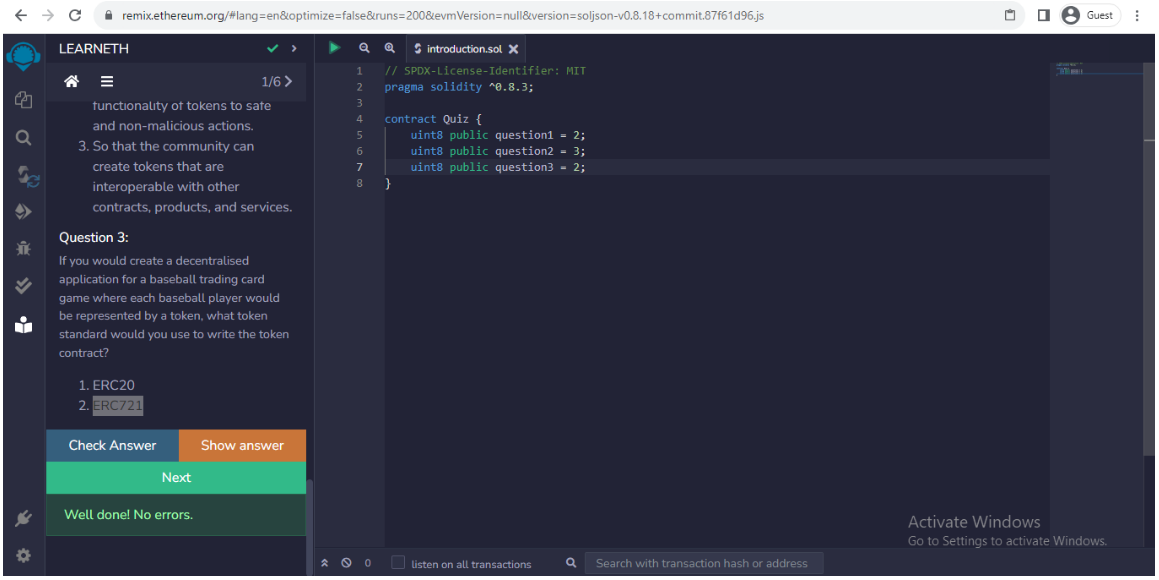
Name: Om Gaydhane **Class/Roll No :** D16AD/15

BLOCKCHAIN HONORS DEGREE

Tutorial

1. Solidity ERC20 Token Course

Introduction


remix.ethereum.org/#lang=en&optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.18+commit.87f61d96.js
Guest
LEARNETH
1/6
introduction.sol
Check Answer
Show answer
Next
listen on all transactions
Search with transaction hash or address
Activate Windows
Go to Settings to activate Windows.

functionality of tokens to safe and non-malicious actions.

3. So that the community can create tokens that are interoperable with other contracts, products, and services.

Question 3:

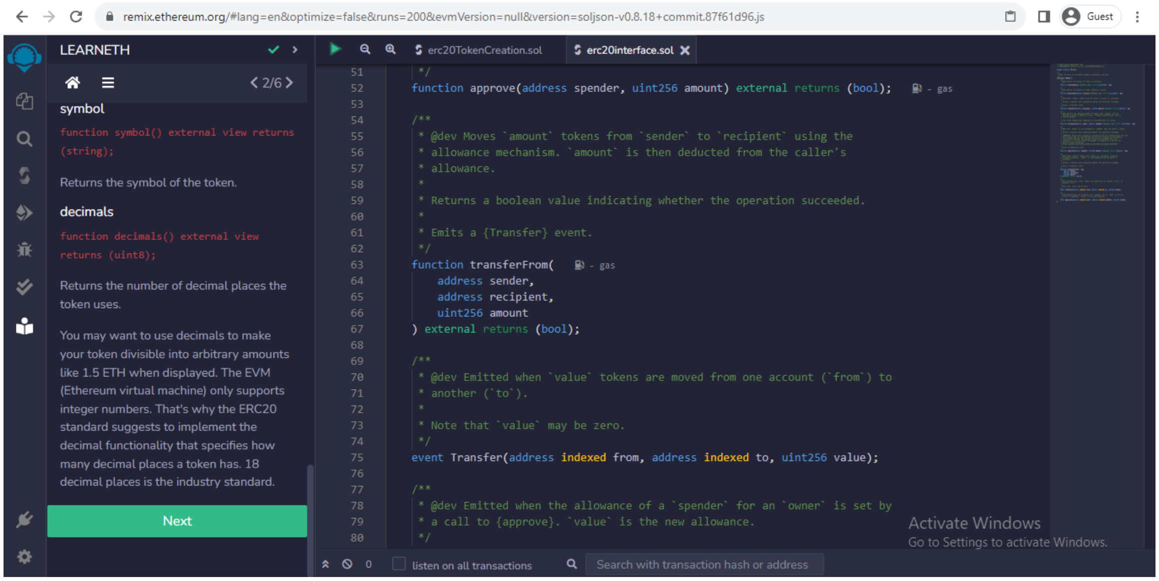
If you would create a decentralised application for a baseball trading card game where each baseball player would be represented by a token, what token standard would you use to write the token contract?

1. ERC20
2. ERC721

Well done! No errors.

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.3;

contract Quiz {
    uint8 public question1 = 2;
    uint8 public question2 = 3;
    uint8 public question3 = 2;
}
```

Interface


remix.ethereum.org/#lang=en&optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.18+commit.87f61d96.js
Guest
LEARNETH
2/6
erc20TokenCreation.sol
erc20interface.sol
Next
listen on all transactions
Search with transaction hash or address
Activate Windows
Go to Settings to activate Windows.

symbol

```
function symbol() external view returns (string);
```

Returns the symbol of the token.

decimals

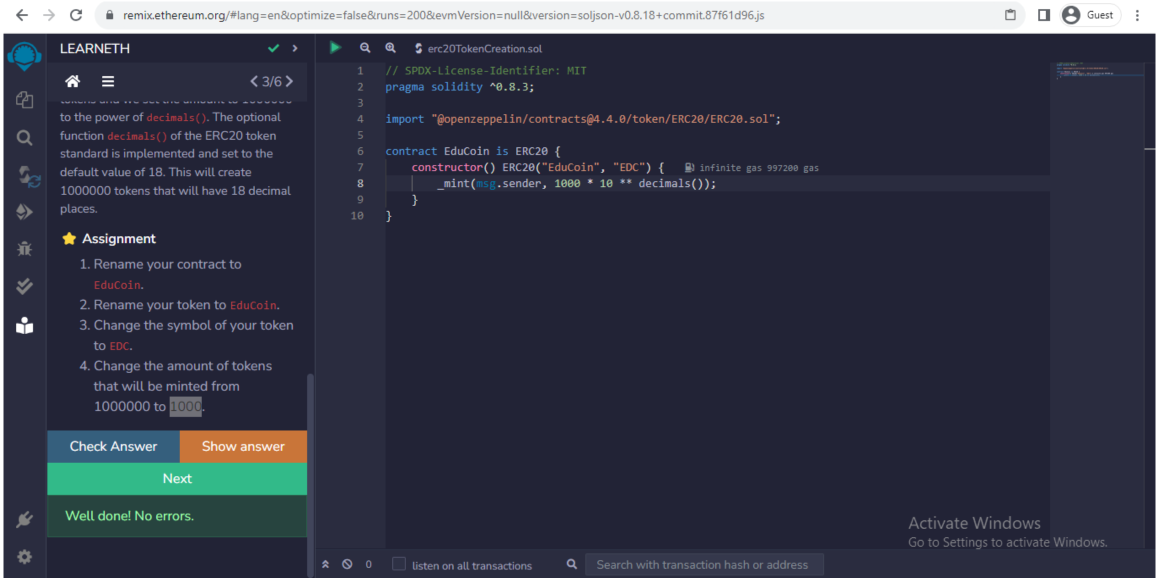
```
function decimals() external view returns (uint8);
```

Returns the number of decimal places the token uses.

You may want to use decimals to make your token divisible into arbitrary amounts like 1.5 ETH when displayed. The EVM (Ethereum virtual machine) only supports integer numbers. That's why the ERC20 standard suggests to implement the decimal functionality that specifies how many decimal places a token has. 18 decimal places is the industry standard.

```
     */
    function approve(address spender, uint256 amount) external returns (bool);   - gas

    /**
     * @dev Moves `amount` tokens from `sender` to `recipient` using the
     * allowance mechanism. `amount` is then deducted from the caller's
     * allowance.
     *
     * Returns a boolean value indicating whether the operation succeeded.
     *
     * Emits a {Transfer} event.
     */
    function transferFrom(   - gas
        address sender,
        address recipient,
        uint256 amount
    ) external returns (bool);

    /**
     * @dev Emitted when `value` tokens are moved from one account (`from`) to
     * another (`to`).
     *
     * Note that `value` may be zero.
     */
    event Transfer(address indexed from, address indexed to, uint256 value);

    /**
     * @dev Emitted when the allowance of a `spender` for an `owner` is set by
     * a call to {approve}. `value` is the new allowance.
     */
```

Token Creation


remix.ethereum.org/#lang=en&optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.18+commit.87f61d96.js
Guest
LEARNETH
3/6
erc20TokenCreation.sol
Check Answer
Show answer
Next
Activate Windows
Go to Settings to activate Windows.
listen on all transactions
Search with transaction hash or address

to the power of decimals(). The optional function decimals() of the ERC20 token standard is implemented and set to the default value of 18. This will create 1000000 tokens that will have 18 decimal places.

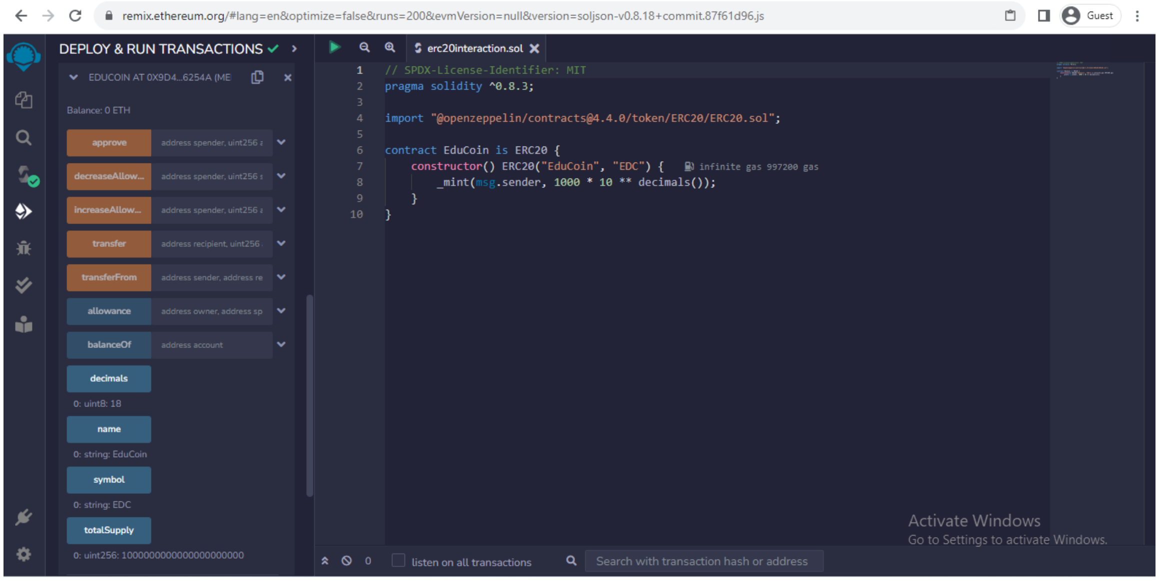
Assignment

1. Rename your contract to EduCoin.
2. Rename your token to EduCoin.
3. Change the symbol of your token to EDC.
4. Change the amount of tokens that will be minted from 1000000 to 1000.

Well done! No errors.

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.3;

import "@openzeppelin/contracts@4.4.0/token/ERC20/ERC20.sol";

contract EduCoin is ERC20 {
    constructor() ERC20("EduCoin", "EDC") {    infinite gas 997200 gas
        _mint(msg.sender, 1000 * 10 ** decimals());
    }
}
```

Interaction


remix.ethereum.org/#lang=en&optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.18+commit.87f61d96.js
Guest
DEPLOY & RUN TRANSACTIONS
EDUCOIN AT 0X9D4...6254A (ME
erc20interaction.sol
approve
address spender, uint256 a
decreaseAllow...
address spender, uint256 s
increaseAllow...
address spender, uint256 a
transfer
address recipient, uint256
transferFrom
address sender, address re
allowance
address owner, address sp
balanceOf
address account
decimals
name
symbol
totalSupply
Activate Windows
Go to Settings to activate Windows.
listen on all transactions
Search with transaction hash or address

Balance: 0 ETH

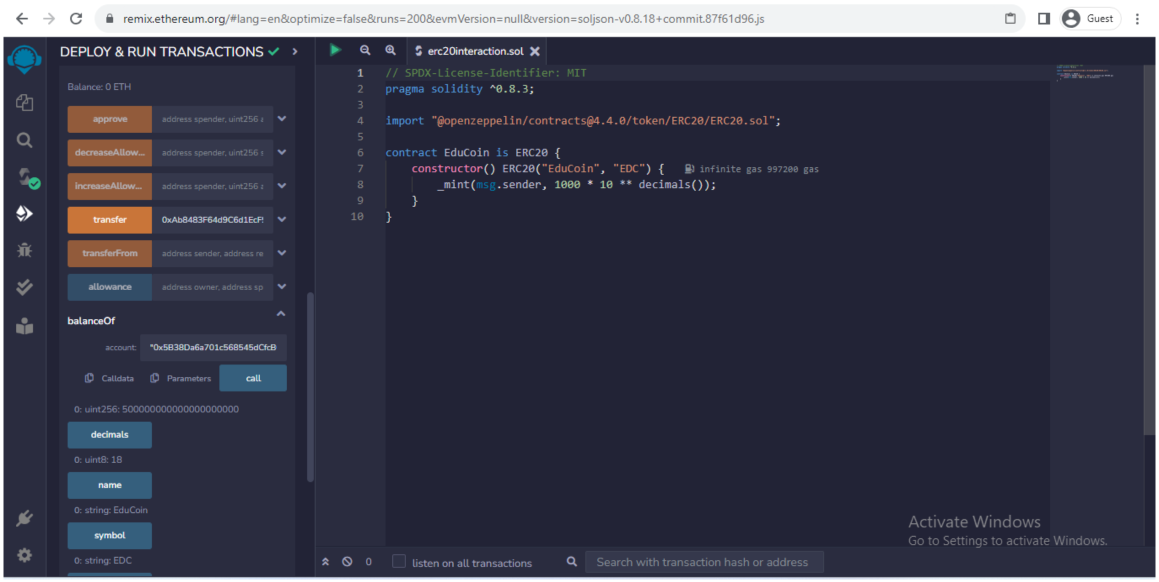
decimals: 0: uint8: 18

name: 0: string: EduCoin

symbol: 0: string: EDC

totalSupply: 0: uint256: 1000000000000000000000

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.3;

import "@openzeppelin/contracts@4.4.0/token/ERC20/ERC20.sol";

contract EduCoin is ERC20 {
    constructor() ERC20("EduCoin", "EDC") {    infinite gas 997200 gas
        _mint(msg.sender, 1000 * 10 ** decimals());
    }
}
```



remix.ethereum.org/#lang=en&optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.18+commit.87f61d96.js

DEPLOY & RUN TRANSACTIONS

Balance: 0 ETH

approve
address spender, uint256 a
decreaseAllow...
address spender, uint256 s
increaseAllow...
address spender, uint256 a

transfer: 0xAb8483F64d9C6d1EcF!

transferFrom
address sender, address re
allowance
address owner, address sp

balanceOf

account: "0x5B38Da6a701c568545dCfcB

0: uint256: 500000000000000000000

decimals

0: uint8: 18

name

0: string: EduCoin

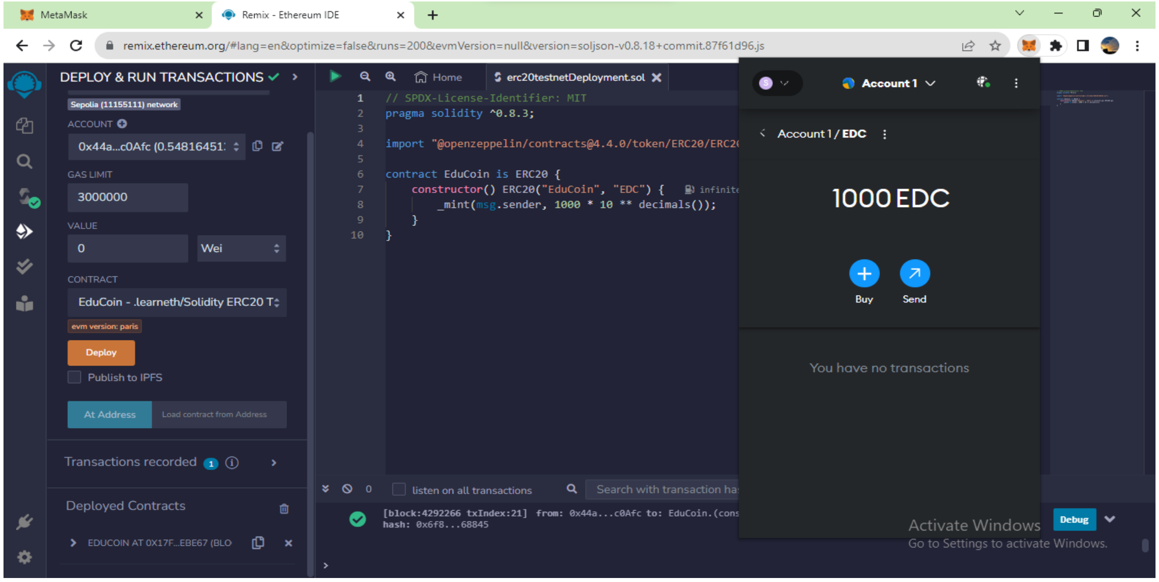
symbol

0: string: EDC

erc20interaction.sol

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.3;

import "@openzeppelin/contracts@4.4.0/token/ERC20/ERC20.sol";

contract EduCoin is ERC20 {
    constructor() ERC20("EduCoin", "EDC") {    infinite gas 997200 gas
        _mint(msg.sender, 1000 * 10 ** decimals());
    }
}
```

0
listen on all transactions
Search with transaction hash or address
Activate Windows
Go to Settings to activate Windows.



MetaMask
Remix - Ethereum IDE

remix.ethereum.org/#lang=en&optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.18+commit.87f61d96.js

DEPLOY & RUN TRANSACTIONS

Sepolia (11155111) network

ACCOUNT: 0x44a...c0Afc (0.54816451:

GAS LIMIT: 3000000

VALUE: 0 Wei

CONTRACT: EduCoin - .learneth/Solidity ERC20 T

evm version: paris

Deploy
Publish to IPFS
At Address
Load contract from Address

Transactions recorded 1

Deployed Contracts

EDUCOIN AT 0X17F...EBE67 (BLO

erc20testnetDeployment.sol

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.3;

import "@openzeppelin/contracts@4.4.0/token/ERC20/ERC2
contract EduCoin is ERC20 {
    constructor() ERC20("EduCoin", "EDC") {    infinite
        _mint(msg.sender, 1000 * 10 ** decimals());
    }
}
```

```
[block:4292266 txIndex:21] from: 0x44a...c0Afc to: EduCoin.(cons
hash: 0x6f8...68845
```

Account 1

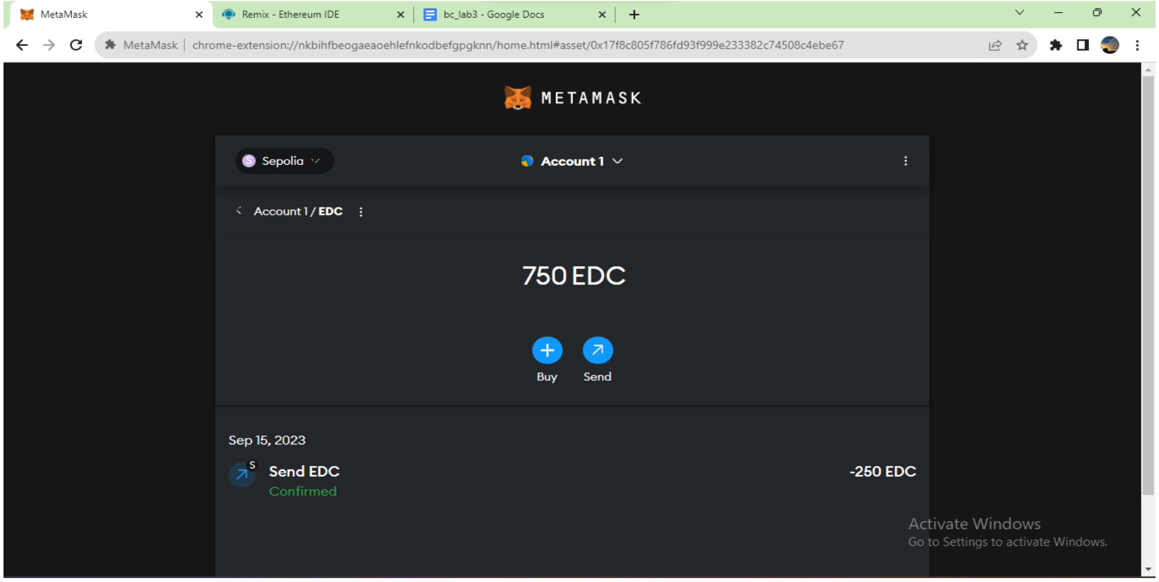
Account 1 / EDC

1000 EDC

You have no transactions

Debug
listen on all transactions
Search with transaction ha
Activate Windows
Go to Settings to activate Windows.

Testnet Deployment



MetaMask | chrome-extension://nkbihfbeogaeaoehlefnkodbefgpgknn/home.html#asset/0x17f8c805f786fd93f999e233382c74508c4ebe67

Sepolia

Account 1

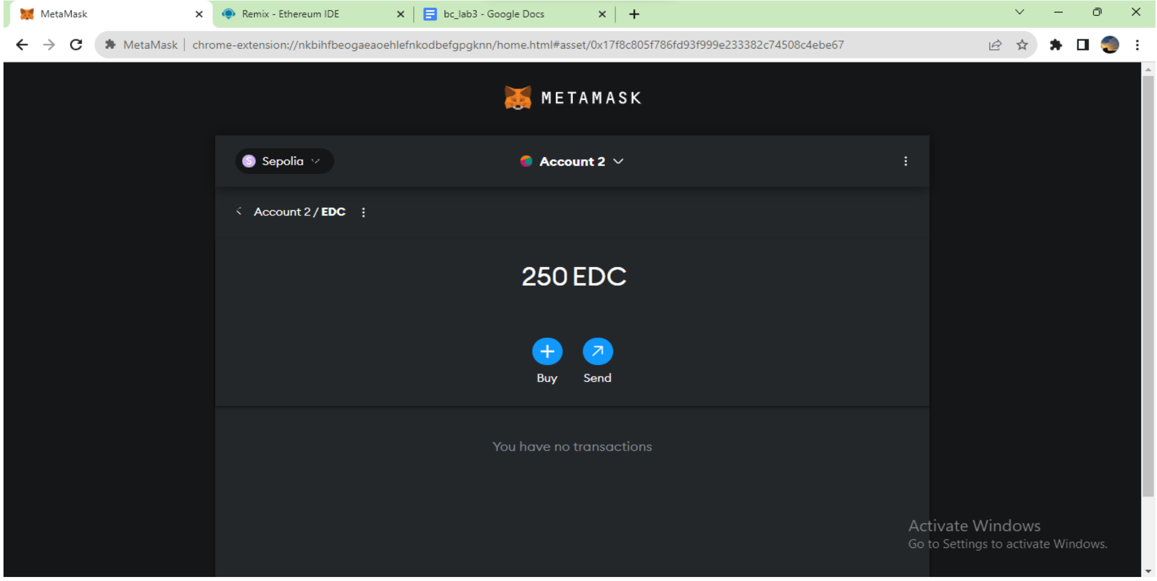
Account 1 / EDC

750 EDC

Sep 15, 2023

Send EDC -250 EDC

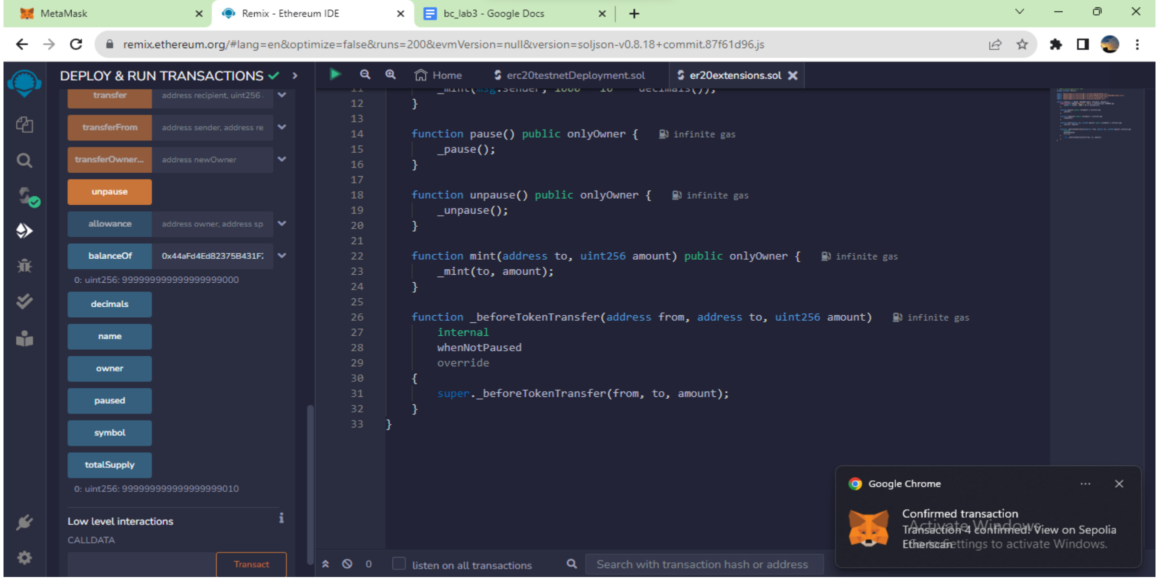
Confirmed


MetaMask
Remix - Ethereum IDE
bc_lab3 - Google Docs
MetaMask | chrome-extension://nkbihfbeogaeaoehlefnkodbefgpgknn/home.html#asset/0x17f8c805f786fd93f999e233382c74508c4ebe67
METAMASK
Sepolia
Account 2
Account 2 / EDC
Buy
Send

250 EDC

You have no transactions

Extensions


MetaMask
Remix - Ethereum IDE
bc_lab3 - Google Docs
remix.ethereum.org/#lang=en&optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.18+commit.87f61d96.js
DEPLOY & RUN TRANSACTIONS
Home
erc20testnetDeployment.sol
er20extensions.sol
transfer
address recipient, uint256
transferFrom
address sender, address re
transferOwner...
address newOwner
unpause
allowance
address owner, address sp
balanceOf
0x44aFd4Ed82375B431F
decimals
name
owner
paused
symbol
totalSupply
Low level interactions
CALLDATA
Transact
listen on all transactions
Search with transaction hash or address

0: uint256: 9999999999999999000

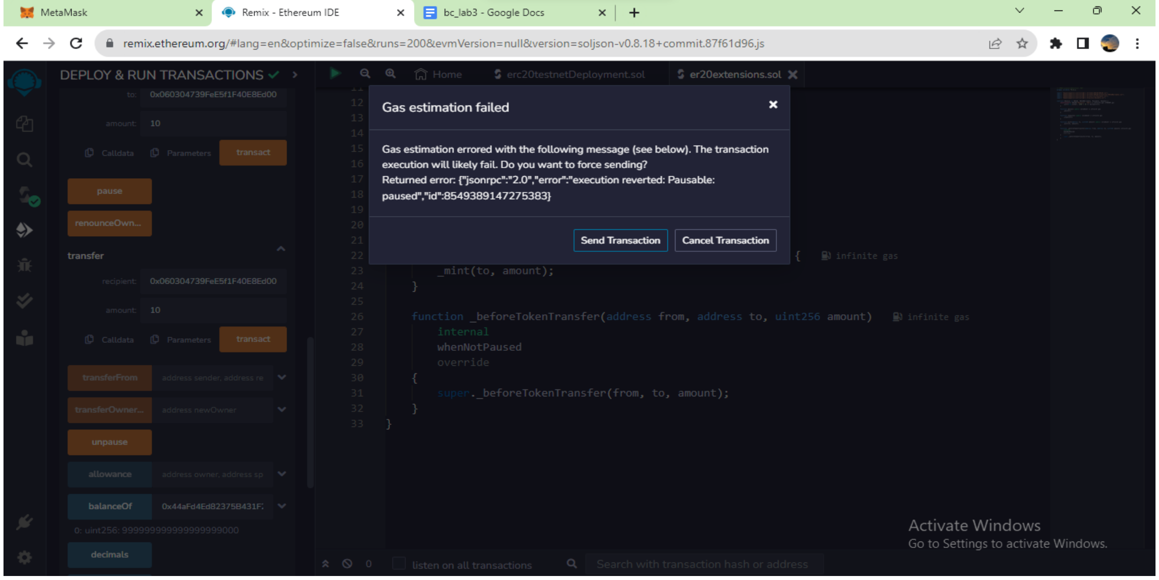
0: uint256: 9999999999999999010

```
    }

    function pause() public onlyOwner {    infinite gas
        _pause();
    }

    function unpause() public onlyOwner {    infinite gas
        _unpause();
    }

    function mint(address to, uint256 amount) public onlyOwner {    infinite gas
        _mint(to, amount);
    }

    function _beforeTokenTransfer(address from, address to, uint256 amount)    infinite gas
        internal
        whenNotPaused
        override
    {
        super._beforeTokenTransfer(from, to, amount);
    }
}
```

Google Chrome

Confirmed transaction

Transaction 4 confirmed! View on Sepolia Etherscan


MetaMask
Remix - Ethereum IDE
bc_lab3 - Google Docs
remix.ethereum.org/#lang=en&optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.18+commit.87f61d96.js
DEPLOY & RUN TRANSACTIONS
Home
erc20testnetDeployment.sol
er20extensions.sol
to:
0x060304739FeE5f1F40E8Ed00
amount:
10
Calldata
Parameters
transact
pause
renounceOwn...
transfer
recipient:
0x060304739FeE5f1F40E8Ed00
amount:
10
Calldata
Parameters
transact
transferFrom
address sender, address re
transferOwner...
address newOwner
unpause
allowance
address owner, address sp
balanceOf
0x44aFd4Ed82375B431F
decimals
listen on all transactions
Search with transaction hash or address
Send Transaction
Cancel Transaction

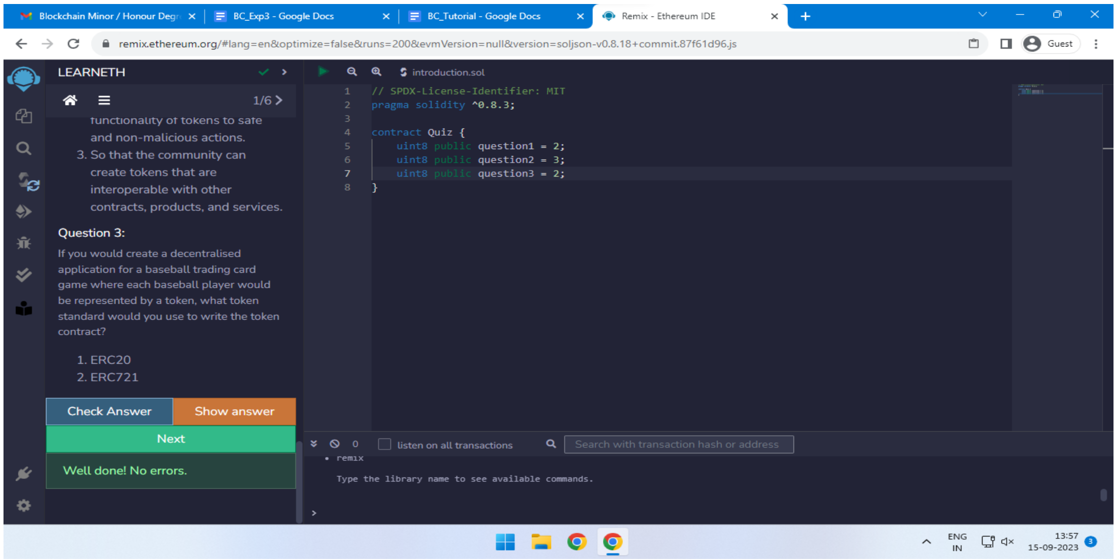
Gas estimation failed

Gas estimation errored with the following message (see below). The transaction execution will likely fail. Do you want to force sending?
Returned error: {"jsonrpc":"2.0","error":"execution reverted: Pausable: paused","id":8549389147275383}

0: uint256: 9999999999999999000

2. Solidity NFT Course :

Introduction


Blockchain Minor / Honour Degr
BC_Exp3 - Google Docs
BC_Tutorial - Google Docs
Remix - Ethereum IDE
remix.ethereum.org/#lang=en&optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.18+commit.87f61d96.js
Guest
LEARNETH
1/6
introduction.sol
ENG
IN
13:57
15-09-2023

functionality of tokens to safe and non-malicious actions.

3. So that the community can create tokens that are interoperable with other contracts, products, and services.

Question 3:

If you would create a decentralised application for a baseball trading card game where each baseball player would be represented by a token, what token standard would you use to write the token contract?

1. ERC20
2. ERC721

Check Answer | Show answer

Next

Well done! No errors.

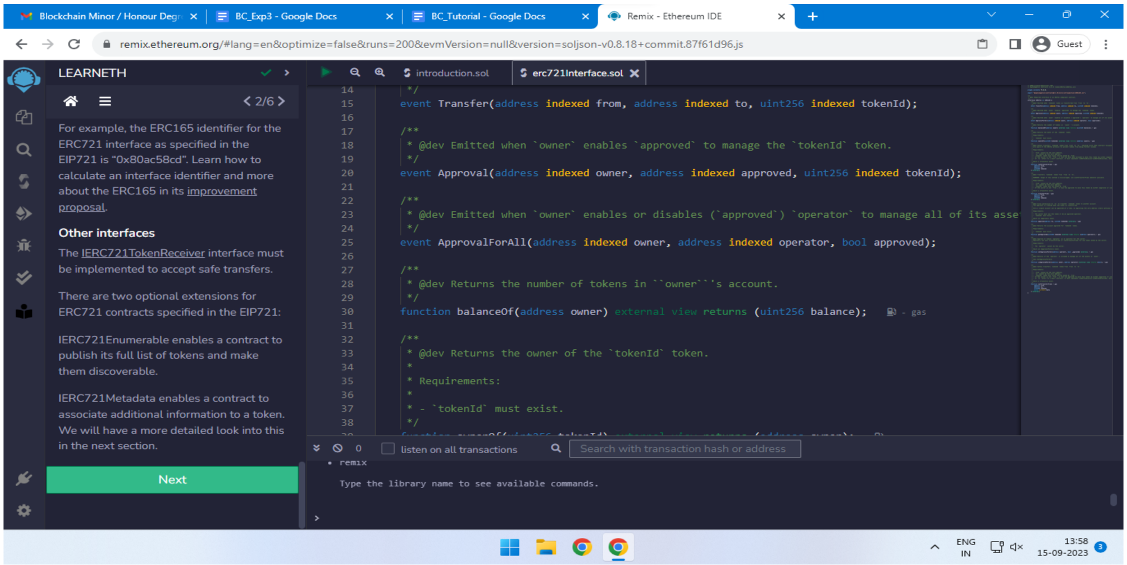
```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.3;

contract Quiz {
    uint8 public question1 = 2;
    uint8 public question2 = 3;
    uint8 public question3 = 2;
}
```

☐ listen on all transactions

Search with transaction hash or address

```
• remix

  Type the library name to see available commands.

>
```

Interface


Blockchain Minor / Honour Degr
BC_Exp3 - Google Docs
BC_Tutorial - Google Docs
Remix - Ethereum IDE
remix.ethereum.org/#lang=en&optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.18+commit.87f61d96.js
Guest
LEARNETH
2/6
introduction.sol
erc721Interface.sol
ENG
IN
13:58
15-09-2023

For example, the ERC165 identifier for the ERC721 interface as specified in the EIP721 is "0x80ac58cd". Learn how to calculate an interface identifier and more about the ERC165 in its improvement proposal.

Other interfaces

The IERC721TokenReceiver interface must be implemented to accept safe transfers.

There are two optional extensions for ERC721 contracts specified in the EIP721:

IERC721Enumerable enables a contract to publish its full list of tokens and make them discoverable.

IERC721Metadata enables a contract to associate additional information to a token. We will have a more detailed look into this in the next section.

Next

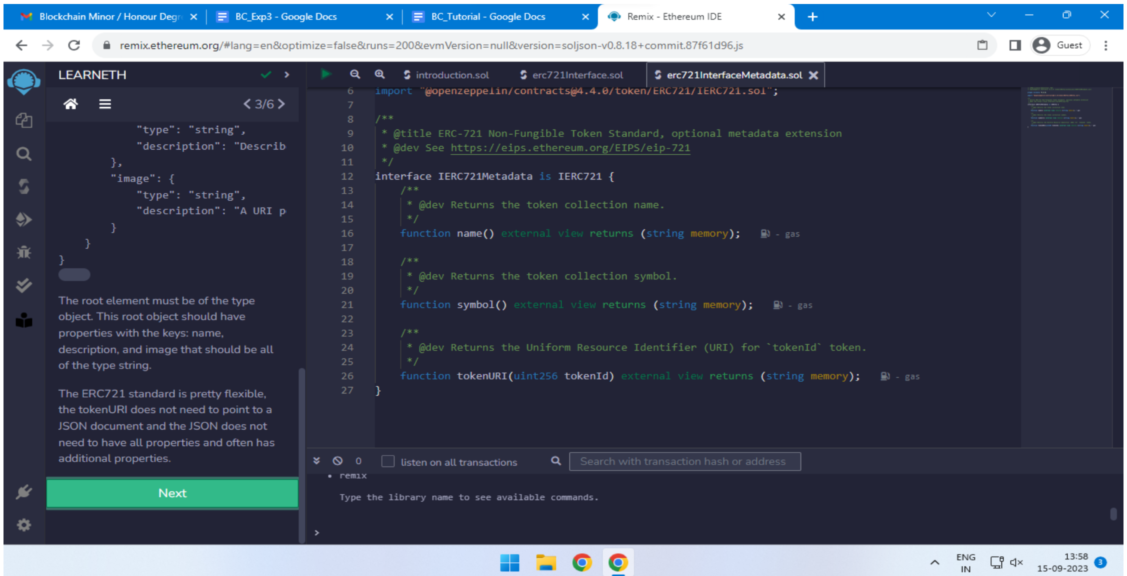
```
     */
    event Transfer(address indexed from, address indexed to, uint256 indexed tokenId);

    /**
     * @dev Emitted when `owner` enables `approved` to manage the `tokenId` token.
     */
    event Approval(address indexed owner, address indexed approved, uint256 indexed tokenId);

    /**
     * @dev Emitted when `owner` enables or disables (`approved`) `operator` to manage all of its asse
     */
    event ApprovalForAll(address indexed owner, address indexed operator, bool approved);

    /**
     * @dev Returns the number of tokens in ``owner``'s account.
     */
    function balanceOf(address owner) external view returns (uint256 balance);    - gas

    /**
     * @dev Returns the owner of the `tokenId` token.
     *
     * Requirements:
     *
     * - `tokenId` must exist.
     */
```

☐ listen on all transactions

Search with transaction hash or address

```
• remix

  Type the library name to see available commands.

>
```

Metadata Interface


Blockchain Minor / Honour Degr
BC_Exp3 - Google Docs
BC_Tutorial - Google Docs
Remix - Ethereum IDE
remix.ethereum.org/#lang=en&optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.18+commit.87f61d96.js
Guest
LEARNETH
3/6
introduction.sol
erc721Interface.sol
erc721InterfaceMetadata.sol
ENG
IN
13:58
15-09-2023

```
                "type": "string",
                "description": "Describ
            },
            "image": {
                "type": "string",
                "description": "A URI p
            }
        }
}
```

The root element must be of the type object. This root object should have properties with the keys: name, description, and image that should be all of the type string.

The ERC721 standard is pretty flexible, the tokenURI does not need to point to a JSON document and the JSON does not need to have all properties and often has additional properties.

Next

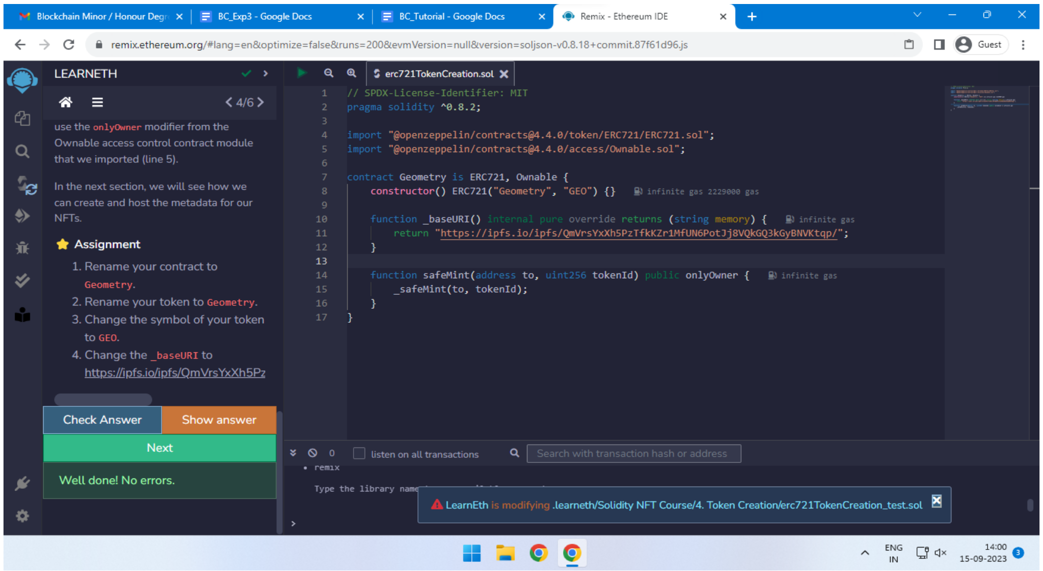
```
import "@openzeppelin/contracts@4.4.0/token/ERC721/IERC721.sol";

/**
 * @title ERC-721 Non-Fungible Token Standard, optional metadata extension
 * @dev See https://eips.ethereum.org/EIPS/eip-721
 */
interface IERC721Metadata is IERC721 {
    /**
     * @dev Returns the token collection name.
     */
    function name() external view returns (string memory);    - gas

    /**
     * @dev Returns the token collection symbol.
     */
    function symbol() external view returns (string memory);    - gas

    /**
     * @dev Returns the Uniform Resource Identifier (URI) for `tokenId` token.
     */
    function tokenURI(uint256 tokenId) external view returns (string memory);    - gas
}
```

☐ listen on all transactions

Search with transaction hash or address

```
• remix

  Type the library name to see available commands.

>
```

Token Creation



remix.ethereum.org/#lang=en&optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.18+commit.87f61d96.js

use the onlyOwner modifier from the Ownable access control contract module that we imported (line 5).

In the next section, we will see how we can create and host the metadata for our NFTs.

Assignment

1. Rename your contract to Geometry.
2. Rename your token to Geometry.
3. Change the symbol of your token to GEO.
4. Change the _baseURI to https://ipfs.io/ipfs/QmVrsYxXh5Pz

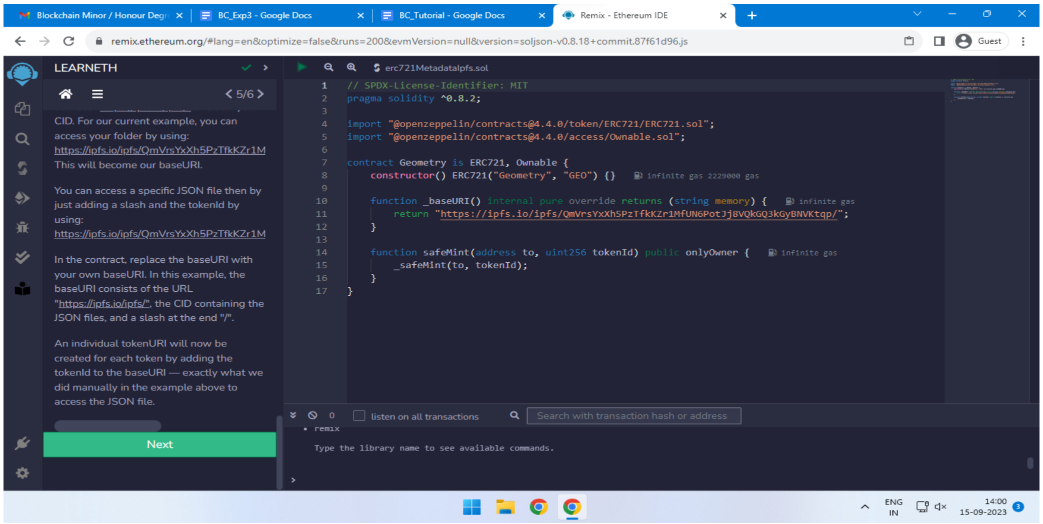
Well done! No errors.

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.2;

import "@openzeppelin/contracts@4.4.0/token/ERC721/ERC721.sol";
import "@openzeppelin/contracts@4.4.0/access/Ownable.sol";

contract Geometry is ERC721, Ownable {
    constructor() ERC721("Geometry", "GEO") {}    infinite gas 2229000 gas

    function _baseURI() internal pure override returns (string memory) {    infinite gas
        return "https://ipfs.io/ipfs/QmVrsYxXh5PzTfkKZr1MfUN6PotJj8VQkGQ3kGyBNVKtqp/";
    }

    function safeMint(address to, uint256 tokenId) public onlyOwner {    infinite gas
        _safeMint(to, tokenId);
    }
}
```

```
• remix

  Type the library name
```

LearnEth is modifying .learneth/Solidity NFT Course/4. Token Creation/erc721TokenCreation_test.sol

Metadata IFPS


Blockchain Minor / Honour Degr
BC_Exp3 - Google Docs
BC_Tutorial - Google Docs
Remix - Ethereum IDE
LEARNETH
< 5/6 >
erc721MetadataIpfs.sol
Next
listen on all transactions
Search with transaction hash or address
ENG IN
14:00 15-09-2023

remix.ethereum.org/#lang=en&optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.18+commit.87f61d96.js

CID. For our current example, you can access your folder by using: https://ipfs.io/ipfs/QmVrsYxXh5PzTfkKZr1M This will become our baseURI.

You can access a specific JSON file then by just adding a slash and the tokenId by using: https://ipfs.io/ipfs/QmVrsYxXh5PzTfkKZr1M

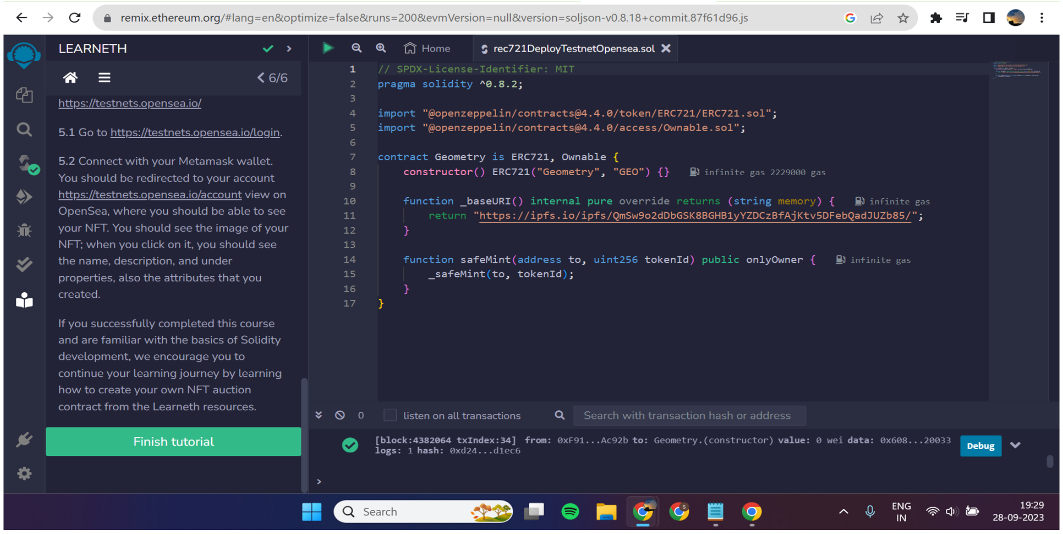
In the contract, replace the baseURI with your own baseURI. In this example, the baseURI consists of the URL "https://ipfs.io/ipfs/", the CID containing the JSON files, and a slash at the end "/".

An individual tokenURI will now be created for each token by adding the tokenId to the baseURI — exactly what we did manually in the example above to access the JSON file.

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.2;

import "@openzeppelin/contracts@4.4.0/token/ERC721/ERC721.sol";
import "@openzeppelin/contracts@4.4.0/access/Ownable.sol";

contract Geometry is ERC721, Ownable {
    constructor() ERC721("Geometry", "GEO") {}    infinite gas 2229000 gas

    function _baseURI() internal pure override returns (string memory) {    infinite gas
        return "https://ipfs.io/ipfs/QmVrsYxXh5PzTfkKZr1MfUN6PotJj8VQkGQ3kGyBNVKtqp/";
    }

    function safeMint(address to, uint256 tokenId) public onlyOwner {    infinite gas
        _safeMint(to, tokenId);
    }
}
```

```
• remix

  Type the library name to see available commands.
```

Deploy to Testnet


LEARNETH
< 6/6
Home
rec721DeployTestnetOpensea.sol
Finish tutorial
listen on all transactions
Search with transaction hash or address
Debug
Search
ENG IN
19:29 28-09-2023

remix.ethereum.org/#lang=en&optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.18+commit.87f61d96.js

https://testnets.opensea.io/

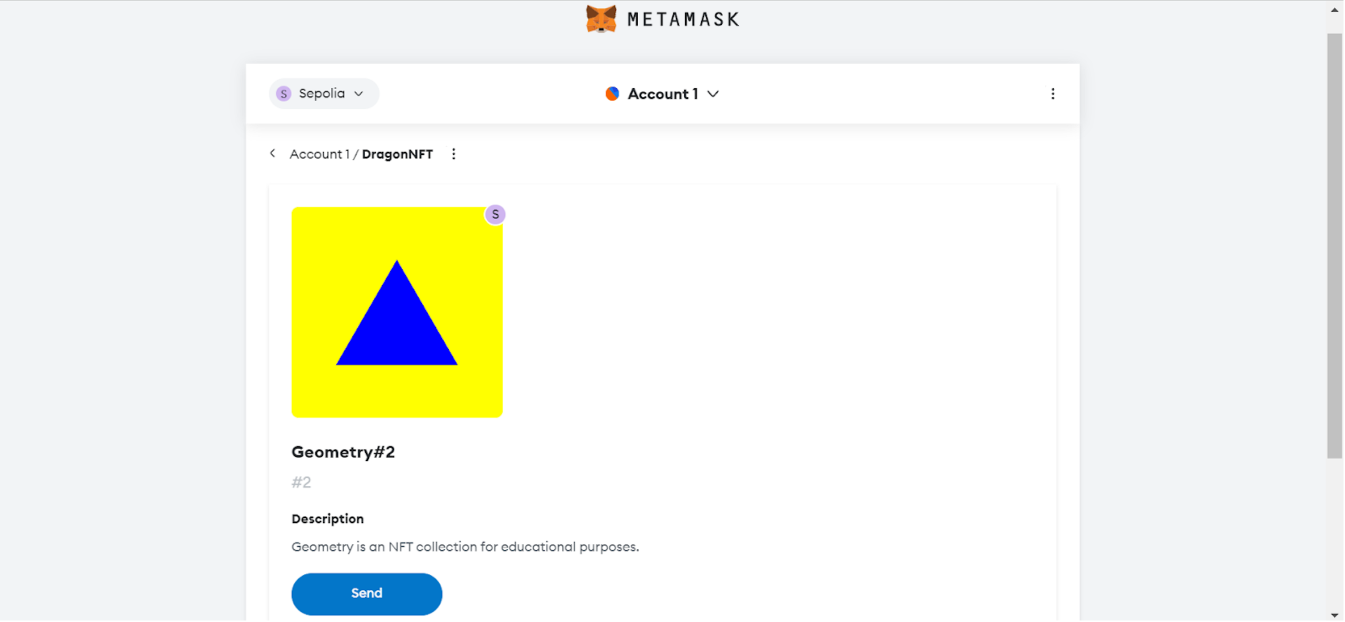
5.1 Go to https://testnets.opensea.io/login.

5.2 Connect with your Metamask wallet. You should be redirected to your account https://testnets.opensea.io/account view on OpenSea, where you should be able to see your NFT. You should see the image of your NFT; when you click on it, you should see the name, description, and under properties, also the attributes that you created.

If you successfully completed this course and are familiar with the basics of Solidity development, we encourage you to continue your learning journey by learning how to create your own NFT auction contract from the Learneth resources.

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.2;

import "@openzeppelin/contracts@4.4.0/token/ERC721/ERC721.sol";
import "@openzeppelin/contracts@4.4.0/access/Ownable.sol";

contract Geometry is ERC721, Ownable {
    constructor() ERC721("Geometry", "GEO") {}    infinite gas 2229000 gas

    function _baseURI() internal pure override returns (string memory) {    infinite gas
        return "https://ipfs.io/ipfs/QmSw9o2dDbGSK8BGHB1yYZDCzBfAjKtv5DFebQadJUZb85/";
    }

    function safeMint(address to, uint256 tokenId) public onlyOwner {    infinite gas
        _safeMint(to, tokenId);
    }
}
```

```
[block:4382064 txIndex:34] from: 0xF91...Ac92b to: Geometry.(constructor) value: 0 wei data: 0x608...20033
logs: 1 hash: 0xd24...d1ec6
```



Geometry#2

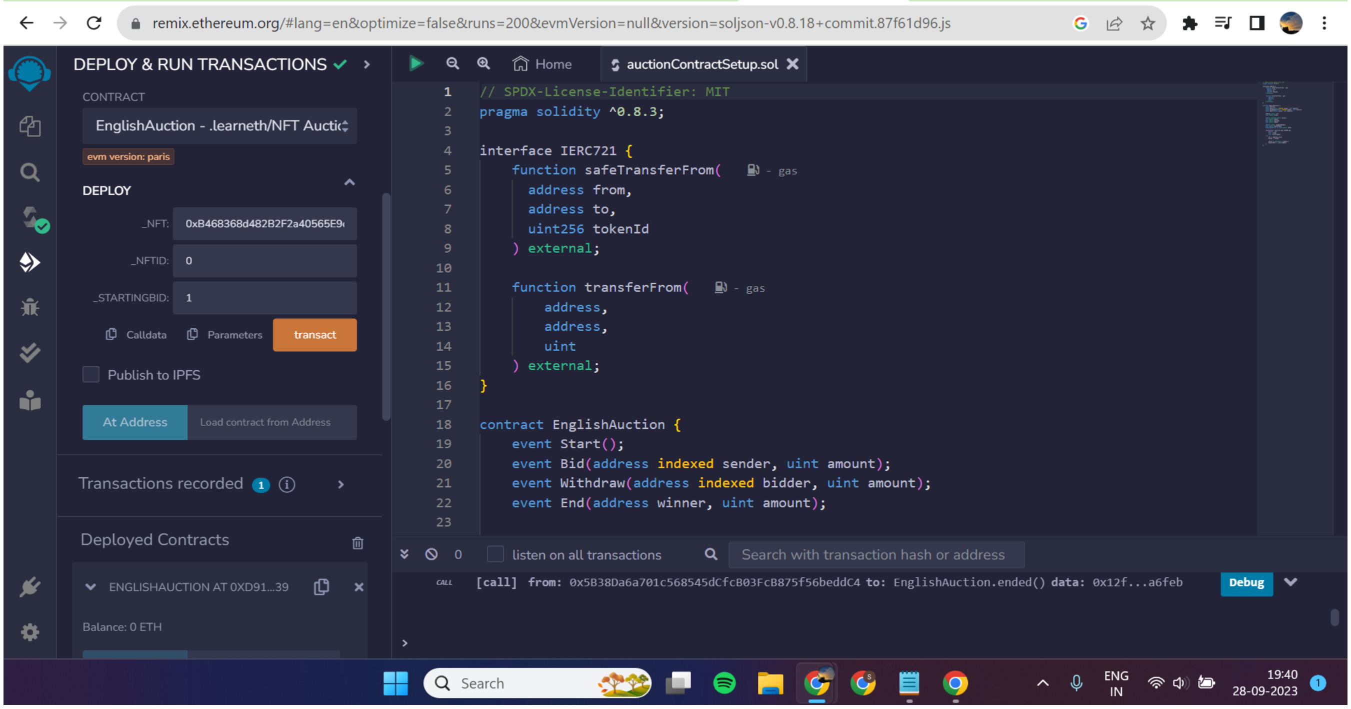
#2

Description

Geometry is an NFT collection for educational purposes.

3. NFT Auction Contract

Contract Setup



remix.ethereum.org/#lang=en&optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.18+commit.87f61d96.js

DEPLOY & RUN TRANSACTIONS

CONTRACT: EnglishAuction - .learneth/NFT Auctic

evm version: paris

DEPLOY

_NFT: 0xB468368d482B2F2a40565E9

_NFTID: 0

_STARTINGBID: 1

Calldata | Parameters | transact

□ Publish to IPFS

At Address | Load contract from Address

Transactions recorded 1

Deployed Contracts

ENGLISHAUCTION AT 0XD91...39

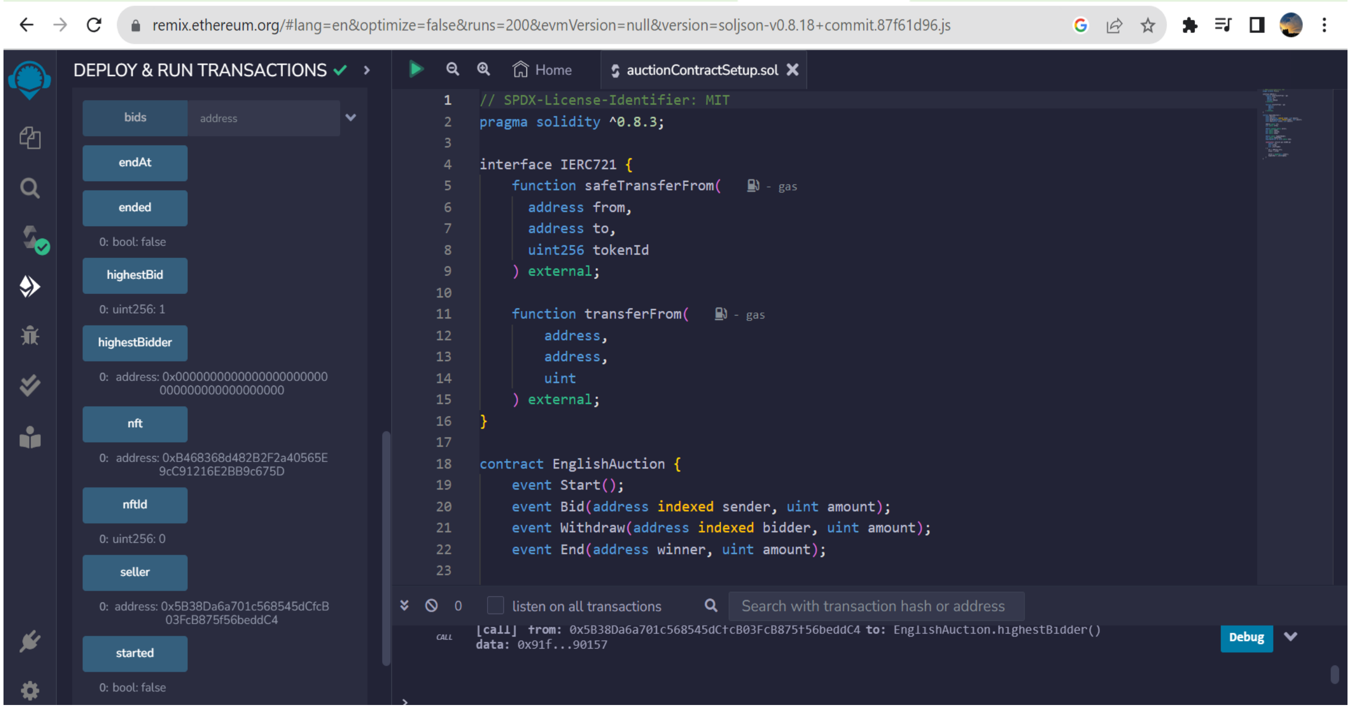
Balance: 0 ETH

auctionContractSetup.sol

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.3;

interface IERC721 {
    function safeTransferFrom(
        address from,
        address to,
        uint256 tokenId
    ) external;

    function transferFrom(
        address,
        address,
        uint
    ) external;
}

contract EnglishAuction {
    event Start();
    event Bid(address indexed sender, uint amount);
    event Withdraw(address indexed bidder, uint amount);
    event End(address winner, uint amount);

```

```
[call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: EnglishAuction.ended() data: 0x12f...a6feb
```

Home
listen on all transactions
Search with transaction hash or address
Debug
Search
ENG
IN
19:40
28-09-2023



remix.ethereum.org/#lang=en&optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.18+commit.87f61d96.js

DEPLOY & RUN TRANSACTIONS

bids: address

endAt

ended
0: bool: false

highestBid
0: uint256: 1

highestBidder
0: address: 0x0000000000000000000000000000000000000000

nft
0: address: 0xB468368d482B2F2a40565E9cC91216E2BB9c675D

nftId
0: uint256: 0

seller
0: address: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4

started
0: bool: false

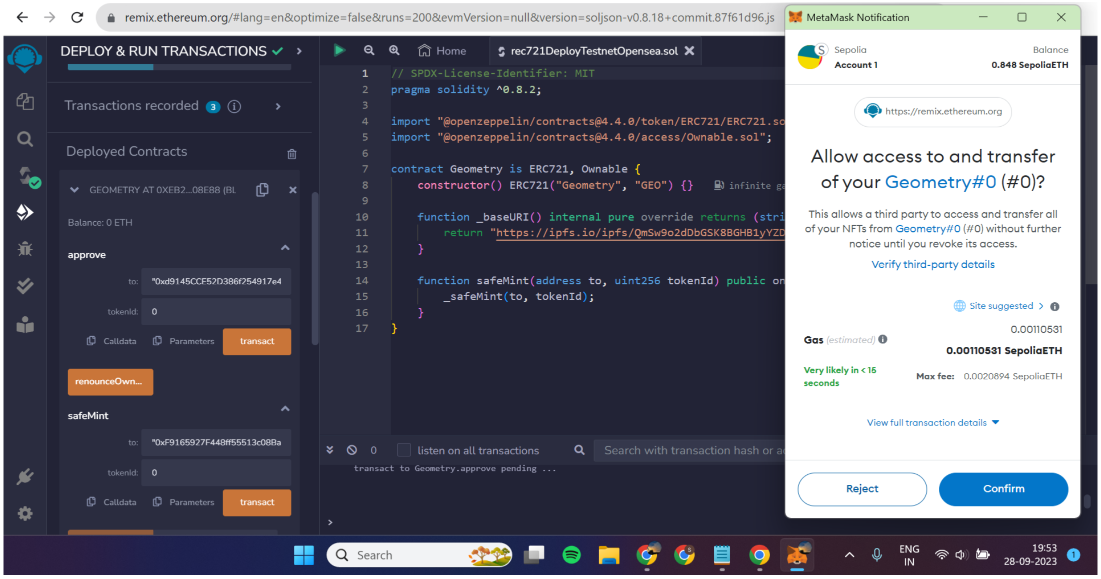
auctionContractSetup.sol

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.3;

interface IERC721 {
    function safeTransferFrom(
        address from,
        address to,
        uint256 tokenId
    ) external;

    function transferFrom(
        address,
        address,
        uint
    ) external;
}

contract EnglishAuction {
    event Start();
    event Bid(address indexed sender, uint amount);
    event Withdraw(address indexed bidder, uint amount);
    event End(address winner, uint amount);

```

```
[call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: EnglishAuction.highestBidder()
data: 0x91f...90157
```

Home
listen on all transactions
Search with transaction hash or address
Debug

Start and Bid Functions



remix.ethereum.org/#lang=en&optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.18+commit.87f61d96.js

DEPLOY & RUN TRANSACTIONS

Transactions recorded 3

Deployed Contracts

GEOMETRY AT 0XEB2...08E88 (BL

Balance: 0 ETH

approve

to: "0xd9145CCE52D386f254917e4

tokenId: 0

renounceOwn...

safeMint

to: "0xF9165927F448ff55513c08Ba

tokenId: 0

rec721DeployTestnetOpensea.sol

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.2;

import "@openzeppelin/contracts@4.4.0/token/ERC721/ERC721.so
import "@openzeppelin/contracts@4.4.0/access/Ownable.sol";

contract Geometry is ERC721, Ownable {
    constructor() ERC721("Geometry", "GEO") {}    infinite ga

    function _baseURI() internal pure override returns (stri
        return "https://ipfs.io/ipfs/QmSw9o2dDbGSK8BGHB1yYZD
    }

    function safeMint(address to, uint256 tokenId) public on
        _safeMint(to, tokenId);
    }
}
```

```
transact to Geometry.approve pending ...
```

MetaMask Notification

Sepolia
Account 1

Balance
0.848 SepoliaETH

https://remix.ethereum.org

Allow access to and transfer of your Geometry#0 (#0)?

This allows a third party to access and transfer all of your NFTs from Geometry#0 (#0) without further notice until you revoke its access.

Verify third-party details

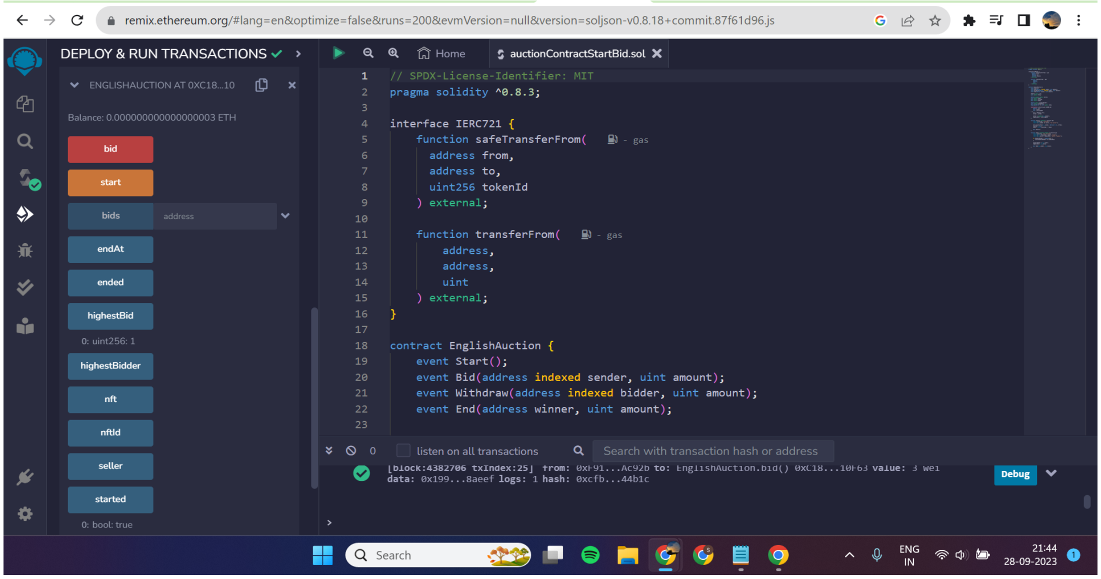
Site suggested

Gas (estimated) 0.00110531
0.00110531 SepoliaETH

Very likely in < 15 seconds

Max fee: 0.0020894 SepoliaETH

View full transaction details



remix.ethereum.org/#lang=en&optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.18+commit.87f61d96.js

DEPLOY & RUN TRANSACTIONS

ENGLISHAUCTION AT 0XC18...10

Balance: 0.000000000000000003 ETH

highestBid

0: uint256: 1

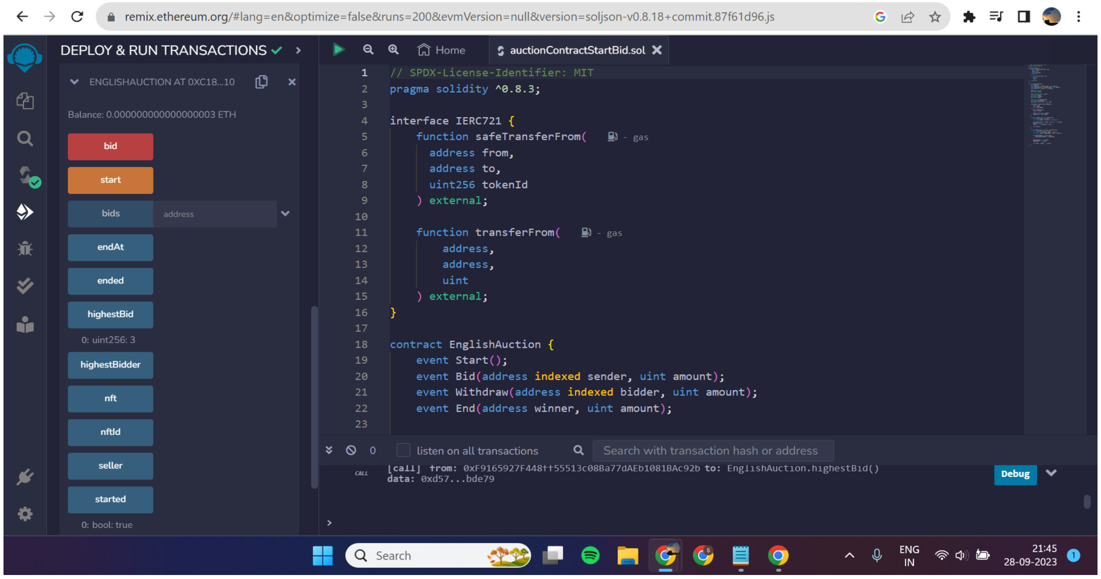
started

0: bool: true

auctionContractStartBid.sol

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.3;

interface IERC721 {
    function safeTransferFrom(    - gas
        address from,
        address to,
        uint256 tokenId
    ) external;

    function transferFrom(    - gas
        address,
        address,
        uint
    ) external;
}

contract EnglishAuction {
    event Start();
    event Bid(address indexed sender, uint amount);
    event Withdraw(address indexed bidder, uint amount);
    event End(address winner, uint amount);

```

```
[block:4382706 txIndex:25] from: 0xF91...Ac92b to: EnglishAuction.bid() 0xC18...10F63 value: 3 wei
data: 0x199...8aeef logs: 1 hash: 0xcfb...44b1c
```



remix.ethereum.org/#lang=en&optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.18+commit.87f61d96.js

DEPLOY & RUN TRANSACTIONS

ENGLISHAUCTION AT 0XC18...10

Balance: 0.000000000000000003 ETH

highestBid

0: uint256: 3

started

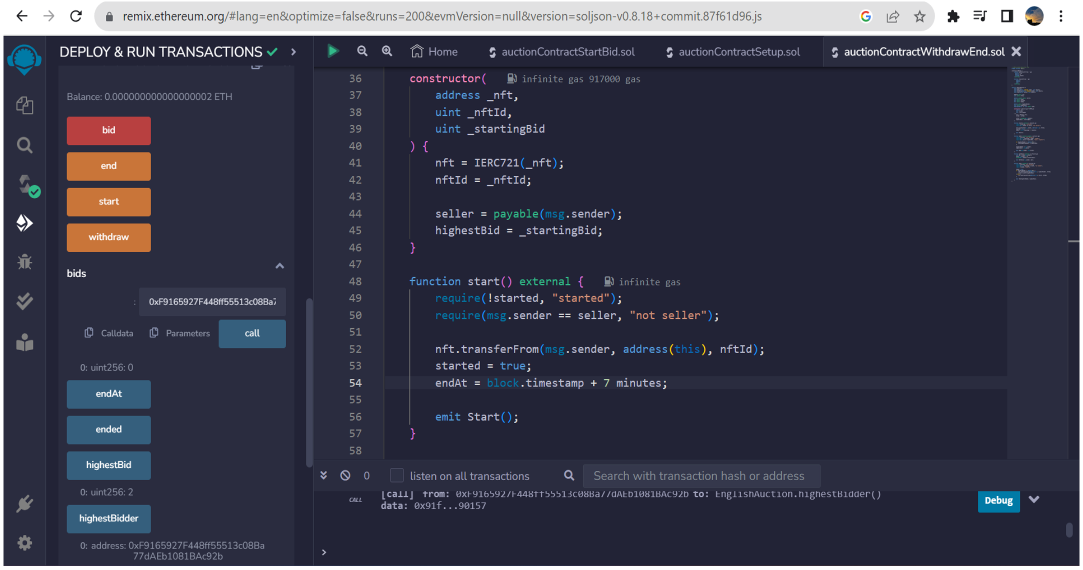
0: bool: true

auctionContractStartBid.sol

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.3;

interface IERC721 {
    function safeTransferFrom(    - gas
        address from,
        address to,
        uint256 tokenId
    ) external;

    function transferFrom(    - gas
        address,
        address,
        uint
    ) external;
}

contract EnglishAuction {
    event Start();
    event Bid(address indexed sender, uint amount);
    event Withdraw(address indexed bidder, uint amount);
    event End(address winner, uint amount);

```

```
CALL  [call] from: 0xF9165927F448ff55513c08Ba77dAEb1081BAc92b to: EnglishAuction.highestBid()
      data: 0xd57...bde79
```

Withdraw and End Functions


DEPLOY & RUN TRANSACTIONS
Home
auctionContractStartBid.sol
auctionContractSetup.sol
auctionContractWithdrawEnd.sol
bid
end
start
withdraw
endAt
ended
highestBid
highestBidder
Calldata
Parameters
call
Debug
listen on all transactions
Search with transaction hash or address

remix.ethereum.org/#lang=en&optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.18+commit.87f61d96.js

Balance: 0.000000000000000002 ETH

bids

: 0xF9165927F448ff55513c08Ba7

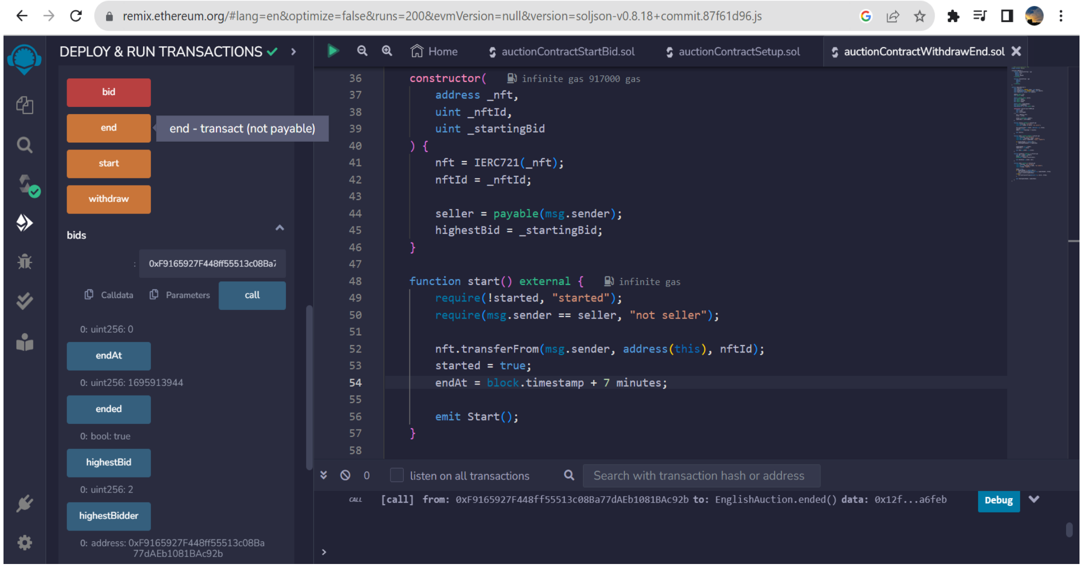
0: uint256: 0

0: uint256: 2

0: address: 0xF9165927F448ff55513c08Ba77dAEb1081BAc92b

```
    constructor(    infinite gas 917000 gas
        address _nft,
        uint _nftId,
        uint _startingBid
    ) {
        nft = IERC721(_nft);
        nftId = _nftId;

        seller = payable(msg.sender);
        highestBid = _startingBid;
    }

    function start() external {    infinite gas
        require(!started, "started");
        require(msg.sender == seller, "not seller");

        nft.transferFrom(msg.sender, address(this), nftId);
        started = true;
        endAt = block.timestamp + 7 minutes;

        emit Start();
    }

```

```
0
CALL [call] from: 0xF9165927F448ff55513c08Ba77dAEb1081BAc92b to: EnglishAuction.highestBidder()
     data: 0x91f...90157
```


DEPLOY & RUN TRANSACTIONS
Home
auctionContractStartBid.sol
auctionContractSetup.sol
auctionContractWithdrawEnd.sol
bid
end
end - transact (not payable)
start
withdraw
endAt
ended
highestBid
highestBidder
Calldata
Parameters
call
Debug
listen on all transactions
Search with transaction hash or address

remix.ethereum.org/#lang=en&optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.18+commit.87f61d96.js

bids

: 0xF9165927F448ff55513c08Ba7

0: uint256: 0

0: uint256: 1695913944

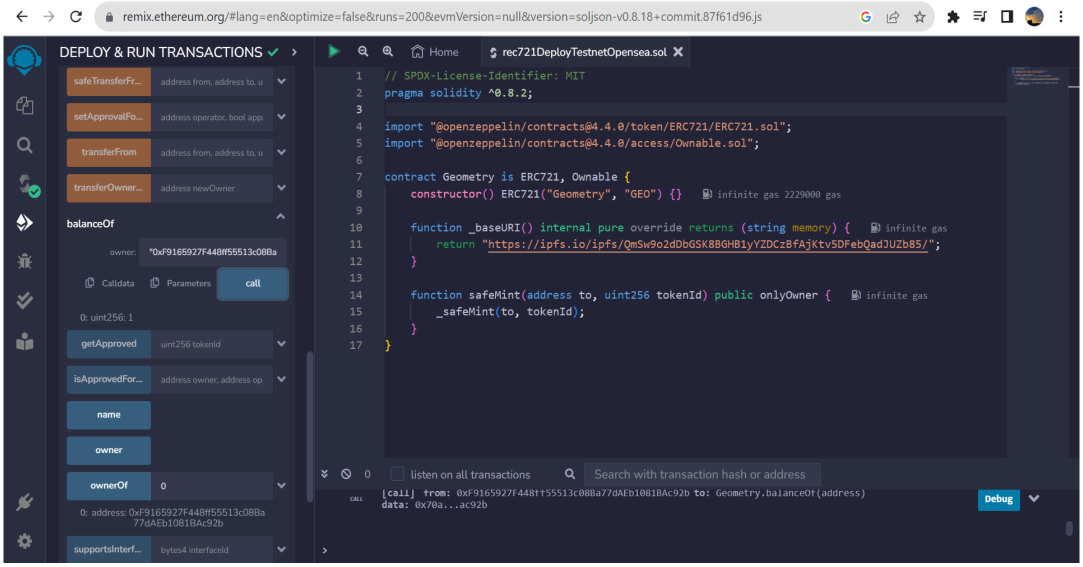
0: bool: true

0: uint256: 2

0: address: 0xF9165927F448ff55513c08Ba77dAEb1081BAc92b

```
    constructor(    infinite gas 917000 gas
        address _nft,
        uint _nftId,
        uint _startingBid
    ) {
        nft = IERC721(_nft);
        nftId = _nftId;

        seller = payable(msg.sender);
        highestBid = _startingBid;
    }

    function start() external {    infinite gas
        require(!started, "started");
        require(msg.sender == seller, "not seller");

        nft.transferFrom(msg.sender, address(this), nftId);
        started = true;
        endAt = block.timestamp + 7 minutes;

        emit Start();
    }

```

```
0
CALL [call] from: 0xF9165927F448ff55513c08Ba77dAEb1081BAc92b to: EnglishAuction.ended() data: 0x12f...a6feb
```



remix.ethereum.org/#lang=en&optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.18+commit.87f61d96.js

safeTransferFr... address from, address to, u

setApprovalFo... address operator, bool appr

transferFrom address from, address to, u

transferOwner... address newOwner

balanceOf

owner: "0xF9165927F448ff55513c08Ba

0: uint256: 1

getApproved uint256 tokenId

isApprovedFor... address owner, address op

ownerOf 0

0: address: 0xF9165927F448ff55513c08Ba77dAEb1081BAc92b

supportsInterf... bytes4 interfaceId

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.2;

import "@openzeppelin/contracts@4.4.0/token/ERC721/ERC721.sol";
import "@openzeppelin/contracts@4.4.0/access/Ownable.sol";

contract Geometry is ERC721, Ownable {
    constructor() ERC721("Geometry", "GEO") {}    infinite gas 2229000 gas

    function _baseURI() internal pure override returns (string memory) {    infinite gas
        return "https://ipfs.io/ipfs/QmSw9o2dDbGSK8BGHB1yYZDCzBfAjKtv5DFebQadJUZb85/";
    }

    function safeMint(address to, uint256 tokenId) public onlyOwner {    infinite gas
        _safeMint(to, tokenId);
    }
}
```

```
0
CALL [call] from: 0xF9165927F448ff55513c08Ba77dAEb1081BAc92b to: Geometry.balanceOf(address)
     data: 0x70a...ac92b
```