

## Part 1:

### Higher order functions

```
1 //High Order Functions
2 function calculate(opera, FirstValue, numbers) { //ฟังก์ชัน calculate รับพารามิเตอร์ opera, FirstValue, numbers
3   let total = FirstValue; //เก็บ FirstValue ในส่วนแปร total
4   for (const number of numbers) { //วนลูปทุกตัวในส่วนแปร number
5     total = opera(total, number); //เรียกใช้ฟังก์ชันจากพารามิเตอร์ opera โดยในฟังก์ชันมีพารามิเตอร์เป็น total กับ number และ คืนค่า total
6   }
7   return total;
8 }
9 function sum(n1, n2) { //ฟังก์ชัน sum รับค่าพารามิเตอร์เป็น n1 กับ n2 และ คืนค่า n1 + n2
10   return n1 + n2;
11 }
12 function multiply(n1, n2) { //ฟังก์ชัน multiply รับค่าพารามิเตอร์เป็น n1 กับ n2 และ คืนค่า n1 * n2
13   return n1 * n2;
14 }
15 function minus(n1, n2) { //ฟังก์ชัน minus รับค่าพารามิเตอร์เป็น n1 กับ n2 และ คืนค่า n1 - n2
16   return n1 - n2;
17 }
18 function devine(n1, n2) { //ฟังก์ชัน devine รับค่าพารามิเตอร์เป็น n1 กับ n2 และ คืนค่า n1 / n2
19   return n1 / n2;
20 }
21
22 console.log(calculate(sum, 3, [1, 2, 4])); //เป็นการเรียกใช้ function calculate โดยรับ sum , 3 และ array ที่เก็บค่าของ [1,2,4] เข้าไปที่ parameter
23 //โดยหลังจาก function calculate ทำงาน ก็จะส่งค่าเข้าไปที่ function sum ต่อ เนื่องจาก parameter ที่รับมานั้น เกิดเป็น function
24 console.log(calculate(multiply, 1, [1, 2, 4]));
25
26
```

### Closure

```
1
2 //Closure
3 //export คือ การส่งออก function เพื่อนำไปสามารถไปใช้ให้ไฟล์อื่น
4 export function member(id, nickName, sex, age){ //สร้าง function ที่ใช้สร้าง/กำหนด property ให้ object
5   this._id = id
6   this._nickName = nickName
7   this._sex = sex
8   this._age = age
9 }
10
11 export function nightClub (mem){ //รับพารามิเตอร์ที่เป็น object
12   let club = []
13   mem._age >= 20 ? club.push(mem) : console.log('อายุของคุณไม่ถึงเกณฑ์ กรุณารออายุ 20 ปีบริบูรณ์แล้วกลับมาใหม่')
14   //เช็คเงื่อนไขว่า object mem ที่รับเข้ามามีค่ามากกว่าหรือเท่ากับ 20 ไหม
15   //ถ้าใช่ให้เพิ่ม mem นั้นลงใน array club ถ้าไม่จะแสดง string กลับไป
16   function addMember(newMember){
17     newMember._age >= 20 ? club.push(newMember) : console.log('อายุของคุณไม่ถึงเกณฑ์ กรุณารออายุ 20 ปีบริบูรณ์แล้วกลับมาใหม่')
18     //เช็คเงื่อนไขว่า object newMember ที่รับเข้ามามีค่ามากกว่าหรือเท่ากับ 20 ไหม
19     //ถ้าใช่ให้เพิ่ม newMember นั้นลงใน array club ต่อหลัง element เดิมที่มีอยู่/ถูกเก็บรักษาไว้ไม่ได้หายไปพร้อม function เมื่อจบการทำงาน
20     //ถ้าไม่แสดง string กลับไป
21     return `Members in the section : ${club.map(m => `${m._id}-${m._nickName}-${m._sex}-${m._age}`)}`
22     //ให้เห็นข้อความ และ ข้อมูลภายใน array โดยการใช้การ map โดยเปลี่ยนรูปแบบการแสดงผลจากเดิมที่เป็น object ให้เป็นตามที่กำหนดใน arrow function
23   }
24   return addMember //ให้คืน function ออกไป
25 }
26
27
28 //Default parameters
29 let products = [ //สร้างสินค้าเป็น array object ขึ้นมา
30   { productName: 'snack', price: 20 },
31   { productName: 'cola', price: 15 },
32   { productName: 'sandwich', price: 25 }
33 ]
34
35 function buy(item, amount = 1) { //รับ parameter เป็น item และ amount = 1
36   let name = products[item].productName //ให้ name เก็บชื่อสินค้า ที่ index ตาม item
37   let totalPrice = products[item].price * amount //ให้ totalPrice เก็บราคาสินค้า * จำนวนที่ซื้อ
38   return `${name} ${amount} ea price ${totalPrice} baht`
39 }
40 //ถ้าเรียกใช้งาน function แต่ไม่ได้ใส่ parameter เมื่อเรียก ค่า default จะถูกใช้เป็นค่า parameter
41 console.log(buy(2)) //ในตัวอย่าง ชื่อ item ที่ 2 แต่ไม่ได้ใส่ amount ก็จะให้ amount default เป็น 1 ซึ่งเรากำหนดให้เป็นขั้นต้นในการซื้อ
42
43 //ถ้าเรียกใช้งาน function แล้วใส่ค่า parameter เมื่อเรียก จะใช้ค่าที่ได้เป็น default parameter
44 console.log(buy(2, 5)) //ในตัวอย่าง ชื่อ item ที่ 2 แล้วใส่ amount ตัว default ก็จะตาม parameter ที่เราใส่ ก็คือ 5
45
46 console.log("\n")
47
```

## Default parameter

```
1 //สร้างสินค้าเป็น array object ขึ้นมา
2 let products = [
3   { productName: 'Doritos', price: 20 },
4   { productName: 'Coke', price: 15 },
5   { productName: 'Farm house', price: 25 },
6   { productName: 'M150', price: 10 },
7   { productName: 'Mentos', price: 10 }
8 ]
9
10 //Default parameters
11 function buy(item, amount = 1) { //รับ parameter เป็น item และ amount โดยให้ amount เป็น Default parameter = 1
12   let name = products[item].productName //ให้ name เก็บชื่อสินค้า ที่ index ตาม item
13   let totalPrice = products[item].price * amount //ให้ totalPrice เก็บราคาสินค้า * จำนวนที่ซื้อ
14   return `${name} ${amount} ea price ${totalPrice} baht`
15 }
16 //ถ้าเรียกใช้งาน function แต่ไม่ได้ใส่ parameter เมื่อเรียก ค่า default จะถูกใช้เป็นค่า parameter
17 console.log(buy(3)) //ในตัวอย่าง ชื่อ item ที่ 2 แต่ไม่ได้ใส่ amount ก็จะมี amount default เป็น 1 ซึ่งเรากำหนดให้เป็นขั้นต่ำในการซื้อ
18 //ถ้าเรียกใช้งาน function แล้วใส่ค่า parameter เมื่อเรียก จะใช้ค่าที่ใส่เป็น default parameter
19 console.log(buy(3, 13)) //ในตัวอย่าง ชื่อ item ที่ 2 แล้วใส่ amount ค่า default ก็จะตาม parameter ที่เราใส่ ก็คือ 5
20 console.log("\n")
```

## Rest parameters

```
22 //Rest parameters
23 function importOrder(...manyItem) { //รับ parameter หลายตัว
24   //reduce คือการเอาพารามิเตอร์แรก และพารามิเตอร์ที่สองเป็นค่าเริ่มต้นสำหรับเมธอดเพื่อทำงานต่อ
25   //โดยในตัวอย่างเราจะเอา prev กับ curr มารวมกัน แล้วคืนด้วย ,
26   return manyItem.reduce(function (prev, curr) {
27     return prev + ", " + curr;
28   });
29 }
30 //โดยเราจะใส่ parameter หรือ itemOrder ของเราเท่าไรก็ได้ ไม่จำกัด
31 console.log("The import order is ")
32 console.log(importOrder("Pepsi", "SevenUp", "Oreo", "Bento", "Pizza", "Chocobo"))
33 console.log("\n")
```

## Destructuring

```
35 //Destructuring
36 //Destruct ภายใน Arguments ของ Function ด้วยการใส่ ชื่อของ Properties ที่เราต้องการ ลงในวงเล็บ
37 function infoItem({productName, price}) { //ต้องการ productName กับ price
38   return `1 piece of ${productName}, price ${price} baht`
39 }
40 //แสดงข้อมูลสินค้าสินค้า Index ที่ 2
41 console.log(infoItem(products[2]))
```

## Part 2:

### Export functions

```
//export คือ การส่งออก function เพื่อนให้สามารถไปดัดแปลงใช้ได้ที่ไฟล์อื่น
export function member(id, nickName, sex, age){ //สร้าง function ที่ใช้สร้าง/กำหนด property ให้ object
    this._id = id
    this._nickName = nickName
    this._sex = sex
    this._age = age
}

export function nightClub (mem){ //รับพารามิเตอร์ที่เป็น object
    let club = []
    mem._age >= 20 ? club.push(mem) : console.log(`อายุของคุณไม่ถึงเกณฑ์ กรุณารออายุ 20 ปีบริบูรณ์แล้วกลับมาใหม่`)
    //เช็คเงื่อนไขว่า object mem ที่รับเข้ามามีค่ามากกว่าหรือเท่ากับ 20 ไหม
    //ถ้าใช่ให้เพิ่ม mem นั้นลงใน array club ถ้าไม่แสดง string กลับไป
    function addMember(newMember){
        newMember._age >= 20 ? club.push(newMember) : console.log(`อายุของคุณไม่ถึงเกณฑ์ กรุณารออายุ 20 ปีบริบูรณ์แล้วกลับมาใหม่`)
        //เช็คเงื่อนไขว่า object newMember ที่รับเข้ามามีค่ามากกว่าหรือเท่ากับ 20 ไหม
        //ถ้าใช่ให้เพิ่ม newMember นั้นลงใน array club ต่อหลัง element เดิมที่มีอยู่/ถูกเก็บรักษาไว้ไม่ได้หายไปพร้อมกับ function เมื่อจบการทำงาน
        //ถ้าไม่แสดง string กลับไป
        return `Members in the section : ${club.map(m => `${m._id}-${m._nickName}-${m._sex}-${m._age}`)}`
        //ให้เรีนข้อความ และ ข้อมูลภายใน array โดยใช้การ map โดยเปลี่ยนรูปแบบการแสดงผลจากเดิมที่เป็น object ให้เป็นตามที่กำหนดใน arrow function
    }
    return addMember //ให้เรีน function ออกไป
}
```

### Import And Test

```
1 //import
2 import {nightClub, member} from './A05.js' //การนำ function จากในไฟล์อื่นมาใช้ที่ไฟล์นี้
3
4
5 let newMember = nightClub(new member('001', 'boss','male',19))
6 //สร้างและส่ง object ที่เป็น member ไปที่ function nightClub จากนั้นเมื่อ function ทำงานเสร็จจะรีเทิร์น function addMember มาเก็บในตัวแปร newMember
7 console.log(newMember(new member ('002', 'jom','male',19)))
8 //สร้างและส่ง object ไปที่ function addMember ซึ่งถูกเก็บใน newMember จากนั้นนำผลลัพธ์ที่เกิดขึ้นมาแสดง
9 console.log(newMember(new member('049', 'tle','male',20)))
10 //สร้างและส่ง object ไปที่ function addMember ซึ่งถูกเก็บใน newMember จากนั้นนำผลลัพธ์ที่เกิดขึ้นมาแสดง
11
12
```