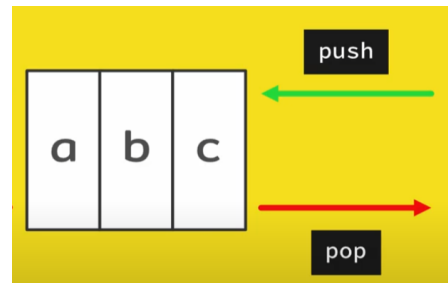


# push()



- **Method Syntax :** `push()`
- **คำอธิบาย :** ใช้เพิ่มข้อมูลในตำแหน่งสุดท้ายของ Array
- **วิธีการใช้ :** `push()` ต้องใส่ parameter เช่น `array.push('Watermelon')` โดยใน parameter จะเป็นข้อมูลที่เรต้องการจะเพิ่มค่าเข้าไปที่ค่าสุดท้ายของ array โดย `push` นั้นจะทำการ mutate array ตัวเก่า
- **ตัวอย่างของแต่ละ syntax :**  
ตัวอย่างที่ 1

```
1 let fruits = ['Apple', 'Banana', 'Carrot'];
2 console.log(fruits);
3
4 fruits.push('Watermelon');
5 console.log(fruits);
```

ตัวอย่างที่ 2

```
104
105
106 let vegetables = ['parsnip', 'potato']
107 let moreVegs = ['celery', 'beetroot']
108
109 // Merge the second array into the first one
110 vegetables.push(...moreVegs);
111
112 console.log(vegetables) // ['parsnip', 'potato',
```

### ตัวอย่างที่ 3

```
114
115
116  ✓ let obj = {
117      length: 0,
118
119      ✓ addElem: function addElem(elem) {
120          // obj.length is automatically incremented
121          // every time an element is added.
122          [].push.call(this, elem)
123      }
124  }
125
126  // Let's add some empty objects just to illustrate.
127  obj.addElem({})
128  obj.addElem({})
129  console.log(obj.length)
130  // → 2
```

PROBLEMS 55 OUTPUT TERMINAL DEBUG CONSOLE

[Running] node "c:\Users\Admin\Desktop\INT201DEVREAL\INT201DEV\ฝึกหัด" 2

# pop()

- **Method Syntax :** `pop()`

- **คำอธิบาย :**

ใช้ดึงค่าสุดท้ายของ array ออกมา

- **วิธีการใช้ :** `pop()` ไม่จำเป็นต้องใส่ parameter ให้ โดยจะทำการดึงข้อมูลค่าสุดท้ายออกมาโดยนำไปต่อหลัง attribute ที่เก็บค่าเป็น array แล้วจะ return ค่าสุดท้ายของ array นั้นออกมา โดยที่ array ตัวนั้นก็จะนำค่าตัวนั้นออกไปด้วย
- **ตัวอย่างของแต่ละ syntax**

ตัวอย่างที่ 1 `pop()`

```
let arr = ['A', 'B', 'C', 'D', 'E', 'F']
let lastArr = arr.pop();
console.log(lastArr); // F
console.log(arr); // ['A', 'B', 'C', 'D', 'E']
```

```
[Running] node "/Users/n11/Desktop/KMUTT/INT201-
F
[ 'A', 'B', 'C', 'D', 'E' ]
```

ตัวอย่างที่ 2 `pop()`

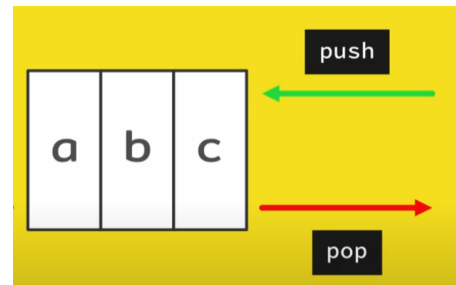
```
let arr = ['A', 'B', 'C', 'D', 'E', 'F']
console.log(`Before: ${arr.length}`) // 6
arr.pop();
console.log(`after ${arr.length}`); // 5
```

```
[Running] node "/Users/n11/Desktop/KMUTT/INT201-
Before: 6
after 5
```

ตัวอย่างที่ 3 `pop()`

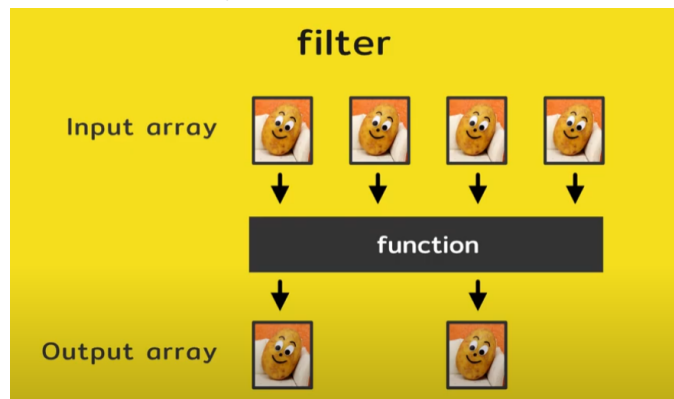
```
let arr = ['A', 'B', 'C', 'D', 'E', 'F'];
let newArr = [];
for (let i = 0; i < arr.length; i++) {
  newArr.push(arr.pop());
  i--;
}
console.log(newArr)
```

```
[Running] node "/Users/n11/Desktop/KMUTT/INT201-
[ 'F', 'E', 'D', 'C', 'B', 'A' ]
```



# filter()

- Method Syntax: `arr.filter(function(element, index, array))`  
*//Arrow Function*  
`arr.filter((element) => { ... } )`  
`arr.filter((element, index) => { ... } )`  
`arr.filter((element, index, array) => { ... } )`
- คำอธิบาย : ใช้คัดเลือกข้อมูลใน array ที่ละตัว



- วิธีการใช้ : `array.filter()` จำเป็นต้องมี function มารองรับเงื่อนไขในการคัดลอกข้อมูลใน array โดย function ใน filter จะสามารถเรียกใช้ `element`, `index`, `array` ของ `arr.filter()` นั้นๆได้ โดยอาจจะนำเอามาใช้เป็นเกณฑ์การคัดเลือกใน function แล้วจะ return ออกมาเป็น `element` ของ array ที่เข้าเงื่อนไขของ function นั้น
- ตัวอย่างการใช้งานของแต่ละ syntax

ตัวอย่างที่ 1 `arr.filter(function(element, index, array))`

```
let num = [0,1,2,3,4,5,6,7,8,9,10,11,12,13];
const evenNum = num.filter(function (element,index,array) {
    return element%2==0 || index==array.length-1
    //return element ที่เป็นเลขคู่ หรือ อยู่ที่ index สุดท้ายของ array
});
console.log(evenNum);
```

```
[Running] node "c:\Users\ST\Desktop\IT2021\INT201_0
[
  0, 2, 4, 6,
  8, 10, 12, 13
]

[Done] exited with code=0 in 0.091 seconds
```

ตัวอย่างที่ 2 `array.filter((element) => { ... } )`

```
let scores = [68,75,70,80,85,90,82];
const checkScore = scores.filter(score => score >= 80);
//return score ที่ >= 80 เท่านั้น
console.log(checkScore);
```

```
[Running] node "c:\Users\ST\Desktop\IT2021\INT201_0
[ 80, 85, 90, 82 ]

[Done] exited with code=0 in 0.092 seconds
```

ตัวอย่างที่ 3 `array.filter((element, index) => { ... } )`

```
let ids = [101,102,103,104,105];
const checkId = ids.filter((id,index) => id < 102 || index == 3);
//return idที่ <102 หรือ มี index อยู่ตำแหน่งที่ 3 เท่านั้น
console.log(checkId);
```

```
[Running] node "c:\Users\ST\Desktop\IT2021\INT201_0
[ 101, 104 ]

[Done] exited with code=0 in 0.112 seconds
```

ตัวอย่างที่ 4 `array.filter((element, index, array) => { ... } )`

```
let animals = ['ant','bird','cat','dog','frog'];
const choose = animals.filter((animal,index,arr)=>{
    arr.pop();          //ดึง frog ออกจาก array นี้ (arr หมายถึง animals)
    arr.push('rat');     //เพิ่ม rat ต่อท้าย array นี้
    return animal.length==3; //return animal ที่มีความยาวเท่ากับ 3 เท่านั้น
});
console.log(choose);
```

```
[Running] node "c:\Users\ST\Desktop\IT2021\INT201_0
[ 'ant', 'cat', 'dog', 'rat' ]

[Done] exited with code=0 in 0.095 seconds
```

ตัวอย่างที่ 5 ของการใช้งาน `filter()` แบบ `callback function` คัดเลือกอายุ

```
//ตัวอย่าง Callblack function
let ages = [10,12,18,22,35,40,45,50,60];
function checkAge(age){ //age คือ element แต่ละตัวที่อยู่ใน ages[]
    return age >=18;     //return age ที่มีค่าอย่างน้อย 18
}
console.log(ages.filter(checkAge)); // [18, 22, 35, 40, 45, 50, 60]
```

```
[Running] node "c:\Users\ST\Desktop\IT2021\INT201_0
[
  18, 22, 35, 40,
  45, 50, 60
]

[Done] exited with code=0 in 0.1 seconds
```