



# Final Class Work

## Part 1 - สร้างไฟล์

ให้สร้างไฟล์ .txt ด้วยวิธีการใดก็ได้ ตั้งชื่ออะไรก็ได้ที่จะคอยเก็บ “จำนวนเหรียญคริปโตที่น้องมีทั้งหมด” โดยให้น้องใส่ข้อมูลลงไปไฟล์นั้นดังนี้

แต่ละบรรทัดจะมีอยู่สองส่วนคือ สัญลักษณ์เหรียญ เช่น BTC, ETH, DOGE เป็นตัวพิมพ์ใหญ่ และปริมาณเหรียญนั้นมีเป็นเลขทศนิยม ซึ่งสองส่วนนี้จะถูกคั่นด้วยเครื่องหมาย tab (“\t”)

ตัวอย่าง ไฟล์ vetit\_crypto.txt

```
BTC 0.4  
ETH 1.5  
ADA 16.8
```

ถ้าใครอยากลองเล่นจริง ๆ ก็ลองใส่ปริมาณที่มีได้เลยครับ

## Part 2 - CryptoClass

ให้ทำการเทรดคริปโตดังต่อไปนี้ (ไม่ได้เทรดจริงหรอกนะครับ)

1. สร้างไฟล์ python ขึ้นมา พร้อมกับสร้าง class ที่รับค่าเริ่มต้นหนึ่งค่าก็คือ path\_to\_crypto หรือเป็น path ที่เอาไว้ใช้เปิดไฟล์นี้นั่นเอง
2. ให้ assert โดยใช้ module os ว่ามีไฟล์ที่ใส่เข้ามาอยู่จริงหรือไม่ ถ้าไม่มีอยู่จริงให้แจ้งเตือนจากโปรแกรมทั้งหมดทันทีโดยห้ามทำงานต่อเด็ดขาด
3. เมื่อคลาสและ path ถูกใส่และสร้างขึ้นมาแล้วให้อ่านไฟล์เก็บปริมาณเหรียญที่เราใส่ (ใน \_\_init\_\_) แล้วนำไปใส่ใน dictionary โดยให้มี key คือสัญลักษณ์เหรียญต่าง ๆ และ value คือปริมาณเหรียญที่เราใส่จากการอ่านไฟล์

```
# ตัวอย่าง attribute ที่เก็บปริมาณและเหรียญไว้  
self.all_coins = {"BTC": 1.2, "ETH": 7.1} # Dictionary
```

4. ให้เพิ่ม attribute ของ class ไปอีกหนึ่งอันคือ

```
self.api_link = "https://api.binance.com/api/v3/ticker/price?symbol="
```

ลิงก์นี้จะถูกเก็บไว้ใช้ในการดึงราคาเหรียญ ณ ปัจจุบันผ่าน Binance API

5. สร้าง method ของ class ที่คืนค่าออกมาเป็นสัญลักษณ์เหรียญทั้งหมดที่เรามีเป็น list

```
# หน้าตาผลลัพธ์  
["BTC", "ETH"]
```

6. สร้าง method ที่รับ arguments เป็นสัญลักษณ์เหรียญ แล้วคืนค่ากลับมาเป็นอัตราและเปลี่ยนระหว่างเหรียญนั้นกับเหรียญ USDT (ดอลลาร์สหรัฐ)

- ให้ใช้ module requests ในการดึงค่าอัตราแลกเปลี่ยนนี้ โดยใช้ `requests.get(API_LINK)`
- ให้ใช้ API link ที่ประกาศไปข้างต้น แต่ให้เติมต่อท้ายลิงก์นั้นด้วยสัญลักษณ์ของเหรียญทั้งสองเหรียญที่ต้องการถามอัตราแลกเปลี่ยน เช่น

```
# หากเราต้องการหาว่า 1 BTC มีค่าเท่ากับกี่ USDT นี่คือ link ในการขอ API  
self.api_link = "https://api.binance.com/api/v3/ticker/price?symbol=BTCUSDT"
```

- เมื่อได้ค่าออกมาต้องแปลงเป็นวัตถุ json โดยนำสิ่งที่ได้มาจาก `requests.get()` ไปใช้ method `.json()` ต่อได้เลย ซึ่ง json object นี้จะเปรียบเหมือน dictionary ก็ไม่ผิดอะไรนะ

```
# ถ้า requests ให้ค่ากลับมาเท่าไร ก็เอาไปทำให้เป็น json เสียก่อน  
result = requests.get(API_LINK)  
json_result = result.json()
```

7. สร้าง method ที่หาจำนวนเงินในหน่วยดอลลาร์สหรัฐทั้งหมดของเหรียญที่เรามีอยู่ (Net worth)

8. สร้าง method ที่**ซื้อเหรียญเพิ่ม** โดยได้รับ 2 arguments: coin: str, amount: float เมื่อทำการเรียกใช้ method นี้ให้เขียนกับไฟล์ที่เก็บปริมาณเหรียญทั้งหมด เปลี่ยนเป็นปริมาณใหม่ที่รวมเหรียญที่ซื้อแล้ว เช่น

```
# vetit_crypto.txt ก่อนที่จะใช้ method ในการซื้อ  
BTC 1.2  
ETH 0.7  
# vetit_crypto.txt หลังจากซื้อเหรียญ BTC จำนวน 0.02 เหรียญเพิ่ม  
BTC 1.22  
ETH 0.7  
# vetit_crypto.txt หลังจากซื้อเหรียญ ADA จำนวน 4 เหรียญเพิ่ม
```

```
BTC 1.22
ETH 0.7
ADA 4.0
```

9. ในทางกลับกันให้สร้าง method ที่เอาไว้ **ขายเหรียญทิ้ง** พร้อมกับปรับค่าในไฟล์เก็บเหรียญค  
รีปโททั้งหมดเหมือนกับข้อ 8.
10. ในแต่ละที่ของโค้ดเราทั้งหมดนี้ มีหลายจุดที่ควรมี assert, try.. except.. หรือแม้กระทั่ง  
raise ให้ลองใส่เพื่อทำให้มั่นใจได้ว่าการทำงานถูกต้อง