

Javascript Array methods (Group 4)

1. Some()

Syntax : some(callback,[thisarg]);

Parameters

Callback:จะเป็น function เงื่อนไขโดยที่ในทุก ๆ element จะถูกนำมาตรวจสอบใน function นี้จนกว่าจะได้ค่าเป็น true หากไม่มีค่าใด return true function นี้จะ return false

ThisArg: จะเป็น optional ไม่จำเป็นต้องกำหนด จะเป็นค่าที่เราสามารถส่งเข้าไปใช้ใน callback function ได้

element: ข้อมูลที่เราต้องการหาใน array

index: จะเป็น Optional เลือกตำแหน่งของ array ของ array นั้นๆ

array: จะเป็น Optional นำ array นั้นใส่ parameter

หน้าที่ : เป็น Method ที่ตรวจสอบค่าใน Array หากมี element อย่างน้อย 1 ค่าที่ผ่านเงื่อนไขจะ return true โดยเราต้องเขียนเงื่อนไขในรูปแบบของ function

วิธีใช้:

เช่น หากต้องการตรวจสอบตัวเลขใน array ว่ามีค่ามากกว่า 10 หรือไม่

```
1 let test_num= [2,4,8,16]
2 let checknum= (num)=>{
  if(num>10){
    return true
  }
}
3 console.log(test_num.some(checknum));
```

1. ประกาศ array ของตัวเลขที่จะตรวจสอบ
2. สร้าง function โดยรับ parameter 1 ค่า เป็นตัวเลขที่ต้องการจะตรวจสอบ หากเลขนั้นมากกว่า 10 จะ return ค่า true
3. ใช้คำสั่ง test_num.some(checknum)
Test_num เป็นอาร์เรย์ที่ต้องการตรวจสอบ , checknum เป็นเงื่อนไขการตรวจสอบ

จะได้ผลลัพธ์

```
true
```

สามารถเขียนลดรูปโดยใช้ arrow function และ anonymous function ได้ เช่น

```
let test_num= [2,4,8,16]
let checknum= test_num.some((num)=>{if(num>10) return true});
console.log(checknum);
```

ตัวอย่างการใช้งาน

1.ตรวจสอบว่าใน name มีคนที่ชื่อเริ่มต้นด้วย J ไหม

```
let name=["Josh","Jame","Alice","Bob"]
let checkname=name.some(e=>{
  return e[0] === "J"
})
console.log(checkname);
```

```
true
```

2. กำหนด 2 parameter สังเกตว่าค่า max,min สามารถถูกนำไปเรียกใช้ใน callback ได้โดยอ้าง attribute this

```
let ages=[14,15,17,19,10,20];
const range={
  min:18,
  max:30
};
let result = ages.some(function (e) {
  return e >= this.min && e <= this.max;
}, range);
console.log(result);
```

Output:

```
true
```

3. สังเกตว่า 2 ตัวอย่างก่อนหน้าจะเป็นการ hard ลงไปในเงื่อนไข ทำให้ Code ไม่ยืดหยุ่น เราสามารถนำ function มาครอบและรับค่าเข้ามาเพื่อให้ code ยืดหยุ่นมากขึ้นได้

```
let score=[4,5,8,9,10]

let checkScore=(arr,value)=>{
  return arr.some(i=> i<value)
}

console.log(checkScore(score,5));
console.log(checkScore(score,3));
```

Output:

```
true
false
```

2. includes()

Syntax: includes(searchElement)
includes(searchElement, fromIndex)

Parameters

searchElement : ค่าที่ต้องการจะตรวจสอบ

fromIndex: Optional ตำแหน่ง Index ที่จะตรวจสอบ

หน้าที่: เป็น method ที่จะ return Boolean โดยจะตรวจสอบใน array ว่ามีค่าเหมือนกับ element ที่ส่งเข้ามาหรือไม่ ถ้ามีจะ return true

วิธีใช้:

เช่น หากเราต้องการตรวจสอบว่าใน array เรามีคนชื่อ Alice หรือไม่

```
1 let name=["Alice","Billy","Caterline","David","Elly"]
2 console.log(name.includes("Alice"));
3 console.log(name.includes("Alice",2));
```

1. ประกาศชื่อโดยเก็บเป็น Array
2. แสดงผลว่าใน array name มีค่าที่เป็น "Alice" หรือไม่
3. แสดงผลว่าใน array name มีค่าที่เป็น "Alice" หรือไม่ โดยเริ่มนับตั้งแต่ตัวที่ 2 ถึงตัวสุดท้ายใน array

Output

```
[Running] node /Users/phanon/Desktop/
true
false
```

Line1: True เพราะ Array index:0 มีค่าเป็น Alice

Line2: False เพราะ เริ่มตรวจสอบ index ที่2 และตั้งแต่ indexที่2 ถึงตัวสุดท้ายไม่มีค่าใดเป็น "Alice"

ตัวอย่างการใช้งาน

- 1.ตรวจสอบว่าใน sports มี Football และ Tabletennis ไหม

```
let sports=["Football","Basketball","Tennis","Badminton","Swimming"]
console.log(sports.includes("Football")&&sports.includes("Tabletennis"));
```

Output

```
false
```

2. ตรวจสอบว่าใน pets มี cat ไหม

```
const pets = ['cat', 'dog', 'bat'];
console.log(pets.includes('cat'));
```

Output

```
true
```

3. ตรวจสอบว่าใน game มี Dota2 ไหม

```
let games = ['Dota2', 'Minecraft', 'Final_Fantasy_XV']
console.log(games.includes('Dota2',1));
```

Output

ได้เป็น false เพราะว่าตำแหน่ง Index ของ Index ที่ 1 ไม่ได้เป็น 'Dota2'

```
false
```