Array.prototype.map()

1. **Method Syntax**

// Arrow function

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

index and array is an optional

// Callback function

```
map(callbackFn, thisArg)
```

thisArg is an optional

// Inline callback function

```
map(function callbackFn(element, index, array) { ... }, thisArg)
```

index, array and thisArg is an optional

2. **Description**

The `map()` method **calls a function callbackFn** once for each element in the array, in order, and **constructs a new array** from the result. `callbackFn` is invoked only for indexes of the array which have assigned values

3. **How to use**

// Arrow function

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

Parameter

element : element being processed in the array
index (optional) : index of the current element
array (optional) : The array `map` was called upon.

Output

A new array with each element being the result of the callback function.

// Callback function

```
map(callbackFn, thisArg)
```

Parameter

`callbackFn` : Function that is called for every element of `arr`. Each time `callbackFn` executes, the returned value is added to `newArray`.
`thisArg` (Optional) : Value to use as `this` when executing `callbackFn`.

Output

A new array with each element being the result of the callback function.

// Inline callback function

```
map(function callbackFn(element, index, array) { ... }, thisArg)
```

Parameter (The description is the same as upper)

callbackFn

-element

-index(Optional)

-array(Optional)

thisArg (Optional)

Output

A new array with each element being the result of the callback function.

4. Example

4.1 Arrow Function

-ใช้ map() เพื่อเข้าถึง เอา element ทุกตัวใน array age มาบวกเพิ่ม 1

```
const age = [18,19,20]
const nextAge = age.map((age) => {
    return age+1
});

console.log(nextAge);
```

-output

```
PROBLEMS    OUTPUT
[ 19, 20, 21 ]
```

4.2 Map Array with value Function

-ใช้ตัวแปรในการเรียก function โดยให้ เอาทุกสมาชิกใน array มาคูณกับ 2

```
let num = [1,2,3,4,5,6];
let mutiplyTwo = num.map(function (item) {
    return item*2;
});
console.log(mutiplyTwo);
```

-output

```
PROBLEMS    OUTPUT    DEBUG CONSOLE
[ 2, 4, 6, 8, 10, 12 ]
```

4.3 Map generically

-ใช้ method call ของ prototype map ในการเรียก parameter ที่เป็น string และ function แล้ว Return เป็น Array ที่แปลงเป็นเลขรหัส Ascii ของตัวอักษรใน method charCodeAt(0) เช่น H เป็น 72

```
let map = Array.prototype.map
let a = map.call('Hello world',function(x){
    return x.charCodeAt(0);
});
console.log(a);
```

-output

```
PROBLEMS   OUTPUT   DEBUG CONSOLE
[
   72, 101, 108, 108,
  111,  32, 119, 111,
  114, 108, 100
]
```

4.4 Map with convertor

-แปลงจาก string เป็น number ในแต่ละ element ของ array โดยใช้ method parseInt

```
let stringArray = ['1','2','3'];
let numberArray = stringArray.map(str => parseInt(str));

console.log(numberArray);
```

-output

```
PROBLEMS   OUTPUT
[ 1, 2, 3 ]
```

Array.prototype.concat()

1. **Method Syntax**

concat()
concat(value0)
concat(value0, value1)
concat(value0, value1, ... , valueN)

valueN is an optional

2. **Description**

This method is used to combine two or more arrays together. This method also does not change any element on the existing array, but instead with a new array.

concat() will copy objects from references into the new array. Both the original and new array refer to the same object. That is, if a referenced object is modified, the changes are visible to both the new and original arrays. This includes elements of array arguments that are also arrays.

3. **How to use**

concat()
concat(value0, value1, ... , valueN)

Parameter

valueN (optional) - Arrays and/or values to concatenate into a new array. If all valueN parameters are omitted, concat returns a shallow copy of the existing array on which it is called.

Output

new array with concat by firstArray and all elements have concatenating.

4. **Example**

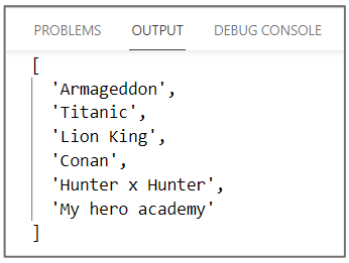
4.1 Concatenating two arrays

-ต้องการสร้างอาร์เรย์ที่แสดงชื่อหนังและอนิเมะโดยนำอาร์เรย์ anime มา concat กับอาร์เรย์ movie

-ต้องการให้ข้อมูลอาร์เรย์ใดขึ้นก่อนให้นำไว้ข้างหน้า ตามด้วย .concat(อาร์เรย์หลัง)

```
let movie = ['Armageddon','Titanic','Lion King'];
let anime = ['Conan','Hunter x Hunter','My hero academy'];
let movieAndAnime = movie.concat(anime)
console.log(movieAndAnime);
```

-output



4.2 Concatenating three arrays

-ต้องการสร้างอาร์เรย์ที่เก็บเลขจำนวนเฉพาะตั้งแต่ 1-1000 จากอาร์เรย์ที่มีอยู่สามอาร์เรย์คือ อาร์เรย์ที่เก็บเลขจำนวนเฉพาะที่มี 1 หลัก สองหลัก และสามหลัก

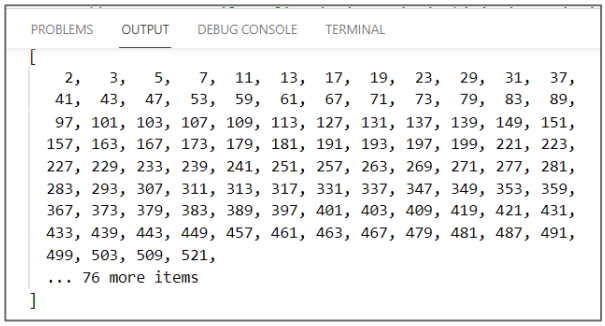
-ในกรณีที่เชื่อมมากกว่าสามอาร์เรย์ใช้ syntax ดังนี้ อาร์เรย์แรก.concat(อาร์เรย์สอง,อาร์เรย์สาม)

```
const prime = [2, 3, 5, 7];
const primeTwoDigit = [ 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97];
const primeThreeDigit = [101, 103, 107, 109, 113, 127, 131, 137, 139, 149, 151, 157, 163, 167, 173, 179, 181,
    191, 193, 197, 199, 221, 223, 227, 229, 233, 239, 241, 251, 257, 263, 269, 271, 277, 281, 283, 293, 307,
    311, 313, 317, 331, 337, 347, 349, 353, 359, 367, 373, 379, 383, 389, 397, 401, 403, 409, 419, 421, 431,
    433, 439, 443, 449, 457, 461, 463, 467, 479, 481, 487, 491, 499, 503, 509, 521, 523, 533, 541, 547, 559,
    563, 569, 571, 577, 587, 593, 599, 601, 607, 611, 613, 617, 619, 631, 641, 643, 647, 653, 659, 661, 673,
    677, 683, 689, 691, 701, 709, 719, 727, 733, 739, 743, 751, 767, 769, 773, 787, 793, 797, 809, 811, 821,
    823, 827, 829, 839, 853, 857, 859, 863, 871, 877, 881, 883, 887, 907, 911, 919, 923, 929, 937, 941, 947,
    949, 953, 967, 971, 977, 983, 991, 997];

const primeOneToThousand = prime.concat(primeTwoDigit, primeThreeDigit);

console.log(primeOneToThousand);
```

-output



4.3 Concatenating values to an array

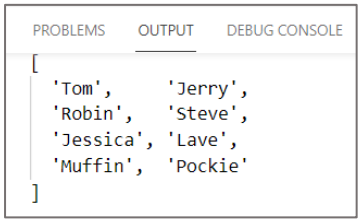
-ต้องการสร้างอาร์เรย์ที่เก็บข้อมูลของสมาชิกภายในครอบครัวของTom และสัตว์เลี้ยงจากอาร์เรย์ TomFamily ที่เก็บรายชื่อสมาชิก(คน)ในครอบครัวของทอม

-เมื่อข้อมูลที่นำมาเชื่อมเป็น value สามารถนำข้อมูลที่ต้องการมาใส่ในวงเล็บของ concat ได้เลย

```
const TomFamily = ['Tom', 'Jerry', 'Robin', 'Steve', 'Jessica', 'Lave'];
const TomMember = TomFamily.concat('Muffin', 'Pockie');

console.log(TomMember);
```

-output



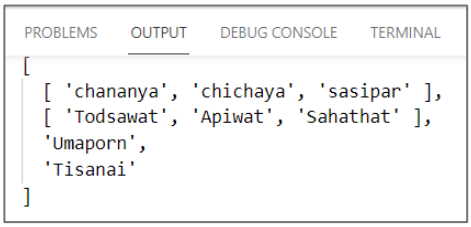
4.4 Concatenating nested arrays

-ต้องการสร้างอาร์เรย์ใหม่จากอาร์เรย์ที่เก็บชื่อของ teacher และ student โดยในอาร์เรย์ student เป็น nest array แบ่งเป็นรายชื่อของนักเรียนชายและนักเรียนหญิง

```
let teacher = ['Umaporn','Tisanai'];
let student = [['chananya','chichaya','sasipar'],['Todsawat','Apiwat','Sahathat']];

let group2 = student.concat(teacher);
console.log(group2);
```

-output



จะเห็นได้ว่า output ที่ได้เป็น nest array

Member Group2

1. 63130500108 Sasipar Maneein
2. 63130500113 Sahathat Yingsakulkiet
3. 63130500130 Apiwat Atittieng
4. 63130500140 Chananya Sinphichit
5. 63130500150 Chichaya Marod
6. 63130500155 Todsawat Somtua