

2023 智能合約實戰演練

2023

Speaker: Cupid Sie 謝銘峰

Copyright & acknowledgement: Photo etc may be from web. Content users
acknowledge it by including this whole page as it is

Speaker Intro

謝銘峰 Sie Ming Fong

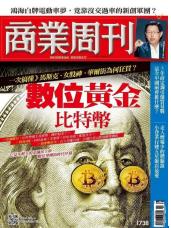
經歷

- SuDo Research Labs Researcher
- Smart Contract Research Forum Research Fellow
- 台灣區塊鏈之父 NTU SUIFT 共識實驗室
- 哥，我塊步行了、區塊勢 Podcaster 靈界偵探
- 區塊鏈大學聯盟理事
- Google University Relations 計劃研究員
- Polonex 交易所 Market Maker 月交易量達 500 萬美金
- 台大資工博士候選人、師大資訊教育研究所碩士

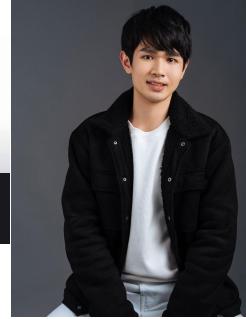
研究領域

- DeFi、公鏈、合約審計、反洗錢、人工智慧

謝銘峰，28 歲，台大資工所博士班
鑑於幫閻國翌，他把投資心法全公布在網上，朋友問他：「你怎麼把秘密都告訴別人了？」他說：「我希望當一個『佈道者』，幫大家走上財務自由之路，發大財。要走得踏實，不能踏空。」



Creditor



近期項目

- 2022 區塊鏈愛好者協會平方募資法海洋區塊鏈 第三名
- 2020 總統盃黑客松 海洋區塊鏈 垃圾變黃金 入圍前六強
- 2020 金管會監理松 打虎抓賊也著親兄弟，希望的種子 AI 企業健檢醫生 入圍全國複賽

媒體報導

- 2021/07 遠見雜誌 [台大資工幣圈高手！百萬虛幣秒蒸發 悟出分散投資之道](#)
- 2021/03 商業週刊 1738 期 數位黃金比特幣
- 2021/04 三立新聞 [成員個個身價百萬.千萬! 揭密比特幣百萬富翁](#)
- 2018/10 數位時代 [韭菜還是玩家？看發幣方和買幣人的歡笑與淚水](#)
- 2017/08 非凡新聞 [從挖礦到ICO 台大生半年賺本金30倍](#)



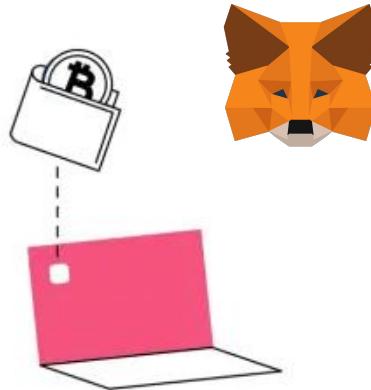
Smart Contract
Research Forum



Types of Wallets

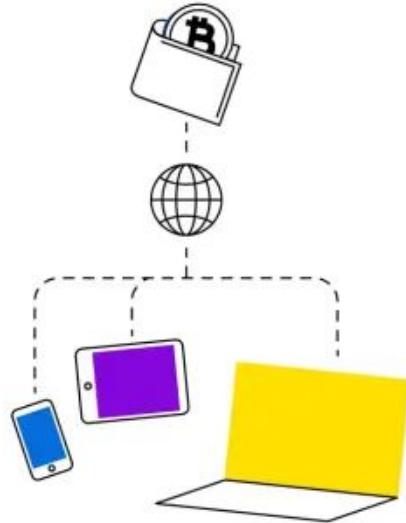
What is a Wallet?

Overview of Wallets



Software Wallet

A software wallet is software used on your desktop or mobile phone similar to a digital bank account for your cryptocurrencies, in order to manage, send and receive your coins.



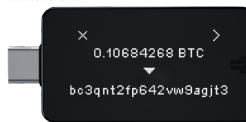
Web Wallet

Web wallets are “hot wallets” and hosted by third parties such as cryptocurrency exchanges online for increased user convenience.



Cold Wallet

A “cold wallet” is any type of wallet that is not connected to the internet, such as a paper wallet or a USB drive.



Hardware Wallet

Devices specifically designed to safely store cryptocurrencies. They are highly secure and probably the best way to store funds.



Source: [Safe Storage](#), [dchained](#), [shiftcrypto](#)

How Wallet Work

 METAMASK

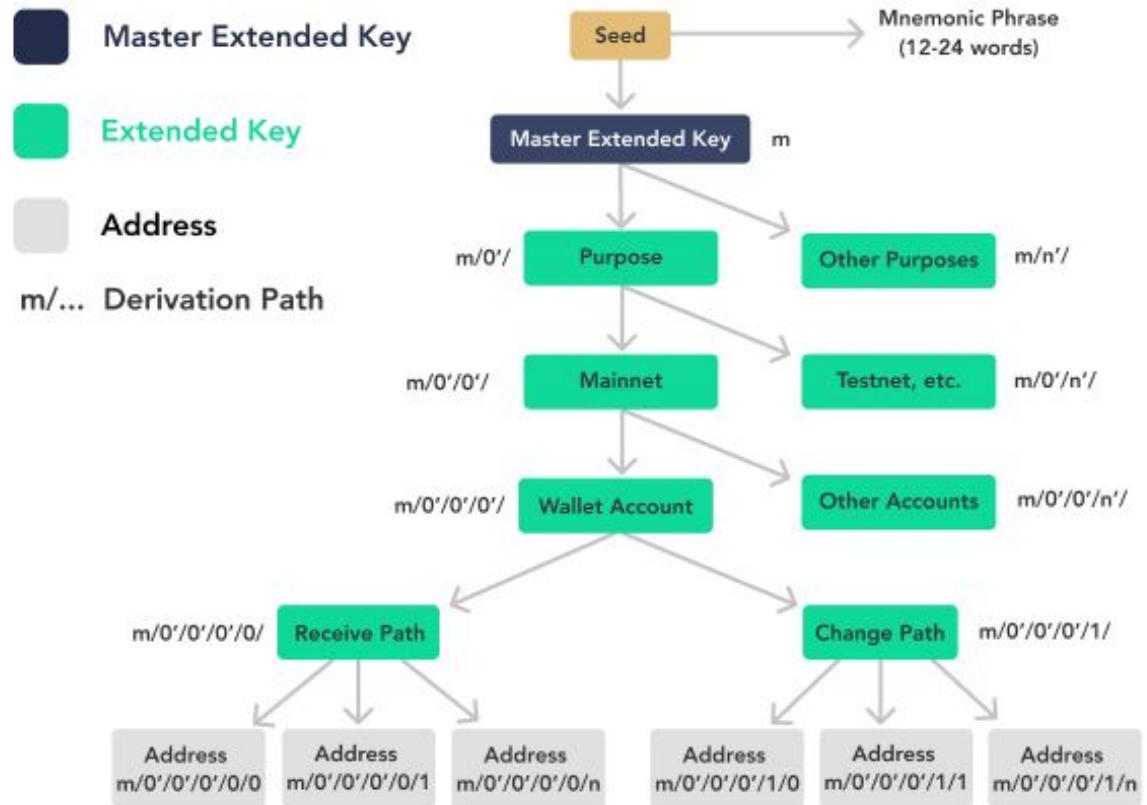
Secret Backup Phrase

Your secret backup phrase makes it easy to back up and restore your account.

WARNING: Never disclose your backup phrase. Anyone with this phrase can take your Ether forever.

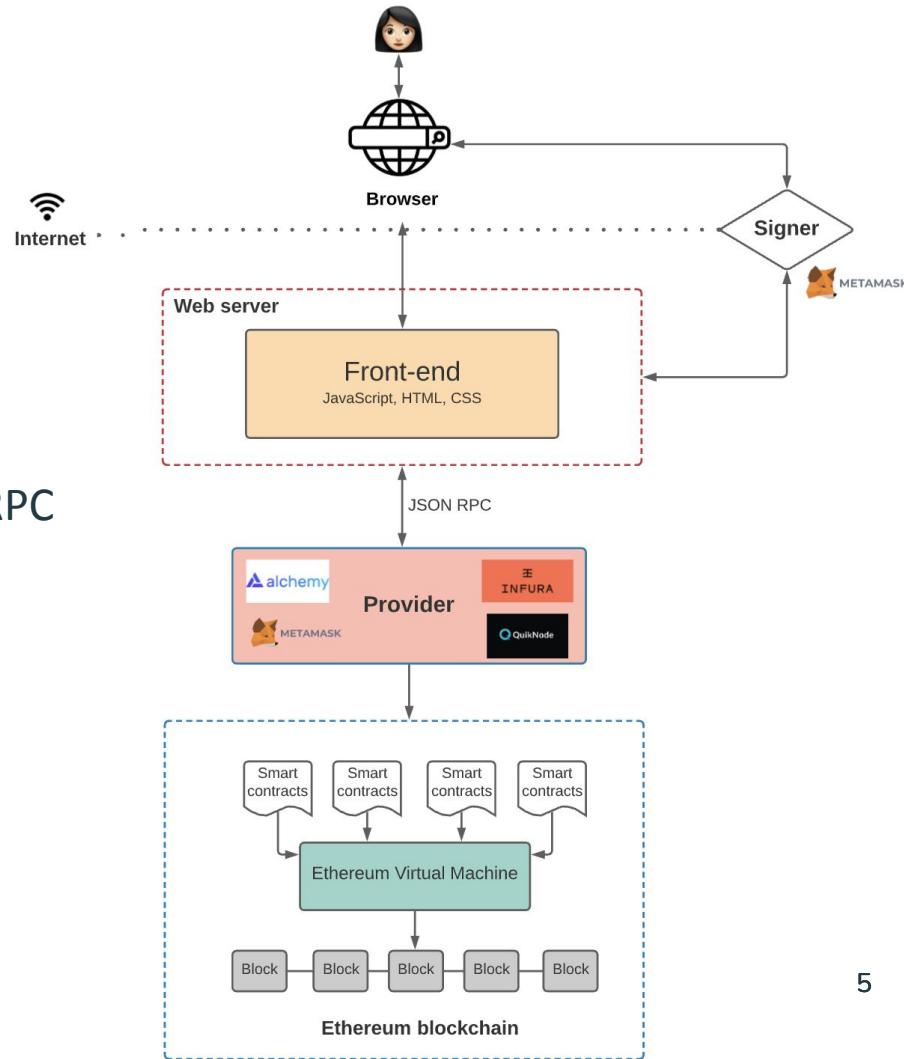
inside rug engine permit peer
squeeze slight aspect sudden traffic
crash giraffe

[Remind me later](#) [Next](#)



Wallet mechanism

- Frontend provide user interface
 - Website
 - App
- Backend - ethereum client provide JSON RPC
 - Infura
 - Alchemy
 - Metamask
 - Quicknode



Source: [Preethi Kasireddy](#)

Cold Wallet - Paper Wallet

- Website Generate
 - Public Key
 - Private Key



Source: [dchained](#), [BitAddress](#)

 **bitaddress.org**

Open Source JavaScript Client-Side Bitcoin Wallet Generator

43% 43% 43% Brain Wallet

43% 43% Wallet Details

Generating Bitcoin Address...
MOVE your mouse around to add some extra randomness... 43%
OR type some random characters into this textbox

```
eea668878dd61abf58c847106584e0f54424f67e3f8efc036699e60a3fe395037
c50fcfa65fd26e396013b77aa4acc606eed389b9bf9cf70a3ea5016066236579
b7afad59a363c6b26e6d901f5afb0edf37cf1a7909a19f1f970df996afbea0
52b56d32448bd456c07d63b0c8add10192d356433991b8928222ee523586a6aec
298f6e21d1ccce296922bcd774c48e0107637b13c24e9d8a62755537c11beac6f
3eabf471eb5b959e2e5f62dc77396f269062ba8112e2953841ed1c9c04bc42f06
c8f3e33442f40a827031d8e1272c119bc126ac0988e153e56df99c84c063345ce
25646b2c83adf9544ee1b4286e49d2a8a25988fd7df0de23eb04d4932
```

How to survived - You Own Your Assets

- Cold Wallet Generate
 - Public Key
 - Private Key
- Different Cold Wallet Companies
 - Trezor founded in 2013
 - Ledger founded in 2014
 - Coolbitx founded in 2014



Cold Wallet Support Chains



Types of Cold Wallets

The most popular cold wallets brands, include:

Name	Purchase Cost	Supported Coins	NFT Support?
Ledger Nano X	\$175	BTC, ETH, XRP, BCH, DOT, LTC, TRX, EOS, XLM, ADA, etc.	Yes
Trezor Model T	\$195	BTC, ETH, USDT, ADA, XRP, DOGE, LTC, XMR, etc.	Yes
Ledger Nano S	\$59	BTC, ETH, XRP, BCH, DOT, LTC, TRX, EOS, XLM, ADA, etc.	Yes
ELLIPAL Titan	\$139	BTC, ETH, LTC, XRP, XLM, BNB, USDT, TRX, DOT, etc.	Yes
CoolWallet Pro	\$149	BTC, ETH, LTC, XRP, XLM, BNB, USDT, TRX, Tezos, Cardano, ETH 2.0, and ERC20 tokens.	Yes
Safepal S1	\$49.99	BTC, ETH, XRP, LTC, XLM, TRX, DOGE, DOT, POLYGON, ADA, etc.	Yes
Keystone Pro	\$169	BTC, ETH, Tether, XRP, Bitcoin Cash, Polkot, LTC, Kucoin, etc.	Yes
Keepkey	\$190	Bitcoin, Bitcoin Cash, Bitcoin Gold, DASH, Dogecoin, Namecoin, Ethereum, Litecoin, etc.	Yes

Source: Horizen.io, consensys

Vitalik: Self-custody is important.



vitalik.eth ✅

@VitalikButerin

...

Self-custody is important. And social recovery and multisig is a great way to do it.

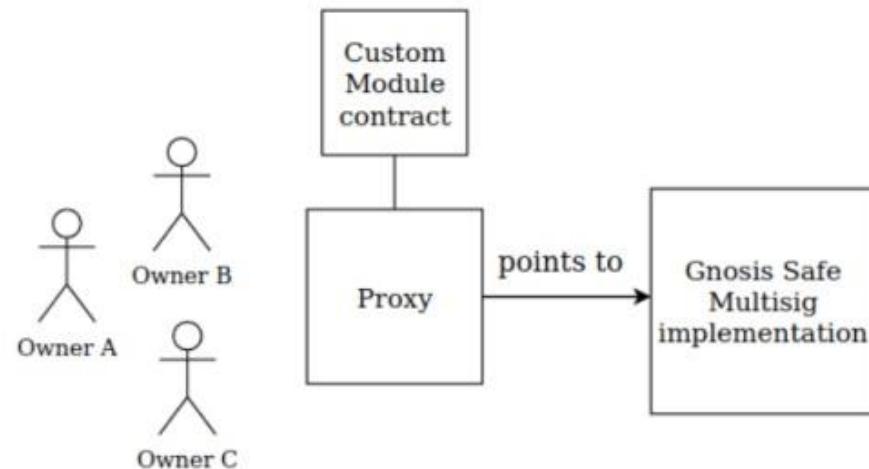
A quick reddit post on how I think about choosing guardians for social recovery and multisig wallets:

old.reddit.com/r/ethereum/com...

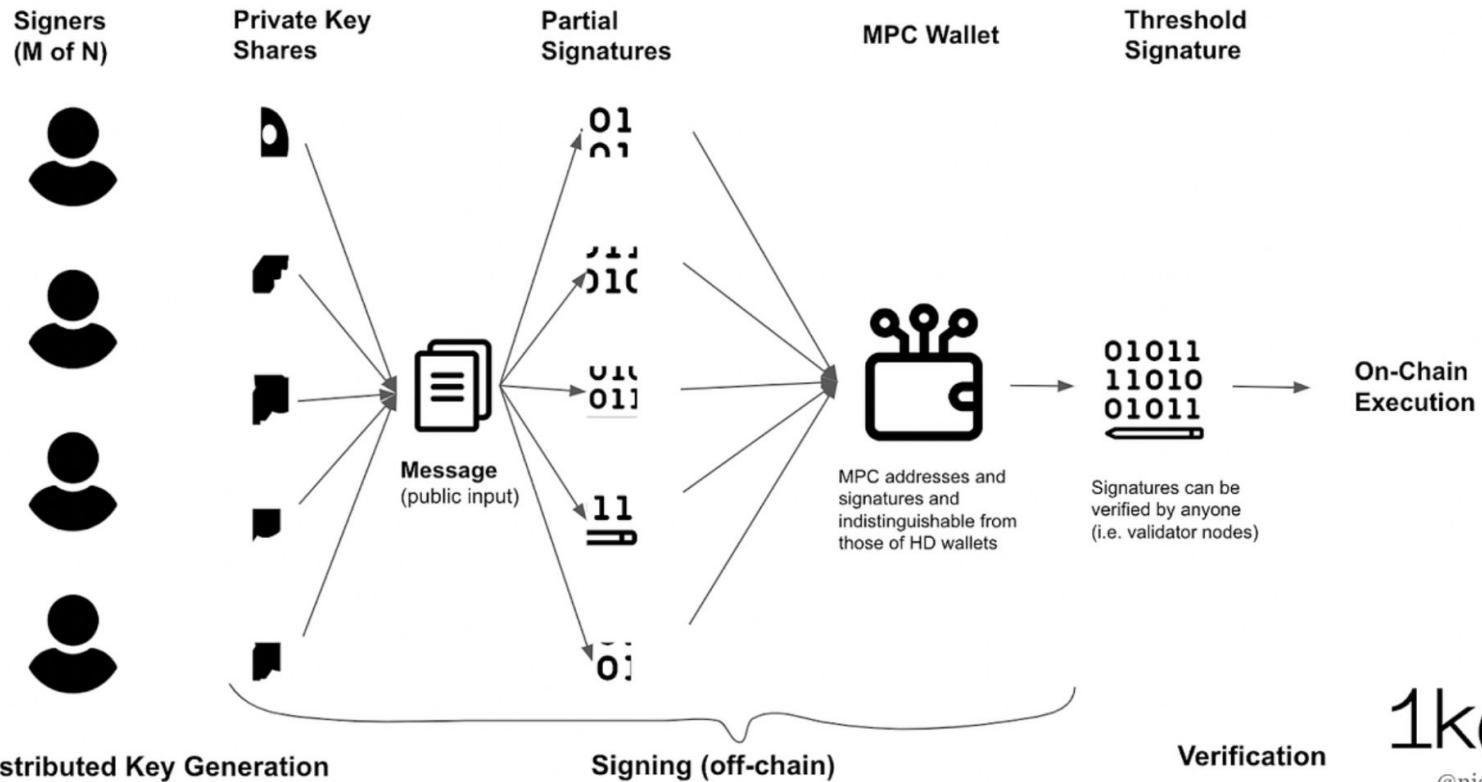
2:01 PM · Mar 17, 2023 · 111.6K Views

Gnosis Safe

- Smart Contract Account
 - An account managed by at least two private keys
 - Access controlled by many private keys enabling co-ownership and stronger security for individuals



Multi-Party Computation(MPC) Wallet

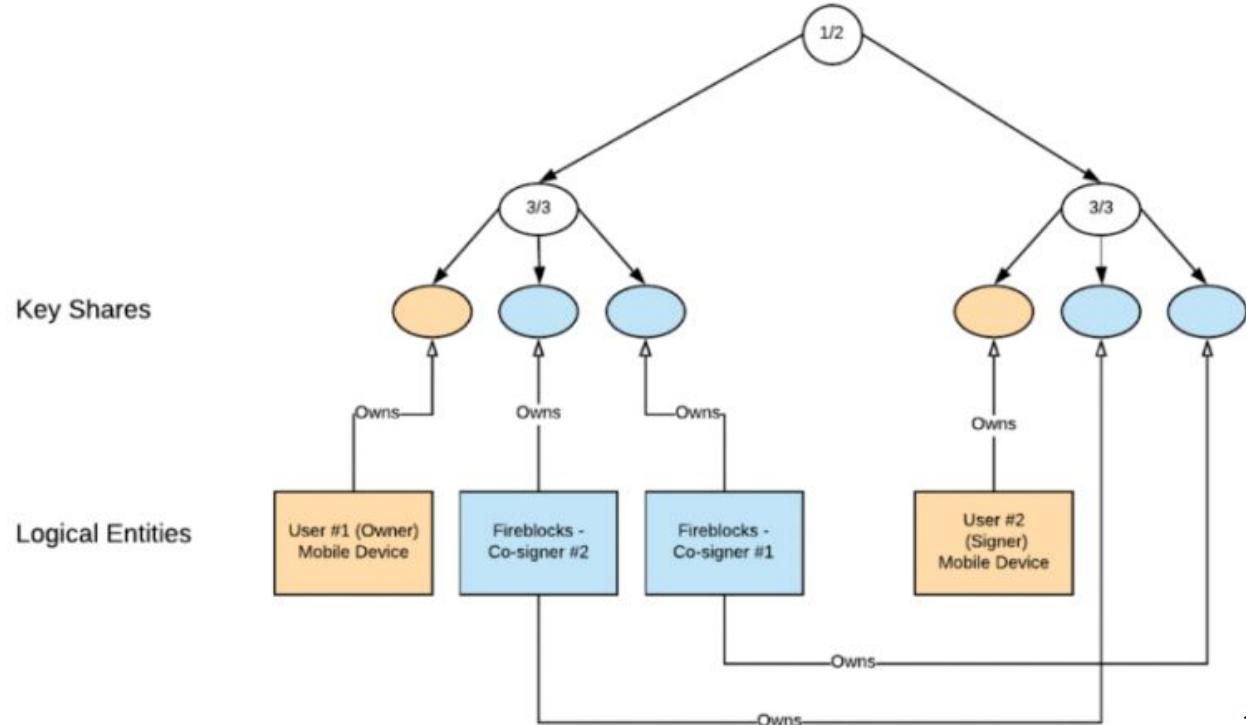


Source: [Seedless Self-Custody: On MPC and Smart Contract Wallets](#)

1k(x)
@nichanank

Fireblocks Multi-Party Computation(MPC) Wallet

- 3 Key Shares
 - A piece on Client Device
 - Two pieces on the platform



OKX MPC Wallet

- No Private Key
- Phone Device, iCloud/Google, OKX Server backup (2 of 3 recover)
- Support 37 chains
- Do **not** support some token Airdrop

無私鑰錢包 新功能

使用交易所賬戶創建 MPC 錢包



啟用面容 ID 便捷使用錢包

啟用後，可通過驗證面容ID 快速解鎖、創建、備份錢包或完成資金操作。為保護資產安全，請確保設備沒有錄入其他面容信息

使用 **375 創建無私鑰錢包**

您可以通過您登錄的交易所賬戶創建一個無私鑰錢包

備份你的錢包到 iCloud

MPC 錢包將私鑰拆分成不同的碎片，其中一份將被存儲在您的 iCloud，由您個人保管。當跨設備恢復錢包或重置錢包時，需登錄 iCloud 獲取私鑰碎片來恢復錢包，請勿刪除。

請確保您的 iCloud 有剩餘空間，否則將會導致您無法在其他設備上雲恢復錢包

助記詞

你需要手動備份助記詞短語



立即啟用

暫不啟用

創建

為什麼備份到 iCloud ?

繼續

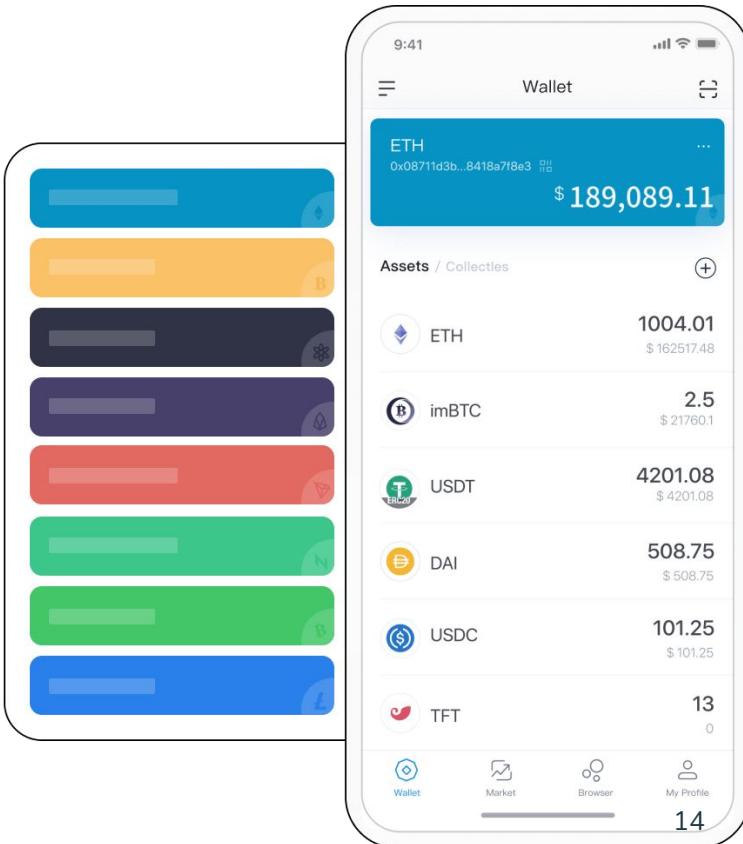
硬件錢包

可通過藍牙連接硬件錢包



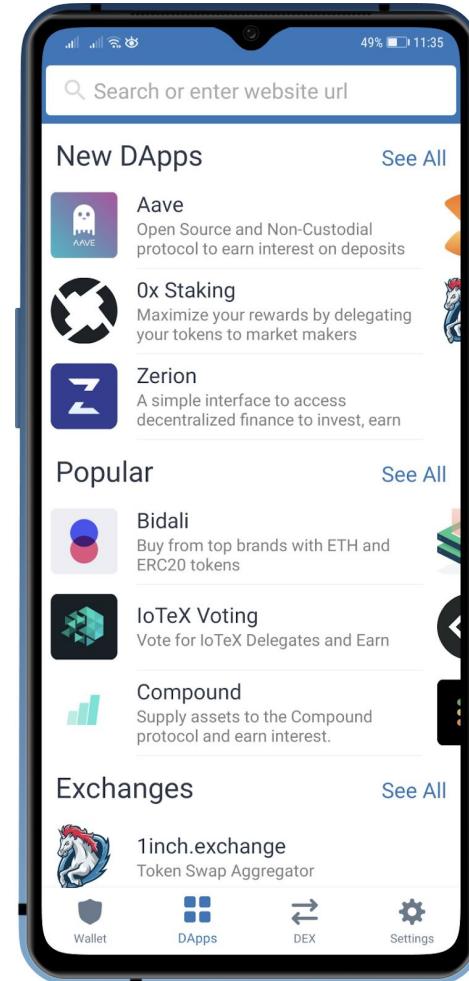
How to survive - You Own Your Assets

- Hot Wallet Generate
 - Public Key
 - Private Key
- Different Hot Wallet Service:
 - [MetaMask](#) founded in 2016
 - [imToken](#) founded in 2016
 - [麥子錢包](#) founded in 2017
 - [Trust Wallet](#) founded in 2017
 - [Coinbase Wallet](#) founded in 2018
 - [ZenGo](#) multiplichain MPC Wallet founded in 2018
 - [Blocto](#) Wallet founded in 2019
 - [OKX](#) multichian MPC Wallet lunched in 2023



DeFi application with your wallet

- Exchange
- Lending / Borrowing
- Staking
- Farming
- Bridge
- Buy coin with your credit card / apple pay / google pay



Source: [Trust Wallet](#)

Wallet Governance Token

- Total Supply: 1,000,000,000 TWT
- BEP 20 Token
- Participate in the voting processes
- Discounts on DEX services and cryptocurrency purchases
- Trust Wallet Token (TWT) **doesn't hold any real value outside its ecosystem**

The screenshot shows the Trust Wallet Token governance interface. At the top, it displays the token's logo (a shield), name, rank (Rank #48), and a snapshot of 10K members. To the right, the current price is listed as \$2.10 with a -0.15% change, along with BTC and ETH prices.

The interface includes a sidebar with navigation links: 提案 (Proposals), 新提案 (New Proposals), 关于 (About), and 设置 (Settings). The main area shows two proposals:

- Ethereum Hard Fork vs. Aptos**
Proposal Summary: Now that the Merge is complete and ETH, a separatist proof-of-work (PoW) blockchain, ETHPoW, forked from Ethereum's Merge, went live o...
Options:
 - Prioritize ETHW support 2M TWT (32.62%)
 - ✓ Continue Aptos Integration 4M TWT (67.38%)
- Dust clean-up of Binance exchange addresses on both chains (bep2 and bep20)**
Proposal Title: Dust clean-up of Binance exchange addresses on both chains (bep2 and bep20)
Proposal Introduction: Welcome to the next governance...
Options:
 - ✓ Yes 8M TWT (94.76%)
 - No 441K TWT (5.24%)

Source: [Coinmarketcap](#), [Trust Governance](#)

Smart Contract Implementation

Blockchain Wallet

- Wallet 支援 CLI(Command Line Interface) 或 GUI(Graphical User Interface) 方式
- 以太坊有兩種不同類型的帳戶：外部擁有帳戶(EOAs)和合約帳戶(Smart Contract Account)
- EOAs 由以太坊以外的軟體(如錢包應用程式)控制
- 合約帳戶由在以太坊虛擬機(EVM)內運行的軟體控制

Source: [Mastering Ethereum](#)

Ethereum Smart Contract

- Smart Contract 是在區塊鏈上的程式，合約與執行結果都會儲存在區塊鏈上
- 以太坊的 Smart Contract 運行於 EVM(Ethereum Virtual Machine) 上，程式語言為 Solidity
- Smart Contract 部署後便不可更改
- 合約運行時需要耗費 gas 若沒有足夠數量的以太幣則無法運行合約
 - e.g. Gas Price:0.001 ether, Total Gas:100



$$\text{Total Cost} = 0.001 * 100 = 0.1\text{ether}$$

Solidity



- Solidity 是一種物件(合約)導向的程式語言、為實現智能合約而創建的高級程式語言。
- Solidity 受到 C++, Python 和 Javascript 影響，設計的目的是能在以太坊虛擬機(EVM)上運行。
- 支援複雜的使用自定義函式, **libraries** 和繼承。
-
- Solidity 可以建立智能合約如投票、審核、群眾募資等合約應用。

Ethereum smart contract

Remix IDE

- <https://remix.ethereum.org>
- 一種網頁式IDE(Intergated Development Environment), 用於開發 Solidity 所撰寫的智能合約。
- 其內建功能包括：
 - 編譯
 - 除錯
 - 合約模擬 (本身提供5個虛擬帳號, 各帳號含100個ether)
 - 合約部署

FILE EXPLORERS

Workspaces 2 tabs

default_workspace 1

contracts scripts tests github .deps artifacts

(GameItems_metadata.json) GameItems.json README.txt ERC1155.sol

Home default_workspace/ERC1155.sol

```
1 // contracts/GameItems.sol
2 // SPDX-License-Identifier: MIT
3 pragma solidity ^0.6.0;
4
5 import "https://github.com/OpenZeppelin/openzeppelin-contracts/blob/v3.2.0/contracts/token/ERC1155/ERC1155.sol";
6
7 contract GameItems is ERC1155 {
8     uint256 public constant GOLD = 0;
9     uint256 public constant SILVER = 1;
10    uint256 public constant THORS_HAMMER = 2;
11    uint256 public constant SWORD = 3;
12    uint256 public constant SHIELD = 4;
13
14
15 constructor() public ERC1155("https://game.example/api/item/{id}.json") {
16     _mint(msg.sender, GOLD, 10**18, "");
17     _mint(msg.sender, SILVER, 10**27, "");
18     _mint(msg.sender, THORS_HAMMER, 1, "");
19     _mint(msg.sender, SWORD, 10**9, "");
20     _mint(msg.sender, SHIELD, 10**9, "");
21 }
22
23 }
```

功能表單

listen on network Search with transaction hash or address

[block:8893937 txIndex:28] from: 0x05b...70DB7 to: 0xaeE...4F4C9 value: 0 wei Debug

[block:8893937 txIndex:30] from: 0x1C7...9902B to: 0xdC1...09FbD value: 0 wei Debug

FILE EXPLORERS

Workspaces 1 default_workspace

- contracts
- scripts
- tests
- github
- .deps
- artifacts
 - Gamettems_metadata.json
 - Gamettems.json
- README.txt
- ERC1155.sol

default_workspace/ERC1155.sol

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.6.0;

import "https://github.com/OpenZeppelin/openzeppelin-contracts/blob/v3.2.0/contracts/token/ERC1155/ERC1155.sol";

contract GameItems is ERC1155 {
    uint256 public constant GOLD = 0;
    uint256 public constant SILVER = 1;
    uint256 public constant THORS_HAMMER = 2;
    uint256 public constant SWORD = 3;
    uint256 public constant SHIELD = 4;

    constructor() public ERC1155("https://game.example/api/item/{id}.json") {
        _mint(msg.sender, GOLD, 10**18, "");
        _mint(msg.sender, SILVER, 10**27, "");
        _mint(msg.sender, THORS_HAMMER, 1, "");
        _mint(msg.sender, SWORD, 10**9, "");
        _mint(msg.sender, SHIELD, 10**9, "");
    }
}
```

檔案管理

listen on network

Search with transaction hash or address

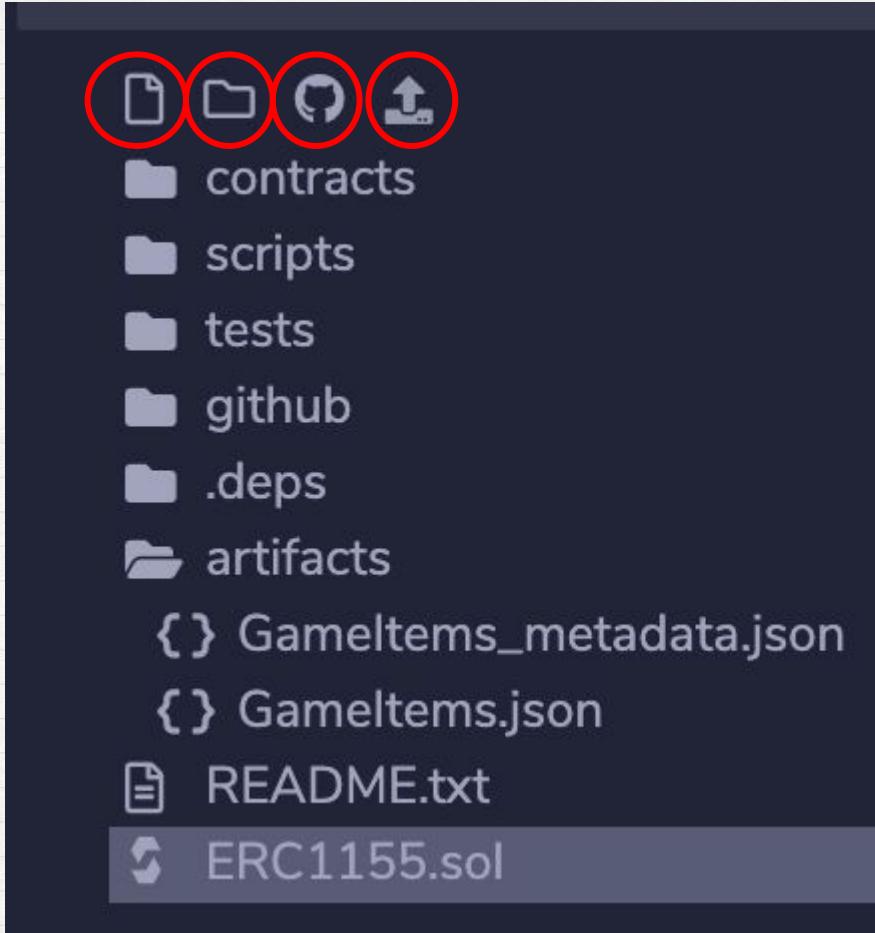
[block:8893937 txIndex:28] from: 0x05b...70DB7 to: 0xaeE...4F4C9 value: 0 wei

[block:8893937 txIndex:30] from: 0x1C7...9902B to: 0xdC1...09FbD value: 0 wei

Debug

Debug

23



1. 新增檔案
2. 新增檔案夾
3. 連結 gist
4. 上傳本機檔案(local host)

The screenshot shows a blockchain development interface with a red box highlighting the code editor area. The code editor displays Solidity code for a contract named `GameItems` that inherits from `ERC1155`. The code defines constants for various items (GOLD, SILVER, THORS_HAMMER, SWORD, SHIELD) and a constructor that mints these items to the msg.sender. A green callout bubble with the text "程式撰寫界面" (Programmable Interface) points to the code editor.

```
// contracts/GameItems.sol
// SPDX-License-Identifier: MIT
pragma solidity ^0.6.0;

import "https://github.com/OpenZeppelin/openzeppelin-contracts/blob/v3.2.0/contracts/token/ERC1155/ERC1155.sol";

contract GameItems is ERC1155 {
    uint256 public constant GOLD = 0;
    uint256 public constant SILVER = 1;
    uint256 public constant THORS_HAMMER = 2;
    uint256 public constant SWORD = 3;
    uint256 public constant SHIELD = 4;

    constructor() public ERC1155("https://game.example/api/item/{id}.json") {
        _mint(msg.sender, GOLD, 10**18, "");
        _mint(msg.sender, SILVER, 10**27, "");
        _mint(msg.sender, THORS_HAMMER, 1, "");
        _mint(msg.sender, SWORD, 10**9, "");
        _mint(msg.sender, SHIELD, 10**9, "");
    }
}
```

The screenshot shows a development environment interface with several panels:

- FILE EXPLORERS** panel on the left, showing a workspace named "default_workspace" containing files like "contracts", "scripts", "tests", "github", ".deps", "artifacts", "GameItems_metadata.json", "GameItems.json", "README.txt", and "ERC1155.sol".
- EDITOR** panel showing a Solidity code editor with the file "ERC1155.sol". The code defines a contract "GameItems" that inherits from "ERC1155". It includes constants for various items (GOLD, SILVER, THORS_HAMMER, SWORD, SHIELD) and a constructor that mints these items to the msg.sender.
- CONSOLE** panel at the bottom, highlighted with a red border. It displays two recent transactions:
 - [block:8893937 txIndex:28] from: 0x05b...70DB7 to: 0xaeE...4F4C9 value: 0 wei
 - [block:8893937 txIndex:30] from: 0x1C7...9902B to: 0xdC1...09FbD value: 0 weiEach transaction entry has a "Debug" button to its right.

A large green thought bubble with the word "console" is positioned above the console panel, and a red arrow points from the bubble towards the console output.

```
// contracts/GameItems.sol
// SPDX-License-Identifier: MIT
pragma solidity ^0.6.0;

import "https://github.com/OpenZeppelin/openzeppelin-contracts/blob/v3.2.0/contracts/token/ERC1155/ERC1155.sol";

contract GameItems is ERC1155 {
    uint256 public constant GOLD = 0;
    uint256 public constant SILVER = 1;
    uint256 public constant THORS_HAMMER = 2;
    uint256 public constant SWORD = 3;
    uint256 public constant SHIELD = 4;

    constructor() public ERC1155("https://game.example/api/item/{id}.json") {
        _mint(msg.sender, GOLD, 10**18, "");
        _mint(msg.sender, SILVER, 10**27, "");
        _mint(msg.sender, THORS_HAMMER, 1, "");
        _mint(msg.sender, SWORD, 10**9, "");
        _mint(msg.sender, SHIELD, 10**9, "");
    }
}
```

0 listen on network Search with transaction hash or address

[block:8893937 txIndex:28] from: 0x05b...70DB7 to: 0xaeE...4F4C9 value: 0 wei

[block:8893937 txIndex:30] from: 0x1C7...9902B to: 0xdC1...09FbD value: 0 wei

等待驗證中的交易

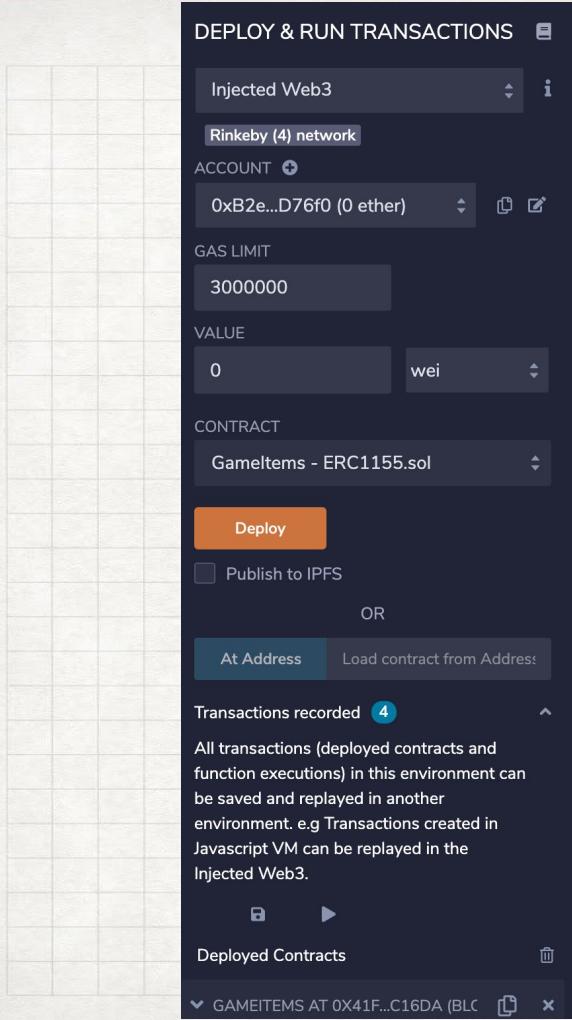
清除畫面

顯示所有交易

交易訊息

The screenshot shows a dark-themed user interface for monitoring blockchain transactions. At the top left, there are three small icons: a dropdown arrow, a circle with a minus sign, and a circle with a checkmark. Next to them is a checkbox labeled "listen on network". To the right is a search bar with the placeholder "Search with transaction hash or address". Below the search bar, there are four transaction entries, each preceded by a green circular icon with a checkmark. Each entry includes the transaction's index in the block (txIndex), the block number, the sender (from), the recipient (to), and the value transferred. A red arrow points from the text "顯示所有交易" to the search bar. Another red arrow points from the text "交易訊息" to the first transaction entry.

txIndex	block	from	to	value
35	8894005	0x3DC...	0xc39...	0 wei
32	8894005	0x068...	0xD40...	0 wei
24	8894005	0xA76...	0x665...	0 wei
34	8894005	0xf40...	0x7a2...	439423999733169455 wei



合約運行環境
e.g. testrpc, testnet, mainnet

交易的 gas 限制

欲部署的合約

已部署的合約

Ethereum smart contract

如何建立合約？

1. 00357c01000000
da47146100c257
1578063e73fa2a
fffffffffffff1fffff1
18082151581526
19150506040518
08215158152602
fffffffffffff1fffff1681
20019150506040
054111561011957
02179055506001
2540000000000000
5766666669054
fffffffff1631908115
6101000a900473
60016003600050
565b6003600050
4. 0160000000000000
00d9980000000000
08473fffff1fffff1fffff1
b6000600090549

```
contract bikeRenting {  
    address public owner;  
    address public currentRenter;  
    uint public expireTime;  
    uint public unitPrice;
```

TxHash:	0xbba7d9b00f0237a69a4c4f1266bb555e535839d87e89c667e47c74de434fd649
Block Height:	1002591 (1501896 block confirmations)
TimeStamp :	253 days 22 hrs ago (Feb-14-2016 11:40:02 AM +UTC)
From:	0xe8a6c59c50eeab5a66c906fbaf45a85e77128cc2
To:	[Contract 0x87320461408c874303a2a22febb5b56e4445fddd Created]
Value:	0 Ether (\$0.00)
Gas:	1000000
Gas Price:	0.00000005 Ether
Gas Used By Transaction:	205911
Actual Tx Cost/Fee:	0.01029555 Ether (\$0.12)
Cumulative Gas Used:	750249
Nonce:	1
Input Data:	0x606060405260026101086000505560405161015638038061015683398101604052805160805160a05191909201919080383815160019081018155600090600160a060020a0332169060029060038390559183525061010260205260408220555b82518110156100eb57828181518110156100025790602001906020020151600160a060020a03166002600050826002016101008110156100025790900160005081905550806002016101026000506000858481518110156100025790602001906020020151600160a060020a031681526020019081526020016000206000

```
        owner.send(this.balance);  
    }  
}
```

快速介紹

- 貨幣單位: ether
- address: 代表一個 account, 可提錢存錢。分為兩種:
 - 使用者
 - 合約:
 - 會有一份 code 和這個 address 繩在一起。
 - 同一份 code 在不同 address 上就是不同合約。
- transaction:
 - 可為單純的送錢
 - 或用來執行合約(同時也可附錢)

快速介紹

- 合約: 分為狀態 (state) 和動作 (function)
 - state: 可永久保留, 使用者用來記錄合約的相關資訊
 - function: 對這合約狀態產生影響的動作
 - 註: Ethereum的設計機制是不鼓勵儲存 state的, 用到state耗費的gas特別高。
 - 因為每個節點都要儲存一份一樣的資料, 如果大家都用到大量的儲存, 會造成所維護的鏈非常笨重。
- gas
 - 每一個函式都是由許多單元運算所組成, 每個運算都有固定的算力消耗, 因此每執行一次函式都要付出相對應該付的費用, 而這個費用是用 ether來付。
 - 當函式寫好, gas耗費就已經固定, 唯一可變的是每單位 gas所價值的ether。你每單位gas給越多ether, 礦工更願意收入你的transaction, 也就更快被放進鏈裡。

快速介紹

- 如果是公鏈，每分錢每分貨，每個狀態都要花成本，所以會斤斤計較
- 如果使用的是私鏈呢？
 - gas好像就沒有這麼重要了？
 - 因為要有多少ether有多少ether
 - 但是狀態的儲存就可以想存多少就存多少嗎？
 - 沒錯，但同樣的，最後儲存的成本還是會轉嫁到每個節點身上。

實戰演練 Solidity

ERC20 Token (Fungible)

- Ethereum Request for Comments 20
- ERC20 是目前在以太坊上的 Token 最主流的規格標準，相對安全性較高
- 所有 ERC20 代幣可以利用 Ethereum 生態內的工具做交易、追蹤，代幣發行商不必額外開發工具



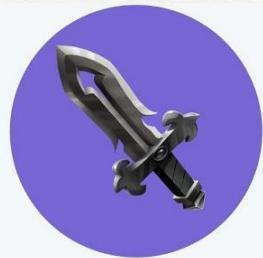
ERC721 Token (Non-Fungible)

- ERC721 是 2017-2019 在以太坊上的 Token 最主流的規格標準
- 所有 ERC721 代幣可以利用 Ethereum 生態內的工具做交易、追蹤，代幣發行商不必額外開發工具
- 可以將影像加入 ERC721 的合約當中，當作卡牌的圖片
- 每個 ERC721 Token 就等於一個限量卡牌包



ERC1155 Token (Non-Fungible)

- ERC1155 是 2019 以來在以太坊上的 Token 最主流的 NFT 規格標準
- 智能合約可以在一次傳輸多種類型的代幣，節省了80-90%的GAS費用。
- 多種類型間的代幣交易，也可以直接基於此標準交易，不需要先核准 (approve) 不同的獨立合約才能互動。
- 人們在單一智能合約內，進行同質性代幣與非同質性代幣間的互動(Batch Operation)



安裝錢包

安裝 MetaMask 瀏覽器插件

1. Chrome 線上商店搜尋 MetaMask
2. 加到 Chrome

The screenshot shows the MetaMask extension page on the Chrome Web Store. At the top, there's a navigation bar with '首頁' > '擴充功能' > 'MetaMask'. Below the navigation, there's a large orange fox head icon next to the text 'MetaMask'. To the right of the icon is a blue button labeled '加到 Chrome'. Underneath the icon, it says '來源網站: <https://metamask.io>'. Below that, there are two pieces of user data: '★★★★★ 2,021 | 實用工具' and '5,000,000+ 位使用者'.

創建帳戶

選擇創建錢包

1. 輸入自定義密碼
2. 備份助憶詞，放在一個沒有人看得見的地方
3. **千萬不要像這樣分享助憶詞**

monkey galaxy catalog age across provide

burden sweet track unable wheat expect

助憶詞

助憶詞將可協助您用更簡單的方式備份帳戶資訊。

警告：絕對不要洩漏您的助憶詞。任何人只要得知助憶詞代表他可以竊取您所有的以太幣和代幣。

monkey galaxy catalog age across
provide burden sweet track unable
wheat expect

稍後提醒我

下一頁



好，我們開始吧！

這將創建新的錢包與助憶詞

創建錢包

開啟 Metamask 測試網路顯示

Metamsak 設定

1. 點選以太坊主網路按鈕
2. 進階
3. Show test networks 開啟





選擇測試網路

- Goerli (LayerZero 把測試幣變真錢了)
 - ~~囤積測試幣的項目開發者已爆富~~
 - PoS (Proof of Stake)
 - deprecated
- Sepolia
 - PoS (Proof of Stake)
 - Infura 水龍頭可以領幣
- Linea Goerli
 - PoS
 - zkEVM

The screenshot shows a wallet interface for the Ethereum network. At the top, it says "TOKEN ETH ON ETHEREUM" and "BALANCE 1". Below that, there's a transaction input field with "Gö" and "TO GOERLIETH ON GOERLI". The transaction hash "11581.469051653792917312" is also visible.

- Goerli 測試網路
- ✓ ● Sepolia test network
- Linea Goerli 測試網路



從水龍頭索取測試以太坊

<https://www.infura.io/faucet/sepolia>

1. 註冊 infura
2. 輸入 sepolia ethereum 地址

**Sepolia ETH
delivered straight
to your wallet.**

Enter your wallet address and sign up for an Infura account to get started.

Enter your wallet address (ex. 0x)

RECEIVE ETH





選擇測試網路

- Polygon Mumbai
 - PoS (Proof of Stake)
 - [Chainlist](#) 搜尋 Polygon 加入 Mumbai RPC
 - [官網水龍頭](#)可以領幣

Mumbai

ChainID Currency

80001(0x13881) MATIC

[Add to Metamask](#)

^

Get Test Tokens

This faucet transfers TestToken on Matic testnets and parent chain. Confirm details before submitting.

Network

Mumbai Goerli Avail Devnet Avail Testnet

Select Token

MATIC Token

Wallet Address

0x77EE45524bEc344c4041435C68C3C883b92D Paste

Submit

未

網路

Show/hide test networks 忽略

- Harmony Mainnet Shard 0 X
- Avalanche Network C-C... X
- ✓ ● Mumbai



Polygon Testnet

- Network: Polygon Mumbai Testnet
 - RPC URL (public endpoint): <https://rpc-mumbai.maticvigil.com>
 - RPC URL (dedicated endpoint):
<https://polygon-mumbai.g.alchemy.com/v2/your-api-key>. You'll need a free API key
 - Chain ID: 80001
 - Currency Symbol: MATIC
 - Block Explorer URL: <https://mumbai.polygonscan.com/>

Source: [Deploy a Smart Contract Using Remix](#)

進入 Remix 線上編譯器

<https://remix.ethereum.org/>

The screenshot shows the Remix Ethereum IDE interface. On the left is a dark-themed file explorer sidebar with various icons for contracts, scripts, and tests. A workspace named "default_workspace" is selected, containing files like "contracts/1_Storage.sol", "scripts/deploy_web3.js", and "tests/4_Ballot_test.sol". The main workspace area features a cartoon character playing a guitar. To the right, there's a "Quicklinks" section with a migration guide, social sharing buttons for Twitter and GitHub, and sections for "Migration tools" and "Help". Below this is a "Featured Plugins" section with cards for Solidity, LearnETH, Solhint Linter, Sourcify, Debugger, and More.

FILE EXPLORERS

Workspaces + 🗑️

default_workspace

- contracts
- 1_Storage.sol
- 2_Owner.sol
- 3_Ballot.sol
- scripts
- deploy_web3.js
- deploy_ethers.js
- tests
- 4_Ballot_test.sol
- README.txt

Home X

1 tabs

Quicklinks

[Guide for migrating the old File System](#)

Migration tools:

- [Basic migration](#)
- [Download all Files](#) as a backup zip
- [Restore files](#) from backup zip

Help:

[Gitter channel](#)

[Report on Github](#)

Featured Plugins

SOLIDITY

LEARNETH

SOLHINT LINTER

SOURCIFY

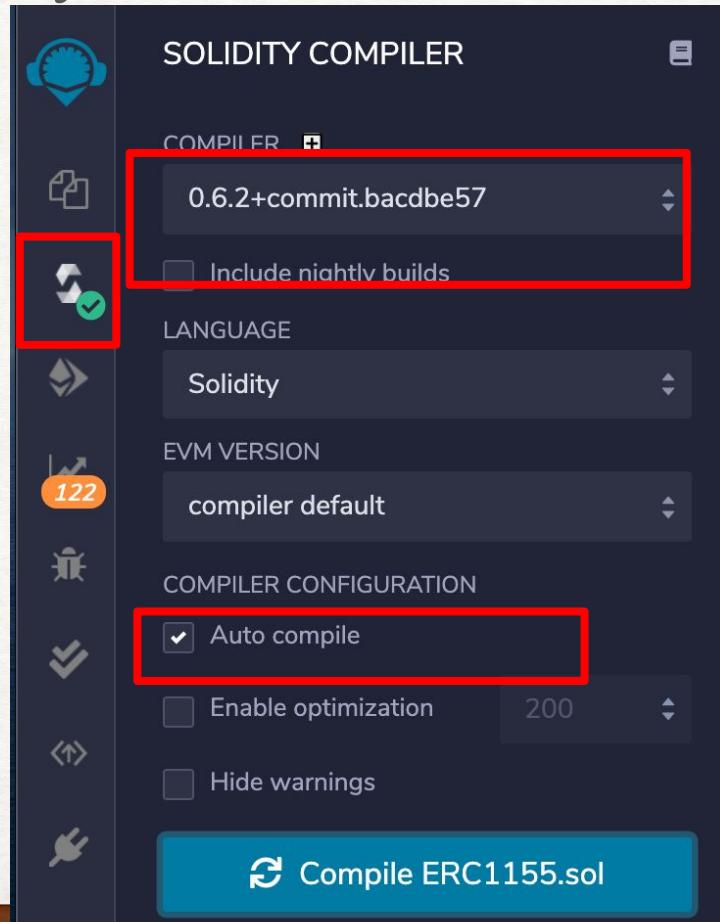
DEBUGGER

MORE

設置 Remix Solidity 編譯環境

<https://remix.ethereum.org/>

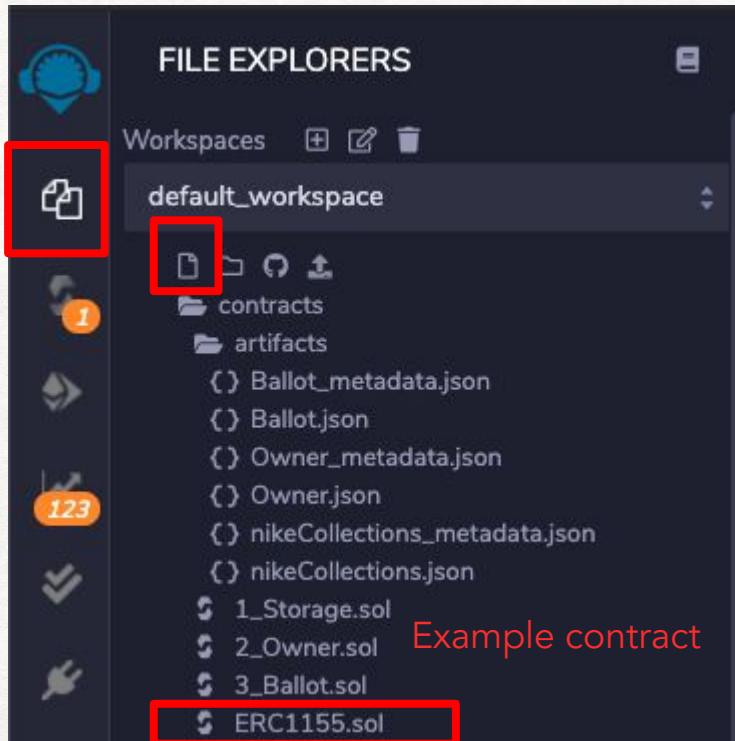
1. Choose Solidity Compiler
2. Choose Compiler version
3. Auto compile



撰寫合約 - 新增 ERC 1155 合約

<https://remix.ethereum.org/>

1. Select File Explorers
2. Add new file
3. Rename it as ERC1155.sol



撰寫 ERC 1155 合約

```
1. // contracts/Gameltems.sol
2. // SPDX-License-Identifier: MIT
3. pragma solidity ^0.6.2;
4.
5. import "https://github.com/OpenZeppelin/openzeppelin-contracts/blob/v3.2.0/contracts/token/ERC1155/ERC1155.sol";
6.
7. contract Gameltems is ERC1155 {
8.     uint256 public constant GOLD = 0;
9.     uint256 public constant SILVER = 1;
10.    uint256 public constant THORS_HAMMER = 2;      Gold is fungible token
11.    uint256 public constant SWORD = 3;
12.    uint256 public constant SHIELD = 4;
13.
14.    constructor() public ERC1155("https://game.example/api/item/{id}.json") {
15.        _mint(msg.sender, GOLD, 10**18, " ");
16.        _mint(msg.sender, SILVER, 10**27, " ");
17.        _mint(msg.sender, THORS_HAMMER, 1, " ");
18.        _mint(msg.sender, SWORD, 10**9, " ");
19.        _mint(msg.sender, SHIELD, 10**9, " ");
20.    }
21. }
```

Thors Hammer is non-fungible token. Only one



撰寫 ERC 1155 合約

```
1. // contracts/Gamelitems.sol
2. // SPDX-License-Identifier: MIT
3. pragma solidity ^0.6.0;
4.
5. import "https://github.com/OpenZeppelin/openzeppelin-contracts/blob/v3.2.0/contracts/token/ERC1155/ERC1155.sol";
6.
7. contract Gamelitems is ERC1155 {
8.     uint256 public constant GOLD = 0;
9.     uint256 public constant SILVER = 1;
10.    uint256 public constant THORS_HAMMER = 2;      Gold is fungible token
11.    uint256 public constant SWORD = 3;
12.    uint256 public constant SHIELD = 4;
13.
14. constructor() public ERC1155("https://abcoathup.github.io/SampleERC1155/api/token/{id}.json") {
15.     _mint(msg.sender, GOLD, 10**18, " ");
16.     _mint(msg.sender, SILVER, 10**27, " ");
17.     _mint(msg.sender, THORS_HAMMER, 1, " ");
18.     _mint(msg.sender, SWORD, 10**9, " ");
19.     _mint(msg.sender, SHIELD, 10**9, " ");
20. }
21. }
```

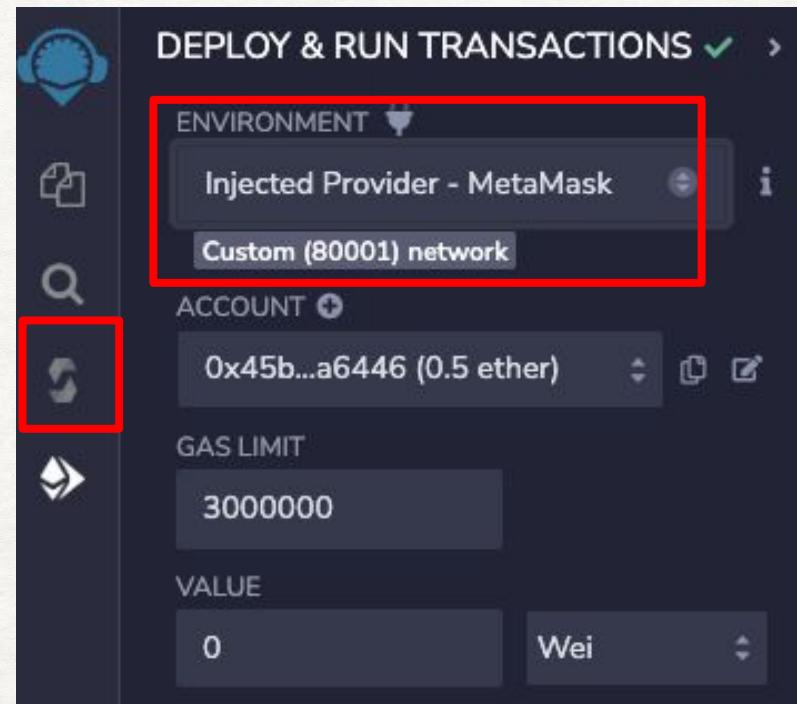
Thors Hammer is non-fungible token. Only one



連結 MetaMask Polygon Mumbai 錢包

<https://remix.ethereum.org/>

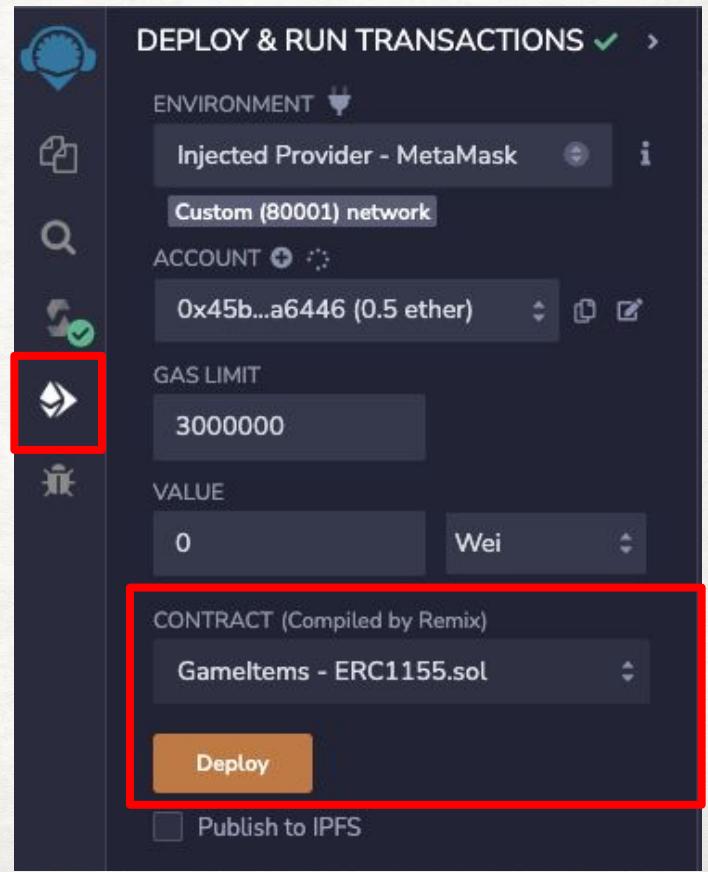
1. Choose Deploy & Run Transaction
2. Metamask choose mumbai testnet
3. Injected Provider
4. Link to your MetaMask Wallet



部署 ERC 1155 合約

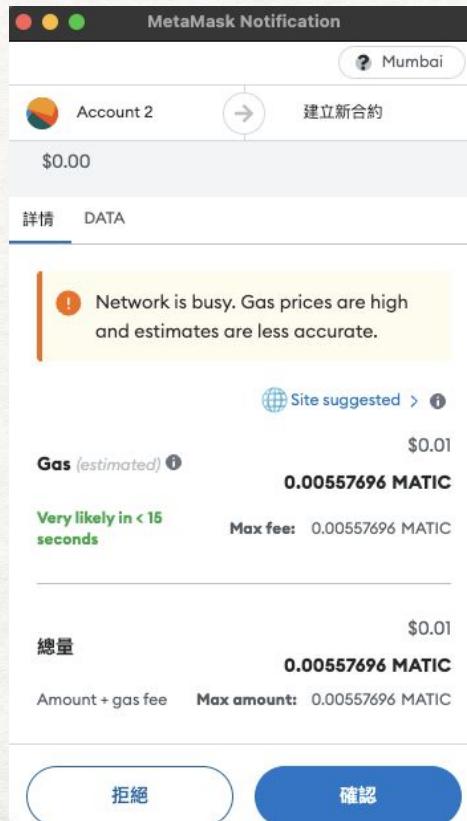
<https://remix.ethereum.org/>

1. Choose Deploy & Run Transaction
2. Check Contract
3. Deploy your ERC1155.sol

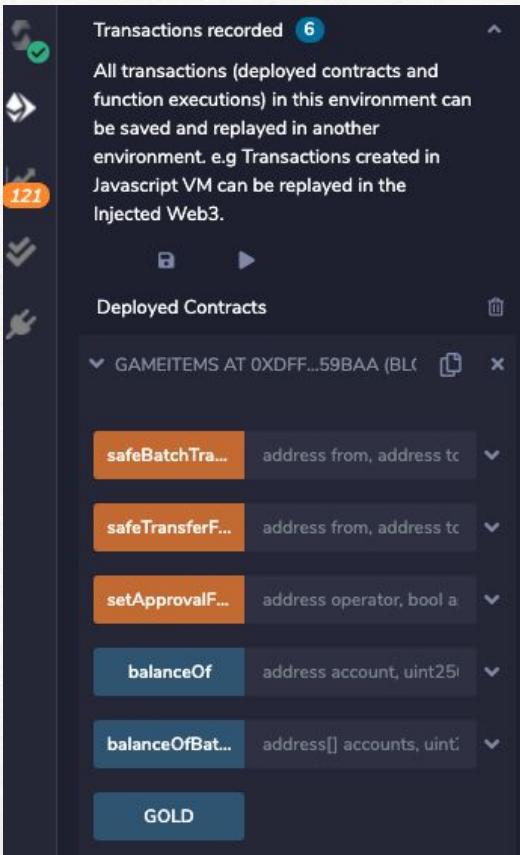


連結 MetaMask 部署合約

1. Confirm MteaMask Network
2. Check your ETH
3. Confirm Signature



ERC1155 合約部署完成



Console

```
creation of GameItems pending...
```

確認鏈上資訊

The screenshot shows the PolygonScan Mumbai interface for a specific Ethereum contract. The top navigation bar includes 'All Filters', a search bar, and tabs for 'Home', 'Blockchain', 'Tokens', 'Misc', and 'Testnet'. The main content area is divided into two main sections: 'Contract Overview' and 'More Info'.

Contract Overview: Displays the contract address (0xdf116413a5e23595343f13bE4fa9cEe7820f270c), balance (0 MATIC), and links for 'View Contract' and 'View Source'.

More Info: Includes fields for 'My Name Tag' (Not Available), 'Contract Creator' (0x45b11644013d1b34c2... at tx 0x53486879fa5316a79e...), and 'Token Tracker' (ERC1155).

Transactions: A table showing the latest transaction from a total of 1 transaction. The table columns include Txn Hash, Method, Block, Age, From, To, Value, and [Txn Fee]. The single transaction listed is 0x53486879fa5316a79e..., which is a 'Contract Creation' from 0x45b11644013d1b34c2... to 0xdf116413a5e23595343f13bE4fa9cEe7820f270c, value 0 MATIC, and fee 0.005576960035 MATIC.

<https://mumbai.polygonscan.com/address/0xdf116413a5e23595343f13bE4fa9cEe7820f270c>

確認 ERC1155 合約寶物數量

balanceOf

account: "0x641dd94Ae964007AB"

id: "2"

 call

0: uint256: 1



balanceOf

account: "0x641dd94Ae964007AB"

id: "3"

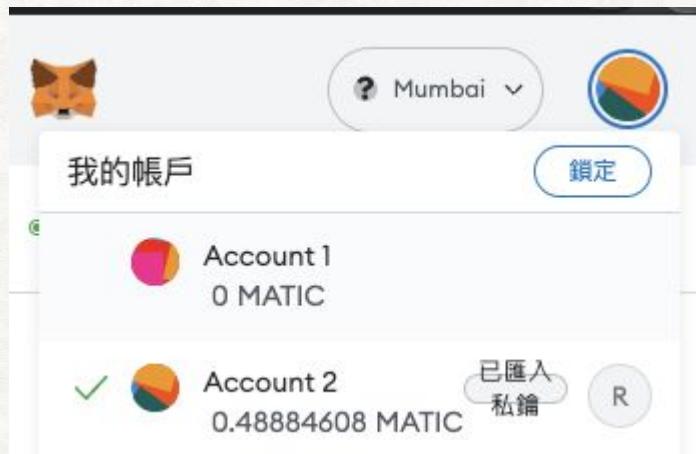
 call

0: uint256: 1000000000



增加 MetaMask 帳戶

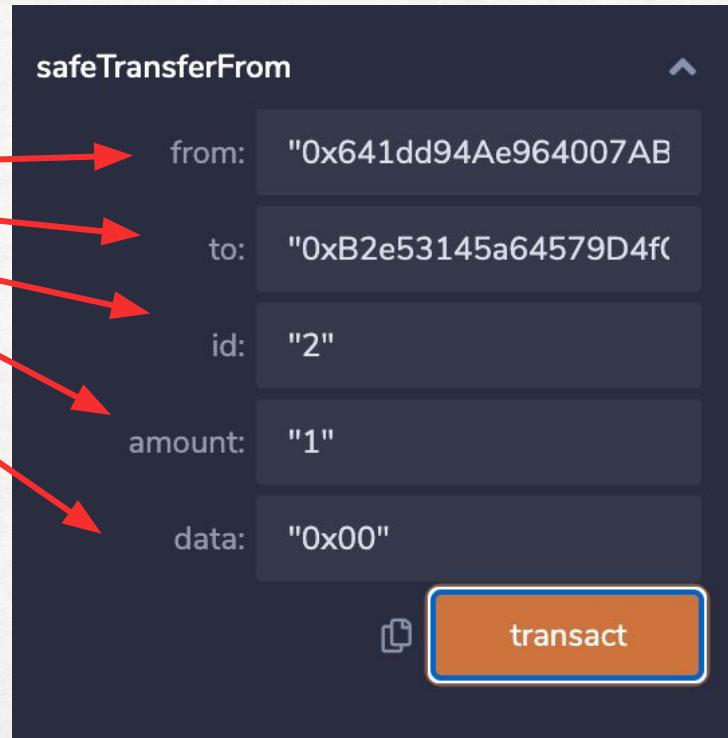
1. 建立帳戶
2. 取得帳戶 2
3. 切換回帳戶 1



傳送索爾的錘子

帳戶 1
帳戶 2
ID 2
數量
Data

合約擁有者
新玩家
錘子
1 (只有一個)
"0x00"





檢查新玩家有沒有收到錘子

帳戶 1 合約擁有者 (部署帳號)
0x45b11644013D1b34C2ba8da474b605A6C66
a6446

balanceOf

account: "0x641dd94Ae964007AB"

id: "2"

 call

0: uint256: 0

帳戶 2 新玩家
0x641dd94Ae964007AB86061933D411550b0d
0711a

balanceOf

account: "0xB2e53145a64579D4fC"

id: "2"

 call

0: uint256: 1

批次傳送寶物

帳戶 1 合約擁有者
帳戶 2 新玩家
多 ID [0, 1, 2, 3, 4]
數量 [50, 100, 1, 1, 1]
Data "0x00"

safeBatchTransferFrom

from: "0x641dd94Ae964007AB"

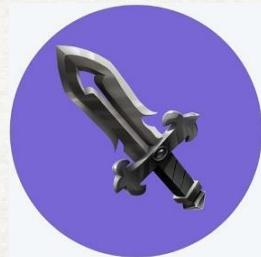
to: "0xB2e53145a64579D4fC"

ids: [0, 1, 2, 3, 4]

amounts: [50, 100, 1, 1, 1]

data: "0x00"

 transact



恭喜你已經學會如何
製作 NFT 寶物了

實戰演練 2

上架 OpenSea

OpenSea Testnet

The screenshot shows the OpenSea Testnet homepage. At the top, there is a navigation bar with the OpenSea logo, a "Testnets" button, "Drops", "Stats", and a search bar. Below the search bar, there are buttons for "All", "Art", "Gaming", "Memberships", "PFPs", and "Photography".

Two NFT listings are displayed:

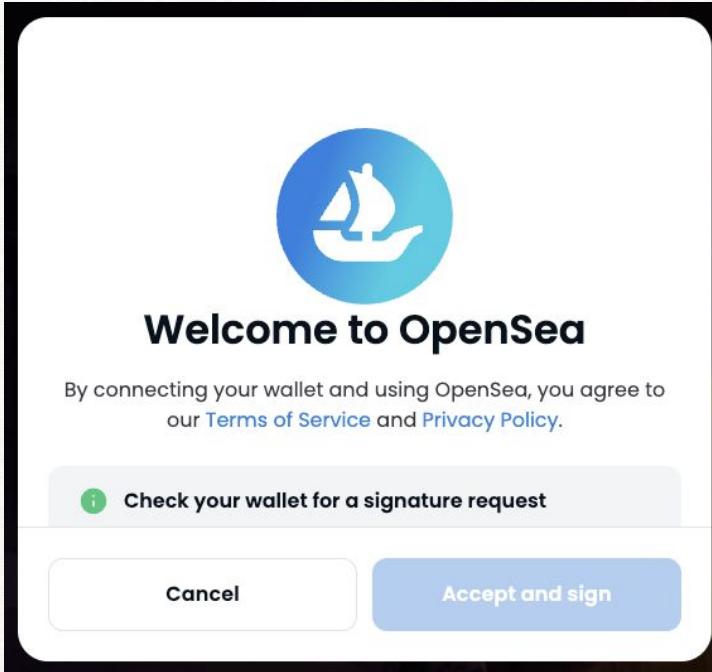
- Left Listing:** A purple-toned illustration of a shop interior with a sign that reads "THE COKED KISSES CARDS • ANTIQUES • PAINTINGS". A "View drop" button is visible at the bottom.
- Right Listing:** A dark, blurry image of a sneaker, identified as an "RTFKT x Nike Air Force 1". The listing is by "marcusbuffett".

Below the listings, there are filters for "Trending" and "Top", and time intervals for "1h", "6h", "24h", and "7d". There are also buttons for "All chains" and "View all".

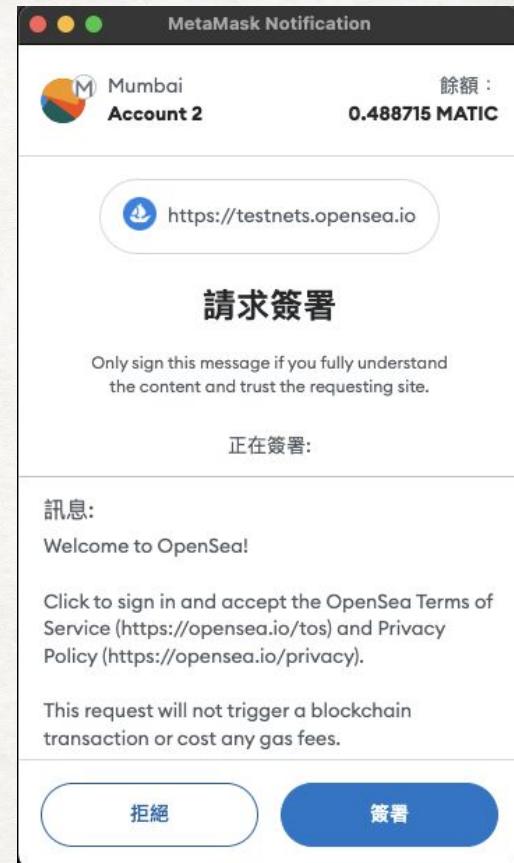
At the bottom, there is a table showing the top collections:

COLLECTION	FLOOR PRICE	VOLUME	COLLECTION	FLOOR PRICE	VOLUME
1 BoredApeYachtClub - GOE...	11.88 ETH	0.00 ETH			

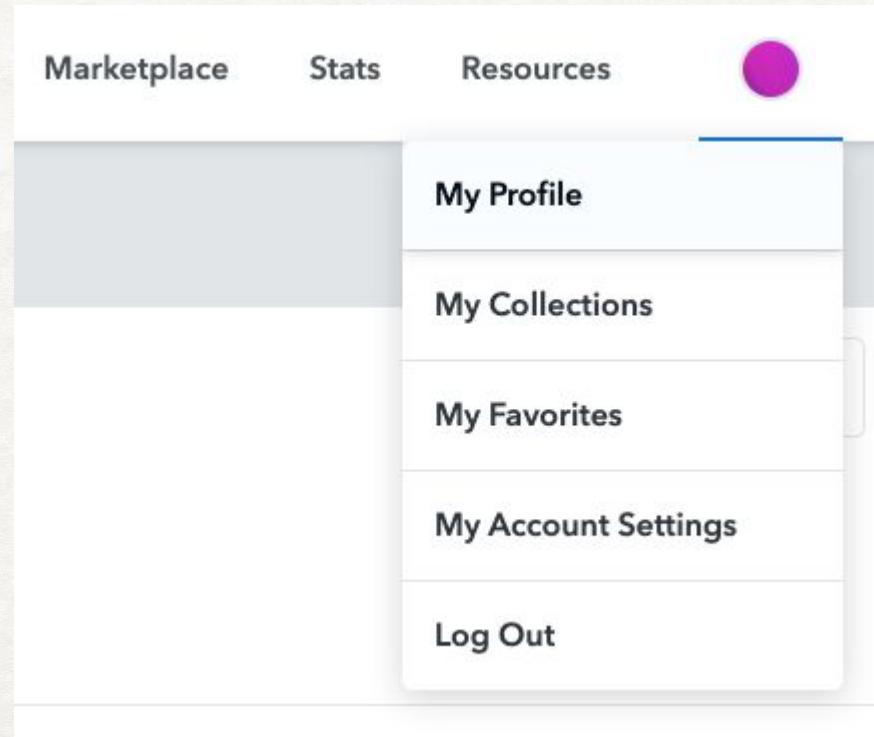
MetaMask 簽署 OpenSea



<https://testnets.opensea.io/>



檢查 My Profile



檢查索爾的錘子數量與 ID

The screenshot shows the OpenSea marketplace interface. At the top, there is a logo for OpenSea, a search bar with placeholder text "搜尋項目、作品集和帳戶", and a navigation bar with tabs for "品項" and "統計資料". On the right side, there are buttons for "0 ETH", "0.4887 MATIC", and a purple circular icon. Below the header, there are several filters: "篩選條件", "狀態", "區塊鏈", "按名稱搜尋", "最近收到", and view mode icons. The main content area displays a grid of NFT items. Each item has a thumbnail, a quantity indicator (e.g., "x1,000,000,000"), a name, and a description. One item, "Thor's Hammer", is highlighted with a red border around its thumbnail.

項目	數量	名稱	說明
Shield	x1,000,000,000	Unidentified contract	
Sword	x1,000,000,000	Unidentified contract	
Gold	x1,000,000,000,000,000,000	Unidentified contract	
Silver	x1,000,000,000,000,000,000	Unidentified contract	
Shield	x1,000,000,000,000,000,000	Unidentified contract	
Silver	x1,000,000,000,000,000,000	Unidentified contract	
Gold	x1,000,000,000,000,000,000	Unidentified contract	
Sword	x1,000,000,000	Unidentified contract	
Thor's Hammer	x1,000,000,000	Unidentified contract	

將索爾的錘子上架到 OpenSea

The screenshot shows the OpenSea interface for listing an NFT. The asset is a hammer, identified by the title "Thor's Hammer". The listing status is "Unidentified contract". The owner is listed as "you". There is one view count. A red box highlights the "上架" (List) button at the top right.

Thor's Hammer

擁有者 you

1 檢視次數

價格歷史記錄

尚未發生任何事件
請稍後再回來查看。

標售清單

報價

<https://testnets.opensea.io/zh-TW/assets/mumbai/0xdf116413a5e23595343f13be4fa9cee7820f270c/2>

設定價格後 Post your listing

標售項目

 **Thor's Hammer**
Unidentified contract

標售價格
1 MATIC
\$0.98 美元

設定價格 ①

MATIC

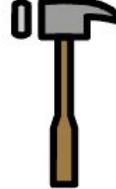
\$0.98

持續時間

 1 month

標售價格 1 MATIC
創作者收益 0%
服務費 2.5%
總潛在收益 0.975 MATIC
\$0.96 USD

[完整標售清單](#)



你的物品已標售！

Unidentified contract 作品集中的 Thor's Hammer 已標售。

分享到.....



[檢視標售清單](#)

檢查索爾的錘子拍賣頁面

The image shows a screenshot of a digital auction page for an NFT. On the left, there is a large, stylized icon of Thor's Hammer (Mjolnir), featuring a grey head and a brown handle. To the right of the icon, the title "Thor's Hammer" is displayed in bold black text. Below the title, it says "擁有者 you". There is a circular icon with the number "2" next to the text "檢視次數" (Viewed 2 times). A clock icon indicates the auction ended on "2023年6月1日 下午7:00" (June 1, 2023, 7:00 PM). The current price is listed as "1 MATIC \$0.98". A link "價格歷史記錄" (Price History) is provided. At the bottom, a clock icon and the text "尚未發生任何事件 請稍後再回來查看。" (No events have occurred yet. Please check back later.) are shown.

<https://testnets.opensea.io/zh-TW/assets/mumbai/0xdf116413a5e23595343f13be4fa9cee7820f270c/2>

檢查索爾的錘子拍賣紀錄

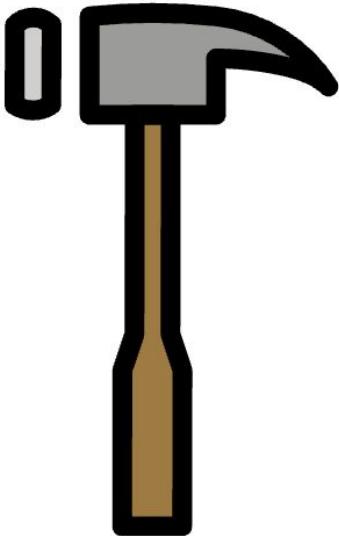
↑↓ 項目活動

筛选條件

活動	價格	從	到	日期
標售	1 MATIC	you		4 分鐘 前
已鑄造		000000	you	55 分鐘 前 <input checked="" type="checkbox"/>

<https://testnets.opensea.io/zh-TW/assets/mumbai/0xdf116413a5e23595343f13be4fa9cee7820f270c/2>

分享給同學與朋友行銷 / ~~舉辦 NFT 活動~~



Unidentified contract

Thor's Hammer

擁有者 [you](#)

🕒 銷售結束 2023年6月1日 下午7:00

目前價格
1 MATIC \$0.98

↗ 價格歷史記錄

尚未發生任何事件

▶ ⌂ ⌄ ...

複製連結

在 Facebook 上分享

在 Twitter 上分享

內嵌項目

恭喜你已經完成了 Vitalik 的夢想



請分享寶物拍賣訊息
Twitter
並分享畫面

實戰演練 3

ERC 721 上架

Opensea

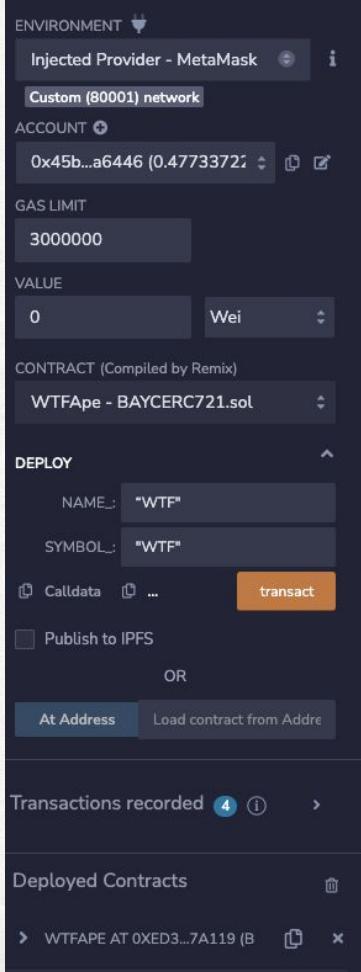
撰寫 ERC 721 合約

```
1. // SPDX-License-Identifier: MIT
2. pragma solidity ^0.8.4;
3. import "@openzeppelin/contracts/token/ERC721/ERC721.sol";
4. contract WTFape is ERC721{
5.     uint public MAX_APES = 10000; // 總量
6.
7.     // 構造函數
8.     constructor(string memory name_, string memory symbol_) ERC721(name_, symbol_){
9. }
10.
11.    //BAYC的baseURI為ipfs://QmeSjSinHpPnmXmspMjwiXyN6zS4E9zccariGR3jxcaWtq/
12.    function _baseURI() internal pure override returns (string memory) {
13.        return "ipfs://QmeSjSinHpPnmXmspMjwiXyN6zS4E9zccariGR3jxcaWtq/";
14.    }
15.
16.    // 鑄造函數
17.    function mint(address to, uint tokenId) external {
18.        require(tokenId >= 0 && tokenId < MAX_APES, "tokenId out of range");
19.        _mint(to, tokenId);
20.    }
21.
22. }
```



BAYC ERC721 is only one NFT

<https://forum.openzeppelin.com/t/create-an-erc1155/4433>



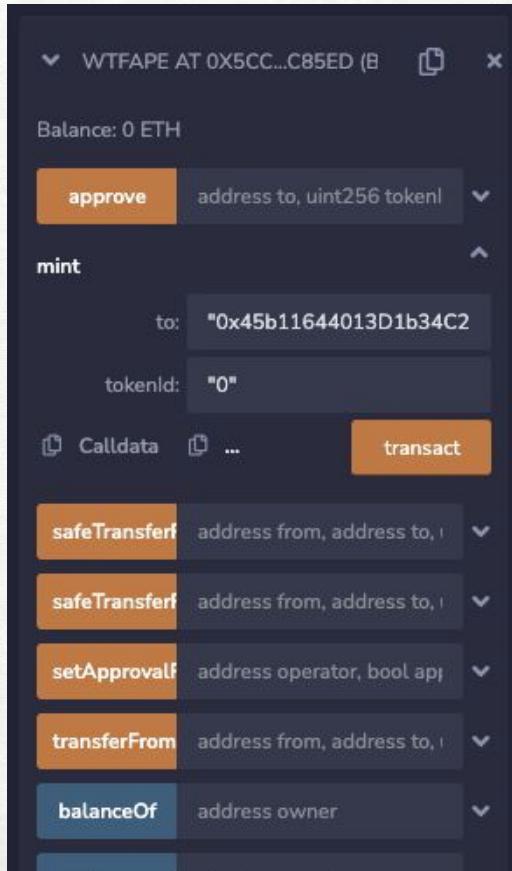
ERC721 合約部署

Console

```
[block:35293233 txIndex:11]
creation of WTFape pending...
```

NFT 系列名字
NFT 代幣名字

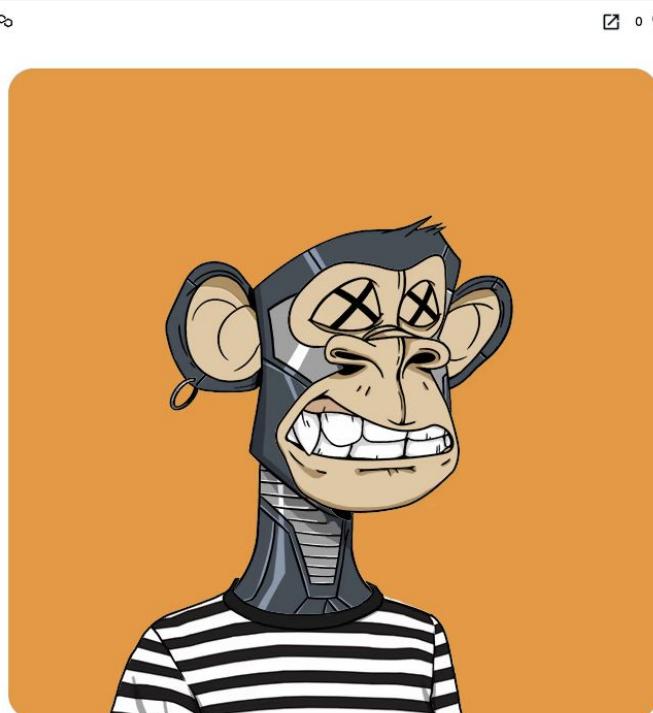
Mint BAYC NFT



自己錢包地址
tokenId 設為 0

[block:35293298 txIndex:2] from:
transact to WTFApe.mint pending ...

檢查 Opensea Testnet



A screenshot of an NFT listing on the Opensea Testnet. The main image is a cartoon-style monkey wearing a black leather collar and a black and white striped shirt, set against an orange background. The monkey has large, expressive eyes with 'X' marks over them. Above the image are icons for a profile picture, a heart, and a share button, followed by the text 'WTF'. To the right of the image, the NFT details are listed:

- #0**
- Owned by you
- 1 view

Below these details are three sections with icons and labels:

- Price History** (clock icon)
- Listings** (key icon)
- Offers** (stacked coins icon)

The entire interface has a light gray header and footer.

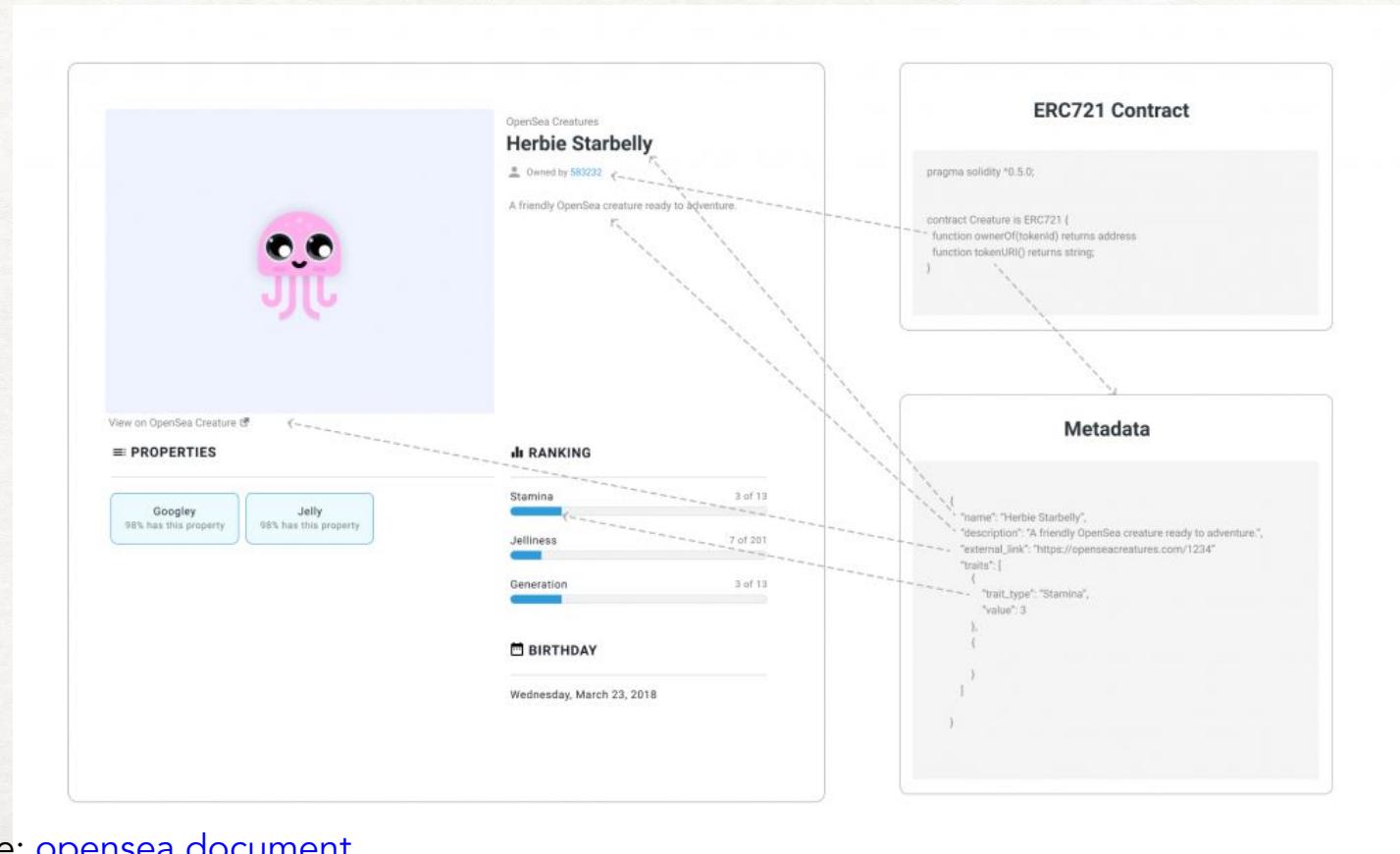
<https://testnets.opensea.io/assets/mumbai/0x5cc92a072796ff2ad14e049e7acf8f9b793c85ed/0>

實戰演練 4

ERC 721 接上

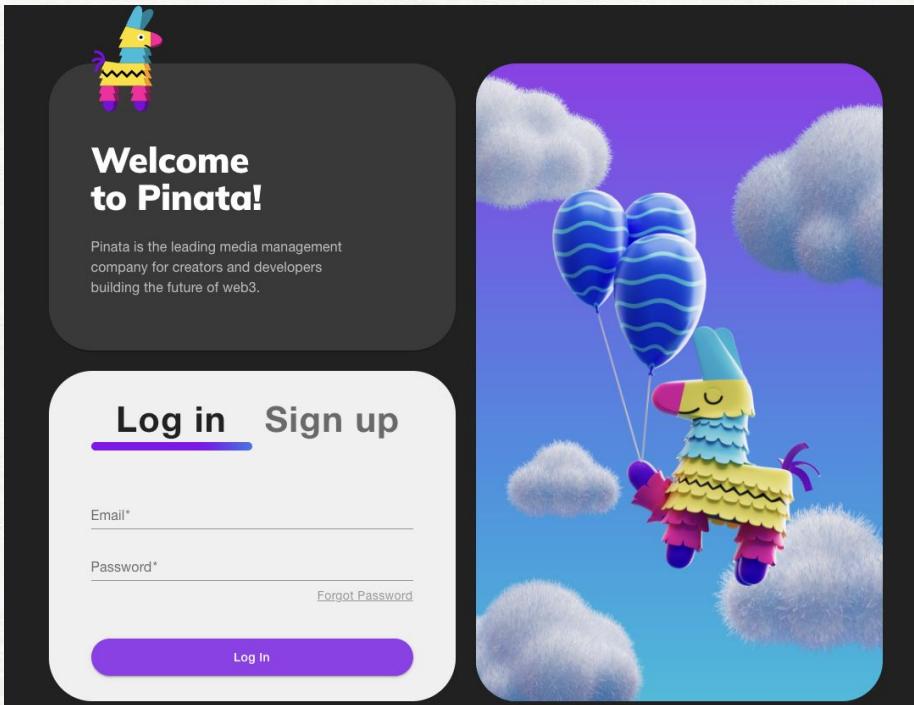
Opensea Metadata、 IPFS上傳圖片

Opensea Metadata



Source: [opensea document](#)

上傳圖片與 JSON 到 IPFS 取得 BaseURI



[Pinata](#) | Effortless IPFS File Management 註冊獲得 1G 容量

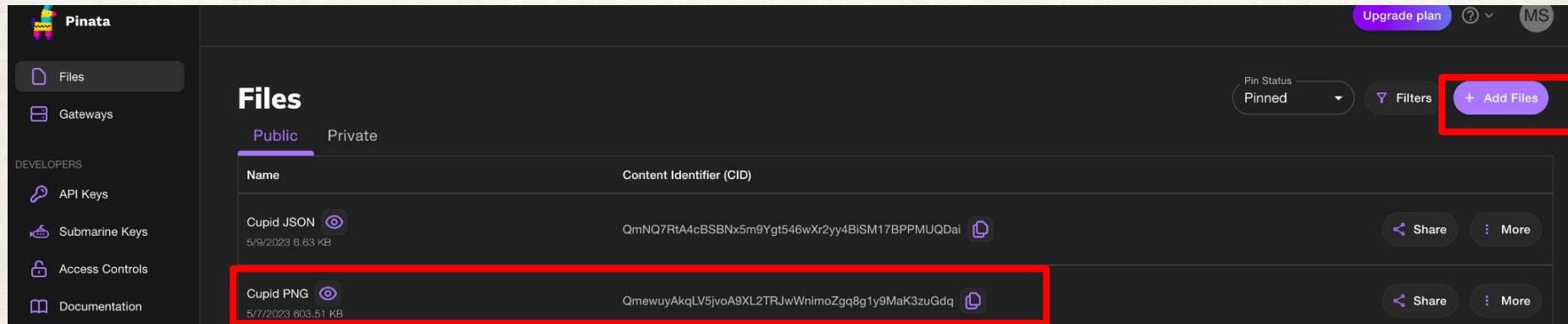
製作圖片

1. 新增資料夾
2. 命名圖片為 0.png



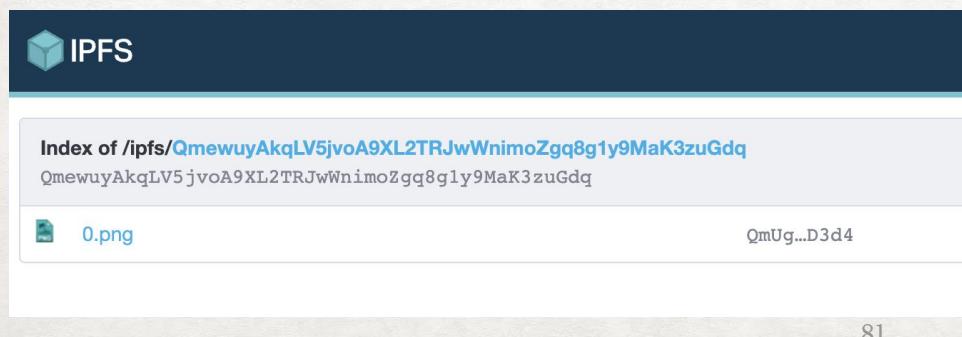
上傳圖片資料夾到 IPFS 取得 BaseURI

上傳資料夾



The screenshot shows the Pinata dashboard interface. On the left, there's a sidebar with options like 'Files', 'Gateways', 'DEVELOPERS', 'API Keys', 'Submarine Keys', 'Access Controls', and 'Documentation'. The main area is titled 'Files' with tabs for 'Public' and 'Private'. It lists two files: 'Cupid JSON' (CID: QmNQ7RtA4cBSBNx5m9Ygt546wXr2yy4BiSM17BPPMUQDai) and 'Cupid PNG' (CID: QmewuyAkqLV5jvoA9XL2TRJwWnimoZgq8g1y9MaK3zuGdq). Both files have a 'Share' and 'More' button. In the top right, there are buttons for 'Pin Status' (set to 'Pinned'), 'Filters', and a prominent purple button labeled '+ Add Files' which is highlighted with a red box.

圖片資料夾 CID (Content IDentifier)



The screenshot shows the IPFS index page for the CID QmewuyAkqLV5jvoA9XL2TRJwWnimoZgq8g1y9MaK3zuGdq. It displays the directory structure: 'Index of /ipfs/QmewuyAkqLV5jvoA9XL2TRJwWnimoZgq8g1y9MaK3zuGdq'. Below it, there's a single file entry: '0.png' with a 'BaseURI' of 'QmUG...D3d4'. There's also a small preview thumbnail of the file.

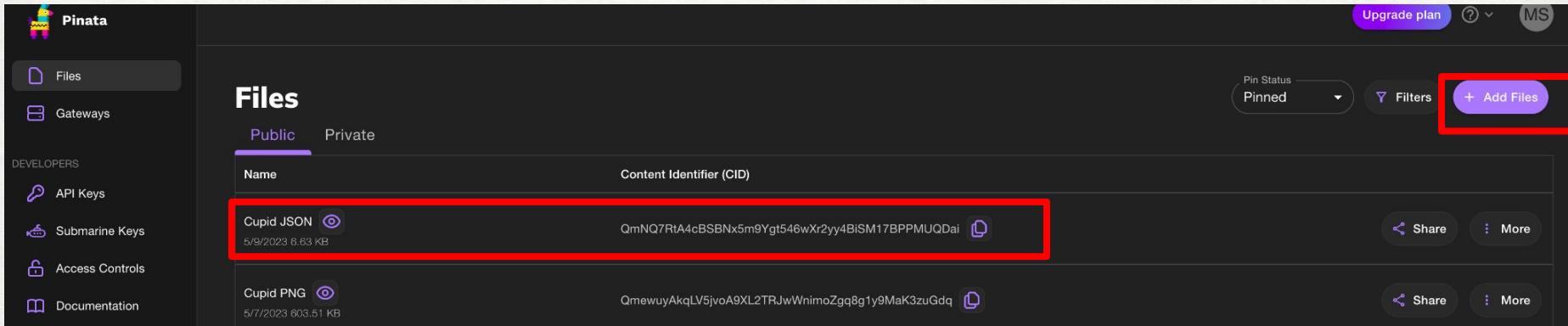
製作 Metadata “0.json” 檔案

```
1.  {
2.    "name": "Cupid",
3.    "image": "ipfs://QmewuyAkqLV5jvoA9XL2TRJwWnimoZgq8g1y9MaK3zuGdq/0.png",
4.    "description": "This NFT collection is created by Cupid",
5.    "external_url": "https://linktr.ee/siemingfong",
6.    "attributes": [
7.      {
8.        "trait_type": "Name",
9.        "value": "Cupid"
10.       },
11.       {
12.         "trait_type": "creator",
13.         "value": "Cupid"
14.       }
15.     ]
16.   }
```

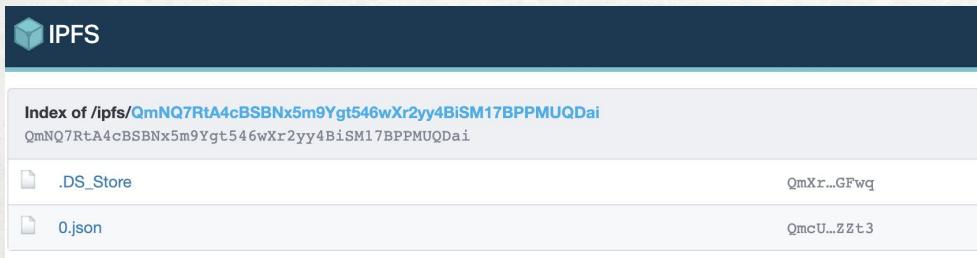
<https://forum.openzeppelin.com/t/create-an-erc1155/4433>

上傳 JSON 資料夾到 IPFS 取得 BaseURI

上傳資料夾



The screenshot shows the Pinata dashboard interface. On the left, there's a sidebar with links for 'Files', 'Gateways', 'API Keys', 'Submarine Keys', 'Access Controls', and 'Documentation'. The main area is titled 'Files' and shows two entries: 'Cupid JSON' and 'Cupid PNG'. 'Cupid JSON' is highlighted with a red box. It has a timestamp of '5/9/2023 6.63 KB' and a Content Identifier (CID) of 'QmNQ7RtA4cBSBNx5m9Ygt546wXr2yy4BiSM17BPPMUQDai'. To the right of the CID is a download icon. Below the CID, there are 'Share' and 'More' buttons. The 'Cupid PNG' entry has a timestamp of '5/7/2023 603.51 KB' and a CID of 'QmewuyAkqLV5jvoA9XL2TRJwWnimoZgq8gTy9MaK3zuGdq'. It also has a download icon and 'Share'/'More' buttons. At the top right of the dashboard, there are buttons for 'Upgrade plan', 'Pin Status' (set to 'Pinned'), 'Filters', and a prominent purple 'Add Files' button, which is also highlighted with a red box.



The screenshot shows the IPFS browser interface. The address bar indicates the URL is 'Index of /ipfs/QmNQ7RtA4cBSBNx5m9Ygt546wXr2yy4BiSM17BPPMUQDai'. The page lists two files: '.DS_Store' and '0.json'. The '.DS_Store' file has a CID of 'QmXr...GFwq' and the '0.json' file has a CID of 'QmcU...zzt3'.

JSON資料夾 [CID](#) (Content IDentifier)



OpenZippelin Contract Wizard

Base URI = IPFS Json CID

ipfs://QmNQ7RtA4cBSBNx5m9Ygt546
wXr2yy4BiSM17BPPMUQDai

ERC20 ERC721 ERC1155 Governor Custom [Copy to Clipboard](#) [Open in Remix](#) [Download](#)

SETTINGS

Name: Cupid Symbol: CUPID

Base URI: ipfs://QmNQ7RtA4cBSBNx5m9Ygt546wXr2yy4BiSM17BPPMUQDai

FEATURES

Mintable [?](#)
 Auto Increment Ids [?](#)
 Burnable [?](#)
 Pausable [?](#)
 Votes [?](#)
 Enumerable [?](#)
 URI Storage [?](#)

ACCESS CONTROL [?](#)

Ownable [?](#)
 Roles [?](#)

UPGRADEABILITY [?](#)

Transparent [?](#)
 UUPS [?](#)

```
/ SPDX-License-Identifier: MIT
pragma solidity ^0.8.9;

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

contract Cupid is ERC721, Ownable {
    using Counters for Counters.Counter;

    Counters.Counter private _tokenIdCounter;

    constructor() ERC721("Cupid", "CUPID") {}

    function _baseURI() internal pure override returns (string memory)
    {
        return "ipfs://QmNQ7RtA4cBSBNx5m9Ygt546wXr2yy4BiSM17BPPMUQDai
    }

    function safeMint(address to) public onlyOwner {
        uint256 tokenId = _tokenIdCounter.current();
        _tokenIdCounter.increment();
        _safeMint(to, tokenId);
    }
}
```

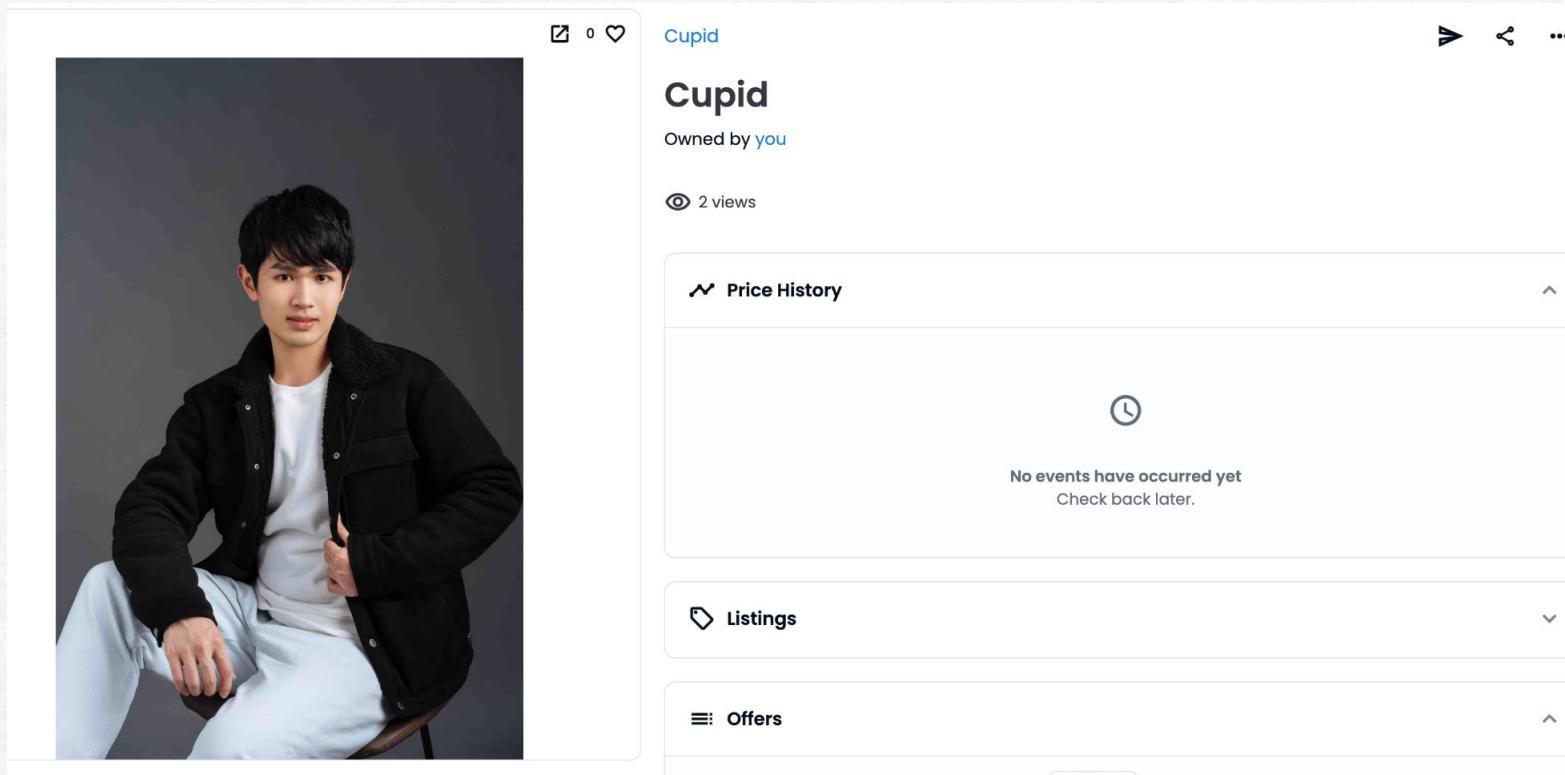
撰寫 Cupid ERC 721 合約

```
1. // SPDX-License-Identifier: MIT
2. pragma solidity ^0.8.9;
3.
4. import "@openzeppelin/contracts@4.8.3/token/ERC721/ERC721.sol";
5. import "@openzeppelin/contracts@4.8.3/access/Ownable.sol";
6. import "@openzeppelin/contracts@4.8.3/utils/Counters.sol";
7.
8. contract Cupid is ERC721, Ownable {
9.     using Counters for Counters.Counter;
10.
11.    Counters.Counter private _tokenIdCounter;
12.
13.    uint256 MAX_SUPPLY = 30;
14.    string baseURI = "ipfs://QmNQ7RtA4cBSBNx5m9Ygt546wXr2yy4BiSM17BPPMUQDai/";
15.    string baseExtension = ".json";
16.
17.    mapping (address => bool) public mintedWallets;
18.
19.    constructor() ERC721("Cupid", "CUPID") {
20.
21.        function tokenURI(uint256 tokenId) public view virtual override returns (string memory){
22.            require(
23.                _exists(tokenId),
24.                "ERC721Metadata: URI query for nonexistent token"
25.            );
26.            return string(abi.encodePacked(baseURI, Strings.toString(tokenId), baseExtension));
27.        }
28.        function safeMint(address to) public onlyOwner {
29.            uint256 tokenId = _tokenIdCounter.current();
30.            _tokenIdCounter.increment();
31.            _safeMint(to, tokenId);
32.        }
33.    }
```

JSON CID

把整個合併變成代幣的 URI
baseURI/ + 代幣標號 + .json

檢查 Opensea Testnet



A screenshot of an NFT listing on the Opensea Testnet. The listing is for an NFT named "Cupid", which is owned by the user. The image shows a young man with dark hair, wearing a black jacket over a white t-shirt and light-colored pants, sitting on a chair. The listing interface includes sections for "Price History" (which shows no events), "Listings" (empty), and "Offers" (empty). There are also buttons for favoriting and sharing the listing.

0 views

Cupid

Owned by you

2 views

Price History

No events have occurred yet
Check back later.

Listings

Offers

<https://testnets.opensea.io/assets/mumbai/0x1ba18df26a694289f85bd466f58ceca7b33c57c4/0>

Q&A

推薦學習

- Solidity 極簡入門 [WTF Solidity](#)
- 建造鐵人賽專屬NFT！[\(五\) 設定圖片/名稱Metadata並上傳至IPFS！](#)