



## Functions

**Part 1** ให้เขียนฟังก์ชันที่ประกอบด้วย features ดังต่อไปนี้ โดยฟังก์ชันนั้นจะรวมทุก features ใน function เดียวกันหรือแยกหลายฟังก์ชันก็ได้

- Higher order function
- Closure
- Default parameter และ/หรือ Rest Parameters และ/หรือ Arguments Parameters และ/หรือ destructuring

**Part 2** ให้ export functions ใน Part 1 และจัดการ import เพื่อใช้ประโยชน์ฟังก์ชันที่ได้ ออกแบบไว้ใน Part 1



## ข้อกำหนด

1. ให้ copy source code จาก VSCode มาวางใน pdf file **พร้อมคำอธิบาย**การใช้คำสั่งสำคัญต่าง ๆ กำกับในแต่ละบรรทัดของโปรแกรม
2. ให้ทำการทดสอบด้วยเคสข้อมูลที่แตกต่างกันอย่างน้อย 3 เคส
3. ให้ capture output จาก VSCode มาวางใน pdf file ในข้อ 1
4. ให้ใส่ git link ใน pdf file ด้วย

**All file > <https://github.com/salmonindaeyo/INT201G8/tree/master/Assignment05>**

# Part1

## 1. Higher-Order Functions

```
//1.Higher-Order Functions >> รับ param เป็น fn || return เป็น fn
let listCoffee = ['Americano','Cappuccino','Espresso','Latte','Mocha'];
//Higher-Order Functions
function excute(array,fn,other){
    return fn(array,other);
}
//Other function
//นับจำนวนสมาชิกใน array
let countElement = (x)=> x.length;
//เพิ่มสมาชิกใน array
let addElement = function(array,newProduct){
    array.push(newProduct);
    return array;
}
//ค้นหาสมาชิกใน array ตาม keyword
let searchName = function(array,keyword){
    return array.find((name)=>name === keyword);
}
console.log(excute(listCoffee,countElement)); //มีทั้งหมด 5 ชิ้น
console.log(excute(listCoffee,addElement,'Tea')); //เพิ่ม Tea ใน
listCoffee
console.log(excute(listCoffee,searchName,'Espresso')); //ค้นหาชื่อ Espresso
ใน listCoffee
console.log(excute(listCoffee,countElement)); //มีทั้งหมด 6 ชิ้น
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

Code



[Running] node

"c:\Users\ST\Desktop\IT2021\INT201\_Client-side\MyProject\_int201\INT201-631:\myTestJS\tempCodeRunnerFile.js"

5

[ 'Americano', 'Cappuccino', 'Espresso', 'Latte', 'Mocha', 'Tea' ]

Espresso

6

[Done] exited with code=0 in 0.11 seconds

## 2. Closure and Rest parameters

```
//2.Closure + Rest parameters
//Outer Function
function createProduct(){
  let products = []; //สร้าง Array เปล่า ไว้เก็บcollectionของสินค้า
  //Inner Function
  function addNewProduct(ids,name,price,...description){
    //...'...description' คือ rest parameter
    //set ค่า parameter ทั้งหมดลงใน object
    let newProduct = {};
    newProduct.id = ids;
    newProduct.name = name;
    newProduct.price = Number(price);
    newProduct.detail = description;

    //เช็คว่าสินค้าใหม่ที่เพิ่มนี้มี id ที่ซ้ำกันหรือไม่ ถ้าไม่ซ้ำจะเพิ่มสินค้านี้ใน products
    let findProduct = products.find(({id})=>id === ids);
    if(findProduct===undefined){
      products.push(newProduct);
    }
    return products; //return สินค้าทั้งหมด
  }
  return addNewProduct; //<---Closure : เรียกชื่อ Inner Function
  //Closure จะช่วยให้เพิ่มสินค้าใหม่ใน array เดิมได้ (เป็นค่าสะสมไปเรื่อยๆ)
}

//การเรียกใช้ Closure จะเรียกจาก Outer function
let iPhone = createProduct();
iPhone(110,'iphone 11',24900,'RAM 4GB', 'ROM 64GB', '4G', 'Pink');
iPhone(110,'iphone 11',24900,'RAM 4GB', 'ROM 64GB', '4G', 'Red'); //สินค้านี้
//ไม่ถูกเพิ่ม เพราะเลข id ซ้ำ
iPhone(120,'iphone 12',29900,'RAM 4GB','ROM 64GB','4G','Grey');
iPhone(122,'iphone 12 Pro',36100,'RAM 6GB','ROM 128GB','4G','Black');
console.log(iPhone(132,'iphone 13 Pro',38900,'RAM 6GB','ROM
128GB','5G','Blue')); //สุดท้ายมีสินค้าทั้งหมด 4 ชนิด
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

Code



[Running] node

"c:\Users\ST\Desktop\IT2021\INT201\_Client-side\MyProject\_int201\INT201-6313  
\myTestJS\tempCodeRunnerFile.js"

```
[
  {
    id: 110,
    name: 'iphone 11',
    price: 24900,
    detail: [ 'RAM 4GB', 'ROM 64GB', '4G', 'Pink' ]
  },
  {
    id: 120,
    name: 'iphone 12',
    price: 29900,
    detail: [ 'RAM 4GB', 'ROM 64GB', '4G', 'Grey' ]
  },
  {
    id: 122,
    name: 'iphone 12 Pro',
    price: 36100,
    detail: [ 'RAM 6GB', 'ROM 128GB', '4G', 'Black' ]
  },
  {
    id: 132,
    name: 'iphone 13 Pro',
    price: 38900,
    detail: [ 'RAM 6GB', 'ROM 128GB', '5G', 'Blue' ]
  }
]
```

[Done] exited with code=0 in 0.123 seconds

### 3. Default parameter

```
//3.Default parameter >> กำหนดค่าใน param
//function คำนวณราคารวมของสินค้า
function findTotal(price=0,quantity=1){ //ถ้าไม่มีส่งค่า parameter มา จะ
กำหนดค่าเริ่มต้นให้ price=0 และ quantity=1
    return price*quantity;
}
console.log(findTotal(150,4)); //600 ()
console.log(findTotal(200)); //200 (มาจาก 200*1 โดย price=200 แต่ 1
คือ ค่าเริ่มต้นของ quantity)
console.log(findTotal()); //0 (มาจาก 0*1 ที่ได้จากค่าเริ่มต้นของ
price*quantity)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL Code

```
[Running] node
"c:\Users\ST\Desktop\IT2021\INT201_Client-side\MyProject_int201\INT201-631:
\myTestJS\tempCodeRunnerFile.js"
600
200
0

[Done] exited with code=0 in 0.149 seconds
```

### 4. Arguments (จะลองเอาไปรวมกับข้อ3 เลยก็ได้นะ)

```
//function เพิ่มสินค้าลงตะกร้า
function basket() {
    let userBasket = []; //สร้าง Array เปล่า ไว้เตรียมเก็บของ
    //Arguments Parameters
    function addToBasket() {
        //เพิ่มสินค้าทั้งหมดลงตะกร้า โดยการเรียกดู arguments parameter ทั้งหมดมาเพิ่มใน
userBasket
        for (let i = 0; i < arguments.length; i++) {
            userBasket.push(arguments[i]);
        }
        return userBasket;
    }
    return addToBasket;
}
const user = basket();
console.log(user('Phone13', 'Macbook Pro', 'Apple Watch', 'Ipad mini'));
```

```
[Running] node "/Users/n11/Documents/JavascriptProjects/Bas
[ 'Phone13', 'Macbook Pro', 'Apple Watch', 'Ipad mini' ]

[Done] exited with code=0 in 0.06 seconds
```

## **Part2**

### **Export**

```
//Part2.js
//-----Function part 1 Higher-Order Functions -----
export function excute(array,fn,other){
    return fn(array,other);
}
export let countElement = (x)=> x.length;
export let addElement = function(array,newProduct){
    array.push(newProduct);
    return array;
}
export let searchName = function(array,keyword){
    return array.find((name)=>name === keyword);
}
//-----Function part 2 - Closure + Rest parameters -----
export function createProduct(){
    let products = [];
    function addNewProduct(ids,name,price,...description){
        let newProduct = {};
        newProduct.id = ids;
        newProduct.name = name;
        newProduct.price = Number(price);
        newProduct.detail = description;
        let findProduct = products.find(({id})=>id === ids);
        if(findProduct===undefined){
            products.push(newProduct);
        }
        return products;
    }
    return addNewProduct;
}
```

```

//-----Function part 3 Default parameter -----
export function findTotal(price=0,quantity=1){
    return price*quantity;
}

//-----Function part 4 Arguments in function -----
export function basket() {
    let userBasket = [];
    function addToBasket() {
        for (let i = 0; i < arguments.length; i++) {
            userBasket.push(arguments[i]);
        }
        return userBasket;
    }
    return addToBasket;
}

```

## Import

```

//main.js
import {excute,countElement,addElement,searchName,
    createProduct,
    findTotal,
    basket} from './Part2.js'

//Function part 1
let listCoffee = ['Americano','Cappuccino','Espresso','Latte','Mocha'];
console.log(listCoffee);
console.log(excute(listCoffee,countElement));
console.log(excute(listCoffee,addElement,'Tea'));
console.log(excute(listCoffee,searchName,'Espresso'));
console.log(excute(listCoffee,countElement));

//Funciton part 2
let iPhone = createProduct();
iPhone(110,'iphone 11',24900,'RAM 4GB', 'ROM 64GB', '4G', 'Pink');
iPhone(110,'iphone 11',24900,'RAM 4GB', 'ROM 64GB', '4G', 'Red');
iPhone(120,'iphone 12',29900,'RAM 4GB','ROM 64GB','4G','Grey');
iPhone(122,'iphone 12 Pro',36100,'RAM 6GB','ROM 128GB','4G','Black');
console.log(iPhone(132,'iphone 13 Pro',38900,'RAM 6GB','ROM
128GB','5G','Blue'));

```

```
//Function part 3
console.log(findTotal(150,4));
console.log(findTotal(200));
console.log(findTotal());

//Funciton part 4
const user = busket();
user('Phone13');
user('Macbook Pro');
console.log(user('Apple Watch', 'Ipad mini'));
```