



# Smart Contract

บล็อกเชนและการสร้างสัญญาอัจฉริยะ

ตอนที่ 1

# Blockchain

# เทคโนโลยี Blockchain

Blockchain คือ เทคโนโลยีการจัดเก็บข้อมูลแบบ Shared Database หรือที่รู้จักกันในชื่อ **Distributed Ledger Technology (DLT)** เป็นรูปแบบการเก็บข้อมูลที่มีความปลอดภัยสูงซึ่งใช้วิธีการตรวจสอบข้อมูลโดยการเช็คจากข้อมูลที่ถูกจัดเก็บไปก่อนหน้านี้ ข้อมูลดังกล่าวไม่สามารถที่จะเปลี่ยนแปลงหรือแก้ไขได้ ซึ่งผู้ใช้ที่อยู่ในเครือข่ายจะเห็นข้อมูลชุดเดียวกันทั้งหมดเป็นและผู้ตรวจสอบข้อมูลโดยใช้หลักการ Cryptography (การเข้ารหัส)



# การประยุกต์ใช้ Blockchain

- เงินดิจิทัล (Cryptocurrency)
- สัญญาอัจฉริยะ (Smart Contract)
- ระบบ/บริการอัตโนมัติ (Decentralized Autonomous Systems/ Services)
- บริการพิสูจน์ทราบ (Proof of Services)

# การประยุกต์ใช้ Blockchain

- เงินดิจิทัล (Cryptocurrency)
- สัญญาอัจฉริยะ (Smart Contract)
- ระบบ/บริการอัตโนมัติ (Decentralized Autonomous Systems/ Services)
- บริการพิสูจน์ทราบ (Proof of Services)

**Teachnology**

**(Layer 1)**

**Protocol / Coin**

**(Layer 2)**

**Token**

**(Layer 3)**



# Teachnology

เทคโนโลยี Blockchain

## Protocol / Coin

## Token



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



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

## Teachnology

เทคโนโลยี Blockchain

## Protocol / Coin

ข้อกำหนดหรือข้อตกลงในการสื่อสารระหว่างคอมพิวเตอร์

## Token



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



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



## Teachnology

เทคโนโลยี Blockchain

## Protocol / Coin

การนำเอาเทคโนโลยี Blockchain มาใช้งาน แต่ละโปรโตคอลจะมีสกุลเงินของตัวเอง ซึ่ง Bitcoin ก็ถือว่าเป็นโปรโตคอลอีกรูปแบบหนึ่ง

## Token



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



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

## Teachnology

เทคโนโลยี Blockchain

## Protocol / Coin

การนำเอาเทคโนโลยี Blockchain มาใช้งานแต่ละโปรโตคอลจะมีสกุลเงินของตัวเอง ซึ่ง Bitcoin ก็ถือว่าเป็นโปรโตคอลอีกรูปแบบหนึ่ง

## Token

ชุดข้อมูลเสมือนที่ถูกเข้ารหัส ใช้แทนข้อมูลที่ต้องการความปลอดภัยสูงและหลีกเลี่ยงการแลกเปลี่ยนข้อมูลนั้นโดยตรง เช่น เลขบัตรเครดิต เป็นต้น



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



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

**Teachnology**

**Blockchain**

**Protocol / Coin**

**Bitcoin**

**Ethereum**

**ZCoin**

**Token**

**HT**

**USDT**

**MKR**



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



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

**Teachnology**

**Blockchain**

**Protocol / Coin**

**Bitcoin**

**Ethereum**

**ZCoin**

**Token**

**HT**

**USDT**

**MKR**



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



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



Bitcoin คือ สกุลเงินดิจิทัลซึ่งมีมูลค่า  
เหมือนกับธนบัตรในสกุลเงินประเทศต่าง ๆ และ  
ถือว่าเป็นโปรโตคอลที่นำเอาเทคโนโลยี  
Blockchain  
ที่มีความปลอดภัยสูงต่อการรับส่งข้อมูลมาใช้  
เป็นระบบแลกเปลี่ยนเงินบนโลกอินเทอร์เน็ต

ตอนที่ 2

# Ethereum

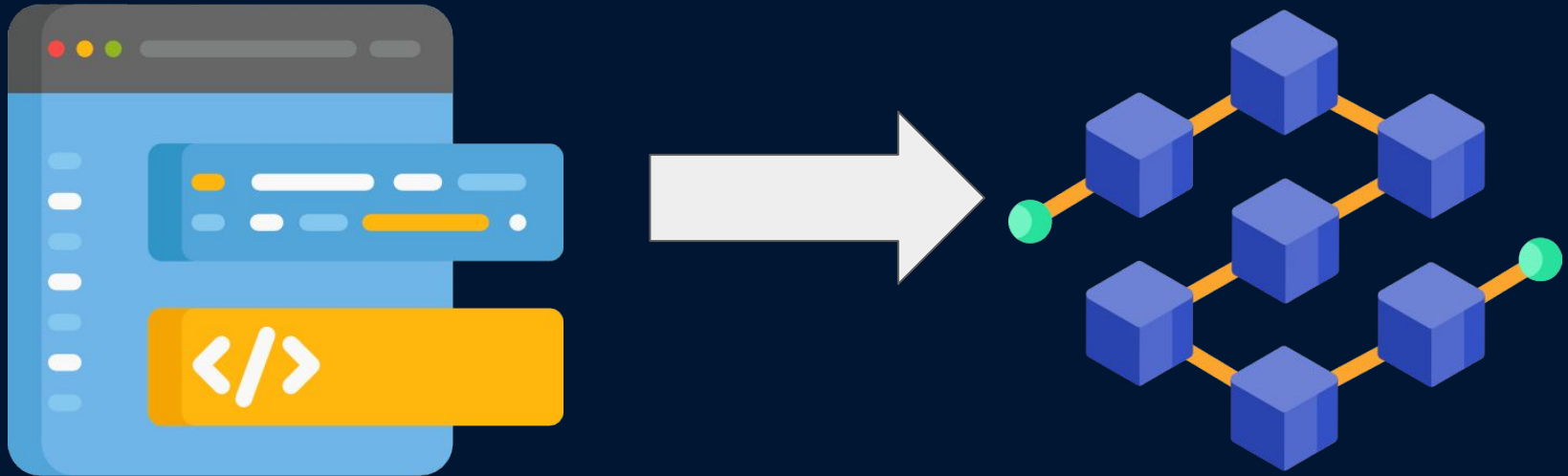
# ข้อจำกัดของ Bitcoin

ระบบของ Bitcoin ที่ถูกสร้างขึ้นมานั้น ทำหน้าที่เป็น  
แค่สกุลเงินดิจิทัล (Cryptocurrency) ซึ่งเน้นการทำ  
ธุรกรรมด้านการเงินเพียงอย่างเดียว

“ ถ้าอยากรจะทำระบบอื่นๆขึ้นมาโดยอาศัย  
เทคโนโลยี Blockchain และมีการทำ  
ธุรกรรมทางการเงินด้วยจะต้องทำอย่างไร ”



# อยากรันโปรแกรมใน Blockchain ได้



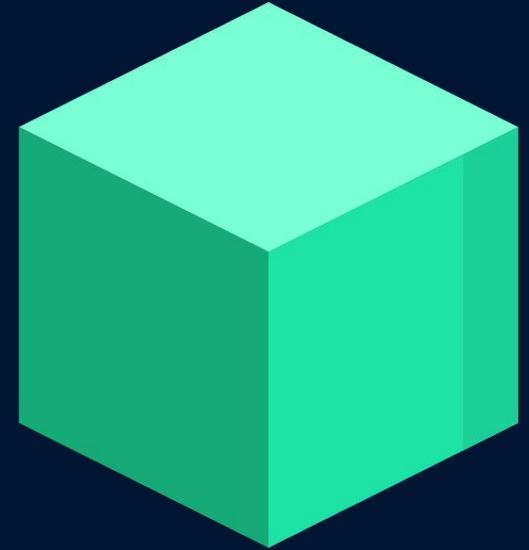
Block #ID

Nonce : xxx

Data : ????????????

Prev Hash : 27235ad4f82

Hash : ????



Block #ID

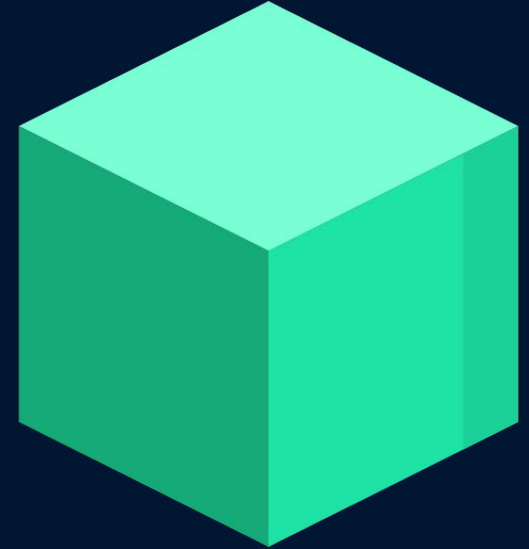
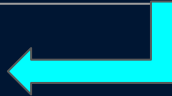
Nonce : xxx

**Data : Source Code**

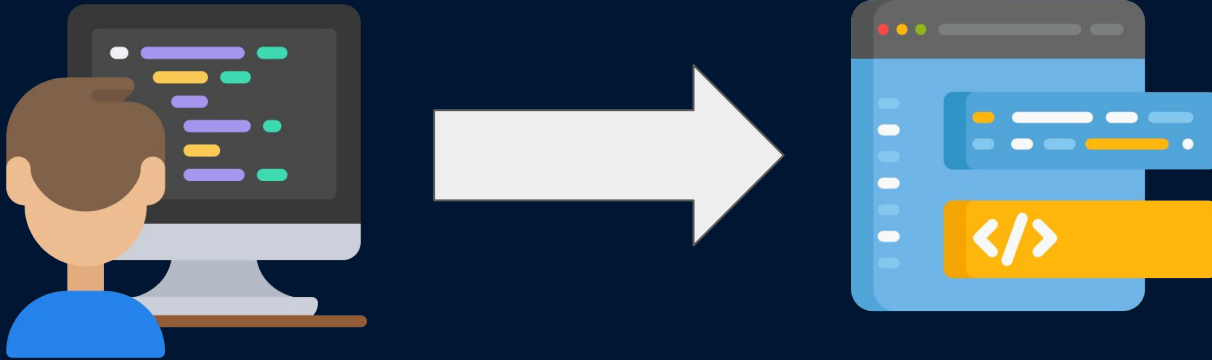


Prev Hash : 27235ad4f82

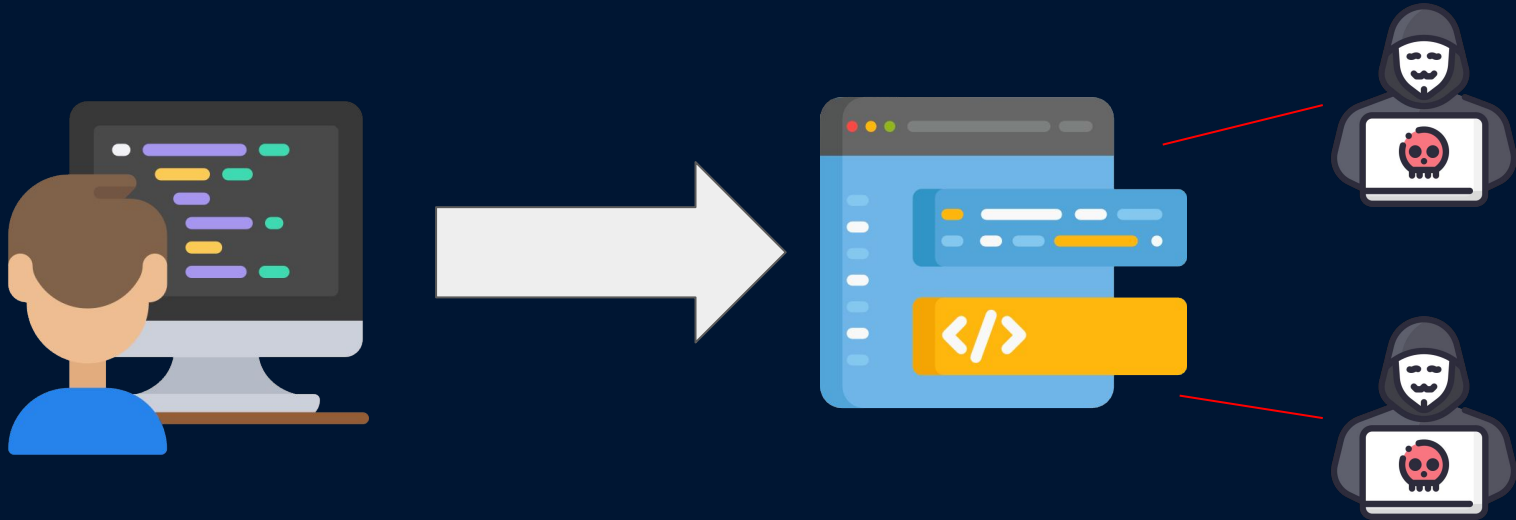
Hash : ??? (Target)



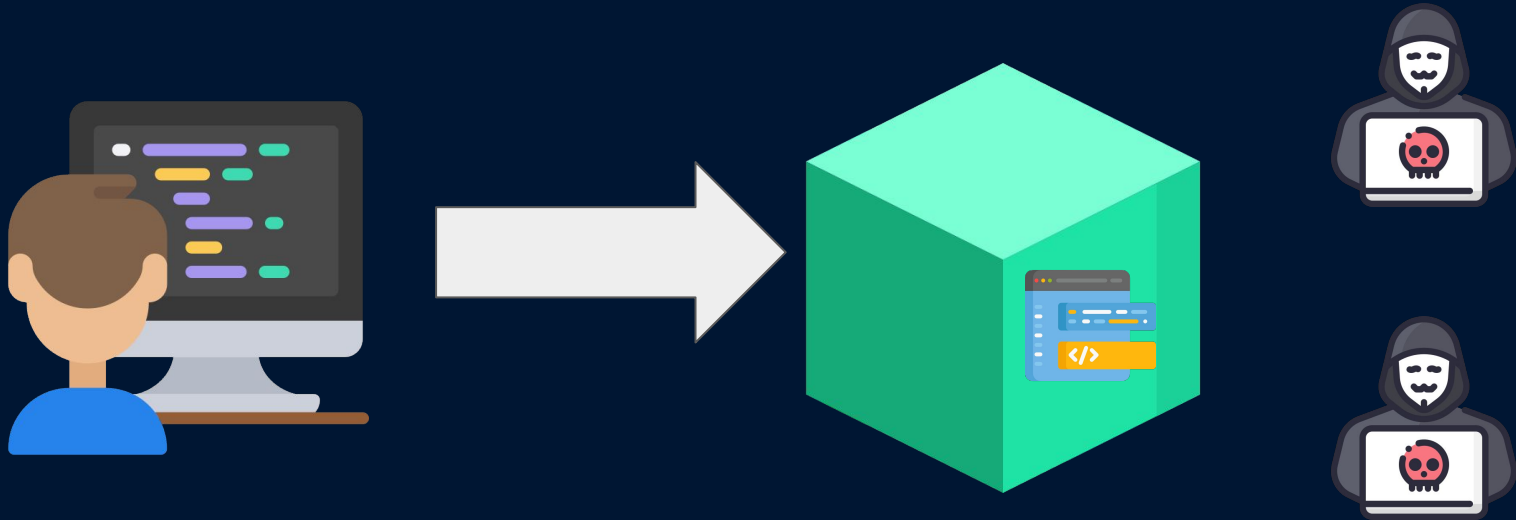
# อยากรันโปรแกรมใน Blockchain ได้



# อยากรันโปรแกรมใน Blockchain ได้



# อยากรันโปรแกรมใน Blockchain ได้



ethereum  
ETHERBROWSER  
PEER-TO-PEER MESSAGING  
GENERALIZED BLOCKCHAIN  
PROGRAM ANYTHING

## Vitalik Buterin

ผู้สร้างและคิดค้นโปรเจกต์ Ethereum



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



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

**Teachnology**

**Blockchain**

**Protocol / Coin**

**Bitcoin**

**Ethereum**

**ZCoin**

**Token**

**HT**

**USDT**

**MKR**



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



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



# Ethereum คืออะไร

- Open Source Platform ที่สร้างขึ้นด้วยเทคโนโลยี Blockchain
- รองรับระบบการเงินเหมือน Bitcoin มีความปลอดภัยและไม่ถูกควบคุมจากกลุ่มบุคคลใดบุคคลหนึ่ง
- มี Network เหมือน Bitcoin ซึ่งภายใน Network ดังกล่าวสามารถรันโปรแกรมได้ด้วย เรียกว่า **Smart Contract (สัญญาอัจฉริยะ)**
- มีเหรียญหรือสกุลเงินที่ใช้ใน Network ชื่อว่า **Ether Coin (ETH)**



# คุณสมบัติของ Ethereum Network

- สามารถรับส่งเหรียญใน Network ได้
- สามารถเก็บข้อมูล (Data) ได้
- ทุกๆ Node ใน Network สามารถรันโปรแกรมได้
- ทุกๆ Node มีข้อมูลเหมือนกัน



# Decentralized Applications (Dapps)



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



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

# Ethereum คืออะไร

- **Open Source Platform** ที่สร้างขึ้นด้วยเทคโนโลยี Blockchain
- รองรับระบบการเงินเหมือน Bitcoin มีความปลอดภัยและไม่ถูกควบคุมจากกลุ่มบุคคลใดบุคคลหนึ่ง
- มี Network เหมือน Bitcoin ซึ่งภายใน Network ดังกล่าวสามารถรันโปรแกรมได้ด้วย เรียกว่า **Smart Contract (สัญญาอัจฉริยะ)**
- มีเหรียญหรือสกุลเงินที่ใช้ใน Network ชื่อว่า **Ether Coin (ETH)**



# Open-Source Platform

คือ Platform ที่สามารถนำเอา Source Code มาทำงานได้ ซึ่งแอปพลิเคชันที่รันบน Ethereum Platform จะถูกทำงานในระบบ Blockchain หรือเรียกว่า

“ Decentralized Applications  
(Dapps) ”



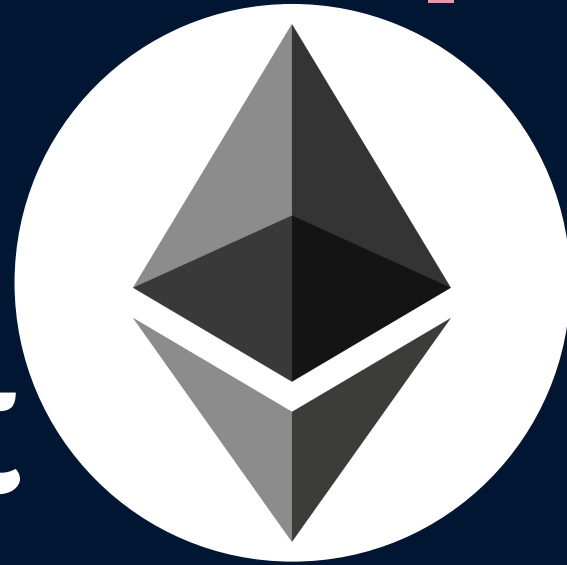


Decentralized Applications ต่างกับแอปทั่วไป (Traditional Application)  
คือ การทำธุรกรรมต่างๆ ภายในแอปพลิเคชัน จะไม่ผ่านคนกลาง (บริษัท  
ต่างๆ ที่ให้บริการผ่านแอปพลิเคชัน)

ใน Ethereum Platform จะมีส่วนที่เรียกว่า Smart Contract สำหรับนำมาทำเป็น Dapps ซึ่งมีรูปแบบการพัฒนาแอป ดังนี้

- แอปพลิเคชันสำหรับแลกเปลี่ยนเงินโดยตรง
- แอปพลิเคชันที่เกี่ยวกับการเงิน
- แอปพลิเคชันอื่นๆ เช่น ระบบพินัยกรรม , ระบบโหวต เป็นต้น

# Smart Contract





# Smart Contract คืออะไร

## ความหมายที่ 1

Smart Contract หรือ สัญญาอัจฉริยะ เป็นโปรแกรมคอมพิวเตอร์ ที่ทำงานใน Ethereum Blockchain โดยจะเก็บเงื่อนไขหรือข้อตกลงของ สัญญาต่าง ๆ ไว้ในรูปแบบ Code คอมพิวเตอร์ ซึ่งจะทำงานอัตโนมัติทันที ที่เกิดเหตุการณ์ตามเงื่อนไขในสัญญา โดยไม่ต้องมีคนกลาง เช่น ธนาคาร รัฐบาล เป็นต้น

# Smart Contract คืออะไร

## ความหมายที่ 1

สามารถแก้ปัญหาคำถามไม่ไว้ การบิดเบือนสัญญา หรือการโกงระหว่างคู่สัญญาได้ เนื่องจากสัญญาดังกล่าวที่ถูกเขียนขึ้นและเก็บใน Blockchain นั้นไม่สามารถแก้ไขได้ในภายหลังและทุกคนในเครือข่ายสามารถเข้าไปตรวจสอบข้อตกลงของสัญญาต่าง ๆ ได้ ยกตัวอย่าง เช่น การทำระบบสัญญาซื้อ-ขาย, การทำระบบพินัยกรรม เป็นต้น

# Smart Contract คืออะไร

## ความหมายที่ 2

- เป็นการเก็บ Source Code ไว้ในแต่ละ Block ที่อยู่ใน Blockchain
- รันโค้ดหรือคำสั่งที่เก็บภายใน Block ได้
- โค้ดที่ถูกเก็บใน Block ไม่สามารถแก้ไขได้จึงเรียกว่า สัญญาอัจฉริยะ
- มีกระบวนการตรวจสอบหรือ Mining Block ได้
- ใช้ภาษา Solidity ในการสร้าง Smart Contract



Block #ID

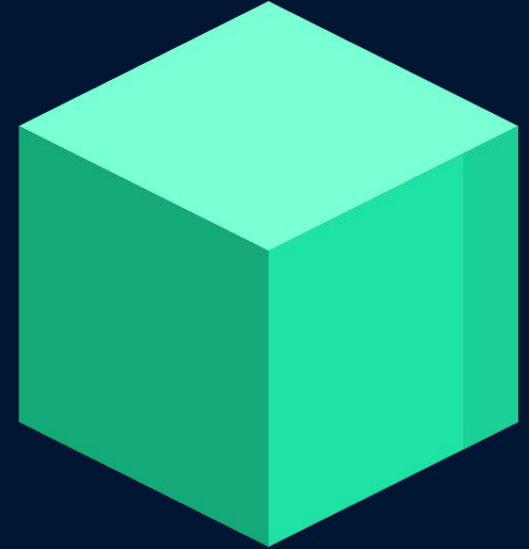
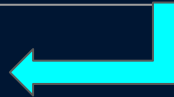
Nonce : xxx

**Data : Source Code**



Prev Hash : 27235ad4f82

Hash : ??? (Target)



# จุดเด่นของ Smart Contract

# จุดเด่นของ Smart Contract

- **ความปลอดภัย (Security)** เนื่องจาก Smart Contract ที่ถูกพัฒนาโดยใช้เทคโนโลยี Blockchain นั้นจะถูกกระจายไปยังสมาชิกอื่น ๆ ที่อยู่ในเครือข่าย (Node) จึงทำให้มั่นใจได้ว่า Smart Contract ดังกล่าวจะไม่สูญหายหรือถูกเปลี่ยนแปลงเงื่อนไขการทำงานโดยไม่ได้รับอนุญาต

# จุดเด่นของ Smart Contract

- **ความเป็นอัตโนมัติ (Automation)** Smart Contract จะดำเนินการตามข้อตกลงโดยอัตโนมัติทันทีเมื่อเกิดเหตุการณ์ ตามเงื่อนไขในสัญญา ซึ่งได้มีการระบุไว้ล่วงหน้าแล้ว โดยไม่ต้อง มีคนกลางมาเกี่ยวข้อง เช่น การทำสัญญาพินัยกรรม เป็นต้น

# จุดเด่นของ Smart Contract

- **ความเป็นมาตรฐาน (Standardization)** การนำ Smart Contract มาใช้งานในระบบใด ๆ นั้น หมายความว่าระบบนั้น จะต้องทำงานภายใต้เงื่อนไข หรือข้อตกลงต่างที่เป็นมาตรฐานเดียวกันทั้งระบบ ตามที่ได้กำหนดไว้ใน Smart Contract



# จุดด้อยของ Smart Contract

# จุดด้อยของ Smart Contract

- ความผิดพลาดที่เกิดจากมนุษย์ (Human Error) เนื่องจากการสร้าง Smart Contract คือ การเขียนโปรแกรม หากมีการทดสอบไม่เพียงพอ Smart Contract ที่สร้างขึ้นก็สามารถทำงานผิดพลาดได้ และอาจส่งผลกระทบในวงกว้าง

# จุดด้อยของ Smart Contract

- ไม่สามารถแก้ไขย้อนหลังได้ (Immutable) หลังจากที่เราสร้าง Smart Contract แล้วจะไม่สามารถหรือปรับปรุงโค้ดหรือโปรแกรมดังกล่าวได้ นั่นหมายถึงโปรแกรมที่พัฒนาขึ้นและสร้างเป็น Smart Contract จะมีแค่เวอร์ชันเดียว



# Ethereum Virtual Machine (EVM)



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



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

# Smart Contract คืออะไร

## ความหมายที่ 2

- เป็นการเก็บ Source Code ไว้ในแต่ละ Block ที่อยู่ใน Blockchain
- **รันโค้ดหรือคำสั่งที่เก็บภายใน Block ได้**
- โค้ดที่ถูกเก็บใน Block ไม่สามารถแก้ไขได้จึงเรียกว่า สัญญาอัจฉริยะ
- มีกระบวนการตรวจสอบหรือ Mining Block ได้
- ใช้ภาษา Solidity ในการสร้าง Smart Contract



# อะไรทำให้ได้ครั้น ใน Blockchain ได้

ethereum  
ETHERBROWSER  
PEER-TO-PEER MESSAGING  
GENERALIZED BLOCKCHAIN  
PROGRAM ANYTHING

## Vitalik Buterin

ผู้สร้างและคิดค้นโปรเจกต์ Ethereum



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



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

ใน Ethereum Blockchain จะมีส่วนที่เรียกว่า

# Ethereum Virtual Machine (EVM)

สำหรับใช้รันโปรแกรมใน Blockchain  
ซึ่งได้แนวคิดมาจากภาษา Java



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



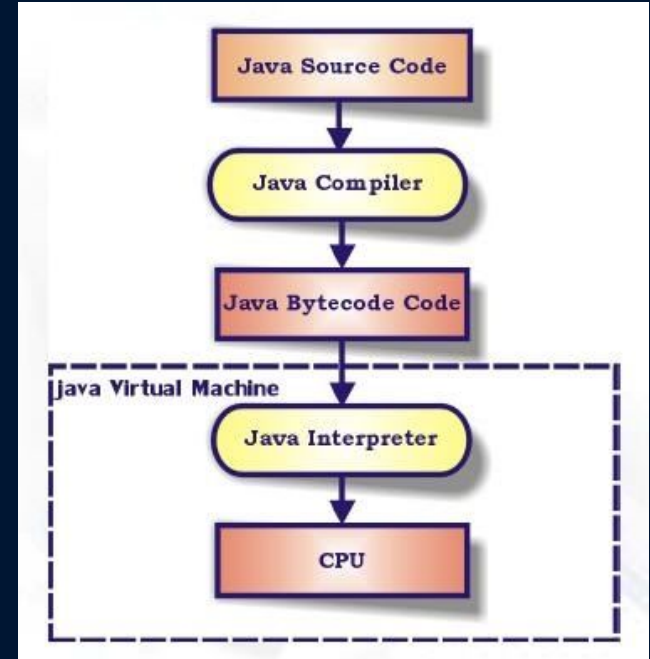
<https://www.facebook.com/KongRuksiamTutorial/>



# แนวคิดของภาษา Java

ภาษา Java มีแนวคิด คือจะมีการเปลี่ยนคำสั่งที่ถูกเขียนทั้งหมดนั้น แปลงเป็น Java Byte Code

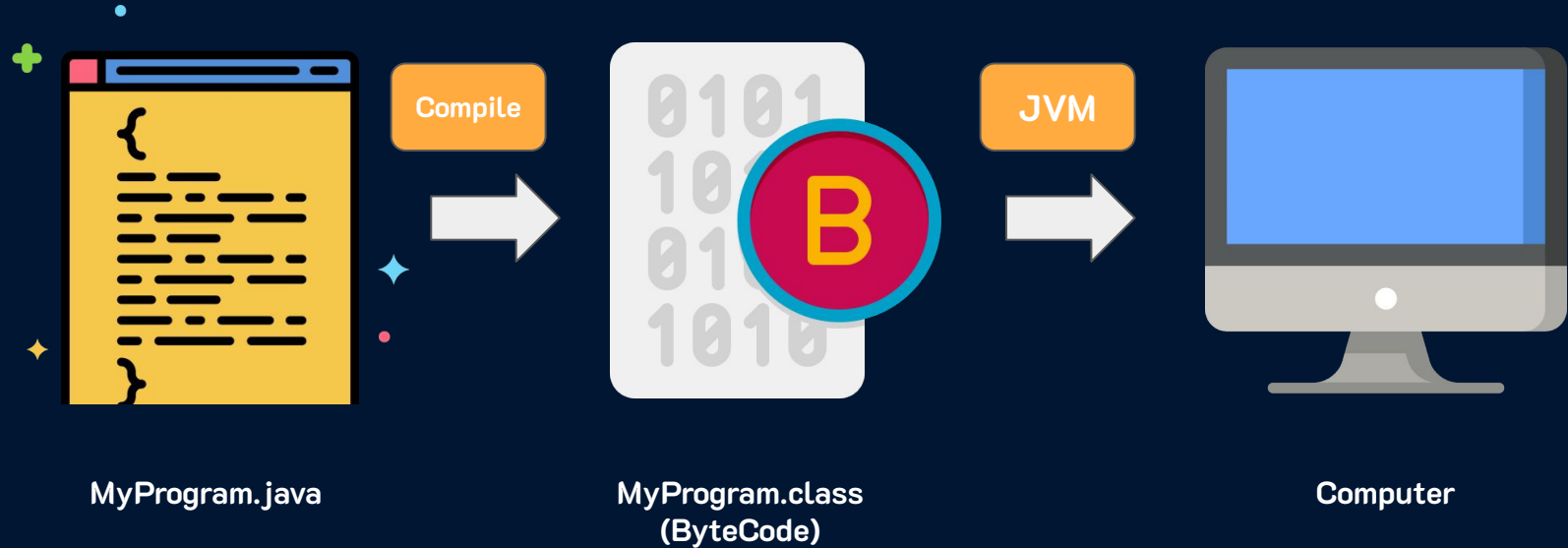
เมื่อถึงเวลาที่ต้องทำงานจริงในคอมพิวเตอร์ Java Byte Code จะถูกนำไปคอมไพล์อีกครั้งให้เป็นภาษาเครื่องเฉพาะอุปกรณ์ชนิดนั้นๆ โดยการคอมไพล์ในครั้งนี้จะใช้สิ่งที่เรียกว่า **Java Virtual Machine (JVM)** เพื่อให้โปรแกรมที่ถูกพัฒนาด้วยภาษา Java นั้นสามารถรองรับการทำงานได้ใน Platform นั้นๆ ได้



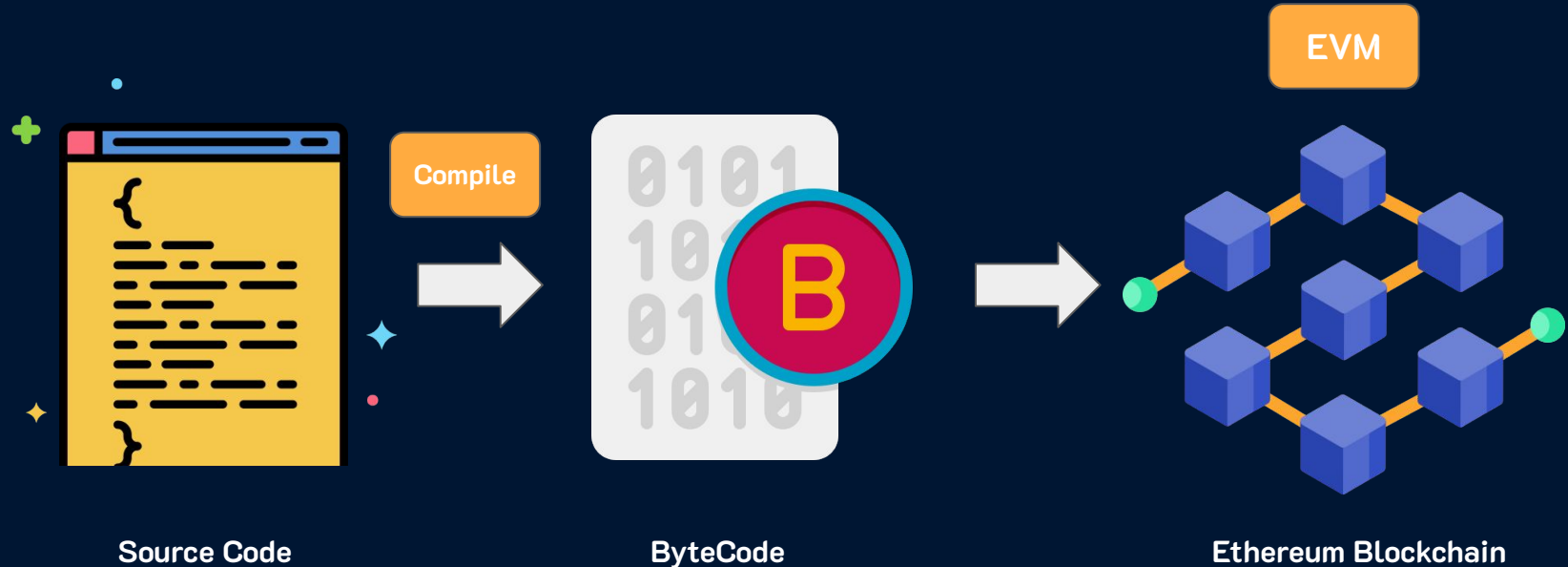
# แนวคิดของภาษา Java

ใน Ethereum อยากให้มีโปรแกรมที่สามารถทำงานและรันใน Blockchain Network ได้ (Smart Contract) จึงได้คิดค้น Ethereum Virtual Machine (EVM) ขึ้นมา โดยอาศัยแนวคิดของ ภาษา Java โดยใช้ Java Virtual Machine (JVM) นั้นเอง

# Java Virtual Machine (JVM)



# Ethereum Virtual Machine (EVM)



Java	Smart Contract
Java Virtual Machine (JVM)	Ethereum Virtual Machine (EVM)
ภาษา Java (.java)	ภาษา Solidity (.sol)
เขียนโปรแกรมเชิงวัตถุ	เขียนโปรแกรมเชิงวัตถุ
Class	Contract
คอมไพล์แล้วได้ Bytecode	คอมไพล์แล้วได้ Bytecode และนำไปเก็บบน Blockchain
ไม่มีการ Verify	มีการ Verify ผ่าน License
ไม่เสียค่าธรรมเนียม	เสียค่าธรรมเนียม (Gas)



Java	Smart Contract
Java Virtual Machine (JVM)	Ethereum Virtual Machine (EVM)
ภาษา Java (.java)	ภาษา Solidity (.sol)
เขียนโปรแกรมเชิงวัตถุ	เขียนโปรแกรมเชิงวัตถุ
Class	Contract
คอมไพล์แล้วได้ Bytecode	คอมไพล์แล้วได้ Bytecode และนำไปเก็บบน Blockchain
ไม่มีการ Verify	มีการ Verify ผ่าน License
ไม่เสียค่าธรรมเนียม	<b>เสียค่าธรรมเนียม (Gas)</b>



ต้องมีพื้นฐานอะไรบ้าง ?

# การเขียนโปรแกรมเชิงวัตถุ

## (Object Oriented Programming)



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

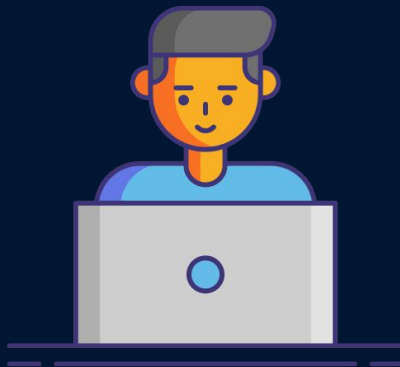


<https://www.facebook.com/KongRuksiamTutorial/>



# ความรู้พื้นฐานที่สำคัญ

- Class , Method
- Constructor
- Access Modifier
- อื่นๆ



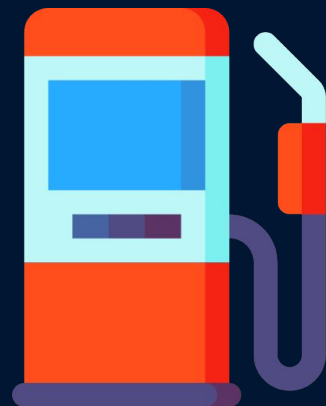
## MyClass.java

```
class MyClass {  
  
    //statement  
  
}
```

## MyContract.sol

```
contract MyContract {  
  
    //statement  
  
}
```

รู้จักกับ Gas



# Gas คืออะไร

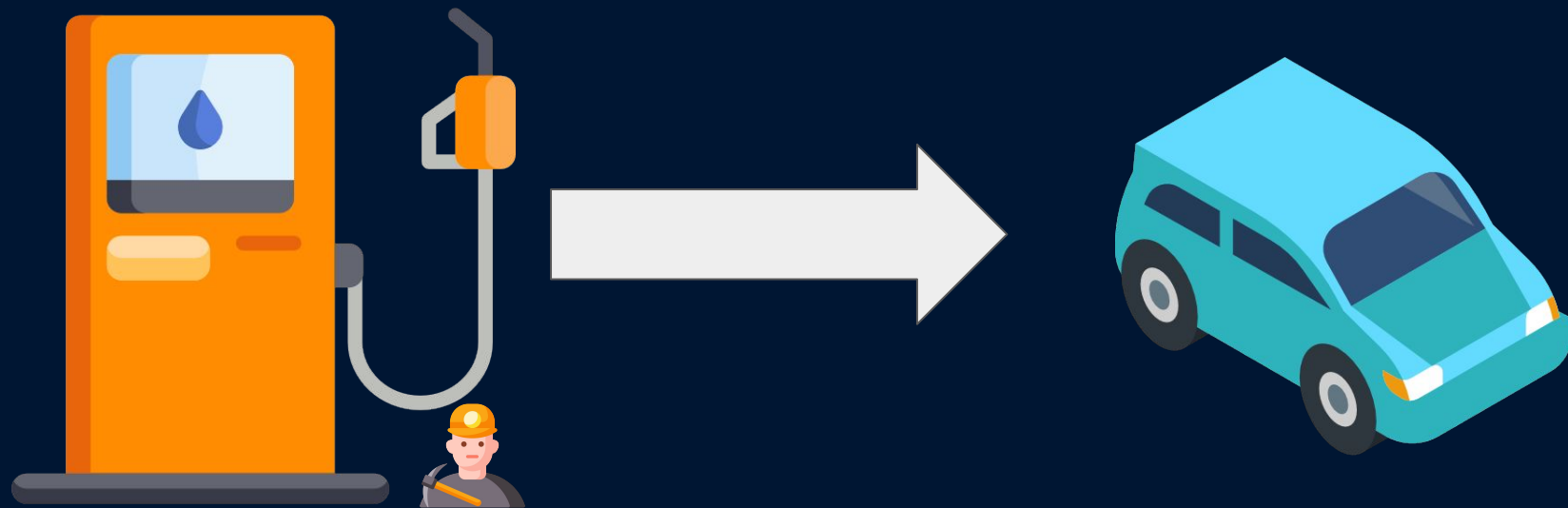
- การนำเอา Smart Contract ไปรันบน Ethereum นั้นต้องจ่ายเงินหรือค่าธรรมเนียมที่เรียกว่า “ Gas ”
- Gas คือ สิ่งที่ถูก Smart Contract ที่ทำงานบน Ethereum ต้องจ่าย เนื่องจากการนำ Smart Contract ขึ้นไปในระบบนั้นต้องมีการตรวจสอบ โดย Miner เช่น การตรวจสอบธุรกรรมต่างๆ ซึ่งผลตอบแทนของ Miner ก็คือค่าธรรมเนียมหรือ Gas ที่จ่ายไปนั่นเอง

# Gas คืออะไร

- แต่ละ Smart Contract มีการใช้พลังงานในการประมวลผล (Mining) ที่ไม่เท่ากัน ดังนั้นค่าธรรมเนียมหรือ Gas ก็จะไม่เท่ากันด้วย
- Gas จะแบ่งออกเป็น 2 รูปแบบ คือ
  - Gas Price
  - Gas Limit



# Gas คืออะไร



# Gas Price



- เป็นการกำหนดค่าธรรมเนียมหรือ Gas ได้ตามความต้องการ เช่น อยากให้ระบบทำงานเร็วขึ้น ใช้พลังงานเยอะขึ้น ก็จะเสียค่า Gas เยอะขึ้นตามไปด้วย เป็นต้น



# Gas Limit



- เป็นการกำหนดขอบเขตค่าธรรมเนียมหรือ Gas ไม่ให้ใช้พลังงานเกินที่กำหนดไว้ ถ้าใช้เกิน ระบบก็จะหยุดทำงานทันที ซึ่งค่าธรรมเนียมที่ต้องจ่ายก็จะคิดตามจริง เช่น กำหนดขอบเขตไว้ 300 แต่ใช้ไปแค่ 200 ก็จะจ่ายแค่เพียง  $200 \times \text{Gas Price}$  นั้นเอง





# Cryptocurrency Wallet



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



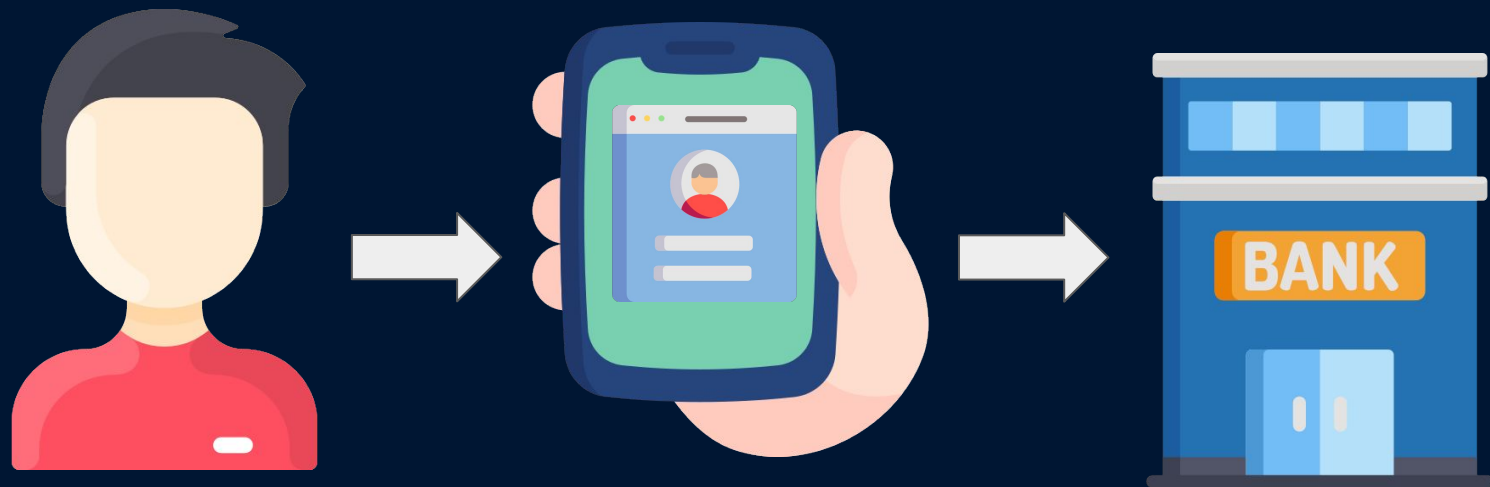
<https://www.facebook.com/KongRuksiamTutorial/>

# กระเป๋ากै๊บเหรียญคริปโต

# การเก็บเงินในระบบแบบเดิม

การทำธุรกรรมทางการเงินในปัจจุบัน เช่น ฝากเงิน โอน หรือ ถอนเงิน ต้องดำเนินการผ่านตัวกลาง คือ ธนาคารเป็นผู้ให้บริการ ผ่านแอปพลิเคชันด้านการเงิน ซึ่งผู้ใช้บริการ (User) จะมีการยืนยันตัวตนในการเข้าสู่ระบบผ่าน **Username / Password** โดยใช้**เลขที่บัญชี**ในการดำเนินการธุรกรรมต่างๆภายในแอปพลิเคชันดังกล่าว ซึ่งสิทธิในการเข้าถึงข้อมูลนั้นจะอยู่ที่ผู้ใช้บริการทั้งหมด

# การเก็บเงินในระบบแบบเดิม

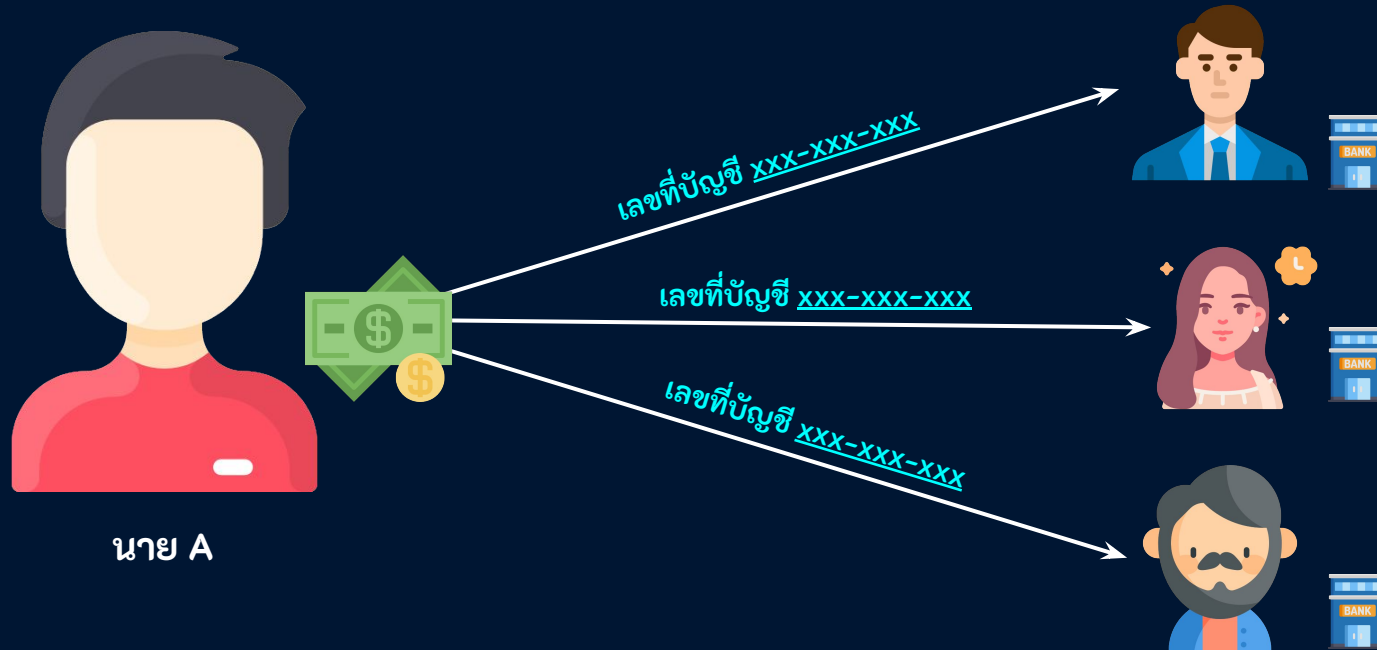


Username / Password

# การโอนเงินในระบบแบบเดิม



# การโอนเงินในระบบแบบเดิม



# กระเป๋าเก็บเหรียญคริปโต (Crypto Wallet)

คือ ซอร์ฟแวร์ที่ใช้สำหรับติดต่อกับ Crypto Network สำหรับการโอนหรือรับเหรียญคริปโตซึ่งในกระเป๋า (Wallet) มีส่วนประกอบอยู่ 2 ส่วนได้แก่

1. Public key หมายถึง ที่อยู่หรือหมายเลขบัญชี (Address)
2. Private key หมายถึง key สำหรับยืนยันการเป็นเจ้าของบัญชีหรือเจ้าของกระเป๋า



# กระเป๋าเก็บเหรียญคริปโต (Crypto Wallet)

1. **Private Key (Username / Password)** เป็นคีย์ที่เจ้าของกระเป๋าจะต้องเก็บไว้เป็นความลับ สำหรับยืนยันตัวตนว่าเป็นเจ้าของกระเป๋าและสำหรับสร้าง public key ห้ามให้คนอื่นรู้ ถ้า private key หายเงินในกระเป๋าก็จะหายไปด้วย
2. **Public Key (Address / หมายเลขบัญชี)** เป็นคีย์ที่เจ้าของกระเป๋าสามารถบอกคนอื่นได้ เมื่อคนอื่นรู้ Public Address ก็สามารถที่จะโอนเหรียญคริปโตมายังกระเป๋าหรือ Wallet ตาม Address ดังกล่าวได้



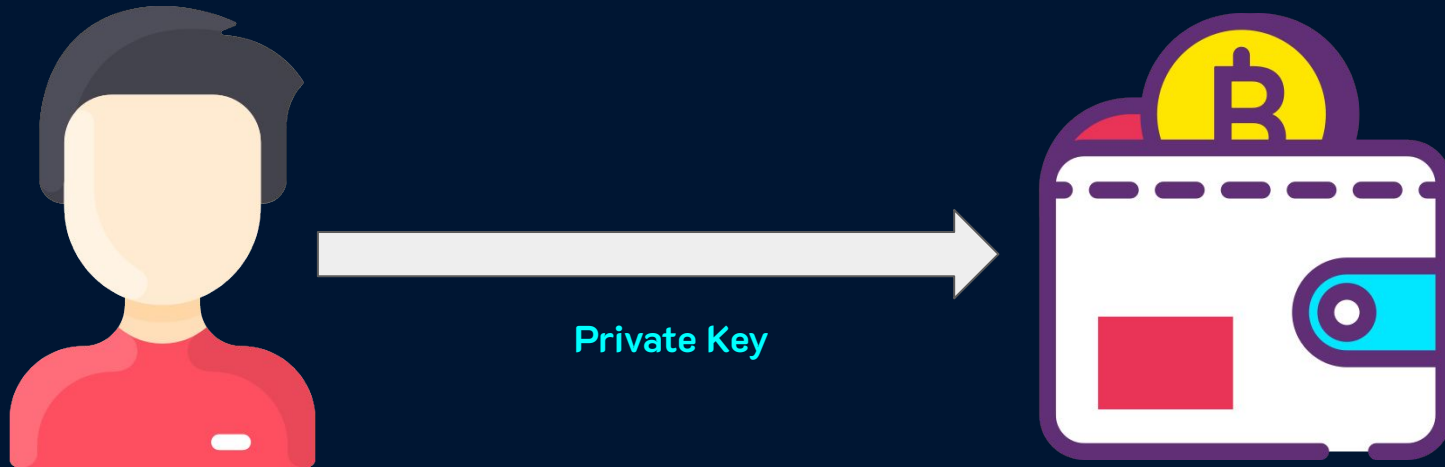


# กระเป๋าเก็บเหรียญคริปโต (Crypto Wallet)

“ ภายใน 1 Wallet สามารถมีได้หลาย Public Address ไม่ว่าจะโอนไปที่ Public Address ใด เงินทั้งหมดที่โอนมาก็จะเข้ามาที่ Wallet เดียวกัน ”



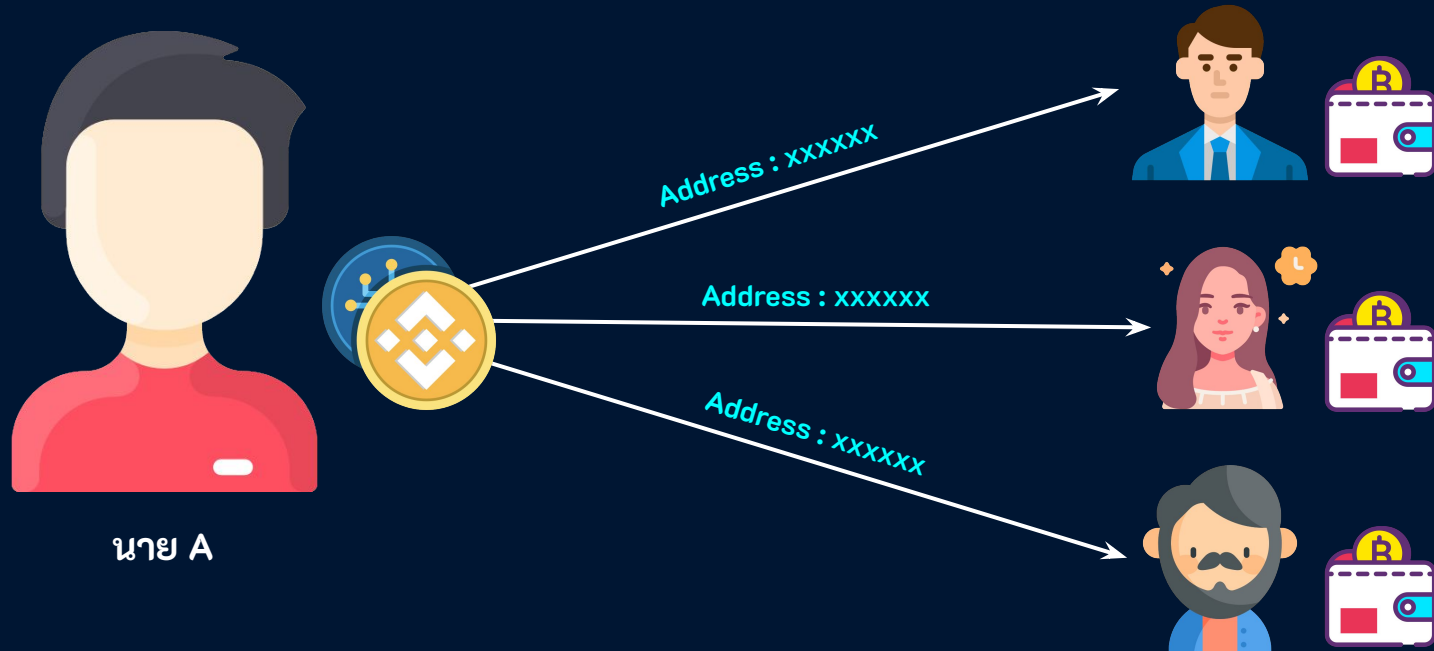
# กระเป๋าเก็บเหรียญคริปโต (Crypto Wallet)



# กระเป๋าเก็บเหรียญคริปโต (Crypto Wallet)



# กระเป๋าเก็บเหรียญคริปโต (Crypto Wallet)



# Remix – Ethereum IDE



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



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

# ไวยากรณ์ภาษา Solidity

# คำอธิบาย (Comment)

วิธีที่ 1 โดยใช้เครื่องหมาย Slash ( / ) ใช้ในการอธิบายคำสั่งสั้นๆในรูปแบบบรรทัดเดียว

วิธีที่ 2 เขียนคำอธิบายไว้ในเครื่องหมาย /\* ... \*/ ใช้ในการอธิบายคำสั่งยาวๆหรือแบบหลายบรรทัด



# 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 ตัวแปรที่ทำงานในฟังก์ชัน
3. Global variable คือ ตัวแปรที่ใช้รับค่าข้อมูลเกี่ยวกับ Blockchain เช่น `msg.sender` เป็นต้น





# ชนิดข้อมูล (Type)

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



# Access Modifier

คือ รูปแบบการเข้าถึงข้อมูลหรือคุณสมบัติที่อยู่ใน Smart Contract มีประโยชน์อย่างมากในเรื่องของการกำหนดระดับการเข้าถึง สิทธิในการใช้งาน การซ่อนข้อมูล และอื่น ๆ ซึ่งประกอบด้วย

- **public** เป็นการประกาศระดับการเข้าถึงที่เข้มงวดน้อยที่สุดหรือกล่าวได้ว่าบุคคลภายนอกหรือใคร ๆ ก็สามารถเข้าถึงและเรียกใช้งานได้
- **private** เป็นการประกาศระดับการเข้าถึงที่เข้มงวดที่สุดกล่าวคือ จะมีแต่ Contract ของตัวมันเองเท่านั้นที่มีสิทธิ์ใช้งานได้



# Access Modifier

## โครงสร้างคำสั่ง

- [type , access\_modifier , ชื่อตัวแปร;]
- [type , access\_modifier , ชื่อตัวแปร = ค่าเริ่มต้น;]

## ตัวอย่าง

- string public name; //public
- uint private \_balance; //private



# Constructor

คือ ฟังก์ชันที่จะถูกเรียกใช้และทำงานอัตโนมัติเพียงครั้งเดียว  
ในตอนเริ่มต้น รัน Smart Contract หรือทำการ Deploy

# ฟังก์ชัน (Function)

เป็นกลุ่มคำสั่งหรือการแบ่งส่วนการทำงานที่อยู่ใน Smart Contract

โครงสร้าง :

```
function ชื่อฟังก์ชัน(<parameter types>) {public|private}  
[pure|view|payable] [returns (<return types>)]
```

# ฟังก์ชัน (Function)

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



# ตัวอย่างฟังก์ชัน (Function)

- `function deposit(){`  
    `// private function`  
    `}`
- `function deposit() public {`  
    `//public function`  
    `}`
- `function deposit public (uint amount){ } // parameter`



# ตัวอย่างฟังก์ชัน (Function)

```
function getBalance() public view returns (uint balance){  
    return _balance; // ดึงค่า _balance จาก storage  
}
```

```
function getBalance() public pure returns (uint balance){  
    return 50; // ค่าคงที่  
}
```





# เอกสารประกอบการสอน

<https://github.com/kongruksiamza/learning-smart-contract>




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



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

# เข้าเรียนเนื้อหา



ปูพื้นฐานการสร้าง Smart Contract  
บล็อกเชนและการสร้างสัญญาอัจฉริยะ  
เล่นทั้งหมด

สร้าง Smart Contract ด้วย Solidity | สำหรับผู้เริ่มต้น [2022]

วิดีโอ 18 รายการ • ภาควิdeo 223 ครั้ง • อัปเดตแล้ววันนี้

ผู้เรียนต้องมีพื้นฐาน Blockchain ด้วยภาษา Python เบื้องต้นมาก่อน

KongRuksiam Official

ติดตามแล้ว

1 Smart Contract & Solidity ตอนที่ 1 - เทคโนโลยี Blockchain  
KongRuksiam Official 9:49

2 Smart Contract & Solidity ตอนที่ 2 - รู้จักกับ Ethereum  
KongRuksiam Official 6:25

3 Smart Contract & Solidity ตอนที่ 3 - Decentralized Applications (Dapps)  
KongRuksiam Official 3:37

4 Smart Contract & Solidity ตอนที่ 4 - Smart Contract คืออะไร  
KongRuksiam Official 10:01

5 Smart Contract & Solidity ตอนที่ 5 - Ethereum Virtual Machine (EVM)  
KongRuksiam Official 8:28



สแกนเพื่อเข้าเรียน