

Homework-1: Programming a Faucet
contract on the Sepolia testnet using Remix

110321006 蔡秉翰

110321007 劉竑毅

我們的這個智能合約是一個簡單的水龍頭（Faucet）合約，用於分發以太幣給其他地址。以下是程式的結構及流程說明：

合約擁有者（Owner）：合約中有一個 `owner` 變數，用來記錄合約的擁有者地址。擁有者是在合約部署時設置的，一旦設置後便無法更改。

接收以太幣的函式：合約中有一個 `receive` 函式，它用來接收以太幣轉入合約的操作。這個函式使用了 `external` 可見性，表示它只能被外部調用，而不能在合約內部被調用。

提取以太幣的函式：`withdraw` 函式允許合約的外部地址提取以太幣。提取的金額由函式的參數指定。在提取前，會檢查合約的餘額是否足夠。

存入以太幣的函式：`deposit` 函式用來向合約存入以太幣。存入的金額由函式調用時所傳入的以太幣數量指定。在存入時，會檢查傳入的金額是否大於零。

合約管理者提取所有資金：`withdrawAll` 函式允許合約的擁有者提取合約內所有的以太幣。這是一個管理功能，只有合約的擁有者才有權限執行這個操作。

這些函式一起組成了一個簡單的水龍頭合約，允許存款、提款和管理合約內的資金。

Powered by alchemy

Ethereum Sepolia Ethereum Goerli Base Sepolia Optimism Sepolia Arbitrum Sepolia Polygon Mumbai Polygon Amoy Welcome, 耳殘 Logout

ETHEREUM SEPOLIA FAUCET

Fast and reliable. 0.5 Sepolia ETH/day.

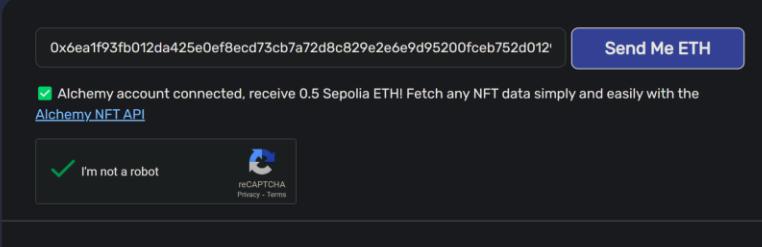
Ops! This request failed and we have notified the team. Please try again later or reach out to us on our [Discord](#) or fillout this [form](#) to share your feedback or issue.

Alchemy account connected, receive 0.5 Sepolia ETH! Fetch any NFT data simply and easily with the [Alchemy NFT API](#)

I'm not a robot  Privacy + Terms

[Send Me ETH](#)

Your Transactions Time



FILE EXPLORER

WORKSPACES

Playground

- contracts
- artifacts
- HelloWorld.sol
- scenario.json
- scripts
- prettierc.json

Home HelloWorld.sol X

```
// 正義合同擁有者地址 - 以及設置可支付的
address payable public owner;

// 合約建構函式，設置合約擁有者為合約部署者
constructor() {
    owner = payable(msg.sender);
}

// 接收以太幣的函式
receive() external payable {} 30384 gas 278800 gas

// 提取以太幣的函式，參數為提取金額
function withdraw(uint256 amount) public payable {
    // 確保水龍頭合約有足夠的餘額以提取
    require(address(this).balance >= amount, "Insufficient balance in the faucet");
    // 轉移提取金額到叫喚者地址
    payable(msg.sender).transfer(amount);
}

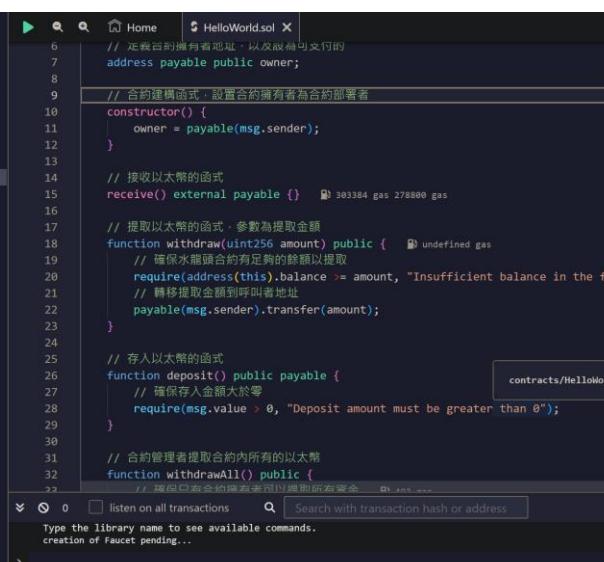
// 存入以太幣的函式
function deposit() public payable {
    // 確保存入金額大於零
    require(msg.value > 0, "Deposit amount must be greater than 0");
}

// 合約管理者提取合約內所有的以太幣
function withdrawAll() public payable {
    // 確保合約內所有以太幣轉移給合約擁有者
    owner.transfer(address(this).balance);
}
```

contracts/Helloworld.sol 20:14

listen on all transactions Search with transaction hash or address

Type the library name to see available commands. creation of Faucet pending...



Remix - Ethereum IDE

https://remix.ethereum.org/#lang=en&optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.24+commit:e1b9e...

Deploy & Run Transactions

ENVIRONMENT

Injected Provider - MetaMask

Sepolia (11155111) network

ACCOUNT

0x57d...D06F9 (0.4747580681)

GAS LIMIT

3000000

VALUE

0 Wei

CONTRACT

Faucet - contracts/Helloworld.sol

evm version: shanghai

Deploy

Publish to IPFS

At Address Load contract from Address

Transactions recorded 1

No saved contracts found

Home HelloWorld.sol X

```
// 確保只有合約擁有者可以提取所有資金 493 gas
require(msg.sender == owner, "Only the owner can withdraw");
// 將合約內所有的以太幣轉移給合約擁有者
owner.transfer(address(this).balance);
```

MetaMask Notification

Account 1

https://remix.ethereum.org

部署合約

詳情 HEX

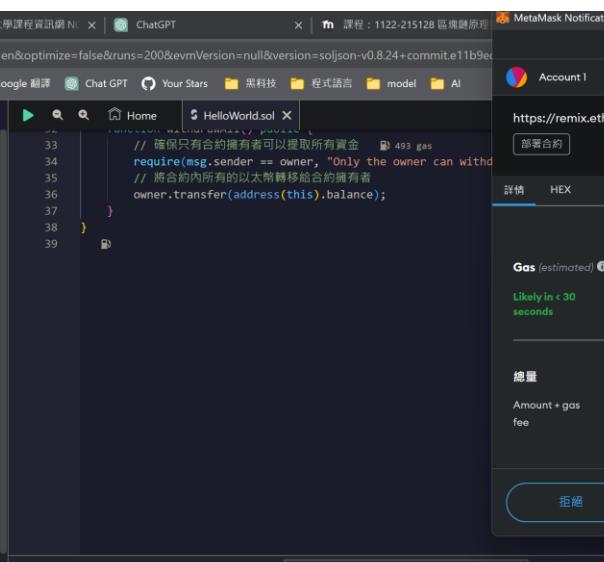
Gas (estimated) 0.00159473 SepoliaETH

Likely in < 30 seconds Max fee: 0.00195777 SepoliaETH

總量 0.00159473 SepoliaETH

Amount + gas fee 0.00195777 SepoliaETH

拒絕 確認



The screenshot shows the Truffle UI interface. On the left, the "DEPLOY & RUN TRANSACTIONS" sidebar is open, displaying account details (0x57d...D06F9), gas limit (3000000), and a deployed contract named "Faucet - contracts/HelloWorld.sol". A "Deploy" button is visible. Below it, sections for "Transactions recorded" and "Saved Contracts" are shown, both currently empty.

The main right panel displays the Solidity code for "HelloWorld.sol". The code defines a contract with a constructor that sets the owner, a withdraw function that requires the owner to be the sender, and a transfer function that sends all funds to the owner. A transaction record is listed at the bottom:

```
// 只有合約擁有者可以提取所有資金
require(msg.sender == owner, "Only the owner can withdraw all funds");
// 將合約內所有的以太幣轉移給合約擁有者
owner.transfer(address(this).balance);
```

[block:5482113 txIndex:9] from: 0x57d...d06f9 to: Faucet.(constructor) value: 0 wei data: 0x608...80033 logs: 0 hash: 0xa75...28a49

This screenshot shows the Truffle UI after a transaction has been deployed. The "DEPLOY & RUN TRANSACTIONS" sidebar now lists a deployed contract under "Deployed Contracts" with the address FAUCET at 0xA70...F253B. It includes buttons for "deposit", "withdraw", and "withdrawAll".

The main right panel shows the same Solidity code for "HelloWorld.sol" as the first screenshot. Below the code, a transaction record is displayed:

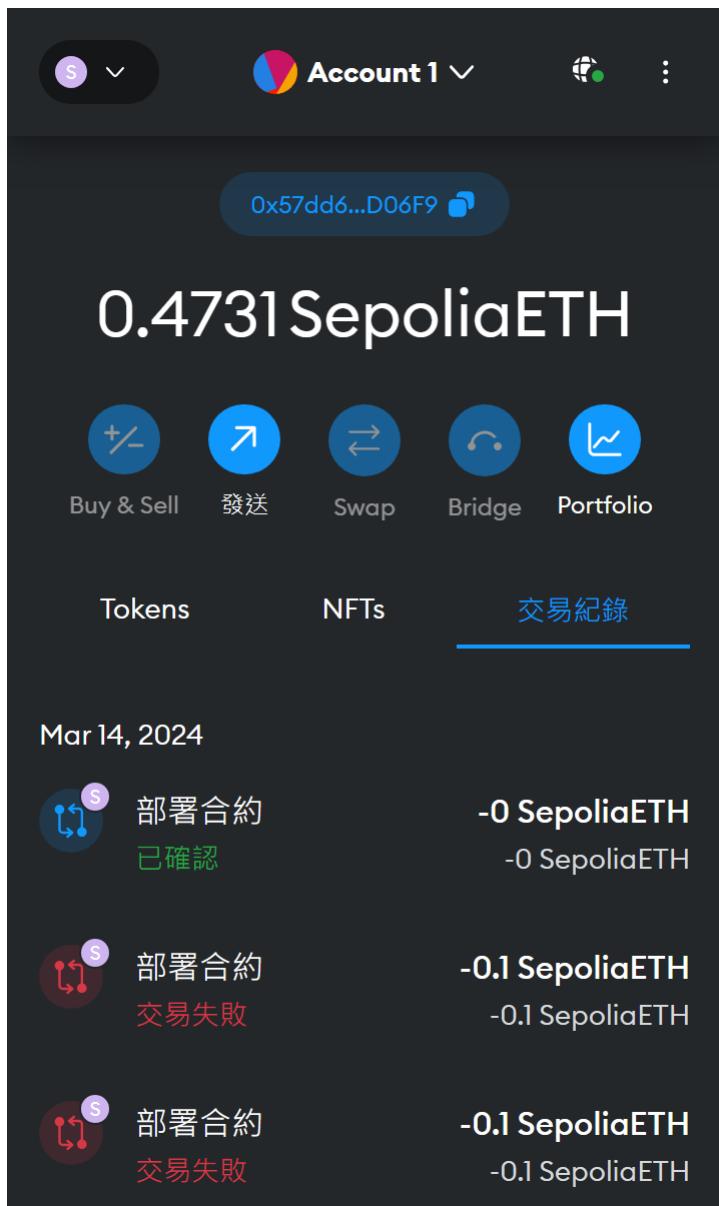
```
// 設置合約擁有者地址，以及設置可支付的
address payable public owner;
// 合約建構函式，設置合約擁有者為合約部署者
constructor() {
    owner = payable(msg.sender);
}

// 接收以太幣的函式
receive() external payable {}  303384 gas 278800 gas

// 提取以太幣的函式，參數為提取金額
function withdraw(uint256 amount) public {
    // 確保水龍頭合約有足夠的餘額以提取
    require(address(this).balance >= amount, "Insufficient balance in the faucet");
    // 轉移提取金額到呼叫者地址
    payable(msg.sender).transfer(amount);
}

// 存入以太幣的函式
function deposit() public payable {
    // 確保存入金額大於零
    require(msg.value > 0, "Deposit amount must be greater than 0");
}

// 合約管理者提取合約內所有的以太幣
function withdrawAll() public {
    // 確保合約內所有資金被提取
    owner.transfer(address(this).balance);
}
```



Account 1

0x57dd6...D06F9

0.4731 SepoliaETH

Buy & Sell 發送 Swap Bridge Portfolio

Tokens NFTs 交易紀錄

Mar 14, 2024

	部署合約 已確認	-0 SepoliaETH -0 SepoliaETH
	部署合約 交易失敗	-0.1 SepoliaETH -0.1 SepoliaETH
	部署合約 交易失敗	-0.1 SepoliaETH -0.1 SepoliaETH

在使用 Remix IDE 時，遇到的兩個警告是關於缺少 SPDX license identifier 和未指定所需的編譯器版本。

解決這些問題的方法如下：

添加 SPDX license identifier：在程式碼的開頭添加// SPDX-License-Identifier: MIT 註釋，表示程式碼採用 MIT 許可證發布。

指定所需的編譯器版本：在程式碼的開頭添加 pragma solidity ^0.8.0;
或 pragma solidity ^0.8.24