



智能合約於 修課歷程之應用

--以政大為例--

Group 2

106306057 資管四 江晨榕

106306062 資管四 郭宇雋

106306083 資管四 黃羽婕

Motivation

1. 確保候課紀錄與證書的真實性，提供企業與學校在徵才時的參考性
2. 免去證書申請與頒發的冗長過程，減少時間成本
3. 減少大量紙本證書的不方便以及不環保



Content

學生入學時，學校會為學生建立新合約，
同時紀錄著學生資料和學籍狀態。
此系統上會紀錄所有修課紀錄及學籍狀態。
學校可透過更新合約增加新的修課紀錄及畢業狀態，
學生也能透過系統檢查修課紀錄及檢驗畢業條件。



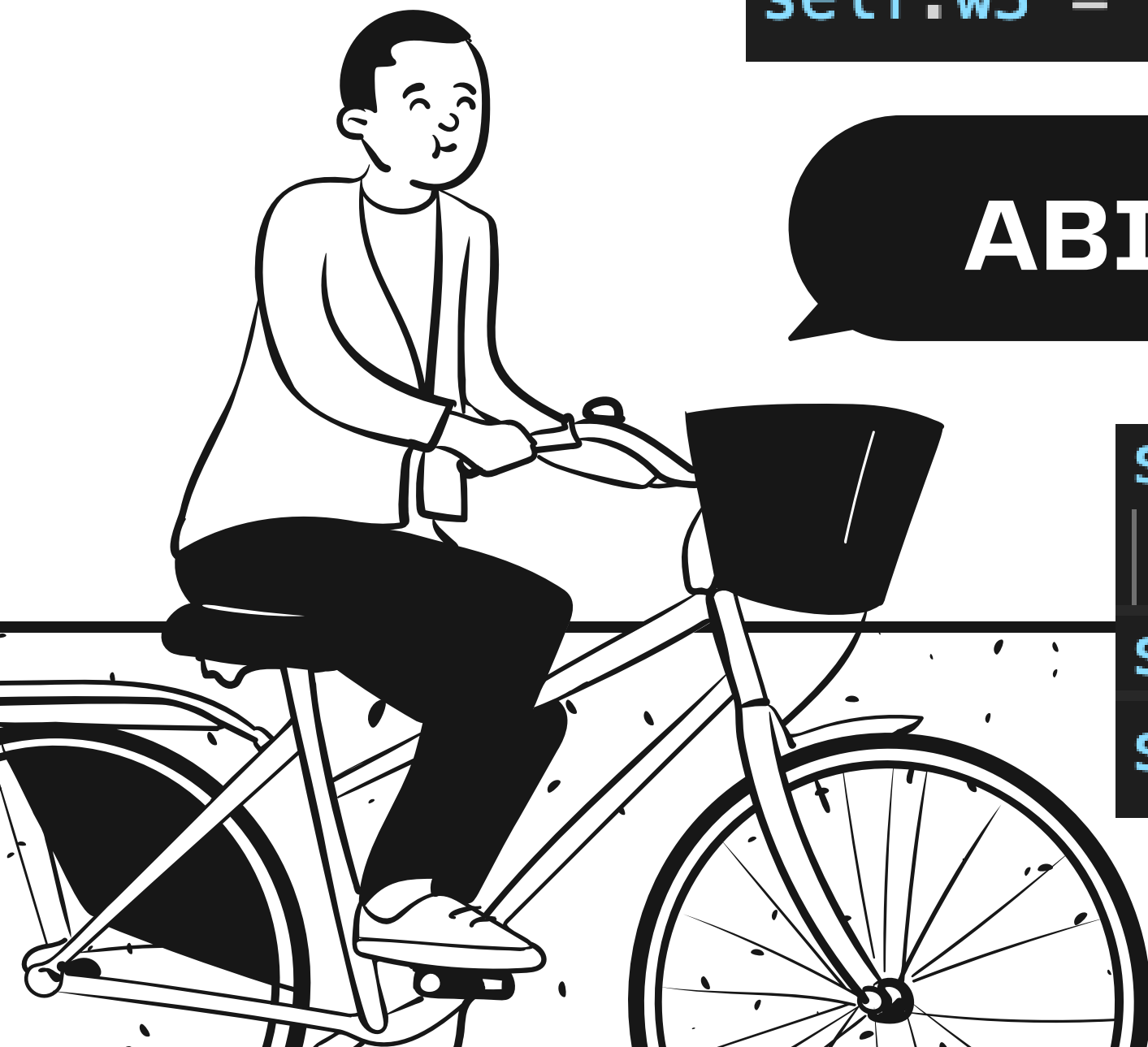
Initial

連到Ethereum node

```
self.w3 = Web3(HTTPProvider("http://localhost:7545"))
```

ABI和Bytecode內容

```
self.TRUFFLE_FILE = json.load(  
    open('./build/contracts/Certificate.json'))  
self.ABI = self.TRUFFLE_FILE['abi']  
self.BYTECODE = self.TRUFFLE_FILE['bytecode']
```



初始化合約及帳戶物件

```
contract = self.w3.eth.contract(bytecode=self.BYTECODE, abi=self.ABI)
```

```
school_account = self.w3.eth.account.from_key(school_key)
```

初始化合約instance object

```
contract_instance = self.w3.eth.contract(
    abi=self.ABI, address=contract_address)
return contract_address, contract_instance
```

建立一個交易

```
construct_txn = contract.constructor(student_name, student_address,
school_name, major, minor, enroll_year).buildTransaction({
    'from': school_account.address,
    'nonce': self.w3.eth.getTransactionCount(school_account.address),
})
```



用私鑰簽署一個交易

```
signed = self.w3.eth.account.sign_transaction(  
    construct_txn, school_account.key)
```

將此合約部署上去並得到其**transaction hash**

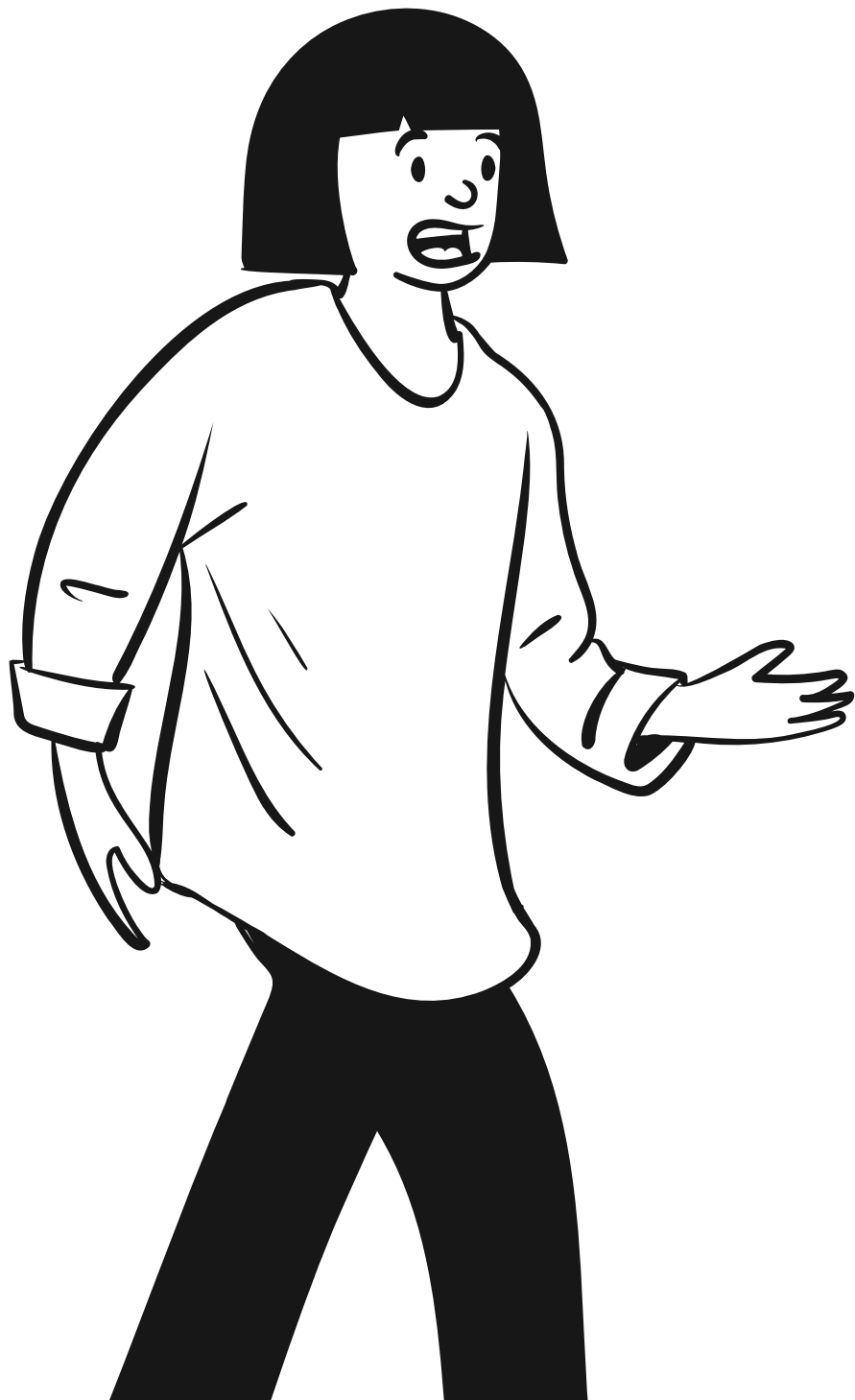
```
tx_hash = self.w3.eth.sendRawTransaction(signed.rawTransaction)  
print(tx_hash.hex())
```

得到其**transaction receipt**

```
tx_receipt = self.w3.eth.waitForTransactionReceipt(tx_hash)  
print("Contract Deployed At:", tx_receipt['contractAddress'])  
contract_address = tx_receipt['contractAddress']
```



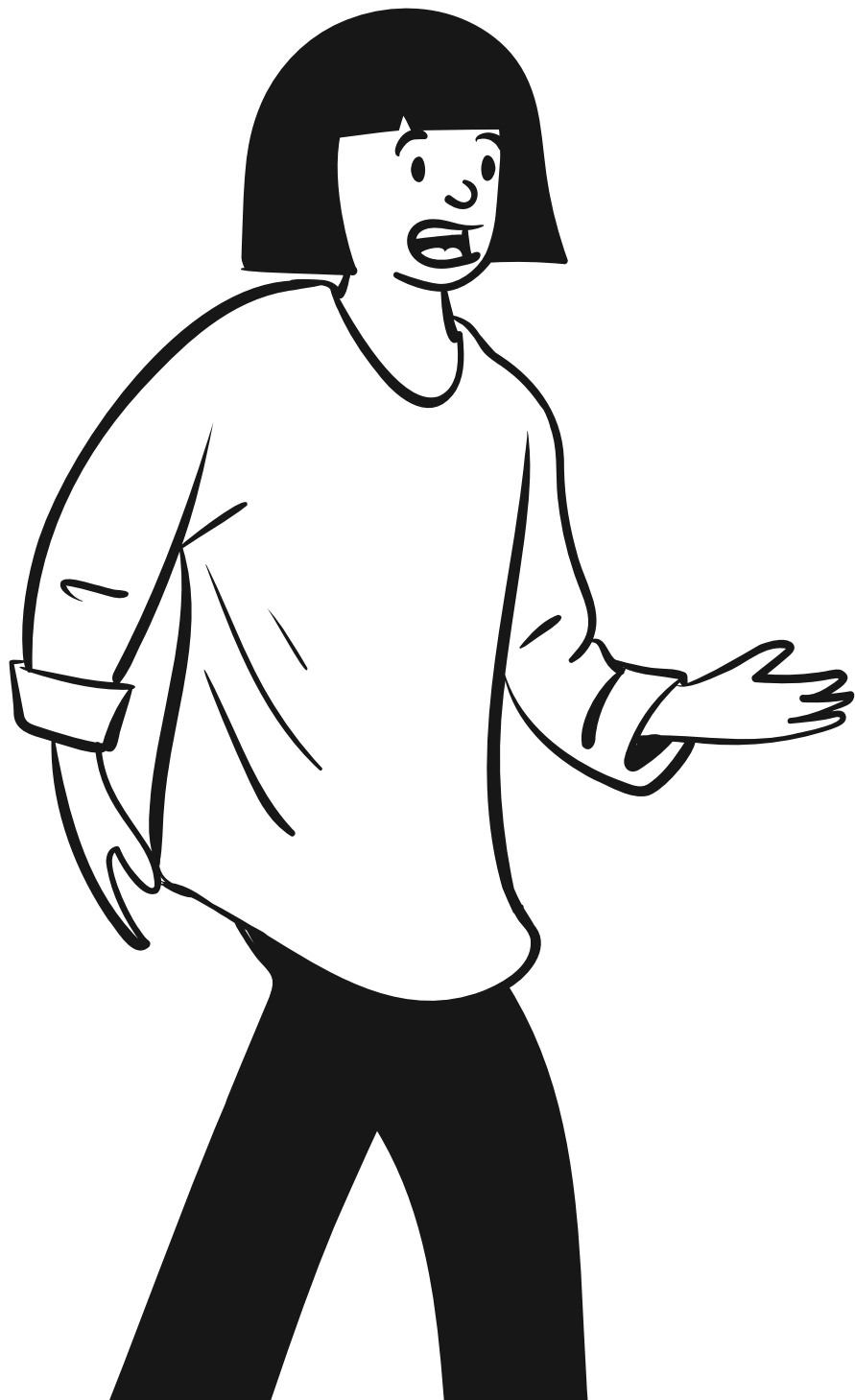
取得所有課程數量



```
function getCourseCount() public view returns(uint) {  
    return courses.length;  
}
```

```
# 查詢某合約中存放的課程數量  
def get_course_count(self, contract_address: str):  
    contract_instance = self.w3.eth.contract(  
        abi=self.ABI, address=contract_address)  
    course_count = contract_instance.functions.getCourseCount().call()  
    return course_count
```


取得某課程資訊



```
function getCource(uint index) public view IndexValidator(index, getCourceCount())
    returns(string memory, string memory, string memory, uint8) {
    Course memory course = courses[index];
    return(course.name, course.content, course.comment, course.grade);
}
```

```
# 查詢某 index 位置的課程資訊
def get_course(self, contract_address: str, index: int):
    contract_instance = self.w3.eth.contract(
        abi=self.ABI, address=contract_address)
    course_info = contract_instance.functions.getCource(index).call()
    return course_info
```


上傳某課程紀錄



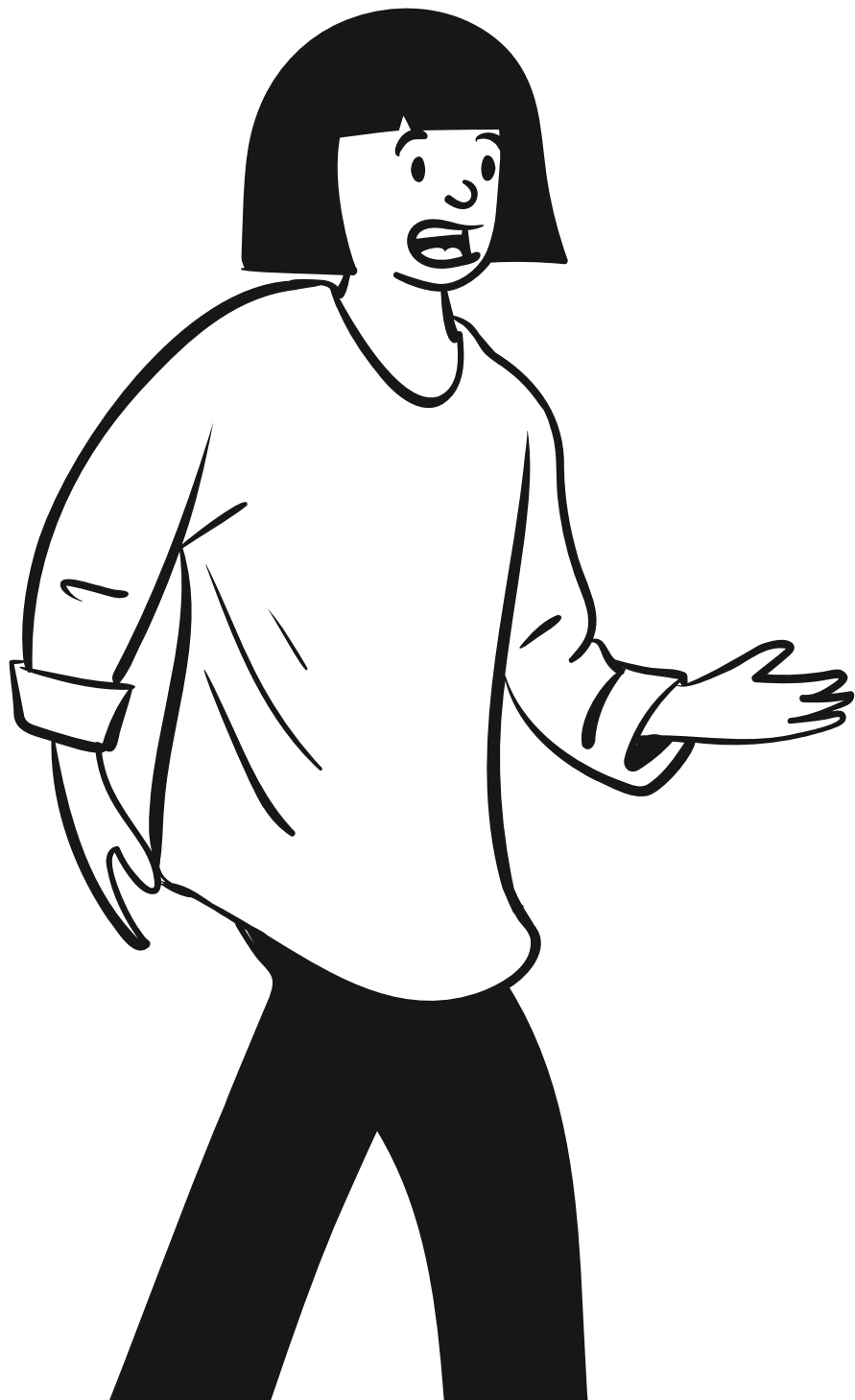
```
function setCourse(string memory name, string memory content, string memory comment, uint8 grade) public checkSchool {  
    courses.push(Course({ name: name, content: content, comment: comment, grade: grade }));  
    emit done(DoneCode.setCourse, "Set Course");  
}
```

```
def set_course(self, school_key: str, contract_address: str, name: str, content: str, comment: str, grade: int):  
    contract_instance = self.w3.eth.contract(...  
    school_account = self.w3.eth.account.from_key(school_key)  
    construct_txn = contract_instance.functions.setCourse(name, content, comment, grade).buildTransaction({  
        'from': school_account.address,  
        'nonce': self.w3.eth.getTransactionCount(school_account.address),  
    })  
    signed = self.w3.eth.account.sign_transaction(  
        construct_txn, school_account.key)  
    tx_hash = self.w3.eth.sendRawTransaction(signed.rawTransaction)  
    return tx_hash.hex()
```

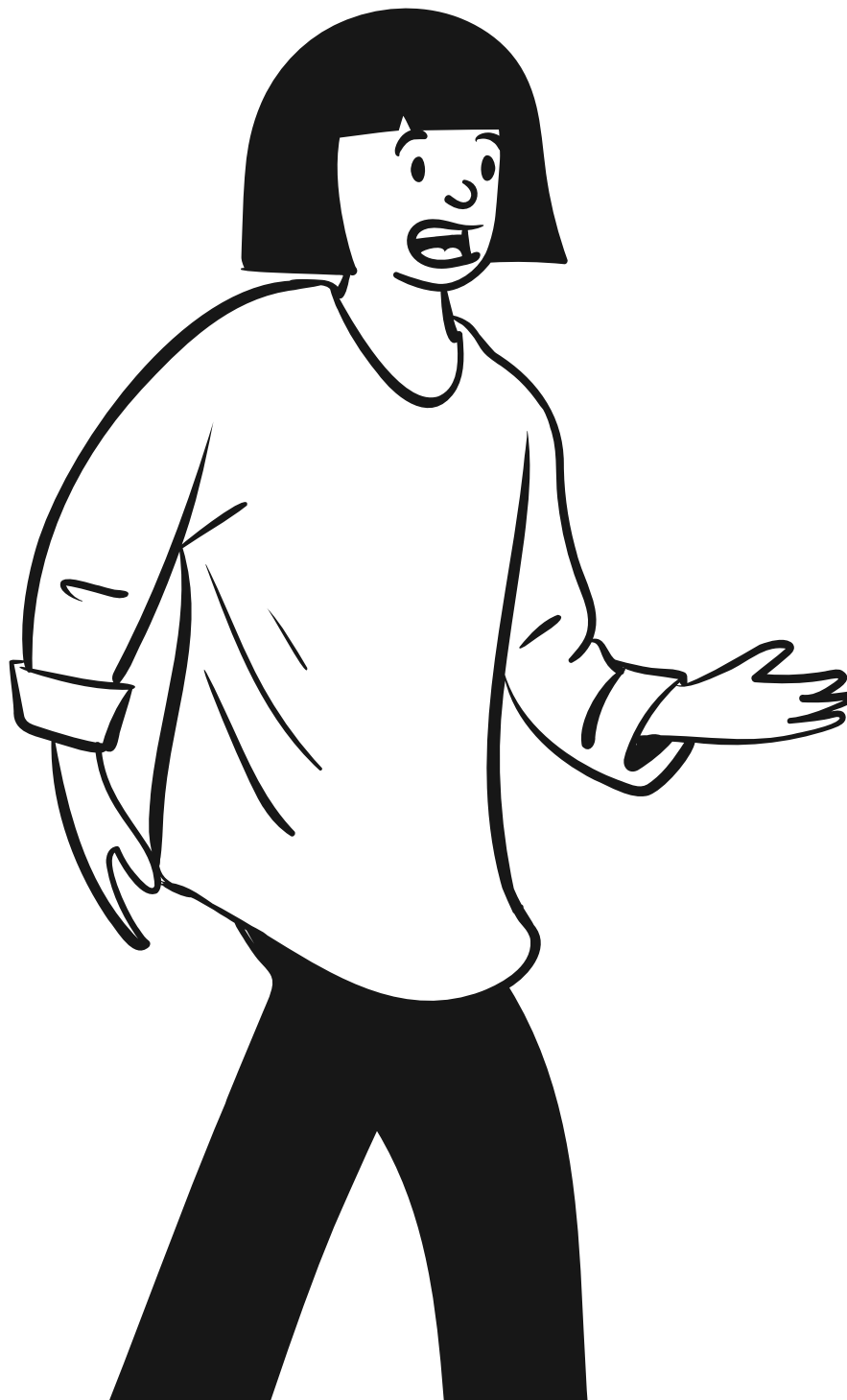
是否滿足證書條件

```
function checkFinishCertificate() public returns(bool){  
    if(courses.length > 2){  
        return true;  
    } else {  
        return false;  
    }  
}
```

```
def check_finish_certificate(self, contract_address: str):  
    contract_instance = self.w3.eth.contract(  
        abi=self.ABI, address=contract_address)  
    check_finish = contract_instance.functions.checkFinishCertificate().call()  
    return check_finish
```



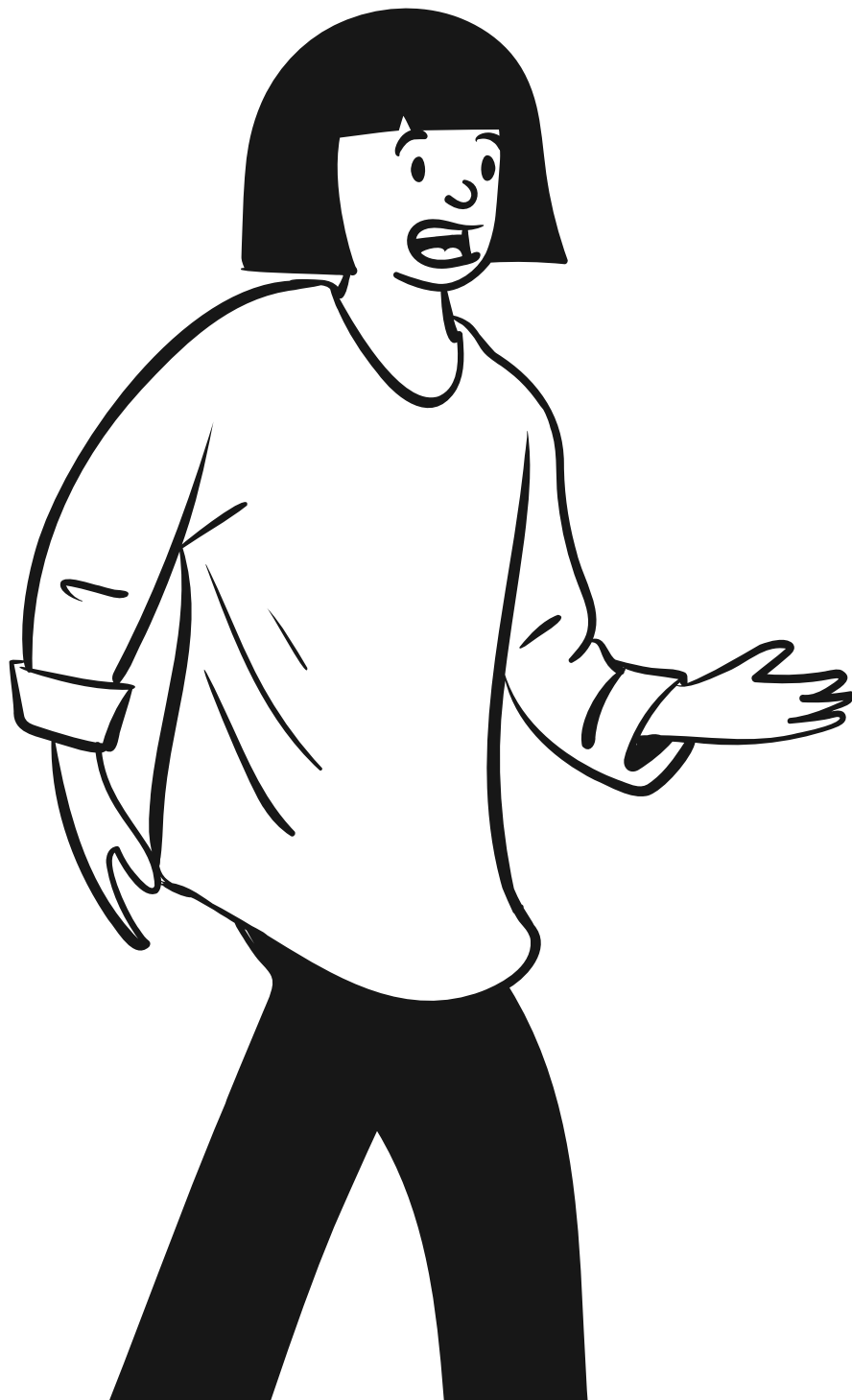
查看目前學籍狀態



```
function getEducationStatus() public view returns(string memory) {  
    if(educationStatus == EducationStatus.undergraduate){  
        return "undergraduate";  
    } else if(educationStatus == EducationStatus.learning){  
        return "learing";  
    } else {  
        return "graduate";  
    }  
}
```

```
def get_education_status(self, contract_address: str):  
    contract_instance = self.w3.eth.contract(  
        abi=self.ABI, address=contract_address)  
    education_status = contract_instance.functions.getEducationStatus().call()  
    return education_status
```

修改學籍狀態（畢業）



```
function setCertificate() public checkFinish{
    educationStatus = EducationStatus.graduate;
    emit done(DoneCode.setCertificate, "Set Certificate");
}
```

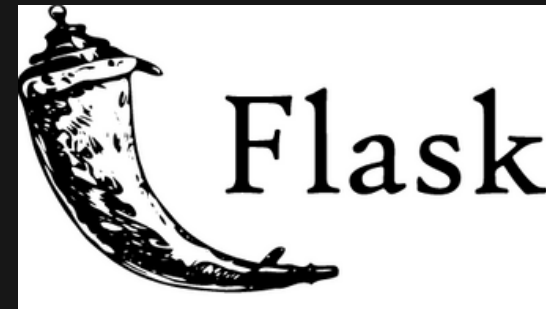
```
def set_certificate(self, student_key: str, contract_address: str):
    contract_instance = self.w3.eth.contract(
        abi=self.ABI, address=contract_address)
    student_account = self.w3.eth.account.from_key(student_key)
    construct_txn = contract_instance.functions.setCertificate().buildTransaction({
        'from': student_account.address,
        'nonce': self.w3.eth.getTransactionCount(student_account.address),
    })
    signed = self.w3.eth.account.sign_transaction(
        construct_txn, student_account.key)
    tx_hash = self.w3.eth.sendRawTransaction(signed.rawTransaction)
    return tx_hash.hex()
```

Tools

合約設計



前端及以太坊點連結



帳戶建立



網頁設計、資料庫



Thank You

