เรียบเรียงและสอนโดย KongRuksiam

Smart Contract

บล็อกเชนและการสร้างสัญญาอัจฉริยะ

Data Type	คำอธิบาย
boolean	ค่าทางตรรกศาสตร์
integer	Int (ตัวเลขจำนวนเต็ม)
	uint (ตัวเลขที่มีเฉพาะจำนวนเต็มค่าบวก)
string	ข้อความ
struct	โครงสร้างข้อมูล
array	ชุดข้อมูล
mapping	<key,value>
address	ใช้เก็บ address ของบัญชีที่สนใจ

Data Type	คำอธิบาย
boolean	ค่าทางตรรกศาสตร์
integer	Int (ตัวเลขจำนวนเต็ม)
	uint (ตัวเลขที่มีเฉพาะจำนวนเต็มค่าบวก)
string	ข้อความ
struct	โครงสร้างข้อมูล
array	ชุดข้อมูล
mapping	<key,value>
address	**ใช้เก็บ address ของบัญชีที่สนใจ**

ตัวแปร (Variable)

ตัวแปร (variable) คือ ชื่อที่ถูกนิยามขึ้นมาเพื่อใช้เก็บค่าข้อมูลสำหรับ นำไปใช้งานใน Smart Contract ประกอบด้วย 3 รูปแบบ

1. State variable (Attribute) คือ ตัวแปรที่เก็บค่าถาวรใน Smart Contract
2. Local variable คือ ตัวแปรที่ทำงานในฟังก์ชั่น
3. Global variable คือ ตัวแปรที่ใช้รับค่าข้อมูลเกี่ยวกับ Blockchain เช่น msg.sender เป็นต้น

ตัวแปร (Variable)

ตัวแปร (variable) คือ ชื่อที่ถูกนิยามขึ้นมาเพื่อใช้เก็บค่าข้อมูล สำหรับ นำไปใช้งานใน Smart Contract ประกอบด้วย 3 รูปแบบ

1. State variable (Attribute) คือ ตัวแปรที่เก็บค่าถาวรใน Smart Contract
2. Local variable คือ ตัวแปรที่ทำงานในฟังก์ชั่น
3. Global variable คือ ตัวแปรที่ใช้รับค่าข้อมูลเกี่ยวกับ Blockchain เช่น msg.sender เป็นต้น

ฟังก์ชั่น (Function)

- Pure : เป็นการแจ้งว่าฟังก์ชันนี้ใช้งานกับค่าคงที่เท่านั้นไม่มีการยุ่งเกี่ยวกับการเปลี่ยนแปลงค่า storage
- View : เป็นการแจ้งว่าฟังก์ชันนี้มีการยุ่งเกี่ยวกับค่าใน storage หรือสามารถอ่านค่าจาก storage ได้เพียงอย่างเดียว
- Payable : เป็นการบ่งบอกว่าฟังก์ชันนี้มีการรับเงินหรือการเรียกเก็บเงิน (Ether) ก่อนจะทำงานในฟังก์ชัน

อาร์เรย์ (Array)

Data Type	คำอธิบาย
boolean	ค่าทางตรรกศาสตร์
integer	Int (ตัวเลขจำนวนเต็ม)
	uint (ตัวเลขที่มีเฉพาะจำนวนเต็มค่าบวก)
string	ข้อความ
struct	โครงสร้างข้อมูล
array	ชุดข้อมูล
mapping	<key,value>
address	ใช้เก็บ address ของบัญชีนั้นๆ

Data Type	คำอธิบาย
boolean	ค่าทางตรรกศาสตร์
integer	Int (ตัวเลขจำนวนเต็ม)
	uint (ตัวเลขที่มีเฉพาะจำนวนเต็มค่าบวก)
string	ข้อความ
struct	โครงสร้างข้อมูล
array	**ชุดข้อมูล**
mapping	<key,value>
address	ใช้เก็บ address ของบัญชีนั้นๆ

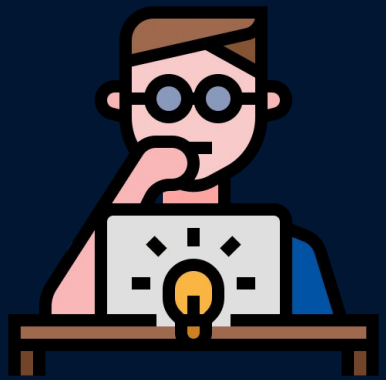
การประกาศตัวแปรแต่ละครั้ง

ตัวแปร 1 ตัวสามารถเก็บข้อมูลได้แค่ 1 ค่าเท่านั้น เช่น

```
uint number = 1;
```

ถ้าอยากเก็บเลข 10 ค่าต้องทำอย่างไร ?

ต้องประกาศตัวแปร 10 ตัวแปร หรือไม่ ?



อาร์เรย์คืออะไร

1. ชุดของตัวแปรที่อยู่ในรูปลำดับใช้เก็บค่าข้อมูลให้อยู่ในกลุ่มเดียวกัน โดยข้อมูลภายในอาร์เรย์จะถูกเก็บในตำแหน่งที่ต่อเนื่องกัน

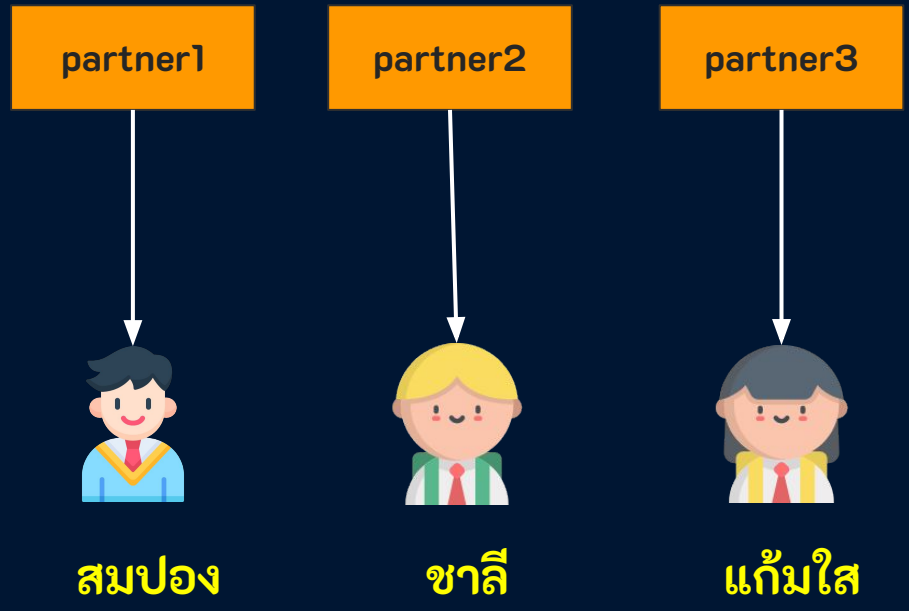
2. เป็นตัวแปรที่ใช้ในการเก็บข้อมูลที่มีลำดับที่ต่อเนื่อง ซึ่งข้อมูลมีค่าได้หลายค่าโดยใช้ชื่ออ้างอิงได้เพียงชื่อเดียว และใช้หมายเลขกำกับ (index) ให้กับตัวแปรเพื่อจำแนกความแตกต่างของค่าตัวแปรแต่ละตัว

คุณสมบัติของอาร์เรย์

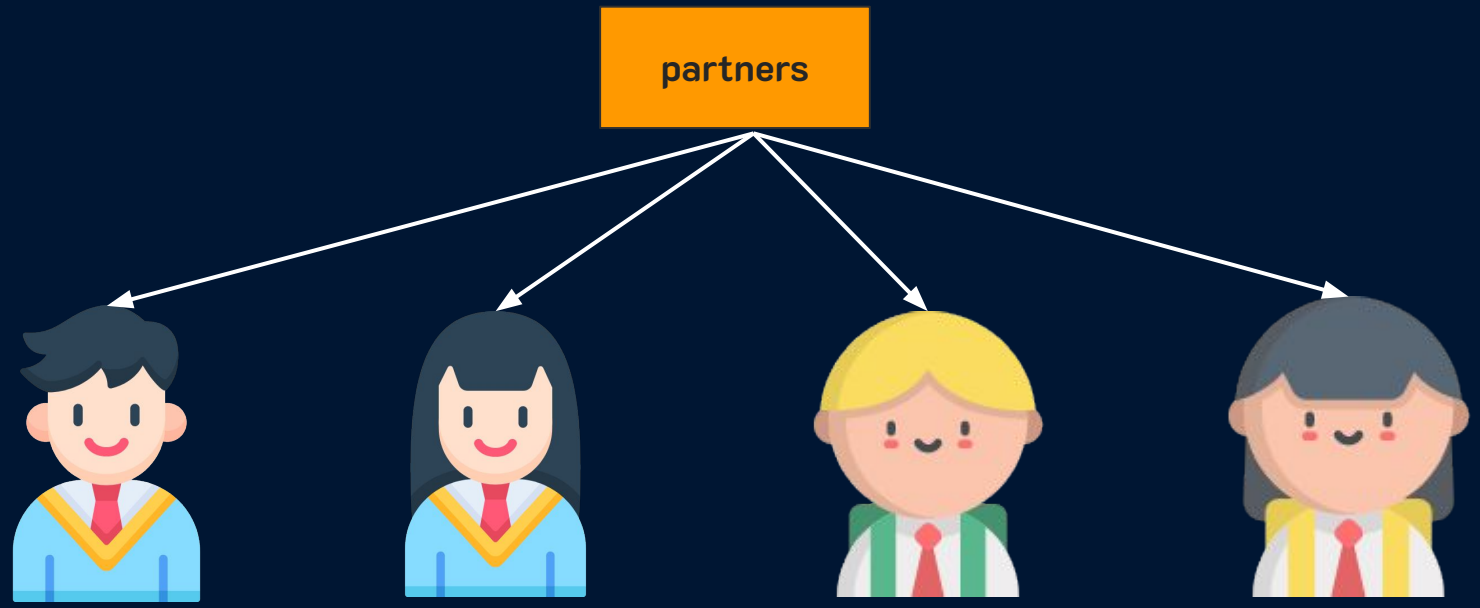
1. ใช้เก็บกลุ่มของข้อมูล
2. ข้อมูลที่อยู่ในอาร์เรย์จะเรียกว่าสมาชิก หรือ อิลิเมนต์ (element)
3. แต่ละอิลิเมนต์ (element) จะเก็บค่าข้อมูล (value) และ อินเด็กซ์ (Index) เอาไว้
4. Index หมายถึงคีย์ของอาร์เรย์ใช้อ้างอิงตำแหน่งของ element **เริ่มต้นที่ 0**
5. สมาชิกใน array ต้องมี**ชนิดข้อมูลเหมือนกัน**
6. สมาชิกใน array จะถูกคั่นด้วยเครื่องหมาย comma

ตัวแปรแบบปกติ

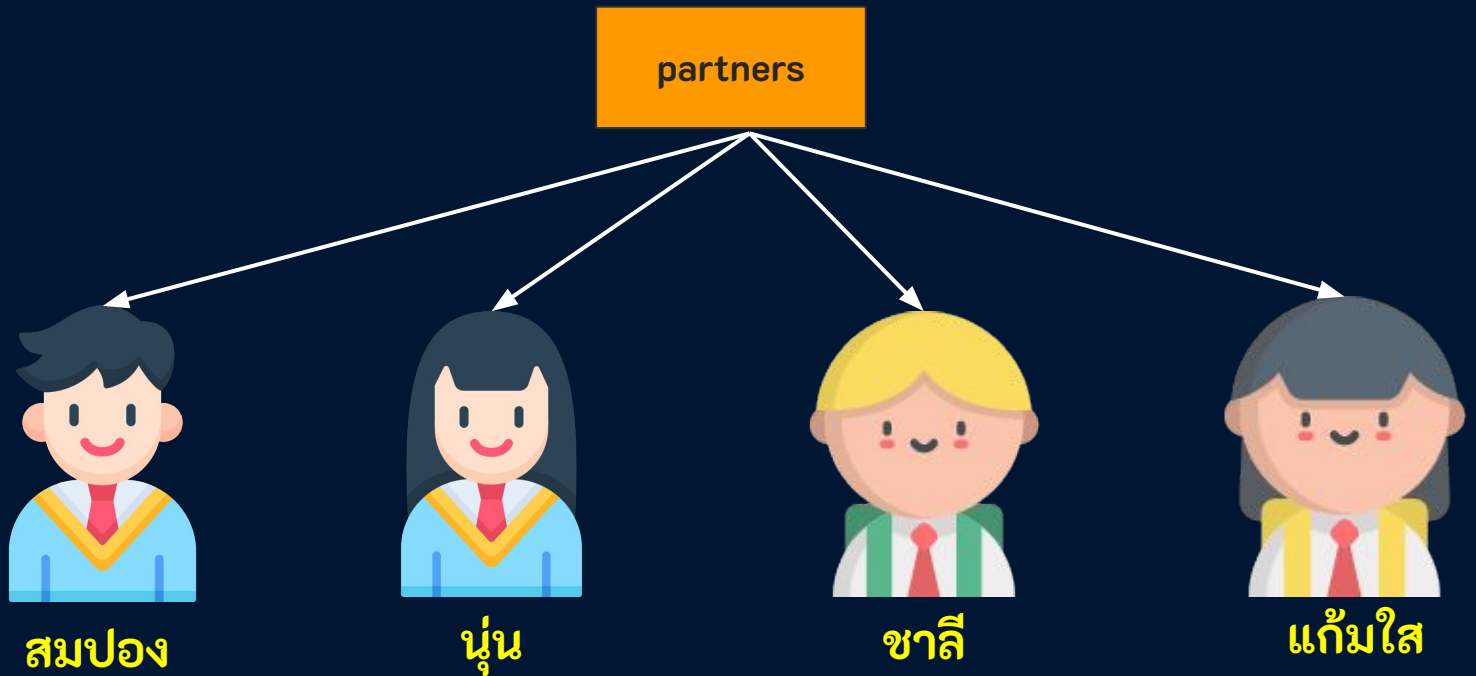
```
string partner1 = “สมปอง”
string partner2 = “ชาลี”
string partner3 = “แก้มใส”
```



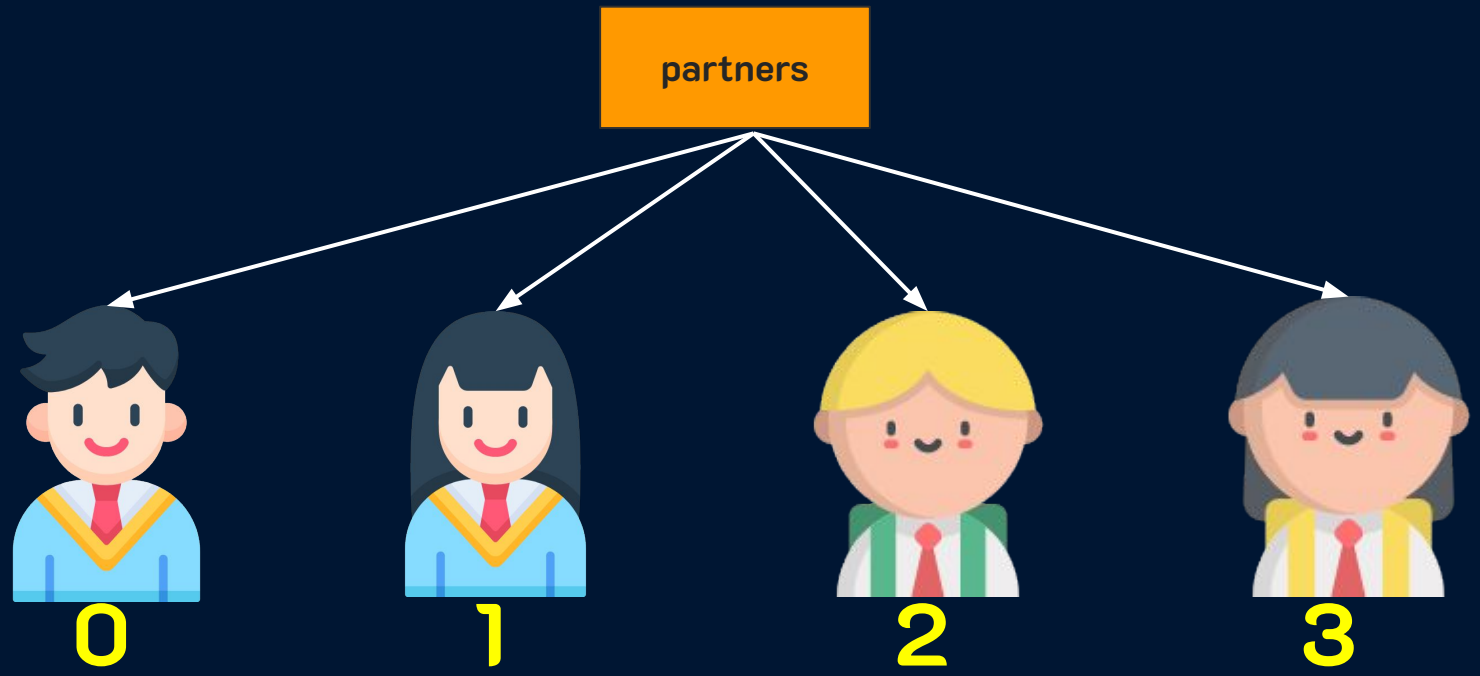
ตัวแปรแบบอาร์เรย์



ตัวแปรแบบอาร์เรย์

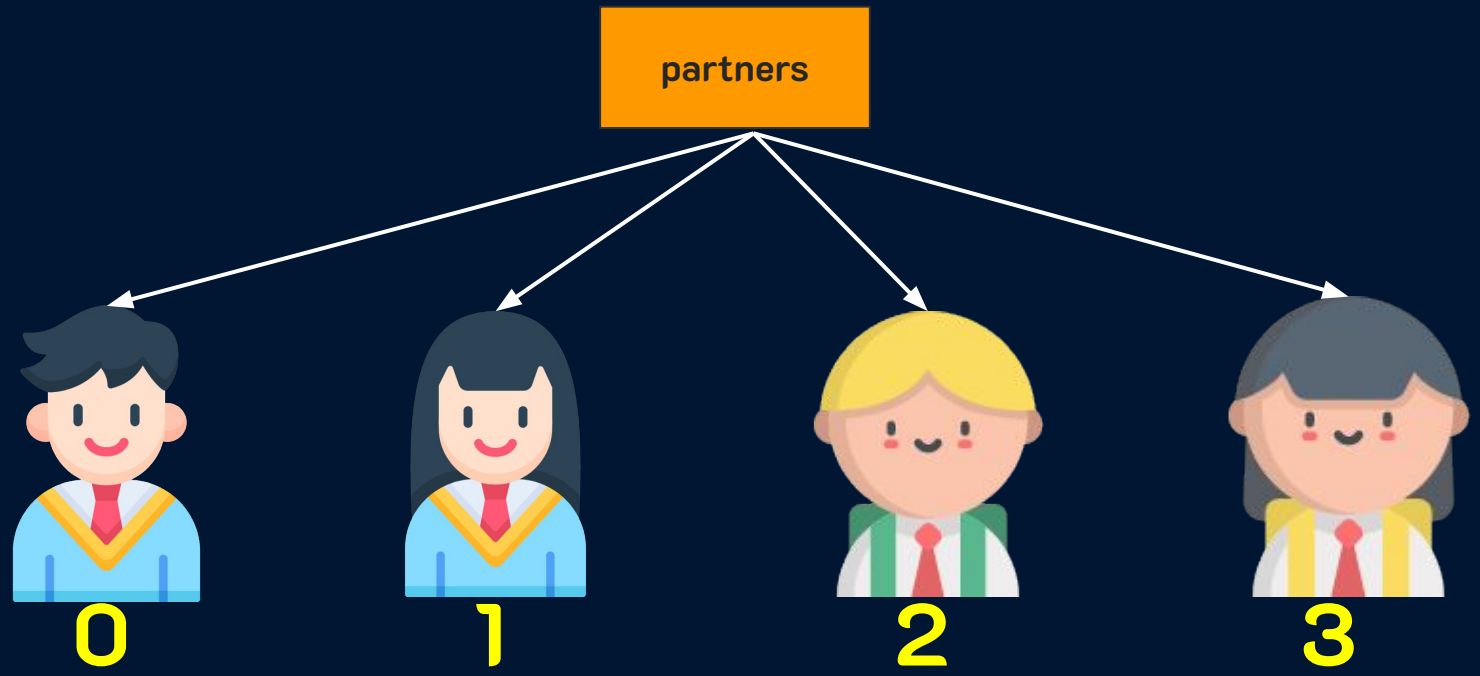


ตัวแปรแบบอาร์เรย์



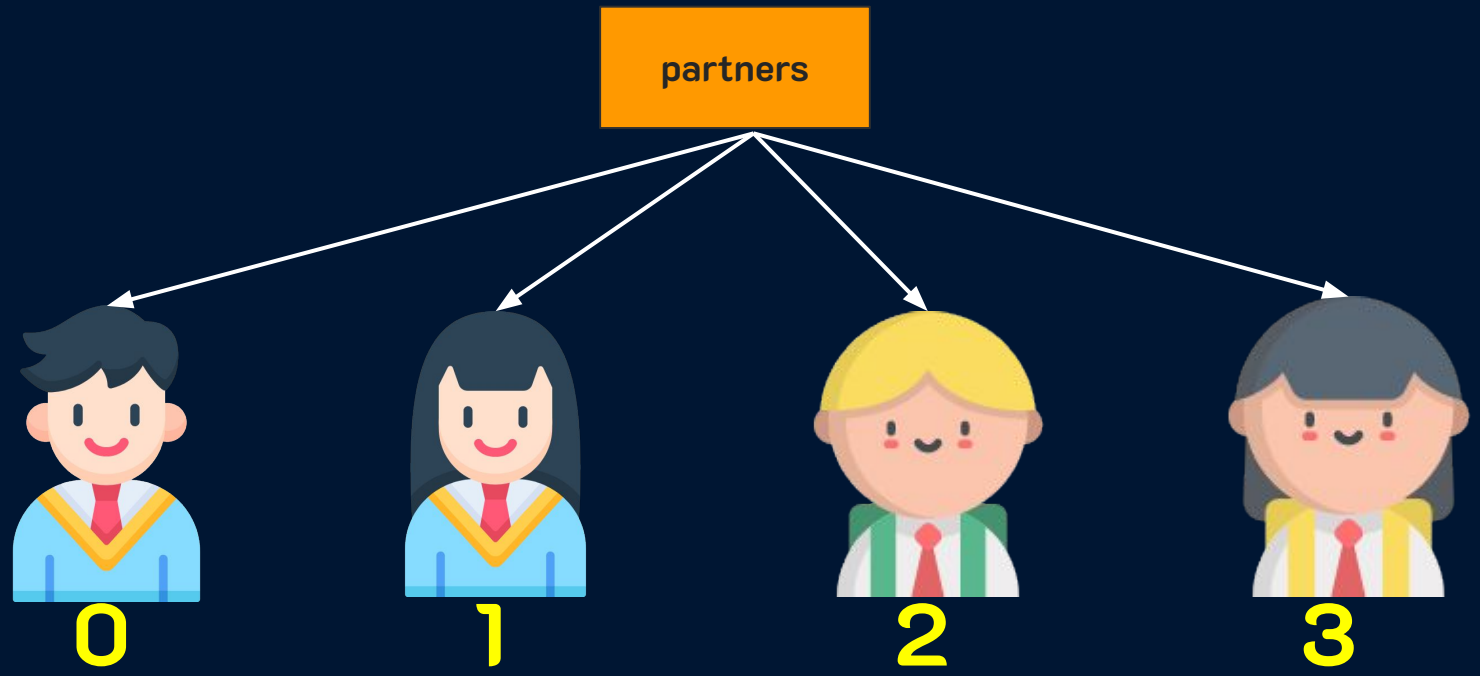
ใช้หมายเลขกำกับอ้างอิงข้อมูล (เป็นลำดับ)

ตัวแปรแบบอาร์เรย์



มีขนาดที่แน่นอน (Fixed Size)

ตัวแปรแบบอาร์เรย์



ใช้จัดเก็บกลุ่มข้อมูลที่มีชนิดข้อมูลเหมือนกัน

สรุปอาร์เรย์

1. ใช้เก็บกลุ่มของข้อมูล **ที่มีชนิดข้อมูลเดียวกัน**
2. ใช้ตัวแปรชื่อเดียวกัน
3. ใช้หมายเลขกำกับเพื่ออ้างอิงตำแหน่งของข้อมูลในอาร์เรย์

รูปแบบอาร์เรย์

1. จำกัดจำนวนสมาชิก (Fixed Size)
2. ไม่จำกัดจำนวนสมาชิก (Dynamic Size)

การสร้างอาร์เรย์แบบกำหนดขนาดเริ่มต้น

int[4] number = [10, 20, 30, 40];

string [2] partners = [“สมชาย”,”แก้วตา”];

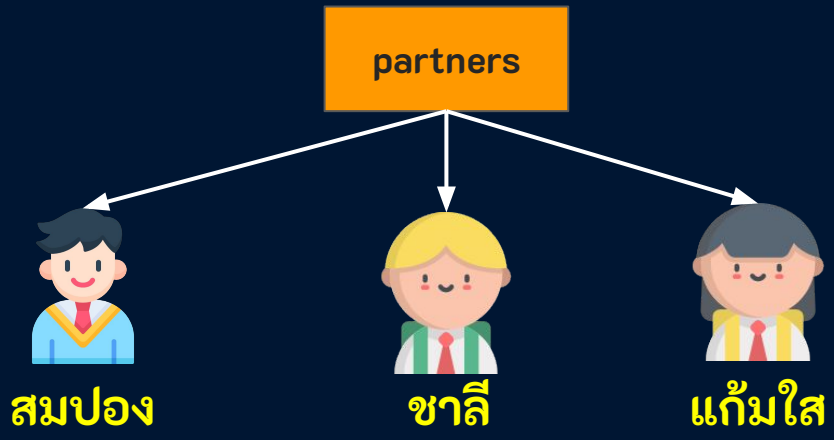
การเข้าถึงสมาชิกอาร์เรย์

int[4] number = [10, 20, 30, 40];

string [2] partners = [“สมชาย”,”แก้วตา”];

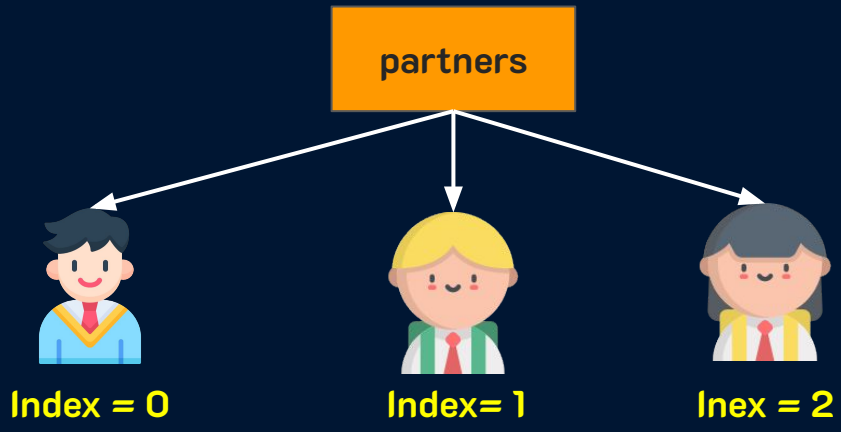
ตัวอย่างที่ 1

string [3] partners=[“สมปอง”,”ชาลี”,”แก้มใส”]



ตัวอย่างที่ 1

string [3] partners=[“สมปอง”,”ชาลี”,”แก้มใส”]



จำนวนสมาชิกในอาร์เรย์ (Length)

int[4] number = [10, 20, 30, 40];

นับจำนวนสมาชิกในอาร์เรย์ (Length)

```
int[4] number = [10, 20, 30, 40];
number.length;
string [2] pets = [“แมว”,”กระต่าย”];
pets.length;
```

Storage และ Memory

Storage และ Memory

Storage คือ การจัดเก็บข้อมูลถาวรบน Blockchain ไม่สามารถเปลี่ยนแปลงได้ การเปลี่ยนแปลงสถานะแต่ละครั้งนั้น ต้องเสียค่าใช้จ่ายในการเปลี่ยนแปลง ตัวอย่างของ Storage เช่น State Variable

Memory คือ การจัดเก็บข้อมูลชั่วคราวซึ่งข้อมูลเหล่านี้จะทำงานเมื่อตอนที่มีการเรียกใช้งานฟังก์ชั่นเท่านั้นและจะถูกลบออกไปหลังจากที่มีการใช้งานแล้ว ซึ่งค่าใน Memory จะไม่ถูกบันทึกลงใน Blockchain เช่น อาร์กิวเมนต์ของฟังก์ชั่น , Local Variable หรือตัวแปรที่ทำงานในฟังก์ชั่น

Mapping

Data Type	คำอธิบาย
boolean	ค่าทางตรรกศาสตร์
integer	Int (ตัวเลขจำนวนเต็ม)
	uint (ตัวเลขที่มีเฉพาะจำนวนเต็มค่าบวก)
string	ข้อความ
struct	โครงสร้างข้อมูล
array	ชุดข้อมูล
mapping	<key,value>
address	ใช้เก็บ address ของบัญชีนั้นๆ

Data Type	คำอธิบาย
boolean	ค่าทางตรรกศาสตร์
integer	Int (ตัวเลขจำนวนเต็ม)
	uint (ตัวเลขที่มีเฉพาะจำนวนเต็มค่าบวก)
string	ข้อความ
struct	โครงสร้างข้อมูล
array	ชุดข้อมูล
mapping	<key,value>
address	ใช้เก็บ address ของบัญชีนั้นๆ

Mapping คืออะไร

ตัวแปรที่เก็บข้อมูลในรูปแบบคู่ key , value มีลักษณะคล้ายกับอาร์เรย์ แต่จะใช้ key เป็น index เพื่อเชื่อมโยง value ข้อมูลที่เก็บใน key นั้นๆ ถ้าทราบ key ก็สามารถเข้าถึง value หรือข้อมูลได้นั่นเอง

การนิยาม Mapping

Mapping (key => value) ชื่อตัวแปร;

Mapping (string => string) language;

Array

string [2] language [“Thailand”,”England”];

Thailand (0)	England (1)

ใช้เลข index ที่เป็นตัวเลขจำนวนเต็มในการอ้างอิงข้อมูล

Mapping

```
mapping(string=>string) country;
country[“TH”] = “Thailand”;
country[“EN”] = “England”;
```

TH=>Thailand	EN=>England

Mapping

```
mapping(string=>string) coin;
coin[“ETH”] = “Ether”;
coin[“BTC”] = “Bitcoin”;
```

ETH=>Ether	BTC=>Bitcoin

Mapping

```
mapping(string=>uint) population;
population[“Thailand”] = 70;
population[“England”] = 55;
```

Thailand => 70	England => 55

Mapping

```
mapping(address=>uint) balance;
balance[0xxx4] = 10;
balance[0xxx3] = 5;
```

0xxx4 => 10	0xxx5 => 5

Structure

สตรัคเจอร์ (Structure)

ข้อจำกัดของ Array ในกรณีที่มีการเก็บข้อมูลลงไปใน Array สมาชิกทุกตัวที่อยู่ใน Array ต้องมีชนิดข้อมูลเหมือนกัน

แล้วถ้าต้องการอยากเก็บข้อมูลที่มีชนิดข้อมูลต่างกันจะทำอย่างไร ?

สตรัคเจอร์ (Structure)

คือ ข้อมูลแบบโครงสร้างที่นำเอาข้อมูลที่มีชนิดข้อมูลต่างกันมารวบรวมเข้าด้วยกัน แต่มีความสัมพันธ์ของข้อมูลแบบต่อกัน มาเก็บไว้ภายในโครงสร้างเดียวกัน

เปรียบเสมือนกับสร้างชนิดข้อมูลขึ้นมาใช้งานเอง

การนิยามสตรัคเจอร์

```
struct ชื่อสตรัคเจอร์{
        ชนิดข้อมูลตัวที่ 1 ตัวแปรที่ 1;
        ชนิดข้อมูลตัวที่ 2 ตัวแปรที่ 2;
        .....
}
```

เก็บข้อมูลผู้จัดการ (Manager)

- ชื่อผู้จัดการ(string)
- อายุ (uint)
- เลขบัญชี (address)

Modifier

Modifier คืออะไร

คือ การนิยามฟังก์ชั่นสำหรับตรวจสอบหรือกำหนดสิทธิ์ในการใช้งานข้อมูลหรือฟังก์ชั่นภายใน Contract

Enum

Enum (Enumerator)

Enumerator คือ สิ่งที่ระบุขึ้นเอง หรือหมายถึงตัวแปรที่เป็นรูปแบบตัวเลขจำนวนเต็ม (integer) ที่มีการตั้งชื่อเฉพาะขึ้นมาเพื่อเป็นตัวแทนของกลุ่มข้อมูล (นิยามชื่อชนิดข้อมูลเอง)

การนิยาม Enum (Enumerator)

```
enum ชื่อenum{
    value1,
    value2,
    value3
    ...
}
```

ตัวอย่าง

```
enum State{
    Open,
    Close
}
```

ตัวอย่าง

```
enum Rating{
        VeryBad,
        Bad,
        Good,
        Great,
        Excellent
}
```

Break!!

https://www.youtube.com/c/KongRuksiamOfficial/

https://www.facebook.com/KongRuksiamTutorial/

พินัยกรรม
(Testament Contract)

บทบาทผู้ใช้งาน Contract

เจ้าของมรดก / ผู้เขียนพินัยกรรม (Owner)

ผู้จัดการมรดก (Manager)

ผู้รับมรดก (heir)

บทบาทผู้ใช้งาน Contract

เจ้าของมรดก / ผู้เขียน
พินัยกรรม (Owner)

10 ETH

ผู้รับมรดก (heir)

บทบาทผู้ใช้งาน Contract

เจ้าของมรดก / ผู้เขียนพินัยกรรม (Owner)

ผู้รับมรดก (heir)

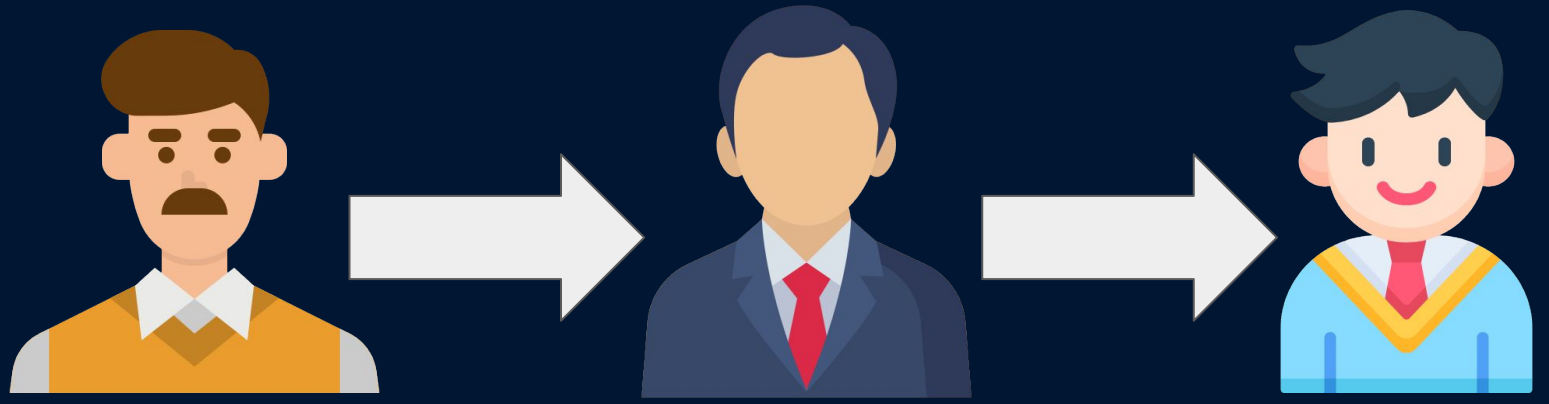
บทบาทผู้ใช้งาน Contract

เจ้าของมรดก / ผู้เขียน
พินัยกรรม (Owner)

10 ETH

ผู้รับมรดก (heir)

บทบาทผู้ใช้งาน Contract



เจ้าของมรดก / ผู้เขียนพินัยกรรม (Owner)

ผู้จัดการมรดก (Manager)

ผู้รับมรดก (heir)

บทบาทผู้ใช้งาน Contract

เจ้าของมรดก / ผู้เขียนพินัยกรรม (Owner)

ผู้จัดการมรดก (Manager)

ผู้รับมรดก (heir)

ผู้จัดการมรดก (Manager)

- เตรียมพินัยกรรม (Contract / Deploy)
- แจ้งการเสียชีวิตของเจ้าของมรดก (Owner)

เจ้าของมรดก / เจ้าของพินัยกรรม (Owner)

- เขียนพินัยกรรม (Create)
- ระบุทายาทผู้รับมรดกว่าเป็นใคร (Heir)
- ระบุจำนวนทรัพย์สินที่จะมอบให้ทายาท (ETH)

ผู้รับมรดก (Heir)

- ได้รับมรดกตามจำนวนทรัพย์สินที่ระบุในพินัยกรรม
- ได้รับมรดกทันทีเมื่อผู้จัดการมรดก (Manager) แจ้งว่าเจ้าของมรดก (Owner) นั้นได้เสียชีวิตแล้ว (ได้รับอัตโนมัติ)
- แต่ถ้าเจ้าของมรดกยังไม่เสียชีวิต ทายาทก็จะไม่ได้รับมรดก

โครงสร้างของ Contract (Manager)

1. เก็บข้อมูลของผู้จัดการมรดก (Manager)
2. เก็บข้อมูลของเจ้าของมรดก (Owner) และทายาท (Heir)
   ถ้าทราบข้อมูลเจ้าของมรดกก็จะทราบว่าทายาทเป็นใคร
3. จำนวนเงินที่เจ้าของมรดกต้องการจะมอบให้ทายาท
4. สร้างพินัยกรรมโดยเจ้าของมรดก (Owner)
5. แจ้งการเสียชีวิตโดยผู้จัดการมรดก (Manager)
6. โอนเงินตามจำนวนที่ระบุในพินัยกรรมไปให้ทายาท

โครงสร้างของ Contract (Manager)

1. Address manager
2. mapping(address owner => address heir)
3. balance = 10 ETH (balance > 0 ETH)
4. Create (address heir)
5. ReportOfDeath(address owner)
6. Transfer(address heir,balance)

เครื่องมือที่ใช้งาน

1. Remix - Ethereum IDE
2. Rinkeby Test Network
3. Metamask
    - Account1 => Manager
    - Account2 => Owner
    - Account3 => Heir

ลอตเตอรี่
(Lottery Contract)

บทบาทผู้ใช้งาน Contract

ผู้จัดการลอตเตอรี่

(Manager)

กลุ่มผู้ซื้อลอตเตอรี่

(Player)

ผู้จัดการลอตเตอรี่ (Manager)

- มีจำนวน 1 คนเท่านั้น
- กำหนดราคาลอตเตอรี่ (ใบละ 1 ETH)
- จัดการระบบลอตเตอรี่ (Contract / Deploy)
- สามารถซื้อลอตเตอรี่ได้ (Manager / Players)
- ประกาศผลรางวัล (Select Winner)

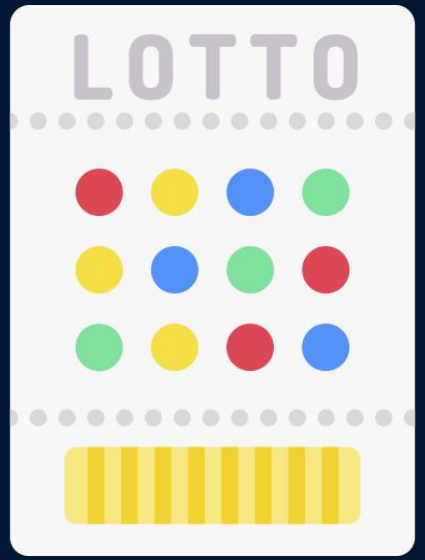
กลุ่มผู้ซื้อลอตเตอรี่ (Players)

- มีจำนวนหลายคน
- แต่ละคนสามารถซื้อลอตเตอรี่ได้หลายใบ
- ลอตเตอรี่แต่ละใบมีราคา 1 ETH
- เงินรางวัลทั้งหมดมาจากกลุ่มคนซื้อลอตเตอรี่
- มีแค่ 1 คนเท่านั้นที่จะได้รับเงินรางวัล
  ซึ่งจะได้จากการสุ่มจากจำนวนผู้ซื้อทั้งหมด (Winner)

ผู้จัดการลอตเตอรี่ (Manager)

ผู้จัดการลอตเตอรี่

(Manager)

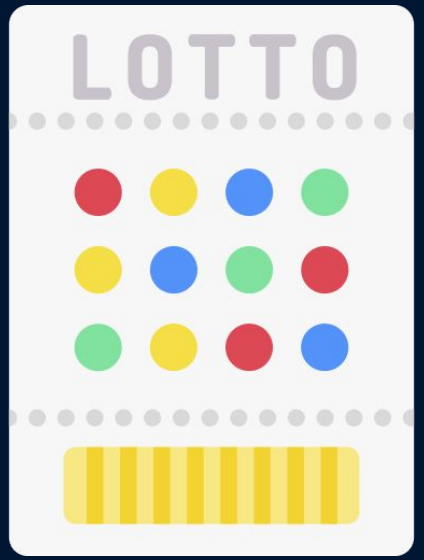


กลุ่มผู้ซื้อลอตเตอรี่ (Players)

กลุ่มผู้ซื้อลอตเตอรี่

(Player)

1 ETH



กลุ่มผู้ซื้อลอตเตอรี่ (Players)



กลุ่มผู้ซื้อลอตเตอรี่ (Players)

Account 1

Account 2

Account 3

กลุ่มผู้ซื้อลอตเตอรี่ (Players)

Account 1

Account 2

Account 3

กลุ่มผู้ซื้อลอตเตอรี่ (Players)

เงินรางวัลรวมทั้งหมด = 9 ETH

ผู้จัดการลอตเตอรี่ (Manager)

ผู้จัดการลอตเตอรี่

(Manager)

เงินรางวัล = 9 ETH

ผู้จัดการลอตเตอรี่ (Manager)

ผู้จัดการลอตเตอรี่

(Manager)

เงินรางวัล = 9 ETH

สุ่มหาผู้ที่จะได้รับเงินรางวัล (Winner)

ผู้จัดการลอตเตอรี่ (Manager)

ผู้จัดการลอตเตอรี่

(Manager)

เงินรางวัล = 9 ETH

ผู้ที่จะได้รับเงินรางวัล (Winner)

โครงสร้างของ Contract (Manager)

1. เก็บข้อมูลของผู้จัดการลอตเตอรี่ (Manager)
2. เก็บข้อมูลของกลุ่มคนซื้อลอตเตอรี่ (Players)
3. เก็บยอดเงินรางวัลทั้งหมด (Balance)
4. เก็บจำนวนกลุ่มคนซื้อลอตเตอรี่ทั้งหมด (Length)
5. ฟังก์ชั่นสุ่มตัวเลข (Random)
6. เลือกผู้ได้รับรางวัล (Winner) จำนวนลอตเตอรี่มากกว่า 2 ใบ
7. มอบเงินรางวัลและเคลียร์ข้อมูลกลุ่มคนซื้อลอตเตอรี่

โครงสร้างของ Contract (Manager)

1. address manager
2. address payable[] players
3. getbalance()
4. getlength()
5. random()
6. select Winner
7. transfer Reward

เครื่องมือที่ใช้งาน

1. Remix - Ethereum IDE
2. Rinkeby Test Network
3. Metamask
    - Account1 => Manager / Player
    - Account2 => Player
    - Account3 => Player

ระบบเลือกตั้ง
(Election Contract)

บทบาทผู้ใช้งาน Contract

ผู้จัดการเลือกตั้ง/เจ้าหน้าที่

(Manager)

ผู้สมัครเลือกตั้ง

(Candidate)

ผู้มีสิทธิเลือกตั้ง

(Voter)

ผู้จัดการเลือกตั้ง/เจ้าหน้าที่ (Manager)

- มีจำนวน 1 คนเท่านั้น
- เพิ่มข้อมูลผู้สมัครเลือกตั้ง (Add Candidate)
- ลงทะเบียนผู้มีสิทธิ์เลือกตั้ง (Register Voter)

ผู้สมัครเลือกตั้ง (Candidate)

- มีได้หลายคน (Array Candidate)
- ข้อมูลผู้สมัครเลือกตั้งจะถูกเพิ่มเข้าสู่ระบบโดยเจ้าหน้าที่ (Manager) ซึ่ง ประกอบด้วย
    1. หมายเลขผู้สมัครเลือกตั้ง (index)
    2. ชื่อผู้สมัครเลือกตั้ง (Candidate Name)
    3. คะแนนเลือกตั้ง (Vote Count)

ผู้มีสิทธิเลือกตั้ง (Voter)

- ต้องลงทะเบียนเลือกตั้งกับเจ้าหน้าที่ ถ้าไม่ลงทะเบียนจะไม่สามารถเลือกตั้งได้ (Register)
- ข้อมูลที่ลงทะเบียนประกอบด้วย
  - รหัสประจำตัวของผู้มีสิทธิเลือกตั้ง (Address)
  - สถานะการเลือกตั้ง (Check Voted)
  - หมายเลขผู้สมัครที่เลือก (Candidate Index)
- 1 คนสามารถเลือกได้แค่ 1 หมายเลขเท่านั้น (1 สิทธิ์ = 1 เสียง)

ผู้จัดการเลือกตั้ง (Manager)

ผู้จัดการเลือกตั้ง(Manager)

เพิ่มข้อมูลผู้สมัครเลือกตั้ง

ผู้สมัครเลือกตั้ง(Candidate)

ผู้จัดการเลือกตั้ง (Manager)

ผู้สมัครเลือกตั้ง(Candidate)

ข้อมูลผู้สมัครเลือกตั้ง (Candidate)

- หมายเลขผู้สมัครเลือกตั้ง
- ชื่อผู้สมัครเลือกตั้ง
- จำนวนคะแนนเสียง

- หมายเลขผู้สมัครเลือกตั้ง
- ชื่อผู้สมัครเลือกตั้ง
- จำนวนคะแนนเสียง

- หมายเลขผู้สมัครเลือกตั้ง
- ชื่อผู้สมัครเลือกตั้ง
- จำนวนคะแนนเสียง

ข้อมูลผู้สมัครเลือกตั้ง (Candidate)

- Index = 0
- Name = kong
- Count = 0

- Index = 1
- Name = Jenny
- Count = 0

- Index = 2
- Name = Kam
- Count = 0

ผู้มีสิทธิ์เลือกตั้ง (Voter)

ผู้มีสิทธิ์เลือกตั้ง (Voter)

ลงทะเบียนเลือกตั้ง (Register)

ผู้จัดการเลือกตั้ง(Manager)

ผู้มีสิทธิ์เลือกตั้ง (Voter)

นาย A

ลงทะเบียนเลือกตั้ง (Register)

ผู้จัดการเลือกตั้ง(Manager)

ผู้มีสิทธิ์เลือกตั้ง (Voter)

นาย A

นำ Address ไปลงทะเบียน

ผู้จัดการเลือกตั้ง(Manager)

ผู้มีสิทธิ์เลือกตั้ง (Voter)

ผู้จัดการเลือกตั้ง(Manager)

บันทึกข้อมูลนาย A

ชื่อ : นาย A (address)
สถานะ : ยังไม่โหวตเลือกตั้ง
หมายเลขผู้สมัครที่เลือก : -
(ลงทะเบียนเรียบร้อย)

ผู้มีสิทธิ์เลือกตั้ง (Voter)

ชื่อ : นาย A (address)
สถานะ : ยังไม่โหวตเลือกตั้ง
หมายเลขผู้สมัครที่เลือก : -
(ลงทะเบียนเรียบร้อย)

สามารถเลือกตั้งได้

การเลือกตั้ง (Vote)

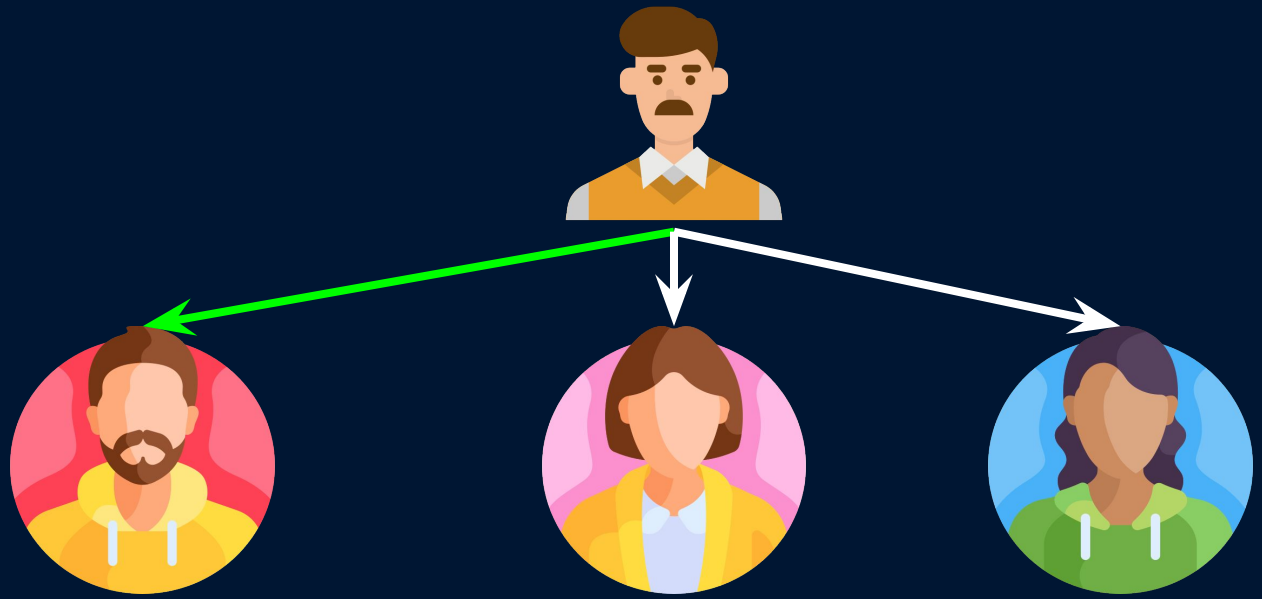
ชื่อ : นาย A (address)
สถานะ : ยังไม่โหวตเลือกตั้ง
หมายเลขผู้สมัครที่เลือก : -
(ลงทะเบียนเรียบร้อย)

- Index = 0
- Name = kong
- Count = 0

- Index = 1
- Name = Jenny
- Count = 0

- Index = 2
- Name = Kam
- Count = 0

การเลือกตั้ง (Vote)

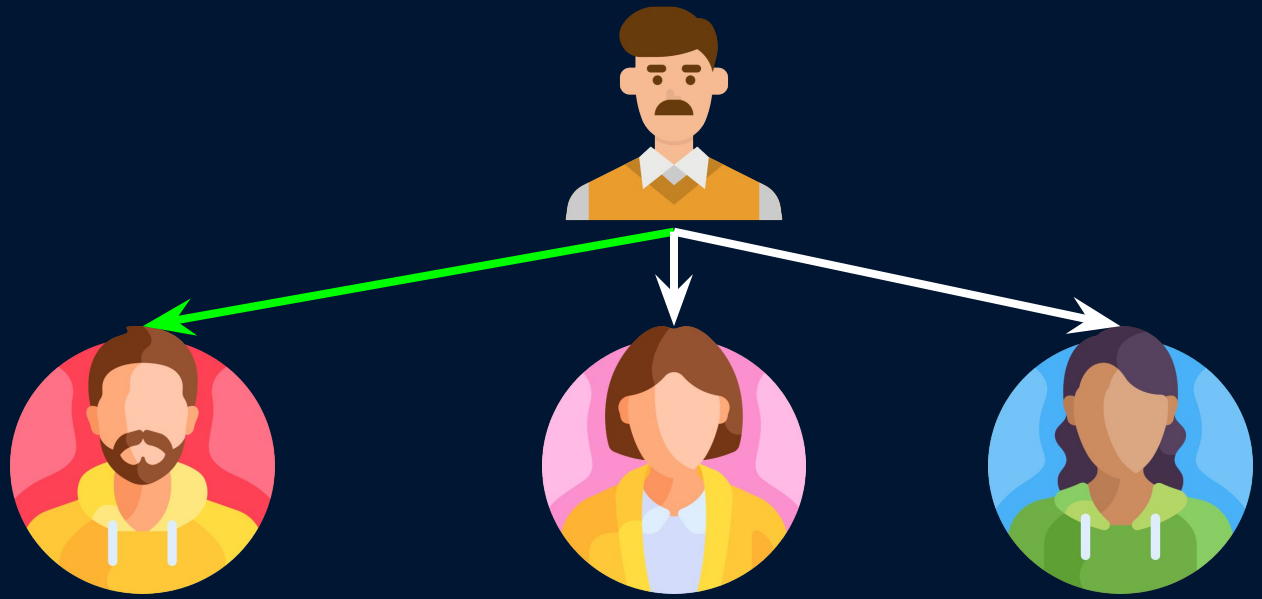


- Index = 0
- Name = kong
- Count = 0

- Index = 1
- Name = Jenny
- Count = 0

- Index = 2
- Name = Kam
- Count = 0

การเลือกตั้ง (Vote)

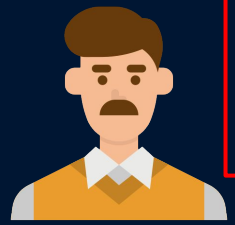


- Index = 0
- Name = kong
- Count = 1

- Index = 1
- Name = Jenny
- Count = 0

- Index = 2
- Name = Kam
- Count = 0

การเลือกตั้ง (Vote)



ชื่อ : นาย A (address)

สถานะ : โหวตแล้ว

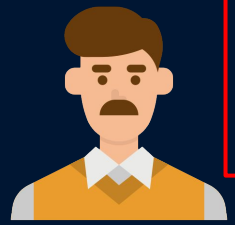
หมายเลขผู้สมัครที่เลือก : 0

- Index = 0
- Name = kong
- Count = 1

- Index = 1
- Name = Jenny
- Count = 0

- Index = 2
- Name = Kam
- Count = 0

การเลือกตั้ง (Vote)



ชื่อ : นาย A (address)

สถานะ : โหวตแล้ว

หมายเลขผู้สมัครที่เลือก : 0

- Index = 0
- Name = kong
- Count = 1

- Index = 1
- Name = Jenny
- Count = 0

- Index = 2
- Name = Kam
- Count = 0

ความรู้พื้นฐาน

1. Constructor
2. Mapping
3. Array
4. Structure
5. Function Modifier