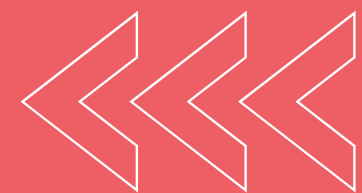


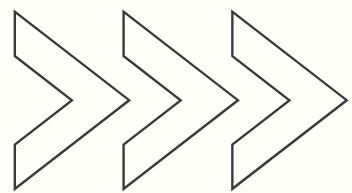
ARRAY METHODS

`push()` & `pop()` & `filter()`

01 push()

เป็น function ที่เพิ่มค่าเข้าไปที่ค่าสุดท้ายของ array โดย push นั้นจะทำการ mutate array ตัวเก่า หรือที่เข้าใจง่ายๆ คือ การเพิ่มสมาชิกใหม่เข้าไปยังตำแหน่งสุดท้ายของอาเรย์





METHOD SYNTAX

push()

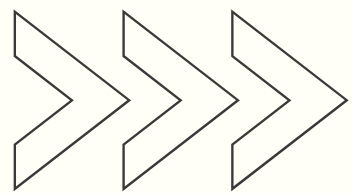
push(element0)

push(element0, element1)

push(element0, element1, ... , elementN)

parameter ของ method push จะเป็นค่าที่เราต้องการเพิ่มให้เป็นสมาชิกของ Array นั้นๆ จะเป็นประเภทอะไรก็ได้ ไม่ว่าจะเป็น primitive types, array, object



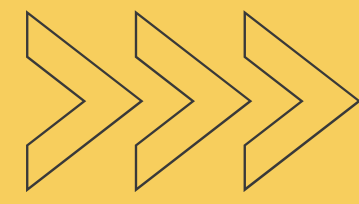


METHOD OUTPUT

push()

เมื่อเรียกใช้ method push() จะทำการเพิ่มสมาชิกตัวใหม่ที่เราใส่
ใน parameter เข้าไป และ return ค่าเป็นจำนวนความยาวของ
array ใหม่ หลังจากที่เราเพิ่มสมาชิกตัวใหม่เข้าไปเรียบร้อยแล้ว





USAGE EXAMPLE

push()

```
const university = ['KMUTT', 'KMITL', 'KMUTNB'];
const count = university.push('KU'); //เรียกใช้ push() โดยส่ง parameter เป็นสมาชิกใหม่ 1 ตัว
console.log(count); // output: 4 -- จะนับจำนวนสมาชิกรวมตัวที่เพิ่ง push เพิ่มเข้ามาใหม่ด้วย
console.log(university); // output: Array ["KMUTT", "KMITL", "KMUTNB", "KU"]
//-----

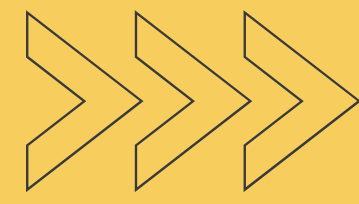
university.push('CU', 'NU', 'KKU'); //เรียกใช้ push() โดยส่ง parameter เป็นสมาชิกใหม่ 3 ตัว
console.log(university); // output: Array ["KMUTT", "KMITL", "KMUTNB", "KU", "CU", "NU", "KKU"]

//-----

const array = [1,2,3,4];
console.log(array); //output: [1,2,3,4]
console.log(array.length); //output:4

//-----

array.push(7);
console.log(array); //output: [1,2,3,4,7]
console.log(array.length); //output:5
```



USAGE EXAMPLE

push()

```
const fruits1 = ["Banana", "Orange"];  
const fruits2 = ["Apple", "Mango"];
```

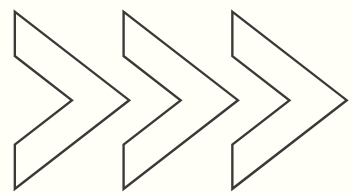
```
fruits1.push(...fruits2); // เป็นนำสมาชิกของ array fruits2 ทั้งหมด ไปเพิ่มเป็นสมาชิกของ fruits1 ผ่าน spread operator  
console.log(fruits1); // output: Array ["Banana", "Orange", "Apple", "Mango"]
```

02

pop()

pop() เป็นการนำค่าใน index สุดท้ายออกมา และ return ค่านั้นออกมา ซึ่ง pop จะทำการ mutate array เช่นเดียวกับกับของ push ทำให้เวลาเราเรียก array เดิม ก็จะแสดงเป็น array ค่าใหม่ หรืออธิบายง่ายๆ คือ ลบสมาชิกที่ตำแหน่งสุดท้ายออกจากอาร์เรย์ และส่งค่าดังกล่าวกลับมา



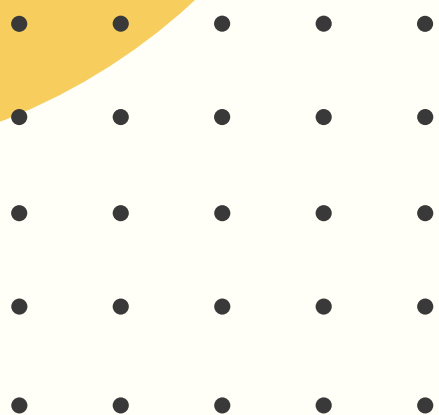


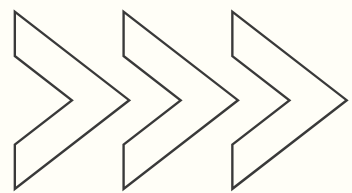
METHOD SYNTAX

pop()

array.pop()

method pop() สามารถเรียกใช้ได้เลยทันที โดยไม่ต้องมี parameter

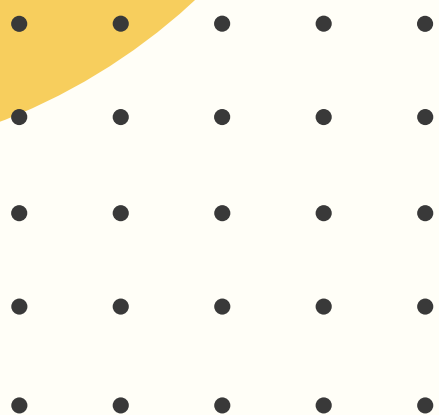


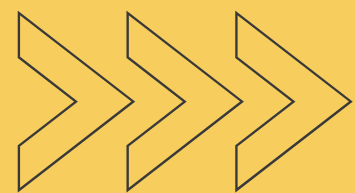


METHOD OUTPUT

pop()

เมื่อเรียกใช้ method pop() จะทำการลบค่าสมาชิกที่ตำแหน่งสุดท้ายออกจาก array และ return ค่าดังกล่าวกลับมา แต่ถ้าหาก เรียกใช้ method pop() ใน array ว่างมันจะ return เป็น undefined





USAGE EXAMPLE

pop()

```
const kpop = ['EXO', 'BTS', 'BLACKPINK', '2NE1', 'TWICE'];
```

```
console.log(kpop.pop()); // output: "TWICE" ทำการลบสมาชิกตัวสุดท้ายออก แล้ว return เป็นค่าของสมาชิกตัวนั้น
```

```
console.log(kpop); // output: Array ["EXO", "BTS", "BLACKPINK", "2NE1"]
```

```
kpop.pop(); //จะทำการลบค่าในตำแหน่งสุดท้ายเสมอ ทำให้เมื่อแสดงค่าจะได้ตามบรรทัดถัดไป
```

```
console.log(kpop); // output: Array ["EXO", "BTS", "BLACKPINK"]
```

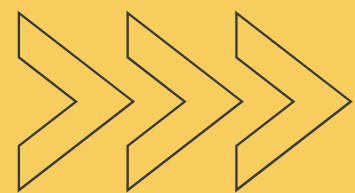
```
//-----
```

```
const num = [1,2,3,4,5];
```

```
console.log(num); //output: [1,2,3,4,5]
```

```
num.pop(); //จะทำการลบค่าในตำแหน่งสุดท้ายเสมอ ทำให้เมื่อแสดงค่าจะได้ตามบรรทัดถัดไป
```

```
console.log(num); //output: [1,2,3,4]
```



USAGE EXAMPLE

pop()

```
const fruits = ["Banana", "Orange", "Apple", {name:"Mango",color:"yellow"}];
```

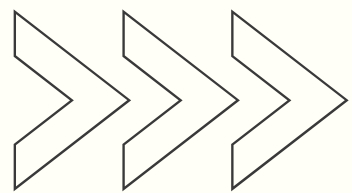
`console.log(fruits.pop());` // output: {name: "Mango", color: "yellow"} เมื่อเรียก `pop()` จะทำการลบสมาชิกตัวสุดท้ายออก แล้ว return ค่าของสมาชิกตัวนั้นออกมา

```
console.log(fruits); // output: ["Banana", "Orange", "Apple"]
```

03 filter()

เป็นการกรองค่าหรือฟิเตอร์ค่าออกมาตามเงื่อนไขที่เราต้องการ
โดย filter นั้นรับ array มาวนเหมือนกับ map
แต่เราจะต้องมีการส่ง condition หรือเงื่อนไขที่เราต้องการที่จะให้
return ค่าอะไรออกมา ซึ่ง filter ก็จะทำการสร้าง array ใหม่เช่นเดียวกัน





METHOD SYNTAX

filter()

`filter(element) => { ... })`

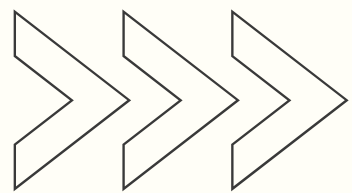
`filter(element, index) => { ... })`

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

parameter ของ method filter จะเป็น arrow function โดยใน arrow function นี้ สามารถกำหนดได้ว่าจะมี parameter ใดบ้าง ได้แก่

- element : สมาชิกตัวปัจจุบันที่กำลังทำงานอยู่ใน filter
- index : ตำแหน่งของสมาชิกปัจจุบันที่กำลังทำงานอยู่ใน filter
- array : เป็น reference ของ array ต้นฉบับที่ถูกเรียกเข้ามา ทำให้เราสามารถเพิ่มเติมหรือแก้ไข array ต้นฉบับ ระหว่างทำงานใน filter ได้

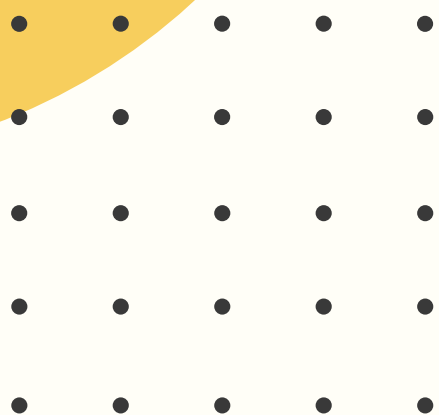


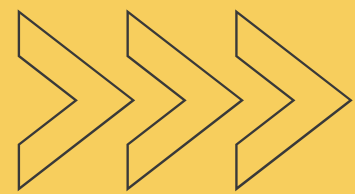


METHOD OUTPUT

filter()

เมื่อเรียกใช้ method filter() จะทำการคัดกรอง array ตามเงื่อนไข
แล้ว return ออกมาเป็น array ตัวใหม่ ที่ทุกสมาชิกตรงตามเงื่อนไข





USAGE EXAMPLE

filter()

```
const words = ['spray', 'limit', 'elite', 'exuberant', 'destruction', 'present'];
```

```
const result = words.filter(word => word.length > 6);
```

//เรียกใช้ filter โดยส่ง parameter 1 ตัว โดยมีเงื่อนไขคือ เอาเฉพาะคำที่มีความยาวตัวอักษรมากกว่า 6 ตัว

```
console.log(result);
```

// output: Array ["exuberant", "destruction", "present"]

จะทำการ return คำที่มีตัวอักษรมากกว่า 6 ตัวออกมา

```
//-----
```

```
function numGreater(value) {
```

```
  return value >= 20;
```

```
}
```

//สร้าง function สำหรับ return เฉพาะค่าที่มากกว่า 20 ขึ้นไป

```
var filtered = [2, 10, 28, 100, 299].filter(numGreater);
```

// เรียกใช้ filter ผ่าน function numGreater ที่สร้างไว้

output [28, 100, 299] จะทำการแสดงค่าที่มากกว่า 20 ออกมา

```
const num = [1,2,3,4,5,6];
```

```
const evenNumber = num.filter(value => {  
  return value % 2 === 0 });
```

 //เรียกใช้ filter โดยส่ง parameter 1 ตัว โดยมีเงื่อนไขคือ เอาเฉพาะค่าที่หารด้วย 2 ลงตัวเท่านั้น

```
console.log(num); // output: [1,2,3,4,5,6]
```

```
console.log(evenNumber); // output: [2,4,6]
```

```
//-----
```

```
const members = [  
  {name: "Numnim", age: 24},  
  {name: "Mos", age: 35},  
  {name: "Billy", age: 15},  
  {name: "Nut", age: 30},  
  {name: "Kookkan", age: 23}  
]
```

```
const result = members.filter((member) => {  
  return member.age > 25
```

```
});
```

 //เรียกใช้ filter โดยส่ง parameter 1 ตัว โดยมีเงื่อนไขคือ object ตัวนั้นต้องมี age มากกว่า 25

```
console.log(members);
```

 // array ตัวเดิมที่ไม่ได้ทำการ filter

```
output: [{ name: 'Numnim', age: 24 }, { name: 'Mos', age: 35 },
```

```
{ name: 'Billy', age: 15 }, { name: 'Nice', age: 30 }, {name: 'Kookkan', age: 23}]
```

```
console.log(result);
```

 // array ตัวใหม่ที่ทำกร filter

```
output: [{name: "Mos", age: 35}, {name: "Nice", age: 30}]
```



```
let words = ["spray", "future", "destruction"];
```

```
const appendedWords = words.filter((word, index, array) => {  
  array.push("new");  
  return word.length > 8;
```

}); //เรียกใช้ filter โดยส่ง paramter 3 ตัว ซึ่งมี array เป็น parameter ด้วย แล้วทำการเพิ่มค่า "new" เข้าไปใน array ตัวที่อ้างอิงมา ทุกครั้งที่ filter ทำงาน
และ filter จะ return เป็น array ตัวใหม่ ที่มีสมาชิกทุกตัวซึ่งมีความยาวของตัวอักษรมากกว่า 8 ตัว

```
console.log(appendedWords); // array ตัวใหม่ที่ทำกร filter
```

output: ["destruction"] เพราะมีสมาชิกเพียงตัวเดียวที่มีความยาวของตัวอักษรมากกว่า 8 ตัว

```
console.log(words) // array ตัวเดิมที่ไม่ได้ทำการ filter
```

output: ["spray", "future", "destruction", "new", "new", "new"] เพราะมีการเพิ่มสมาชิก "new" ทุกครั้งที่ filter ทำงาน ซึ่ง filter จะมีจำนวนครั้งในการทำงานตามสมาชิกของ array words
จึงมี "new" เพิ่มมาทั้งหมด 3 ตัว

ที่มาของข้อมูล



Array - JavaScript | MDN

The JavaScript Array class is a global object that is used in the construction of arrays; which are high-level, list-like objects.

 [mozilla.org](https://developer.mozilla.org)

THANK
YOU