Solidity activities

Here is a simple contract that you can get, increment and decrement the count store in this contract.

*// SPDX-License-Identifier: MIT*

pragma solidity ^0.8.17;

contract Counter {

Enter your code here;

*// Function to get the current count*

function get() public view returns (uint) {

Enter your code here;

}

*// Function to increment count by 1*

function inc() public {

enter your code here;

}

*// Function to decrement count by 1*

function dec() public {

*// This function will fail if count = 0*

Enter your code here;

}

}

**Gas**

**How much ether do you need to pay for a transaction?**

You pay gas spent \* gas price amount of ether, where

* gas is a unit of computation
* gas spent is the total amount of gas used in a transaction
* gas price is how much ether you are willing to pay per gas

Transactions with higher gas price have higher priority to be included in a block.

Unspent gas will be refunded.

**Gas Limit**

There are 2 upper bounds to the amount of gas you can spend

* gas limit (max amount of gas you're willing to use for your transaction, set by you)
* block gas limit (max amount of gas allowed in a block, set by the network)

*// SPDX-License-Identifier: MIT*

pragma solidity ^0.8.17;

contract Gas {

Enter your code here

*// Using up all of the gas that you send causes your transaction to fail.*

*// State changes are undone.*

*// Gas spent are not refunded.*

function forever() public {

*// Here we run a loop until all of the gas are spent*

*// and the transaction fails*

Enter your code here{

i += 1;

}

}

}

## Ether Wallet

An example of a basic wallet.

* Anyone can send ETH.
* Only the owner can withdraw.

*// SPDX-License-Identifier: MIT*

pragma solidity ^0.8.17;

contract EtherWallet {

address payable public owner;

constructor() {

Enter your code here;

}

receive() external payable {}

function withdraw(uint \_amount) external {

Enter your code here

}

function getBalance() external view returns (uint) {

Enter your code here }

}