

(有多傳一份 pdf 檔，如果版面亂掉請老師直接看 pdf 檔案)

1. 請解釋什麼是區塊鏈:

是一個利用點對點網路建立的交易，所以無需第三方信任單位即可對交易進行檢驗。

區塊鏈將每一次的虛擬貨幣的交易放入區塊當中，當完成產生區塊所需要的計算過程之後，將區塊與其他區塊串接一起的，成為區塊鏈

(Blockchain) 一部分。因此區塊鏈就是用來儲存比特幣交易紀錄的帳本，同時也是防衛竄改交易紀錄的防衛系統。

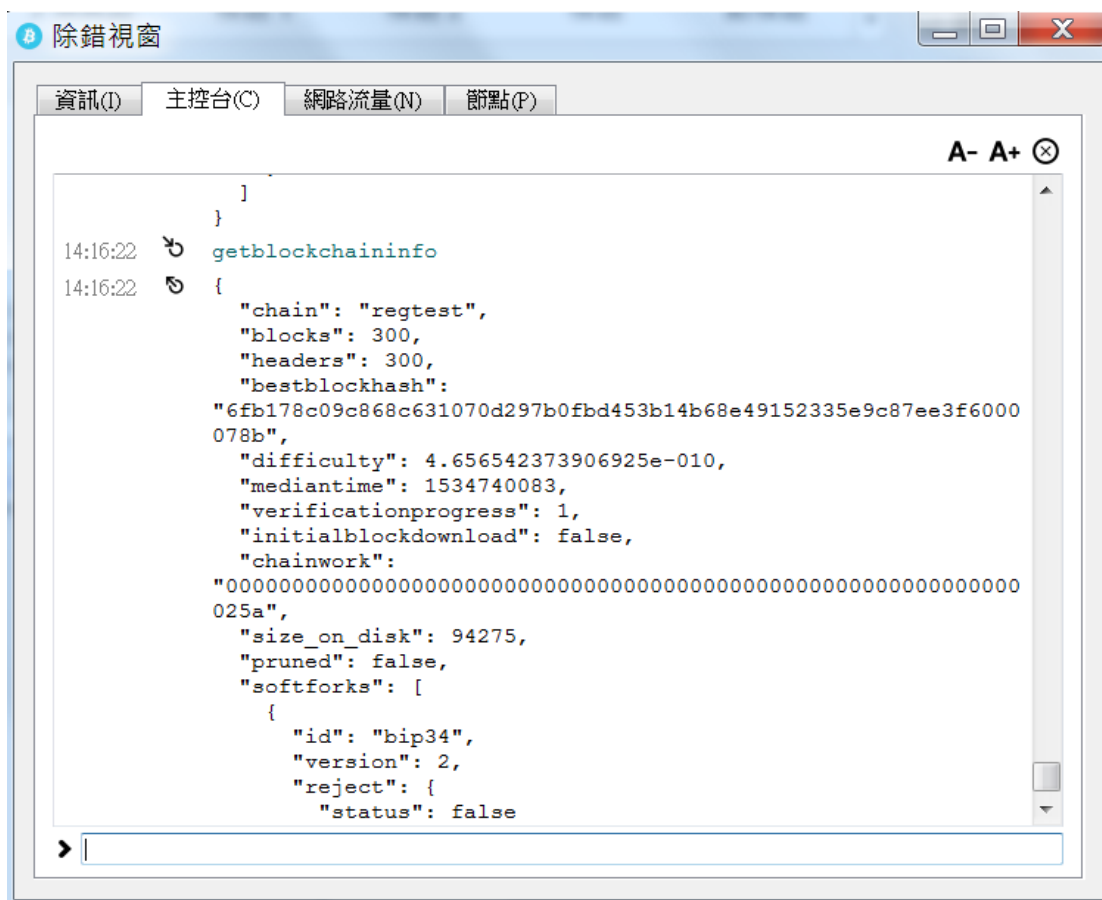
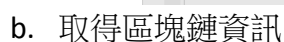
2. 請解釋區塊鏈的交易為何可以避免被竄改:

因為區塊鏈交易會將每次的交易記錄在區塊上，而每筆交易後都會產生一個僅限當筆交易的亂數碼，利用這筆亂碼即可判斷交易資料有沒有經過竄改。

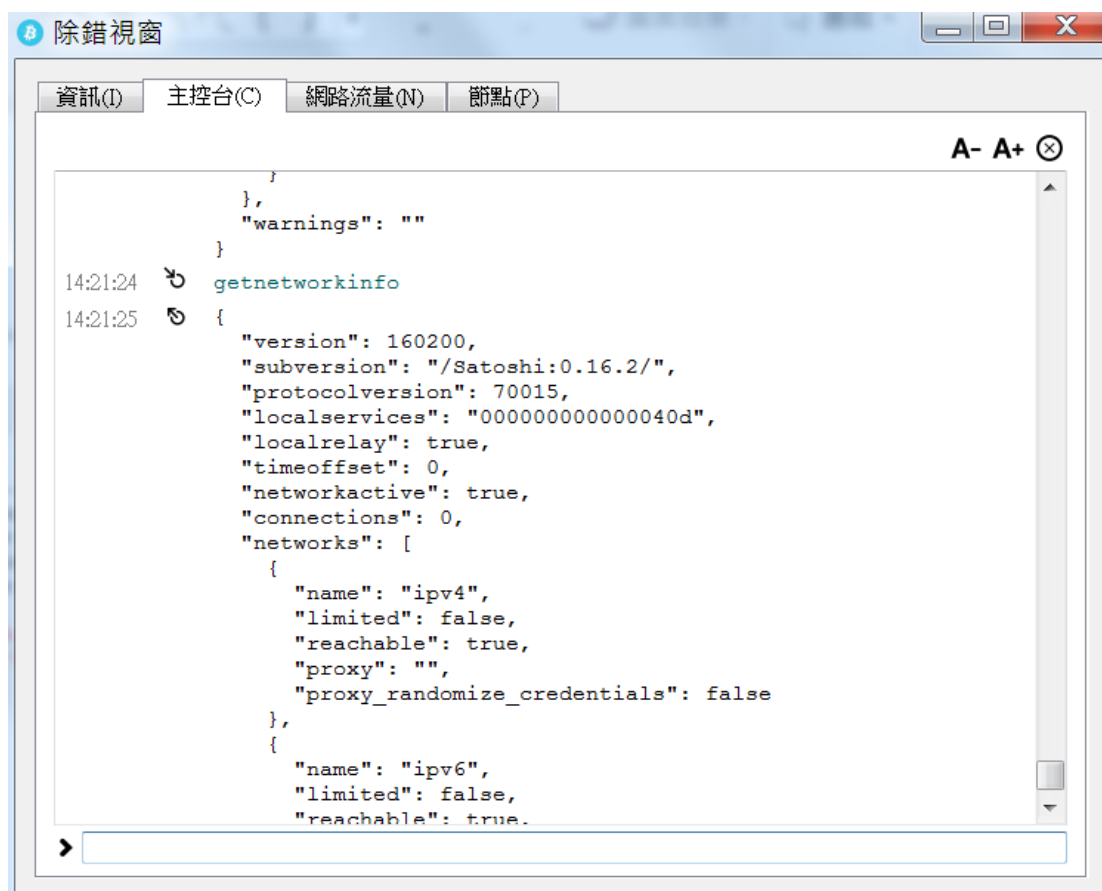
3. 請比較比特幣與以太坊手續費的差異:

比特幣交易由越高手續費者越快速完成交易，以太坊則根據交易價格調整手續費，因此比特幣只要花大錢提高手續費就可以享有不壅擠的網路交易速度，而相反以太坊無法提高手續費，就有可能因為過多垃圾交易造成交易堵塞。

a. 用 regtest 模式進入比特幣錢包



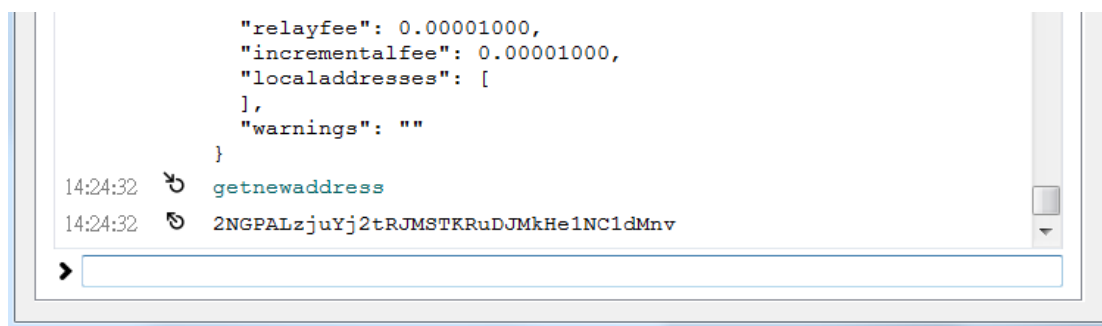
c. 取得區塊鏈網路資訊及發送交易費用



The screenshot shows the Bitcoin console window titled "除錯視窗" (Debug Window). It has tabs for "資訊(I)" (Info), "主控台(C)" (Console), "網路流量(N)" (Network Traffic), and "節點(P)" (Peers). The "主控台(C)" tab is active. The console shows a command prompt with the command `getnetworkinfo` entered at 14:21:25. The output is a JSON object containing network information. The window also has a search bar at the top right with "A- A+ X" and a scroll bar on the right.

```
14:21:24 > getnetworkinfo
14:21:25 {
  "version": 160200,
  "subversion": "/Satoshi:0.16.2/",
  "protocolversion": 70015,
  "localservices": "0000000000000040d",
  "localrelay": true,
  "timeoffset": 0,
  "networkactive": true,
  "connections": 0,
  "networks": [
    {
      "name": "ipv4",
      "limited": false,
      "reachable": true,
      "proxy": "",
      "proxy_randomize_credentials": false
    },
    {
      "name": "ipv6",
      "limited": false,
      "reachable": true
    }
  ]
}
```

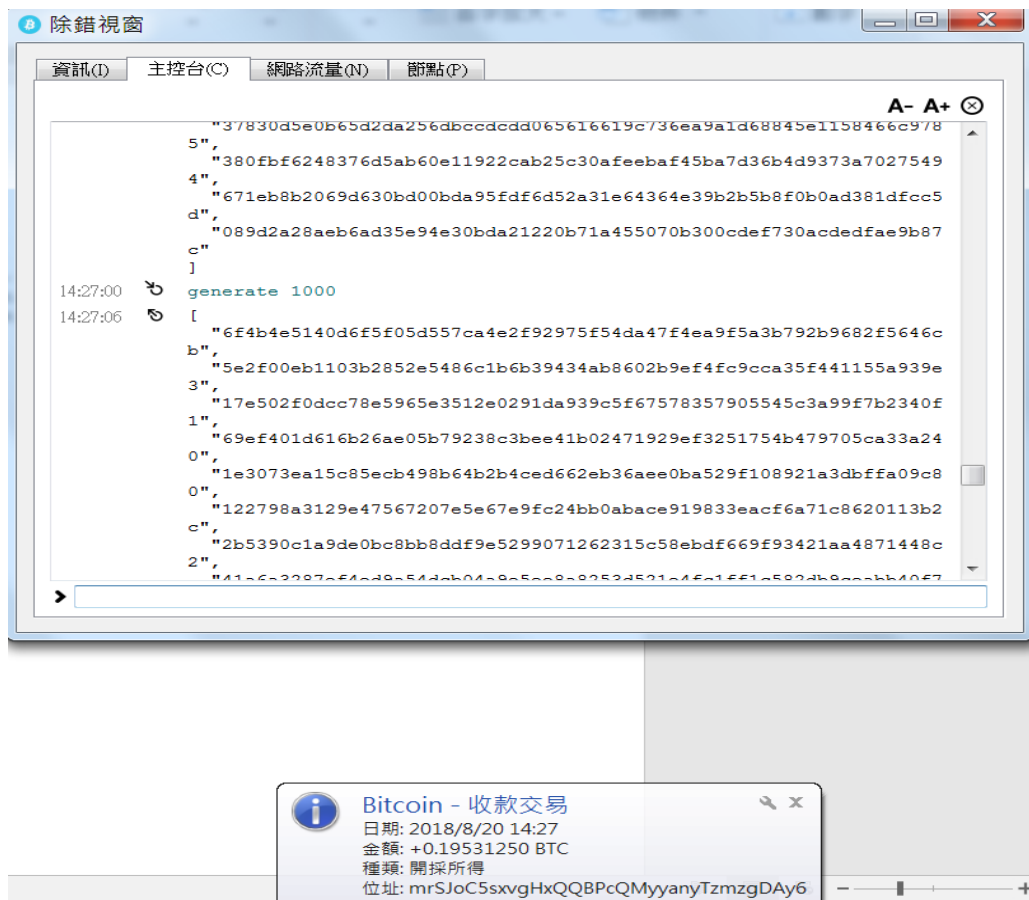
d. 取得新位址



The screenshot shows the Bitcoin console window with the "主控台(C)" tab active. The console shows a command prompt with the command `getnewaddress` entered at 14:24:32. The output is a new Bitcoin address: `2NGPALzjuYj2tRJMSTKRuDJMkHe1NC1dMnv`. The window also has a search bar at the top right with "A- A+ X" and a scroll bar on the right.

```
14:24:32 > getnewaddress
14:24:32 2NGPALzjuYj2tRJMSTKRuDJMkHe1NC1dMnv
```

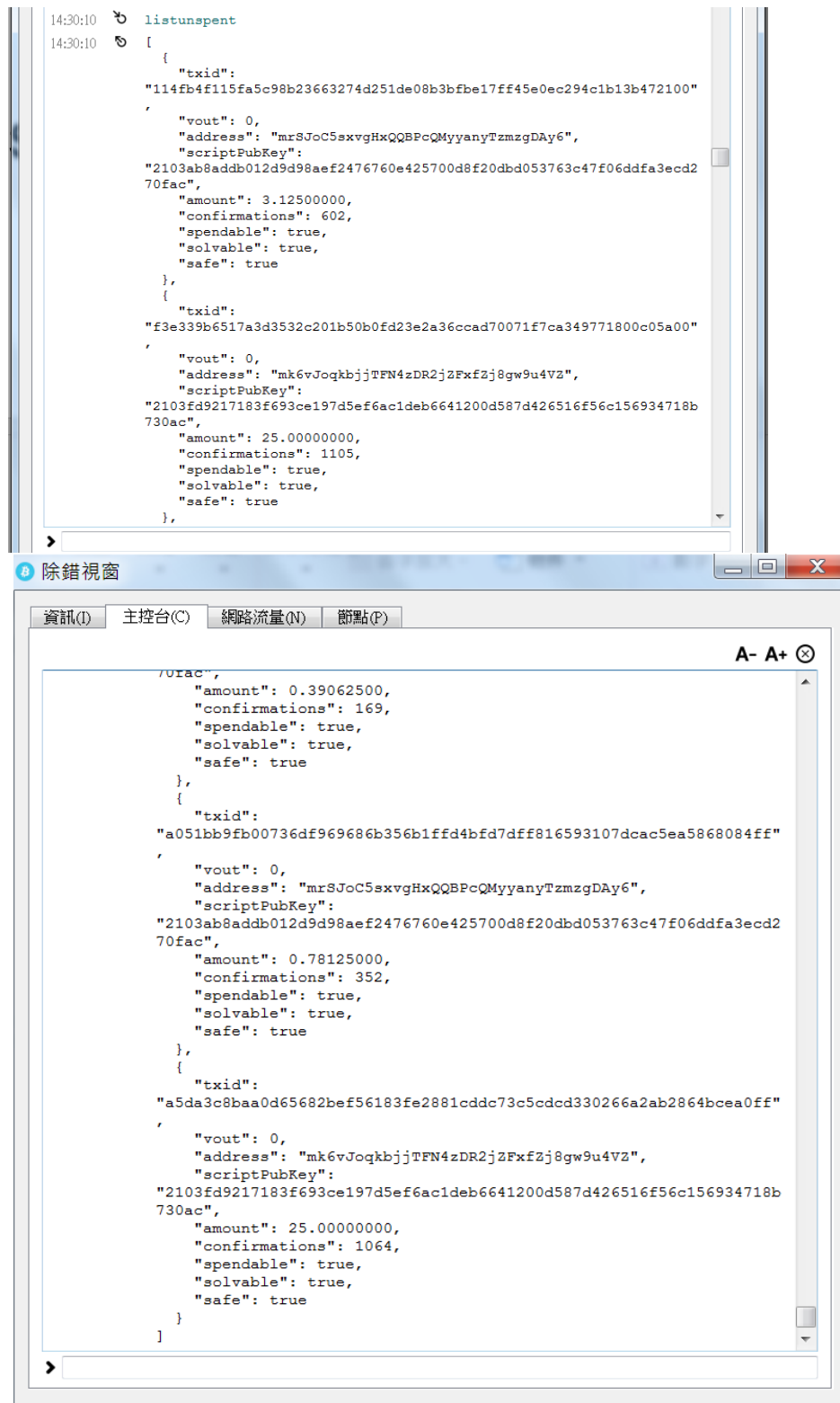
e. 獲取錢幣



f. 轉給自己 10 塊



g. 列出交易紀錄



```
14:30:10 listunspent
14:30:10 [
  {
    "txid":
    "114fb4f115fa5c98b23663274d251de08b3bfbe17ff45e0ec294c1b13b472100"
    ,
    "vout": 0,
    "address": "mrSJoC5xvvgHxQQBPcQMyanyTzmzgDAy6",
    "scriptPubKey":
    "2103ab8adb012d9d98aef2476760e425700d8f20dbd053763c47f06ddfa3ecd2
    70fac",
    "amount": 3.12500000,
    "confirmations": 602,
    "spendable": true,
    "solvable": true,
    "safe": true
  },
  {
    "txid":
    "f3e339b6517a3d3532c201b50b0fd23e2a36ccad70071f7ca349771800c05a00"
    ,
    "vout": 0,
    "address": "mk6vJoqkbjjTFN4zDR2j2FxfZj8gw9u4VZ",
    "scriptPubKey":
    "2103fd9217183f693ce197d5ef6ac1deb6641200d587d426516f56c156934718b
    730ac",
    "amount": 25.00000000,
    "confirmations": 1105,
    "spendable": true,
    "solvable": true,
    "safe": true
  }
],
>
```

除錯視窗

資訊(I) 主控台(C) 網路流量(N) 節點(P)

A- A+ ⊗

```

    "amount": 0.39062500,
    "confirmations": 169,
    "spendable": true,
    "solvable": true,
    "safe": true
  },
  {
    "txid":
    "a051bb9fb00736df969686b356b1ffd4bfd7dff816593107dcac5ea5868084ff"
    ,
    "vout": 0,
    "address": "mrSJoC5xvvgHxQQBPcQMyanyTzmzgDAy6",
    "scriptPubKey":
    "2103ab8adb012d9d98aef2476760e425700d8f20dbd053763c47f06ddfa3ecd2
    70fac",
    "amount": 0.78125000,
    "confirmations": 352,
    "spendable": true,
    "solvable": true,
    "safe": true
  },
  {
    "txid":
    "a5da3c8baa0d65682bef56183fe2881cddc73c5cdcd330266a2ab2864bcea0ff"
    ,
    "vout": 0,
    "address": "mk6vJoqkbjjTFN4zDR2j2FxfZj8gw9u4VZ",
    "scriptPubKey":
    "2103fd9217183f693ce197d5ef6ac1deb6641200d587d426516f56c156934718b
    730ac",
    "amount": 25.00000000,
    "confirmations": 1064,
    "spendable": true,
    "solvable": true,
    "safe": true
  }
]
>
```

(因為中間有測試所以有很多筆記錄，只截了最前面和最後面)

5.

```
//引入 bitcoin 模組
const bitcoin = require("bitcoinjs-lib");

//選用 regtest 網路
const regtest = bitcoin.networks.testnet

//https://github.com/bitcoinjs/bip65
const bip65 = require('bip65')

//輸入私鑰，教學用，請勿隨意公開私鑰
var privateKey =
"cQZts3AqUD4UkSkNLpgbyYWFgo67VZA5yi9Dd5UN2wE2Sm96SE6";

//產生公鑰跟私鑰
const keyPair = bitcoin.ECPair.fromWIF(privateKey,regtest);

//產生付款位址
const { address } = bitcoin.payments.p2pkh({ pubkey: keyPair.publicKey });
console.log(bitcoin.payments.p2pkh({ pubkey: keyPair.publicKey }));
const txb = new bitcoin.TransactionBuilder(regtest);

//用 listunspent 取出最後一筆資料的 txid
txb.addInput('a5da3c8baa0d65682bef56183fe2881cddc73c5cdcd330266a2ab2864bcea0ff', 0);

//用 getnewaddress 取得新的位址
txb.addOutput('2NGPALZjuYj2tRJMSTKRuDJMkHe1NC1dMnv', 25)

//交易簽名
txb.sign(0, keyPair); //第一個位置的是上一筆交易中的第一個支出，第二個欄位
是我們的公鑰與私鑰

//取得交易序號
const transaction_01 = txb.build().toHex();
console.log(transaction_01);
```

```
MINGW64:/c/Program Files/Docker Toolbox
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
From https://github.com/shinzn/examTest
   8968fb0..cb728df  master    -> origin/master
Updating 8968fb0..cb728df
Fast-forward
   one.js | 34 ++++++
   1 file changed, 34 insertions(+)
   create mode 100644 one.js
root@66e4a7ee6b18:~/bitcoin/examTest# nodejs one
{ network:
  { messagePrefix: '\u0018Bitcoin Signed Message:\n',
    bech32: 'bc',
    bip32: { public: 76067358, private: 76066276 },
    pubKeyHash: 0,
    scriptHash: 5,
    wif: 128 },
  address: [Getter/Setter],
  hash: [Getter/Setter],
  output: [Getter/Setter],
  pubkey: <Buffer 03 ec 4b 3c dd d8 e4 58 1f d1 df 8a 4f ba 53 60 09 9e 9b c2 22
ab b3 bb a3 f6 29 72 13 fe cb 19 01>,
  signature: [Getter/Setter],
  input: [Getter/Setter],
  witness: [Getter/Setter] }
0200000001ffa0ce4b86b22a6a2630d3dc5c3cc7dd1c88e23f1856ef2b68650daa8b3cdaa5000000
006a4730440220475816bc8ae7506ac7daf0563e8690bb186245c823c844d5f620337c3867967c02
202d663f7b51936ee85eb35188f365d4f0a6d3beaed3a4642c08ba0bc92ab7c1e4012103ec4b3cdd
d8e4581fd1df8a4fba5360099e9bc222abb3bba3f6297213fecb1901fffffffff0119000000000000
0017a914fdcb2e06b32562dc7a2febe814c2bc1f1a2000d38700000000
root@66e4a7ee6b18:~/bitcoin/examTest#
```