

## forEach()

### 1.Syntax :

```
// Arrow function  
forEach((element) => { ... } )  
forEach((element, index) => { ... } )  
forEach((element, index, array) => { ... } )
```

### 2.ความสามารถของ Method

สามารถวน loop เพื่อเข้าถึง element และ index แต่ละตัวภายใน array ได้ โดยจะเป็นการเข้าถึงทุกตัวเริ่มจาก index ที่ 0 ไปจนถึง index ที่มีค่าเท่ากับ length-1

### 3.วิธีการใช้งาน

เป็น method ที่ใช้งานคู่กับ array object โดย forEach() จะรับพารามิเตอร์ที่เป็น arrow function เท่านั้น

Output/ผลลัพธ์ที่ return ที่ได้จาก forEach() คือ undefined

### 4.ตัวอย่าง

```
//array foreach
let metha = [5, "a", 3, "b", 1];
console.log(metha);

metha.forEach((array) => { //ใช้ arrow function ใน foreach เพื่อสั่งให้ console.log element ใน array metha ออกมา
| console.log(" Value: " + array);
// Value: 5
// Value: a
// Value: 3
// Value: b
// Value: 1
});
console.log("");

metha.forEach((array, index) => { //ใช้ arrow function ใน foreach เพื่อสั่งให้ console.log index ใน array metha ออกมา
| console.log("Index: " + index);
// Index: 0
// Index: 1
// Index: 2
// Index: 3
// Index: 4
});
console.log("");

metha.forEach((array, index) => { //ใช้ arrow function ใน foreach เพื่อสั่งให้ console.log element และ index ของแต่ละตัวออกมาจาก array metha
| console.log("Index: " + index + " Value: " + array);
// Index: 0 Value: 5
// Index: 1 Value: a
// Index: 2 Value: 3
// Index: 3 Value: b
// Index: 4 Value: 1
});
console.log("");
```

## shift()

1.Syntax :

**shift()**

2.ความสามารถของ Method

มีความสามารถในการดึง/return element ตัวแรกภายใน array มาแสดงและมันจะทำการ remove element ตำแหน่งกล่าวออกจาก array และทำการย้าย(ขยับ) element ทุกตัวที่อยู่ถัดไปขึ้นมา (สามารถดึง element ของ array ที่ซ่อนอยู่ภายใต้ array อีกตัวหนึ่งออกมาได้ ซึ่งสิ่งที่จะถูก remove ไป ก็คือ element ตัวดังกล่าว)

### 3.วิธีการใช้งาน

เป็น method ที่ใช้งานคู่กับ array object โดย shift() จะไม่รับพารามิเตอร์ชนิดใดทั้งสิ้นไม่ว่าจะเป็น primitive, array และ object รวมไปถึง arrow function

Output ที่ได้จาก shift() คือข้อมูล/element ที่ดึงมาจาก array โดยจะมีชนิดข้อมูลเป็นอะไรขึ้นอยู่กับชนิดข้อมูลของ element นั้นๆที่เราทำการดึงขึ้นมาจาก array ไม่ว่าจะเป็น String, Number, Object, Boolean, Symbol, Undefined

### 4.ตัวอย่าง

```
let arrayTest = [['Best','Ben','Boss'],true,'Tle','Noey',123]; // array ที่เก็บเป็น array ,boolean ,Number
console.log(arrayTest);
console.log(arrayTest[0].shift()); //แสดงข้อมูลของ array ในตำแหน่งที่ 1 โดยแสดงค่าของ array ในตำแหน่งที่ 1 ซึ่งค่าของ array ของแรก คือ Best โดยหลังจากที่แสดงค่าแล้ว จะทำการเอาข้อมูลนั้นออกจาก array
console.log(arrayTest); //Best ถูกดึงออกจาก array หลังแสดงค่าจาก shift()
console.log(arrayTest.shift()); //แสดงข้อมูลของ array ในตำแหน่งที่ 1 ซึ่งตำแหน่งแรกของ array คือ [Ben,Boss] โดยหลังจากที่แสดงค่าแล้ว จะทำการเอาข้อมูลนั้นออกจาก array
console.log(arrayTest); //array ของแรกถูกขจัด , [Ben ,Boss] ถูกดึงออกจากแสดงค่า shift()
console.log(arrayTest.shift()); //แสดงข้อมูลของ array ในตำแหน่งที่ 1 ซึ่งตำแหน่งแรกของ array คือ true โดยหลังจากที่แสดงค่าแล้ว จะทำการเอาข้อมูลนั้นออกจาก array
console.log(arrayTest); // true ถูกดึงออกจาก array หลังแสดงค่าจาก shift()
```

```
[ [ 'Best', 'Ben', 'Boss' ], true, 'Tle', 'Noey', 123 ]
Best
[ [ 'Ben', 'Boss' ], true, 'Tle', 'Noey', 123 ]
[ 'Ben', 'Boss' ]
[ true, 'Tle', 'Noey', 123 ]
true
[ 'Tle', 'Noey', 123 ]
```

### unshift()

#### 1.Syntax :

```
unshift(element0)
unshift(element0, element1)
unshift(element0, element1, ... , elementN)
```

\*\*element สามารถเป็นได้ทั้ง primitive type, object type, array

## 2.ความสามารถของ Method

มีความสามารถในการเพิ่ม element ลงไปในตำแหน่งแรกของ array โดยสามารถเพิ่มทีละ 1 element หรือทีละหลายๆ element ก็ได้ โดย element แต่ละตัวจะขึ้นด้วยเครื่องหมายคอมม่า(,) โดย element เดิมที่อยู่ในตำแหน่งแรกจะถูกเลื่อน/ขยับถัดไปในตำแหน่ง index ที่มากขึ้น

## 3.วิธีการใช้งาน

เป็น method ที่ใช้งานคู่กับ array object โดย unshift() จะสามารถรับพารามิเตอร์ชนิดที่เป็น primitive, array และ object

Output ที่ได้จาก unshift() คือ length ของ array ที่ได้ทำการเพิ่มข้อมูลเข้าไปแล้ว โดยจะมีชนิดเป็น Number

## 4.ตัวอย่าง

```

let unshift_test = ['a','b','c','d',2];

console.log(unshift_test);

unshift_test.unshift('e');//ใส่ e ที่เป็น String เข้าไปใน array unshift_test
console.log(unshift_test);//มีค่า e ที่เป็น String เพิ่มไปที่ index แรกของ array
//[ 'e', 'a', 'b', 'c', 'd', 2 ]

unshift_test.unshift(true); //ใส่ true ที่ type เป็น boolean เข้าไปใน array unshift_test
console.log(unshift_test);//มี boolean ture อยู่ใน array
// [
//   true, 'e', 'a',
//   'b', 'c', 'd',
//   2
// ]

unshift_test.unshift(55);//ใส่ 55 ที่มี type เป็น Numeric เข้าไปใน array unshift_test
console.log(unshift_test);//มี 55 ที่มี type เป็น Numeric อยู่ใน array
// [
//   55, true, 'e',
//   'a', 'b', 'c',
//   'd', 2
// ]


let i = ['z',false,'y',Object()];
for(j = 0;j<i.length;j++){
    unshift_test.unshift(i[j]);
}
//loop เอา element ใน array i ใส่เข้าไปที่ index แรกสุดของ
//array unshift_test ทีละตัวโดยเริ่มจาก index ที่ 0 ของ array i

console.log(unshift_test);
// [
//   {}, 'y', false,
//   'z', 55, true,
//   'e', 'a', 'b',
//   'c', 'd', 2
// ]

```