

Discussion 3

Exercise

Array Includes

Function `a.includes(x)` returns true if and only if the `a` contains `x`

Exercise

Array Includes

Function `a.includes(x)` returns true if and only if the `a` contains `x`

What's the problem?

Exercise

Array Includes

Function `a.includes(x)` returns true if and only if the `a` contains `x`

What's the problem?

How could we fix it?

What does the following program output?

```
1 function M(f, g, h) {  
2   |   f(g, h);  
3   | }  
4  
5 M(function(v, w) { console.log("A"); w(); },  
6   |   function() { console.log("B"); },  
7   |   function() { console.log("C"); });
```

What does the following program output?

```
1 function M(f, g, h) {  
2   | f(g, h);  
3   | }  
4  
5 M(function(v, w) { console.log("A"); w(); },  
6   | function() { console.log("B"); },  
7   | function() { console.log("C"); });
```

A
C

Consider the following function:

```
function myFunction1(a, b) {  
  return a.map(function(x) { return x.length + b; });  
}
```

Give three distinct pairs of inputs `a, b` for which the function **does not** throw any errors. **Do not use empty arrays**

Fun with Types

1) What is the type of bar

```
function foo(bar, baz) {  
  return bar.filter(baz).reduce(function(acc, x){return acc + x;}, 0);  
}
```


Fun with Types

- 1) What is the type of bar - Array
- 2) What is the type of baz

```
function foo(bar, baz) {  
  return bar.filter(baz).reduce(function(acc, x){return acc + x;}, 0);  
}
```

Fun with Types

- 1) What is the type of bar - Array
- 2) What is the type of baz - Function
- 3) What does this return with the input:

([1, 2, 3, 4], function(x) {return x%2 === 0;})

```
function foo(bar, baz) {  
  return bar.filter(baz).reduce(function(acc, x){return acc + x;}, 0);  
}
```

Fun with Types

- 1) What is the type of bar - Array
- 2) What is the type of baz - Function
- 3) What does this return with the input:

([1, 2, 3, 4], function(x) {return x%2 === 0;})

6

```
function foo(bar, baz) {  
  return bar.filter(baz).reduce(function(acc, x){return acc + x;}, 0);  
}
```

What does this output?

```
function identity(x) {  
  return x;  
}  
  
let exArray = [{x: 1}, {x: 2}, {x: 3}];  
let otherArray = exArray.map(identity);  
otherArray[0].x = 10;  
console.log(exArray[0].x);
```

What does this output?

```
function identity(x) {  
  return x;  
}  
  
let exArray = [{x: 1}, {x: 2}, {x: 3}];  
let otherArray = exArray.map(identity);  
otherArray[0].x = 10;  
console.log(exArray[0].x);
```

What does this output?

```
2 function weird(d) {
3   let f = function(y) {
4     let v = d.x;
5     // d.x += y;
6     d = {x: v + y};
7     return v;
8   }
9   return f;
10 }
11 let a = {x: 8};
12 let b = a;
13 let fa = weird(a);
14 let fb = weird(b);
15 console.log(fa(2));
16 console.log(fa(2));
17 console.log(fb(3));
18 console.log(fb(3));
19 console.log(a);
20 console.log(b);
```

8

10

8

11

{x: 8}

{x: 8}

What does this output?

```
2 function weird(d) {
3   let f = function(y) {
4     let v = d.x;
5     d.x += y;
6     // d = {x: v + y};
7     return v;
8   }
9   return f;
10 }
11 let a = {x: 8};
12 let b = a;
13 let fa = weird(a);
14 let fb = weird(b);
15 console.log(fa(2));
16 console.log(fa(2));
17 console.log(fb(3));
18 console.log(fb(3));
19 console.log(a);
20 console.log(b);
```


8

10

12

15

{x: 18}

{x: 18}