

## ES 6 18 feb 22

মাইলস্টোন ৬ এর টেনটেটিভ আউটলাইন:

ফেব্রুয়ারি ১৮: মডিউল ৩১: JS Recap, ES6 Intro

ফেব্রুয়ারি ১৯: মডিউল ৩২: ES6 Recap, More ES6

ফেব্রুয়ারি ২০: রাত ৯.০০ কন্সেপচুয়াল সেশন

ফেব্রুয়ারি ২০: মডিউল ৩২.৫: Practice/Revision Day

ফেব্রুয়ারি ২১: মডিউল ৩৩: Getting started with API

ফেব্রুয়ারি ২২: মডিউল ৩৪: More API With Example

ফেব্রুয়ারি ২৩: রাত ৯.০০ কন্সেপচুয়াল সেশন

ফেব্রুয়ারি ২৩: মডিউল ৩৪.৫: Practice/Revision Day

ফেব্রুয়ারি ২৪: মডিউল ৩৫: JavaScript Tricky Concepts

ফেব্রুয়ারি ২৫: মডিউল ৩৬: Object Mastering

ফেব্রুয়ারি ২৬: রাত ৯.০০ কন্সেপচুয়াল সেশন

ফেব্রুয়ারি ২৬: মডিউল ৩৬.৫: Practice/Revision Day

ফেব্রুয়ারি ২৭: নো Video: API প্রাকটিস ডে

ফেব্রুয়ারি ২৮: মডিউল ৩৭: এসাইনমেন্ট ৬

মার্চ ১: মডিউল ৩৭.৫: বোনাস মডিউল

আজকের মডিউল থেকে নিচের তিনটা জিনিস মাস্ট বুঝে নিবে

১. কথন const আর কথন let দিয়ে ভেরিয়েবল ডিক্লেয়ার করতে হয়

২. টেম্পলেট স্ট্রিং এর মধ্যে ডাইনামিক ভাবে কিভাবে কোন একটা ভেরিয়েবলকে বসানো যায়

৩. একটা প্যারামিটার ওয়ালা arrow ফাংশন কিভাবে ডিক্রিয়ার করতে হয়।

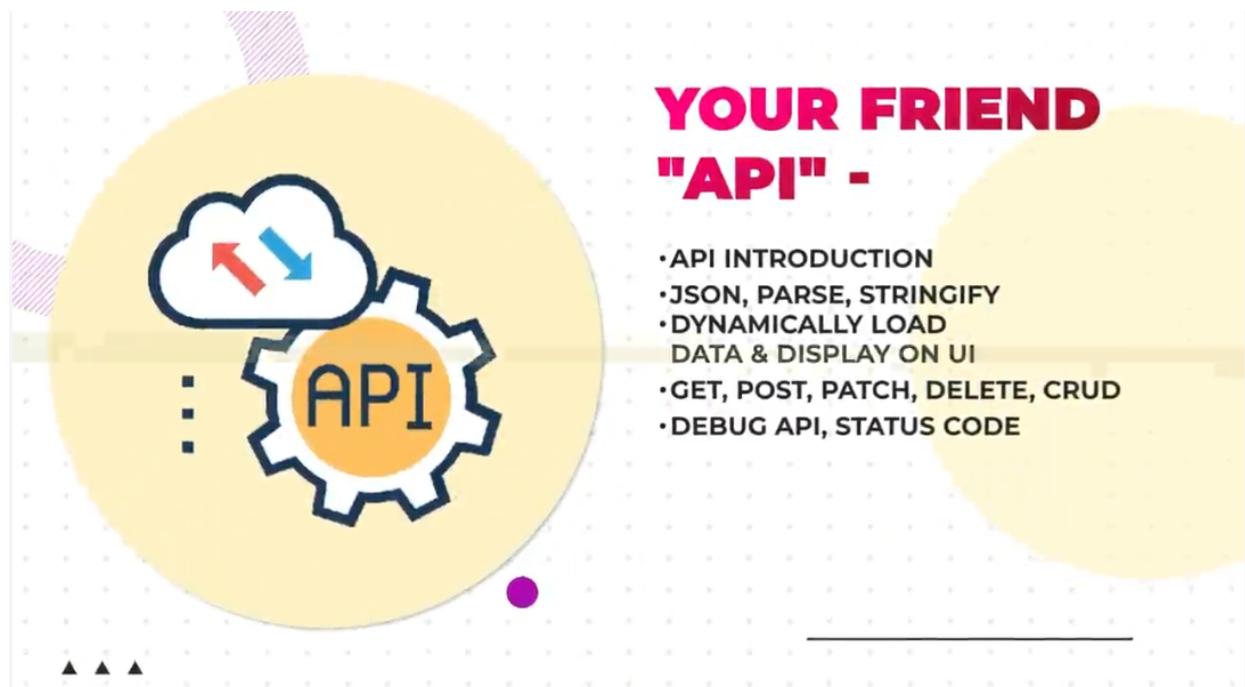
নিচের তিনটা জিনিস বুঝে নিতে পারলে ফিউচারে কাজে লাগবে

১. কিভাবে থি ডট দিয়ে (...) একটা এরে এর সব এলিমেন্ট বসিয়ে দেয়া যায়

২. কোন একটা ফাংশনে এর প্যারামিটার ভ্যালু না দিলে কিভাবে ডিফল্ট প্যারামিটার দিতে হয়

৩. অনেক লাইন ওয়ালা arrow ফাংশন কিভাবে লিখতে হয়। আর কিভাবে সেটা থেকে রিটোর্ন করতে হয়

ব্যস, এই টুকুই।



Parameter er vitor value= value thaka mane call korar somoy value set na korle oita pabe

```
1  function add(num1, num2 = 1000) {  
2      sum = num1 + num2;  
3      return sum  
4  }  
5  const numbers = add(10)  
6  console.log(numbers);
```

### Unlimited parameters to a function using arguments

Onk besi parameter pass korte arguments diye kaj korbo .

```
js arguments.js > ...  
1  function numbers() {  
2      let result = 0  
3      for (const num of arguments) {  
4          result = result + num  
5      }  
6      return result  
7  }  
8  console.log(numbers(3, 5, 42, 100, 200))
```

## Template string, multiple line string, dynamic string

```
template-string.js > multiLine
1 const priya = 'Asif Akbar';
2 const meye = "Meye tumi ki dukho chino";
3 const kobita = `kobita tumi sopno charini`;
4 const multiLine = 'This is my first line. \n' .
5   'this is my second line.' +
6   'third line text here' +
7   'fourth line text here';
8
9 console.log(multiLine);

```

PROBLEMS 12 OUTPU<sup>T</sup> \n এর মানে \n পর্যন্ত যা আছে সেগুলো cmtriazessamit@gmail.com

Try the new c নতুন লাইন খেতে শুরু হয়ে।

PS C:\projects> node template-string.js  
This is my first line.this is my second line.third line text herefourth l

Backtick diye easy vabe kora jay

```
Template literals
S C:\> his is my first line.
his is my second line.
third line text here
ourth line text here
```

এ অনায়াসেই এতাধিক লাইনে লিখে ফেলতে পারবে।

# String Interpolation

emtiazhossainrzs@gmail.com

## Syntax: \${...}

স্ট্রিং এর মধ্যে ভেরিয়েবল এর নাম দিয়ে ভেরিয়েবলের  
মান ঠিকানাতে string interpolation হলো।

```
1>
20 const first = 'Mamun';
21 const last = 'Chowdhury';
22 const fullOld = first + ' ' + last;
23 const fullNew = `${first} ${last}`;
24
```

PROBLEMS 22

OUTPUT

TUTORIAL

DEBUG CONSOLE



Template literals স্ট্রিং এর ভিতর ইচ্ছেমতো জাভাস্ক্রিপ্ট  
এর ট্রেক (expressions) লিখতে পারবে।

More es6

```
const fish = { id: 58, name: 'King Hilsha', price: 9000,  
'017175555555', address: 'Dhaka', category: 'silver' }  
const { name } = user;  
  
// const phone = fish.phone  
// const price = fish.price  
// const address = fish.address  
// const category = fish.category  
  
const { price, address, category } = fish  
  
const user = {  
  name: 'Alex',  
  age: 43  
}  
  
OUTPUT TERMINAL  
55555555 9000  
55555555 58  
55555555 silver  
55555555 9000
```

8    const f, phone, price, address, id, category, fish;  
9    অবজেক্টে এসাধিত প্রপার্টি এর ভ্যালু  
PROBLEMS 5    OUTPUT TERMINAL DEBUG CONSOLE  
এসাথে দেখ দেখ ভেঁড়িয়ে দেখ এ গাখাকে  
0171755555555555 000  
0171755555555555 8  
0171755555555555 silver  
0171755555555555 9000  
10  
11    const company = {  
12     name: 'GP',  
13     ceo: { id: 1, name: 'ajmol', food: 'fuchka' },  
14     web: { work: 'website development', employee: 22, framework:  
15       'react' }  
16 };  
17    |||  
18    |||  
19    |||  
20    |||  
21    |||

```
1 // const numbers = [5, 10, 15, 20, 25]
2 // const output = []
3 /*
4 function doubleOld(number) {
5     return number * 2
6 }
7 const doubleIt = number => number * 2
8 */
9 // for (const number of numbers) {
10 //     const result = doubleIt(number)
11 //     output.push(result)
12 // }
13
14 /* 1.loop throw each element
15 2. element diye function ke call korci
16 3.result ekta array er moddhe push korci */
17
18 // console.log(output)
19
20 //map diye same kaj kora jay
21 const numbers = [5, 10, 15, 20, 25]
22 const value = []
23 const doubleIt = number => number * 2
24 const output = numbers.map(doubleIt) //map uporer kaj gulan e kore
25 console.log(output)
```



**map()** অ্যারের প্রতিটি উপাদান এর জন্য  
ফাংশনকে ঢল ঢরে এবং সেই ফাংশনের ফেজাল্টকে  
একটি নতুন অ্যারে তে নিয়ে পিটোর্ন ঢরে।

# forEach() একটি অ্যারের প্রতিটি

emtiazhossainrzs@gmail.com

উপাদানের জন্য একটি ফাংশন ঢল ঢয়ে।

তিক্ত সোন তিক্ত পিটার্ন ঢয়ে না।

```
1 const numbers = [5, 13, 7, 41, 30, 5, 2, 19];
2 const bigNumbers = numbers.filter(number => number > 20);
3 const smallNumbers = numbers.filter(number => number < 10);
4 console.log(smallNumbers);
```

[]

emtiazhossainrzs@gmail.com

PROBLEMS 15 OUTPUT TERMINAL DEBUG CONSOLE

```
{ name: 'Avengers: Endgame', release: 2019, rating: 8.5 }
PS C:\projects\more-es6> node filter.js
[ { name: 'Avengers: Endgame', release: 2019, rating: 8.5 }
  ]
```

```

//৫. জাভাস্ক্রিপ্ট এ array এর map, forEach, filter, find কোনটা দিয়ে কি হয়। সেটার একটা
সামান্য লিখে ফেলো।
/*
map() array er potiti upadaner jonne function call kore & sob function er result ke
notun array te return kore .

forEach() array er potita upadan er jonner function ke call kore kintu kono kicu
return krore nah .

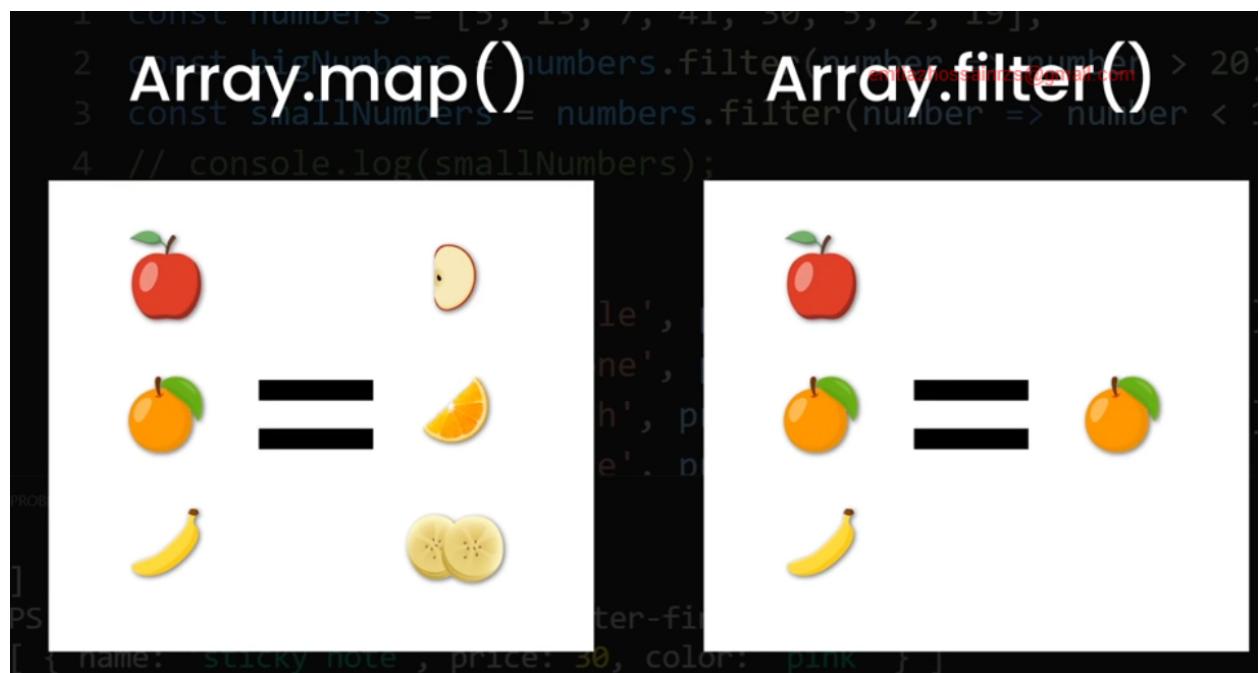
filter() array er potita upadan er muddhe jeta jeta condition fullfill korbe seta ke
ekta array te rekhe seta return korbe

find() array er potita upadan er muddhe jeta first e condition fillup korbe setai just
return korbe

this ==> class er vitore obj er property gulo access korar jonne this use kora hoy .

*/

```



```
# filter-find.js > (0) whitetem > (0) products.find() callback
  15  const cons = log(expensive);
  16  const expensive = products.filter(product => product.color == 'pink');
  17  // console.log(blocks);
  18  [ [red], [purple], [yellow], [blue], [green] ] → [ [red], [purple], [blue] ]
  19  ];
  20  console.log(whiteItem);

find ( )
[ [red], [purple], [yellow], [blue], [green] ] → [red]
{ name: 'mobile phone', price: 15000, color: 'black' }
PS C:\projects\more-es6> node filter-find.js
```



**class এর ভিতরে অবজেক্ট এর প্রপার্টিগুলো  
এতসেস ক্যার জন্য this ইউজ করা হয়।**

### Prototypical inheritance

Javascript e ekta obj theke r ekta obj er property gulo je shared hocce eta ekta chain diye kaj hoy .ta kei **Prototypical inheritance bole** .

**Simply put, prototypical inheritance refers to the ability to access object properties from another object. We use a JavaScript prototype to add new properties and methods to an existing object constructor. ... All JavaScript objects inherit properties and methods from a prototype: Date objects inherit from Date.**

### What will the filter method return?

The filter() method returns **an array containing elements of the parent array that match the set test**.

### **## search google → javascript function declaration vs arrow function**

As you see both functions work same by above example. Now the question comes why do we need regular or arrow function.

Let's discuss below 

## 1. Syntax

## 2. Arguments binding

## 3. this

## 4. new keyword

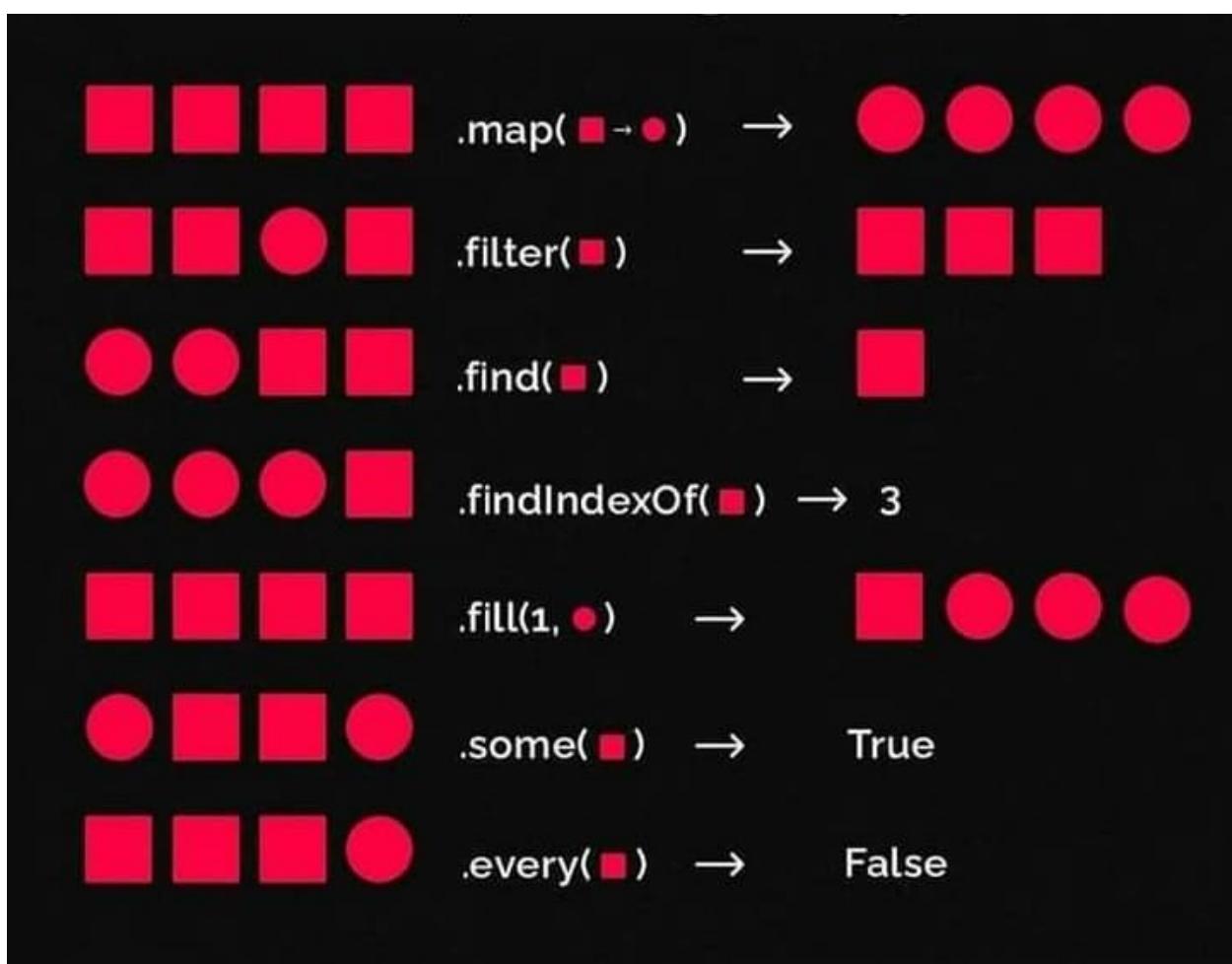
## 5. No duplicate named parameters

## 6. Function Hoisting

## 7. Methods

জাভাস্ক্রিপ্ট এর var, let, const এর মধ্যে প্রার্থক্য কি?

////



GET এবং POST এর মধ্যে পার্থক্য

GET দিলে ফর্মের ডেটা URI এর মধ্যে দেখা যায় যেমন ওগলের সার্চ ফর্মে GET ব্যবহার করা হয়েছে। আর POST দিলে ডেটা লুকানো থাকে। লগিন, রেজিস্টার ফর্ম ইত্যাদির জন্য POST মেথড ব্যবহার করা উচিত।

What is Status code ?

Status codes are **issued by a server in response to a client's request made to the server.**

... The first digit of the status code defines the class of response, while the last two digits do not have any classifying or categorization role.

**To get something you never had, you have to do something you never did.**

```
js api.js          js arrow-function.js •
es6-intro > js arrow-function.js > ...
1  function add(num1, num2) {
2  |    return num1 + num2
3  }
4  //function expression
5  const add2 = function add2(num1, num2) {
6  |    return num1 + num2
7  }
8  //function expression (anonymous function )
9  const add3 = function (num1, num2) {
10 |    return num1 + num2
11 }
12 //arrow function
13 const add4 = (num1, num2) => num1 + num2;
14
15 const sum1 = add(10, 20)
16 const sum2 = add2(20, 20)
17 const sum3 = add3(30, 20)
18 const sum4 = add4(40, 20)
19 console.log(sum1, sum2, sum3, sum4);
```

