

Splice() (insert & replace & delete)

1) Method Syntax

```
splice(start);  
splice(start, deleteCount);  
splice(start, deleteCount, item1);  
splice(start, deleteCount, item1, item2, itemN);
```

2) Method Description

เป็น method ที่ทำหน้าที่เปลี่ยนแปลงข้อมูลที่อยู่ใน Array โดยอาจใช้วิธีลบข้อมูล (delete) หรือแทนที่ข้อมูลที่มีอยู่เดิม (replace) และอาจมีการเพิ่มข้อมูลใหม่ (insert) ลงไปใน Array นั้น ๆ ด้วย

3) How to use method

```
splice(start);  
splice(start, deleteCount);  
splice(start, deleteCount, item1);  
splice(start, deleteCount, item1, item2, itemN);
```

จากภาพจะเห็นได้ว่าจะมี Parameter ทั้งหมดดังนี้

1. start

คือตำแหน่งของข้อมูลใน Array ที่ต้องการเปลี่ยนแปลง

2. deleteCount

คือ จำนวนข้อมูลใน Array ที่ต้องการลบ

3. item1, item2, ... ,itemN

คือ ข้อมูลที่เราต้องการจะเพิ่ม หรือแทนที่เข้าไปใน Array

และการ return ค่า จะมีการ return ค่ากลับมาเป็น Array ของข้อมูลที่ถูกเปลี่ยนแปลงไป

4) Example

Ex.1 การใช้splice() เพื่อ Insert , Replace , Delete ชื่อเดือน

1. Insert

```
let months = ['Jan', 'March', 'April', 'June'];
months.splice(1, 0, 'Feb');
// inserts at index 1
console.log(months);
// expected output: Array ["Jan", "Feb", "March", "April", "June"]
```

Output:

```
[Running] node "c:\Users\CREAM\Desktop\y-2\INT201
Insert
[ 'Jan', 'Feb', 'March', 'April', 'June' ]
```

2. Replace

```
months.splice(4, 1, 'May');
// replaces 1 element at index 4
console.log('Replace')
console.log(months);
// expected output: Array ["Jan", "Feb", "March", "April", "May"]
```

Output:

```
Replace
[ 'Jan', 'Feb', 'March', 'April', 'May' ]
```

3. Delete

```
months.splice(2, 1);  
// remove at index2  
console.log('Delete');  
console.log(months);  
// expected output: Array ["Jan", "Feb", "April", "May"]
```

Output:

```
Delete  
[ 'Jan', 'Feb', 'April', 'May' ]
```

Ex.2 การใช้splice() เพื่อ Insert , Replace , Delete ชื่อสมาชิกในกลุ่ม

1. Insert

```
let names = ['Tong','Bam','View','Oat','Job'];  
names.splice(3,0,'Piyamon');  
//inserts at index3  
console.log('Insert')  
console.log(names);  
//expected output: Array [ 'Tong', 'Bam', 'View', 'Piyamon', 'Oat', 'Job' ]
```

Output:

```
[Running] node "c:\Users\CREAM\Desktop\y-2\INT201\JS  
Insert  
[ 'Tong', 'Bam', 'View', 'Piyamon', 'Oat', 'Job' ]
```

2. Replace

```
names.splice(2,1,'NongHeng');  
//replace at index 2 (remove 1 element at index 2 , and insert "Piyamon")  
console.log('Replace')  
console.log(names);  
//expected output: Array [ 'Tong', 'Bam', 'NongHeng', 'Piyamon', 'Oat', 'Job' ]
```

Output:

```
Replace  
[ 'Tong', 'Bam', 'NongHeng', 'Piyamon', 'Oat', 'Job' ]  
Delete
```

3. Delete

```
names.splice(3,1);  
//remove at index 2  
console.log('Delete');  
console.log(names);  
//expected output: Array [ 'Tong', 'Bam', 'Nongheng', 'Oat', 'Job' ]
```

Output:

```
Delete  
[ 'Tong', 'Bam', 'NongHeng', 'Oat', 'Job' ]
```

Ex.3 การใช้splice() เพื่อ Insert , Replace , Delete ชื่อปลา

1. Insert

```
let fishs = ['angel', 'clown', 'mandarin', 'sturgeon'];  
fishs.splice(2, 0, 'drum');  
//inserts at index 2  
console.log('Insert')  
console.log(fishs);  
//expected output: Array [ 'angel', 'clown', 'drum', 'mandarin', 'sturgeon' ]
```

Output:

```
[Running] node "c:\Users\CREAM\Desktop\y-2\INT201\JS-S  
Insert  
[ 'angel', 'clown', 'drum', 'mandarin', 'sturgeon' ]
```

2. Replace

```
fishs.splice(2, 1, 'trumpet');  
//inserts at index 2 (remove 1 element at index 2 , and insert "trumpet")  
console.log('Replace')  
console.log(fishs);  
//expected output: Array [ 'angel', 'clown', 'trumpet', 'mandarin', 'sturgeon' ]
```

Output:

```
Replace  
[ 'angel', 'clown', 'trumpet', 'mandarin', 'sturgeon' ]
```

3. Delete

```
fishs.splice(3,1);  
// remove at index 3  
console.log('Delete');  
console.log(fishs);  
//expected output: Array [ 'angel', 'clown', 'trumpet', 'sturgeon' ]
```

Output:

```
Delete  
[ 'angel', 'clown', 'trumpet', 'sturgeon' ]
```

Ex.4 การใช้งาน splice() method + คำอธิบาย

Ex.1

```
//index  
const array1 = ['one','two','three'];
```

```
array1.splice(1,0);  
console.log(array1); //ไม่มีอะไรเปลี่ยนแปลง
```

```
[Running] node "c:\Users\LE  
[ 'one', 'two', 'three' ]
```

คำอธิบาย: ที่ไม่เกิดการเปลี่ยนแปลงเนื่องจาก ไม่ได้มีการกำหนด **deleteCount** คือระบุไว้ว่าเป็น 0 และไม่ได้กำหนดค่าที่จะเพิ่มเข้ามา(**itemN**) ซึ่ง output ที่ได้จะไม่มีอะไรเปลี่ยนแปลงไปจากเดิมเลย

Ex.II

```
array1.splice(1,2);  
console.log(array1);//เกิดการลบ items ในตำแหน่งที่ 1 "ดังนั้น! output จะมีเพียง ['one']จะต้องไม่มี ['two','three'] ปรากฏ  
//เนื่องจาก ได้กำหนดว่า howmany = 2 ที่จะทำการลบออกไป ก็คือ ลบไป 2 ตัวตั้งแต่ two นั้นเอง"
```

```
[Running] node "  
[ 'one' ]
```

คำอธิบาย: เกิดการลบ items ใน index ที่ 1 ดังนั้น “output จะมีเพียง [‘one’] จะต้องไม่มี [‘two’,‘three’] ปรากฏ” เนื่องจาก ได้กำหนดว่า deleteCount=2 ที่จะทำการลบออกไป ก็คือลบไป 2 ตัว ตั้งแต่ two นั้นเอง

Ex.III

```
array1.splice(1,0,'Hello');  
console.log(array1);//ทำการเพิ่ม item ที่เป็น String ว่า 'Hello' ณ indexที่1 ซึ่งมันจะหมายความว่า  
//“ก่อนหน้า indexที่1 จะมี 'Hello' เพิ่มเข้ามาอีก1 ตัว”
```

```
[Running] node "c:\Users\LENOVO\Desktop  
[ 'one', 'Hello', 'two', 'three' ]
```

คำอธิบาย: ทำการเพิ่ม item ที่เป็น String ว่า ‘Hello’ ณ index ที่ 1 ซึ่งมันจะหมายความว่า “ก่อนหน้า index ที่ 1 จะมี ‘Hello’ เพิ่มเข้ามาอีก1 ตัว”

Ex.IV

```
array1.splice(1,1,'Hello');  
console.log(array1);  
//ทำการแทนที่ items String 'Hello' ในตำแหน่งที่ 1  
//(จำเป็นต้องระบุจำนวน howmany ด้วยเพื่อกำหนดว่าจะเปลี่ยนแปลงค่า 1 ตัว)  
//โดย output ที่ได้คือ "จะไม่มี 'two' ปรากฏเพราะถูกแทนที่ด้วย 'Hello' นั่นเอง"
```

```
[Running] node "c:\Users\LENOVO\  
[ 'one', 'Hello', 'three' ]
```

คำอธิบาย: จากในภาพ เป็นหลักการของ **replace** คือมันจะนำมาใช้ร่วมกันกับ

delete&insert ด้วยการลบ(**delete**)ไป 1 ค่า แล้วทำการเพิ่ม(**insert**)ค่าที่ต้องการไป

แทน = การแทนที่(**replace**) ซึ่งขึ้นอยู่กับตำแหน่ง **index** ที่เราเลือก

Ex.V

ถ้าจะเพิ่มค่า(**insert**)โดยอยากกำหนดให้มีมากกว่า **1 item** ที่จะเพิ่มเข้าไป ก็สามารถทำได้โดยการ “ใส่เครื่องหมาย ‘,’ คั่นไว้เสมอและกำหนด **item** ที่เราจะเพิ่มเข้าไปได้ตามที่ต้องการ”

```
const array1 = ['one','two','three'];  
array1.splice(1,0,'Hello','Hi','Yo');  
console.log(array1); |
```

```
[Running] node "c:\Users\LENOVO\Desktop\SIT\2564  
[ 'one', 'Hello', 'Hi', 'Yo', 'two', 'three' ]
```

Ex.VI

เช่นเดียวกันกับการแทนที่(replace) โดย output ที่ได้ จะต้องไม่มี ['two'] ปรากฏ เพราะถูกแทนที่ด้วย ['Hello','Hi','Yo'] นั่นเอง

```
const array1 = ['one','two','three'];  
array1.splice(1,1,'Hello','Hi','Yo');  
console.log(array1); |
```

```
[Running] node "c:\Users\LENOVO\Desktop\S  
[ 'one', 'Hello', 'Hi', 'Yo', 'three' ]
```

Ex.5 การใช้งาน splice() method ร่วมกับ Arrow Function

```
let Person = ['a', 'b', 'c', 'd', 'e']
```

มีคนดังนี้: 'a' 'b' 'c' 'd' 'e' จากนั้น ลองจัดการ Array Person ด้วย ฟังก์ชัน (ในรูปแบบของ Arrow Function)

5.1) ฟังก์ชันแทนที่ element

```
replacePerson = (index, itemChange) => Person.splice(index, 1, itemChange)  
replacePerson(2, 'C')
```

ฟังก์ชันชื่อ **replacePerson()** ทำหน้าที่รับพารามิเตอร์ **index, itemChange** เข้ามา เพื่อไปดำเนินการต่อข้างใน **index** คือ **index** ที่ต้องการเปลี่ยน | **itemChange** คือ ค่าที่ต้องการให้เปลี่ยนเป็นแบบอื่น

การทำงานภายในฟังก์ชัน: รับ **index, itemChange** เข้ามาทำงานผ่าน **method splice()** ดังรูป คือ การเปลี่ยนแปลงค่า **index** ที่ **2** แทนค่าด้วย **'C'** (ตัวพิมพ์ใหญ่)

```
[ 'a', 'b', 'C', 'd', 'e' ] ( Output จากคำสั่ง console.log(Person) )
```


5.2) ฟังก์ชันเพิ่มตัวสุดท้าย

```
addPerson = (itemAdd) => Person.splice(Person.length, 0, itemAdd)
addPerson('f')
```

- ฟังก์ชันชื่อ **addPerson()** ทำหน้าที่รับพารามิเตอร์ **itemAdd** เข้ามาเพื่อไปดำเนินการต่อข้างใน
- **itemAdd** คือ ค่าที่ต้องการเพิ่มเข้าไป
- การทำงานภายในฟังก์ชัน: รับ **itemAdd** เข้ามาทำงานผ่าน method **splice()**

ดังรูป คือ การเพิ่มค่า **f** เข้าไปใน index **Person.length** (หรือ index สุดท้าย) ก็จะเป็นการเพิ่ม **f** เข้าไปในตำแหน่งสุดท้ายของ Array

```
[ 'a', 'b', 'c', 'd', 'e', 'f' ] ( Output จากคำสั่ง console.log(Person))
```

5.2) ฟังก์ชันลบตัวสุดท้าย

```
deletePerson = () => Person.splice(Person.length-1, 1)
deletePerson()
```

- ฟังก์ชันชื่อ **deletePerson()** ไม่มีพารามิเตอร์
- การทำงานภายในฟังก์ชัน: ทำงานผ่าน method **splice()** ด้วยการไปลบตัวที่อยู่ใน **Person.length-1** (หรือตัวที่เป็น element สุดท้ายใน Array ปัจจุบัน)

```
[ 'a', 'b', 'c', 'd', 'e' ] ( Output จากคำสั่ง console.log(Person) )
```

Ex.6 การประยุกต์ใช้งาน splice() method

1.ลบตัวเลขที่เป็นเลขคู่

```
let elements = [1, 3, 8, 5, 16, 1, 4];
for (i = elements.length - 1; i >= 0; --i) {
  if (elements[i] % 2 === 0) { //เช็คว่าถ้า mod ด้วย 2 แล้วได้เท่ากับ 0 จะเป็นเลขคู่
    elements.splice(i, 1); //ลบค่าที่เป็นเลขคู่
  }
}
console.log(elements);
```

Output: [1, 3, 5, 1]

2.ลบค่าทั้งหมดที่เท่ากับค่าที่กำหนด

```
let elements = [1, 3, 5, 3, 1, 4];
let removeEl = 3; //กำหนดตัวเลขที่ต้องการลบไว้ในตัวแปร removeEl
let index = elements.indexOf(removeEl);
while (index !== -1) { //ถ้าหาค่าที่กำหนดไว้ใน removeEl ไม่เจอจะทำการ return -1
  //แต่ถ้าหาค่าที่กำหนดไว้ใน removeEl เจอ จะทำการ return index ออกมา
  elements.splice(index, 1); //ทำการลบindexนั้น
  index = elements.indexOf(removeEl);
}
console.log(elements);
```

Output: [1, 5, 1, 4]

3.ลบค่าทั้งหมดที่เท่ากับค่าที่กำหนด

```
let elements = [1, 3, 5, 3, 1, 4];
for (let i = 0; i < elements.length; i++) {
  if (elements[i] == 3) { //กำหนดค่าที่ต้องการลบ
    elements.splice(i, 1); //ลบ i ที่มีค่าเท่ากับ 3
  }
}
console.log(elements);
```

Output: [1, 5, 1, 4]