



## ES6 Cú pháp mới

### Mục tiêu:

- Hiểu được lịch sử của Javascript và phiên bản ES6 của JS
- Hiểu được các feature mới của ES6: Arrow function, Classes, Block scope, Rest/Spread, Destructuring, Template string, Map/Set
- Sử dụng được các cú pháp mới của ES6 để code có thể dễ đọc hơn.

### Exercise 01

Cho đoạn code sau, nhiệm vụ của bạn là thay đổi function declaration/ function expression of ES5 thành cú pháp arrow function của ES6 mà output của function vẫn phải giữ nguyên là **true**

```
1. (function IIFE() {
2.     function foo(x) {
3.         var y = x * 2;
4.
5.         return function bar(z) {
6.             if (z.length > 3) {
7.                 return z.map(function baz(v) {
8.                     if (v > 3) return v + y;
9.                     else return baz(v * 4);
10.                });
11.             } else {
12.                 var obj = [];
13.
14.                 setTimeout(
15.                     function bam() {
16.                         obj.length = 1;
17.                         obj[0] = this.w;
18.                     }.bind(this),
19.                     100
20.                 );
21.
22.                 return obj;
23.             }
24.         };
25.     }
26.
27.     var p = foo(2);
28.     var list1 = [1, 3, 4];
29.     var list2 = list1.concat(6);
30.
31.     list1 = p.call({ w: 42 }, list1);
32.     list2 = p(list2);
33.
34.     setTimeout(function() {
35.         console.log(
36.             list1[0] ===
37.             list2.reduce(function(s, v) {
38.                 return s + v;
39.             }, 0)
40.         );
41.     }, 200);
42. })();
```

## Exercise 02

Sử lại đoạn code sau sao cho output vẫn là `true`.

```
1. var x = 2,
2. fns = [];
3.
4. (function() {
5.   var x = 5;
6.
7.   for (var i = 0; i < x; i++) {
8.     // ..
9.   }
10. })();
11.
12. // DO NOT MODIFY BELOW CODE
13. console.log(x * 2 === fns[x * 2]());
14. // true
```

## Exercise 03

Sử dụng rest/spread operator để đoạn code sau vẫn có output là `true`

```
1. function foo() {}
2.
3. function bar() {
4.   var a1 = [2, 4];
5.   var a2 = [6, 8, 10, 12];
6.
7.   return foo();
8. }
9.
10. // DO NOT MODIFY BELOW CODE
11. console.log(bar().join('') === '281012');
12. // true
```

## Exercise 04

Cho đoạn code sau, sử dụng tính năng ES6 Destructuring để cấu trúc lại data cho function kiểm tra để đầu ra output vẫn là `true`.

```
1. function ajax(url, cb) {
2.   // fake ajax response:
3.   cb({
4.     foo: 2,
5.     baz: [6, 8, 10],
6.     bam: {
7.       qux: 12
8.     }
9.   });
10. }
11.
12. function check(data) {
13.   console.log(
14.     56 ===
15.     data.foo +
16.     data.bar +
17.     data.baz[0] +
18.     data.baz[1] +
19.     data.baz[2] +
20.     data.bam.qux +
```

```
21.         data.bam.qam
22.     );
23. }
24.
25. var defaults = {
26.     foo: 0,
27.     bar: 4,
28.     bam: {
29.         qux: 0,
30.         qam: 14
31.     }
32. };
33.
34. // YOUR CODE HERE
35. function response(...) {
36.     check(...); // true
37. }
38.
39. // DO NOT MODIFY
40. ajax('http://fun.tld', response);
```

### **Exercise 05**

Cho đoạn code sau, hoàn thành **upper** function và sử dụng nó như một tag function để template string sau ``Hello ____(@____), welcome to the ____!!!`` vẫn cho đầu ra là ``true``

```
1. function upper(strings, ...values) {}
2.
3. var name = 'Nguyen Van A',
4.     account = 'ANV',
5.     classname = 'Fresher FrontEnd';
6.
7. console.log(
8.     `Hello ____(@____), welcome to the ____!!!` ===
9.     'Hello NGUYEN VAN A (@ANV), welcome to the FRESHER FRONTEND!!!'
10.);
```