

# JavaScript Array Functions

## filter, map, and reduce

Samuel Rios-Lazo – WDD 230 – Winter 2024

# filter()

- Method that creates a new array
- The new array includes all the elements that pass a certain criteria
- The criteria is specified as a function that we provide



# filter()

```
1 const names = [
2   "samuel",
3   "bob",
4   "austin",
5   "tim",
6   "kirsten",
7   "sue",
8   "ronald",
9   "stephen",
10 ];
11
12 const result = names.filter(word => word.length > 5);
13
14 console.log(result);
15
```

```
1 const names = ["samuel", "austin", "kirsten", "ronald", "stephen"];
2
3 startWithS = names.filter(checkNameS);
4
5 function checkNameS(name) {
6   return name.charAt(0) === "s";
7 }
8
9 console.log(startWithS);
10
```

- (base) samuel@Samuels-MacBook-Air-9 presentation % node second.js  
[ 'samuel', 'stephen' ]
- (base) samuel@Samuels-MacBook-Air-9 presentation %

- (base) samuel@Samuels-MacBook-Air-9 presentation % node one.js  
[ 'samuel', 'austin', 'kirsten', 'ronald', 'stephen' ]
- (base) samuel@Samuels-MacBook-Air-9 presentation %



# map()

- Method that creates a new array
- The new array includes all the elements after a specific function was called for each of them
- The criteria for the transformation that the elements suffer is determined by a function that we provide

```
1 const numbers = [1, 2, 3, 4, 5];
2
3 const map = numbers.map((x) => x * 2);
4
5 console.log(map);
6
```

```
● (base) samuel@Samuels-MacBook-Air-9 presentation % node third.js
[ 2, 4, 6, 8, 10 ]
○ (base) samuel@Samuels-MacBook-Air-9 presentation %
```

```
1 let numbers = [4, 9, 16, 25];
2 let newArr = numbers.map(Math.sqrt);
3
4 console.log(newArr);
5
```

```
● (base) samuel@Samuels-MacBook-Air-9 presentation % node fourth.js
[ 2, 3, 4, 5 ]
○ (base) samuel@Samuels-MacBook-Air-9 presentation %
```

# map()

```
1 names = ["Whitney", "Tendai", "Nefi", "Jessica", "Amanda", "Heather", "Rachel"];
2
3 namesFixed = names.map(addDr);
4
5 function addDr(name) {
6   return `Dr. ${name}`;
7 }
8
9 console.log(namesFixed);
10 |
```

```
● (base) samuel@Samuels-MacBook-Air-9 presentation % node fifth.js
[
  'Dr. Whitney',
  'Dr. Tendai',
  'Dr. Nefi',
  'Dr. Jessica',
  'Dr. Amanda',
  'Dr. Heather',
  'Dr. Rachel'
]
○ (base) samuel@Samuels-MacBook-Air-9 presentation %
```

# reduce()

- Method that returns a single value
- The value it returns is the function's accumulated result
- It takes a reducer function
- It executes the reducer function on each element of the array in order, and then it passes the return value of the calculation on the preceding element



# reduce()

```
1 const numbers = [175, 50, 25];
2
3 // With no initial value, index [0] serves that purpose.
4 result = numbers.reduce(myFunc);
5
6 function myFunc(total, num) {
7   return total - num;
8 }
9
10 console.log(result);
```

```
1 const array = [1, 2, 3, 4];
2
3 const initialValue = 10;
4 const sumWithInitial = array.reduce(
5   (accumulator, currentValue) => accumulator + currentValue,
6   initialValue
7 );
8
9 console.log(sumWithInitial);
10 |
```

- (base) samuel@Samuels-MacBook-Air-9 presentation % node sixth.js  
100
- (base) samuel@Samuels-MacBook-Air-9 presentation %

- (base) samuel@Samuels-MacBook-Air-9 presentation % node seventh.js  
20
- (base) samuel@Samuels-MacBook-Air-9 presentation %