

every()

SYNTAX

```
every((element) => { ... } )  
every((element, index) => { ... } )  
every((element, index, array) => { ... } )
```

DESCRIPTION

every method จะคืนค่า **true** เมื่อ callbackFn ที่เรียกใช้สำหรับทุก **element** ทุกตัวใน array คืนค่าเป็น **true** หากมี **element 1 ตัวหรือมากกว่า** คืนค่าที่เป็น **false** แล้ว **every จะคืนค่า false** กันที.

note!!: การเรียกใช้เมธอดนี้ในอาร์เรย์ที่ว่างเปล่าจะคืนค่า **true** สำหรับเงื่อนไขใด ๆ

HOW TO USE

```
array.every(callbackFn);  
array.every((element, index, array) => { ... });  
[12, 5, 8, 130, 44].every((element, index, array) => { ... });
```

Parameters

every method มี parameter ที่รับ callbackFn ที่จะใช้กับ element ทุกตัว โดย callbackFn รับ 3 arguments ได้แก่

- **element** -> **รับค่าเป็น element**
element จาก array ที่จะประมวลผล
- **index (Optional)** -> **รับค่า INTEGER**
เป็น index ของ element จาก array ที่จะประมวลผล
- **array (Optional)** -> **รับค่าเป็น array**
array ของ element ทั้งหมดที่ถูกเรียกใช้

Return value

Boolean (**TRUE** / **FALSE**)

every()

EXAMPLES

EX. 1

```
function isBigEnough(element, index, array) {  
  return element >= 10;  
}  
[12, 54, 18, 130, 44].every(isBigEnough); // true
```

EX. 2

```
[12, 5, 8, 130, 44].every((element) => element >= 10); // false
```

EX. 3

```
const isBelowThreshold = (currentValue) => currentValue < 40;  
  
const array1 = [1, 30, 39, 29, 10, 13];  
  
console.log(array1.every(isBelowThreshold)); // true
```

indexOf()

SYNTAX

```
indexOf(searchElement);  
indexOf(searchElement, fromIndex);
```

DESCRIPTION

ค้นหาเลข Index ของ Element ที่ต้องการจะหาใน Array จากหน้าไปหลัง โดยจะ Return Index ของค่าที่ต้องการจะหาคลับมาให้

HOW TO USE

```
array.indexOf(searchElement);  
array.indexOf(searchElement, fromIndex);
```

Parameters

Method มี Parameter ที่รับ 2 Arguments ดังนี้

- **searchElement** -> **รับค่าเป็น String**
Element ที่ต้องการจะหาเลข Index ที่อยู่ใน Array
- **fromIndex** (OPTIONAL) -> **รับค่าเป็น Number**
 - เป็นการระบุ Index ที่ต้องการจะเริ่มต้นในการค้นหา Index ของ Element
 - Return **-1** ก็ต่อเมื่อ **"ใส่ค่าเลขมากกว่าหรือเท่ากับขนาดของ Array"**
 - หากใส่ค่าเลขเป็นลบ **"จะเป็นการเอาขนาดของ Array บวกกับค่าเลขที่ใส่ไป"** แล้วค่อยค้นหาเลข Index ของ Element จากหน้าไปหลัง

Return Value

เลข Index ตำแหน่งแรกของค่าที่ต้องการที่อยู่ใน Array หากไม่พบจะ Return -1

indexOf()

EXAMPLES

EX.1 indexOf(searchElement)

```
const arrNum = [1,3,4,5,6]
console.log(arrNum.indexOf(4)) // 2
console.log(arrNum.indexOf(7)) // -1

const arrStr = ['ant', 'bison', 'camel', 'duck', 'bison']
console.log(arrStr.indexOf('bison')) // 1
console.log(arrStr.indexOf('giraffe')) // -1
```

EX.2 indexOf(searchElement, fromIndex)

```
const arrNum = [1,3,4,5,6,7,1,2,3,4]
console.log(arrNum.indexOf(4, 4)) // 9
console.log(arrNum.indexOf(4, 1)) // 2
console.log(arrNum.indexOf(10, 2)) // -1

const arrStr = ['ant', 'bison', 'camel', 'duck', 'bison']
console.log(arrStr.indexOf('bison', 2)) // 4
console.log(arrStr.indexOf('giraffe', 5)) // -1
```

EX.3 indexOf(searchElement, fromIndex)

```
const arrNum = [1,3,4,5,6,7,1,2,3,4] // Length = 10
console.log(arrNum.indexOf(4, -8)) // 2 (fromIndex = 9+(-8) = 1)
console.log(arrNum.indexOf(4, -1)) // 9 (fromIndex = 9+(-1) = 8)
console.log(arrNum.indexOf(4, 10)) // -1
console.log(arrNum.indexOf(4, 15)) // -1
```