint256 k

increaseCount...uint256 k

owner

sayHello

0: string: Hello World

sendMessage

logs

raw logs

call to Hello.getCounter

[call]
from:
0xf5406927254d2da7f7c28a61191e3ff1f2400fe9
to: Hello.getCounter() data: 0x8ad...a066e

Debug

from0xf5406927254d2da7f7c28a61191e3ff1f2400fe9

toHello.getCounter()

0x56B77E686FDd9BEe48255A2086c4C26cCC04eaa

input0x8ad...a066e

output00000000000000000000000000000000

decoded input{}

decoded output{"0": "uint256: 0"}

logs

raw logs

Task 2.b: Invoke a function via transaction

The screenshot shows a blockchain IDE interface. On the left, a sidebar displays the contract state: "HELLO AT 0X417...3EA27 (BLC)" with a balance of 0 ETH. Below this, there are several function buttons: "decreaseCounter...", "increaseCounter" (set to 1), "increaseCounter..." (set to uint256 k), "sendMessage", "getCounter", "getResult" (set to uint256 a, uint256 b), and "owner". The "getCounter" button is highlighted in blue. Below the buttons, the state is shown as "0: uint256: 1". On the right, a "call to Hello.getCounter" transaction is displayed. The transaction details include: "from: 0xf5406927254d2da7f7c28a61191e3ff1f2400fe9", "to: Hello.getCounter()", "input: 0x8ad...a066e", "output: 0001", "decoded input: {}", "decoded output: { '0': 'uint256: 1' }", "logs: []", and "raw logs: []". A "Debug" button is visible next to the transaction details.

From this we can see that the counter indeed go up by 1, Now we implement decreaseCounter

```
function decreaseCounter(uint k) public returns (uint) {  
    counter -= k;  
    return counter;  
}
```

The screenshot shows the same blockchain IDE interface as before, but with the "decreaseCounter" button highlighted in blue. The state is now "0: uint256: 0". On the right, a "call to Hello.getCounter" transaction is displayed. The transaction details include: "from: 0xf5406927254d2da7f7c28a61191e3ff1f2400fe9", "to: Hello.getCounter()", "input: 0x8ad...a066e", "output: 00", "decoded input: {}", "decoded output: { '0': 'uint256: 0' }", "logs: []", and "raw logs: []". A "Debug" button is visible next to the transaction details.

From this we can see that counter gets decreased after calling decreaseCounter

Task 2.c: Under the hood

[illegible]

```

GNU nano 4.8                                script.py
from web3 import Web3
hash = Web3.sha3(text="increaseCounter(uint256)")
print(hash.hex())

[11/30/25] seed@VM:~/.../contract$ nano script.py
[11/30/25] seed@VM:~/.../contract$ python3 script.py
/usr/lib/python3/dist-packages/requests/__init__.py:89: RequestsDep
endencyWarning: urllib3 (2.2.3) or chardet (3.0.4) doesn't match a
supported version!
  warnings.warn("urllib3 ({}), or chardet ({}), doesn't match a suppo
rted "
0x9e80c074933ed616d86c79d926a1dfd97afbe0ba20c42d692de450d20df14d77

```

As we can see 0x9e80c matches with the data field from EtherView

Now we try with low level instructions using
0x9e80c0740005 as
input

The screenshot displays a web interface for interacting with a smart contract. The left sidebar shows the contract details for 'HELLO AT 0X417...3EA27 (BLC)' with a balance of 0 ETH. It includes buttons for 'decreaseCounter', 'increaseCounter', 'increaseCount...', 'sendMessage', 'getCounter', 'getResult', 'owner', and 'sayHello'. The 'increaseCounter' button is highlighted with the value '5'. Below these is a 'Low level interactions' section with a 'CALLDATA' input field containing '1005' and a 'Transact' button. The right sidebar shows the 'Explain contract' view with a search bar and a list of logs. A 'call' event is highlighted, showing a call to 'Hello.getCounter()' with data '0x8ad...a066e'. The 'decoded output' shows '0': 'uint256: 10'.

From this, we were able to successfully send a transaction to increase the counter by 5.

Task 2.d: Emit events

```

Logs
[
    {
        "from":
            "0x41724AcaecCeeE8cEC54e0500f11bD72d43EA27",
        "topic":
            "0x8ff41ca1815f7eac6985bf4ba03a8cf43ca7fcd5354be9797190a614f56b106",
        "event": "MyMessage",
        "args": {
            "0":
                "0xF5406927254d2daF77c28A61191e3ff1f72400fe9",
                "1": "10"
            }
        }
    ]

raw logs
[
    {
        "_type": "log",
        "address":
            "0x41724AcaecCeeE8cEC54e0500f11bD72d43EA27",
        "blockHash":
            "0x6840179f164e32684250675c5068a099fcd99d9d958d21elf779cb55ell6676f",
        "blockNumber": 1123,
        "data":
            "0x0000000000000000000000000000000000000000000000000000000000000000",
        "index": 0,
        "topics": [
            "0x8ff41ca1815f7eac6985bf4ba03a8cf43ca7fcd5354be9797190a614f56b106",
            "0x0000000000000000000000000000000000000000000000000000000000000000",
            "0x1191e3ff1f72400fe9"
        ],
        "transactionHash":
            "0x064f2fe83f8b9b12d4b79aad1144e5cc991e5ff6a9d3ce84379bbac2e607a0bf",
        "transactionIndex": 0
    }
]

```


Transaction	Receipt
hash	0x064f2f83f8b9b12d4b79aad1144e5cc991e5ff6a9d3ce84379bbac2e607a0bf
type	2
accessList	
blockHash	0x6840179ff64e32684250675c5068a099fcd99d9d058d21e1f779cb55e116676f
blockNumber	1123
transactionIndex	0
confirmations	4
from	0xF5406927254d2da7F7c28A61191e3F1f2400fe9
gasPrice	1000000007
maxPriorityFeePerGas	1000000007
maxFeePerGas	1000000007
gasLimit	24965
to	0x41724AcaecCeeEDcEC54e0500f11b1D72d43EA27
value	0
nonce	25
data	0xe5aed28a
r	0xaa6fd67d4a4f2d3637b41bee6e3887a82b3a94e691ab9b14fd3a15acfa6a2c2f
s	0x65aa27f252f5e3ca6119796492b117c6049e6e9c82d2f60e525d237c74cef3
v	1
creates	null
chainid	1337
wait	(omitted)


From this we can see the message logs and raw logs that remix produced.


Task 3: Send Fund to Contract

Task 3.a: Send fund directly to a contract address


Use EtherFaucet.sol, recompile and create a new contract


Account 30eth 


10 network addresses 





\$0.00


+\$0.00 (+0.00%) [Discover](#) 

Buy

Swap


Send

Receive




Jump into Monad


The Monad network is now live on MetaMask



TokensDeFiNFTsActivity

Seed Emulator 

Nov 30, 2025

Sent

L Confirmed

-5 ETH
-\$14,285.20

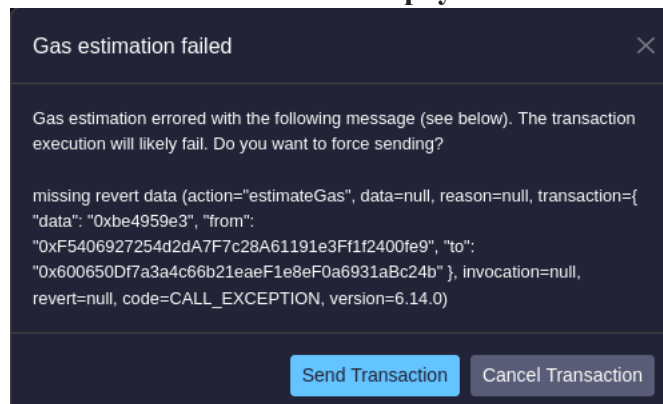
When removing receive(), we can see that Remix will say that it cannot receive any funds and cause an error. Since the smart contract can't receive outside funds.

Task 3.b: Send fund to a payable function


We can see that donationCount and amount have increased

donationCount	amount
0: uint256: 1	0: uint256: 6000000000000000000

Task 3.c: Send fund to a non-payable function



We get this error, but if we continue to still send the transaction, we can see that the transaction failed

 **Failed transaction**
Transaction 31 failed! Transaction dropped or replaced

```
transact to EtherFaucet.donateEtherWrong pending ...

[block:1313 txIndex:-] from: 0xF54...00fe9
to: EtherFaucet.donateEtherWrong()
0x600...Bc24b
value: 10000000000000000000 wei
data: 0xbe4...959e3 logs: 0
hash: 0x49f...e72f4

status      0 Transaction mined but execution failed
transaction hash  0xde871ca95b8b1d8eb52ac270c7b2b6b3d18e0af5560b7f
d9e85400756a239b4
block hash     0x49f51f8969c3c73795fc63dde1e08821a093bb7de055864
a8d37aldcce2e72f4
block number   1313
from           0xF5406927254d2dA7F7c28A61191e3Ff1f2400fe9
to            EtherFaucet.donateEtherWrong()
              0x600650Df7a3a4c66b21eaeF1e8eF0a6931aBc24b
transaction cost  21231 gas
decoded input   {}
decoded output  -
logs           []
raw logs       []
value          10000000000000000000 wei
```

amount
0: uint256: 6000000000000000000

And that the transaction simply just got dropped and ignored.

Task 3.d: Send fund to a non-existing function

GNU nano 4.8

script.py

```
from web3 import Web3
hash = Web3.sha3(text="foo()")
print(hash.hex())
```

```
[11/30/25] seed@VM:~/.../contract$ nano script.py
[11/30/25] seed@VM:~/.../contract$ python3 script.py
/usr/lib/python3/dist-packages/requests/__init__.py:89: RequestsDep
endencyWarning: urllib3 (2.2.3) or chardet (3.0.4) doesn't match a
supported version!
  warnings.warn("urllib3 ({}), or chardet ({}), doesn't match a suppo
rted "
0xc2985578b8f3b75f7dc66a767be2a4ef7d7c2224896a1c86e92ccf30bae678b7
```

Take the first 8 and rest with 0s

0xc2985578b000 as input

We can see that the transaction succeeded as it went to fallback



Confirmed transaction
Transaction 33 confirmed!

✓

[block:1366 txIndex:-] from: 0xF54...00fe9
to: EtherFaucet.(fallback) 0x600...Bc24b
value: 10000000000000000000 wei
data: 0x298...00000 logs: 0
hash: 0xf29...7943b

status 1 Transaction mined and execution succeed

transaction hash 0xf0b8e465d92c1a243012517042032dccc59c3155af0d422fa3483c36cb929923

block hash 0xf293dd47fd8017b2a59f730f2e0b649e88d4dd6c670ef99eeb3c164b1697943b

block number 1366

from 0xF5406927254d2dA7F7c28A61191e3Ff1f2400fe9

to EtherFaucet.(fallback)
0x600650df7a3a4c66b21eaeF1e8eF0a6931a8c24b

transaction cost 31442 gas

decoded input -

decoded output -

logs []

raw logs []

value 10000000000000000000 wei



Debug

amount

0: uint256: 7000000000000000000

Task 4: Send Fund from Contract

First with getEtherViaCall

block hash	0xef9253a9a6e1530566c173d3b6ff56628921ca11d50c3d2a6f5fd3d5ae6c2830 
block number	1398 
from	0xF5406927254d2dA7F7c28A61191e3Ff1f2400fe9 
to	EtherFaucet.sendEtherViaCall(uint256) 0x600650Df7a3a4c66b21eaeF1e8eF0a6931aBc24b 
transaction cost	35849 gas 
decoded input	{ "uint256 _amount": "10000000000000000" }
decoded output	- 
logs	[{ "from": "0x600650Df7a3a4c66b21eaeF1e8eF0a6931aBc24b", "topic": "0x4a9688bc300302f47e5f621829c7f763f79d71f6341040009edf2b28f4ad5ba9", "event": "MyMessage", "args": { "0": "0xF5406927254d2dA7F7c28A61191e3Ff1f2400fe9", "1": "sendEtherViaCall: Fund sent to!", "2": "10000000000000000" } }]
raw logs	[{ "_type": "log", "address": "0x600650Df7a3a4c66b21eaeF1e8eF0a6931aBc24b", "blockHash": "0xef9253a9a6e1530566c173d3b6ff56628921ca11d50c3d2a6f5fd3d5ae6c2830". }]

Where the balance went from 23 ETH to 24 ETH and the balance on the contract went from 7 ETH to 6 ETH

Next with getEtherViaSend

block hash	0x5caf62e52505f5cd2742ed45f36867d8fa6232174e4075454ad52312d8768f4e 🔗
block number	1412 🔗
from	0xF5406927254d2dA7F7c28A61191e3Ff1f2400fe9 🔗
to	EtherFaucet.sendEtherViaSend(uint256) 0x600650Df7a3a4c66b21eaeF1e8eF0a6931aBc24b 🔗
transaction cost	35707 gas 🔗
decoded input	{ "uint256 _amount": "1000000000000000000" }
decoded output	- 🔗
logs	[{ "from": "0x600650Df7a3a4c66b21eaeF1e8eF0a6931aBc24b", "topic": "0x4a9688bc300302f47e5f621829c7f763f79d71f6341040009edf2b28f4ad5ba9", "event": "MyMessage", "args": { "0": "0xF5406927254d2dA7F7c28A61191e3Ff1f2400fe9", "1": "sendEtherViaSend: Fund sent to!", "2": "1000000000000000000" } }]
raw logs	[{ "_type": "log", "address": "0x600650Df7a3a4c66b21eaeF1e8eF0a6931aBc24b", "blockHash": "0x5caf62e52505f5cd2742ed45f36867d8fa6232174e4075"

Similarly, the balance went from 24 ETH to 25 ETH and the balance on the contract went from 6 ETH to 5 ETH

Finally getEtherViaTransfer

block hash	0x88f8c6419bb92ef6a55bf6063159444ea4b68415f8ce233aadb60caa89685515 🔗
block number	1424 🔗
from	0xF5406927254d2dA7F7c28A61191e3Ff1f2400fe9 🔗
to	EtherFaucet.sendEtherViaTransfer(uint256) 0x600650Df7a3a4c66b21eaeF1e8eF0a6931aBc24b 🔗
transaction cost	36037 gas 🔗
decoded input	{ "uint256 _amount": "1000000000000000000" }
decoded output	- 🔗
logs	[{ "from": "0x600650Df7a3a4c66b21eaeF1e8eF0a6931aBc24b", "topic": "0x4a9688bc300302f47e5f621829c7f763f79d71f6341040009edf2b28f4ad5ba9", "event": "MyMessage", "args": { "0": "0xF5406927254d2dA7F7c28A61191e3Ff1f2400fe9", "1": "sendEtherViaTransfer: Fund sent to!", "2": "1000000000000000000" } }]
raw logs	[{ "_type": "log", "address": "0x600650Df7a3a4c66b21eaeF1e8eF0a6931aBc24b", "blockHash": "0x00f0c4110b007a46c55ebfc063150444a~4b60415f8ce233"

Similarly, the balance went from 25 ETH to 26 ETH and the balance on the contract went from 5 ETH to 4 ETH

Task 5: Invoke Another Contract

Use Caller.sol, recompile and create a new contract

First with invokeHello

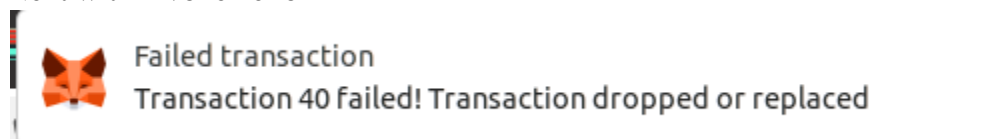
We can see that the transaction succeeded and we were able to receive this output

```
decoded output  -  ⓘ

logs            [
                  {
                    "from":
                    "0x76946d45B246128E5e87c5DF4F1D171E39073B26",
                    "topic":
                    "0x20326840ae505a1d6ab12569149f08fca01b7b95676eb4
                    8007146bafd9bb4aeb",
                    "event": "ReturnValue",
                    "args": {
                      "0":
                      "0xF5406927254d2dA7F7c28A61191e3Ff1f2400fe9",
                      "1":
                      "10000000000000000000"
                    }
                  }
                ] ⓘ

raw logs        [
                  {
                    "_type": "log",
                    "address":
                    "0x76946d45B246128E5e87c5DF4F1D171E39073B26",
                    "blockHash":
                    "0xb5e4ca0771be52b6c134a1e23b60f837c034340529a262
                    a2b298066fadb14c59",
                    "blockNumber": 1514,
                    "data":
                    "0x0000000000000000000000000000000000000000000000000000000000000000
                    000de0b6b3a7640000",
                    "index": 0,
                    "topics": [
                      "0x20326840ae505a1d6ab12569149f08fca01b7b95676eb4
                      8007146bafd9bb4aeb",
                      "0x00000000000000000000000000000000f5406927254d2da7f7c28a
                      61191e3ff1f2400fe9"
                    ],
                    "transactionHash":
                    "0x4833a4936b1d535ab3e68fa2bbe301dfcec950daa49fa
                    b8b54995d32875e464",
                    "transactionIndex": 0
                  }
                ]
```

Next with invokeHello2



x

[block:1529 txIndex:-] from: 0xF54...00fe9
to: Caller.invokeHello2(address,uint256)
0x769...73B26
value: 0 wei data: 0x7da...40000 logs: 0
hash: 0xf8b...b72d7

Debug ^

status

0 Transaction mined but execution failed

transaction hash

0xf67687bfb71d90e3529fde14ad22e0b00d62bd25907899b
ale2caba832097bf0 [🔗](#)

block hash

0xf8b98bf69ad31e449e1edf1ac5852fab6e9bae6241fa36f
765c085dce1b72d7 [🔗](#)

block number

1529 [🔗](#)

from

0xF5406927254d2dA7F7c28A61191e3Ff1f2400fe9 [🔗](#)

to

Caller.invokeHello2(address,uint256)
0x76946d45B246128E5e87c5DF4F1D171E39073B26 [🔗](#)

transaction cost

33627 gas [🔗](#)

decoded input

{
 "address addr":
 "0xC31601AC33A3db21505214eAcAA6BA5fB0b8cd95",
 "uint256 _val": "1000000000000000000"
}

[🔗](#)

decoded output

- [🔗](#)

logs

[] [🔗](#)

raw logs

[] [🔗](#)

We can see that it failed with invokeHello2