

Arrays II



Overview

```
1  /*
2    - more methods
3
4    - nested arrays
5  */
6
7
8
9
10
11
12
13
14
```



[George, Thomas]
[John]

.splice method

```
1  /* .splice mutates the original array; it's used to add or remove elements
2     from the middle of an array (instead of adding or removing from either
3     end */
4
5  /* its first argument is an index; the second is a count of elements to
6     delete from the element, starting at the provided index */
7
8  /* it returns a new array with all of the removed elements */
9  let names = ['George', 'John', 'Thomas'];
10
11 let singleTermPresidents = names.splice(1, 1);
12
13 console.log(names);
14 console.log(singleTermPresidents);
```



```
[ Paul, Ringo ]  
[ George, John ]
```

.splice method

```
1 let names = ['Paul', 'George', 'John', 'Ringo'];  
2  
3 let acrossTheUniverse = names.splice(1, 2);  
4  
5 console.log(names);  
6 console.log(acrossTheUniverse);  
7  
8  
9  
10  
11  
12  
13  
14
```



```
[ Paul, George, Ringo, John ]  
[ Pete ]
```

.splice method

```
1  /* you can optionally add new elements at the given index */  
2  
3  let names = ['Paul', 'George', 'Pete', 'John'];  
4  
5  let formerMembers = names.splice(2, 1, 'Ringo');  
6  
7  console.log(names);  
8  console.log(formerMembers);  
9  
10  
11  
12  
13  
14
```



```
[ Paul, George, Pete, Stuart,  
Ringo, John ]  
[]
```

.splice method

```
1  /* you don't have to remove any elements! */  
2  
3  let names = ['Paul', 'George', 'Ringo', 'John'];  
4  
5  let removedElements = names.splice(2, 0, 'Pete', 'Stuart');  
6  
7  console.log(names);  
8  console.log(removedElements);  
9  
10  
11  
12  
13  
14
```



string
Paul,George,John,Ringo

.join method

```
1  /* .join concatenates the elements of an array into a string; the original
2     array is not changed */
3
4  let names = ['Paul', 'George', 'John', 'Ringo'];
5
6  let joinedString = names.join();
7
8  console.log(typeof joinedString);
9  console.log(joinedString);
10
11
12
13
14
```



.join method

```
1 let names = ['Paul', 'George', 'John', 'Ringo'];
2
3 let joinedString = names.join(' and '); // add a separator
4
5 console.log(joinedString);
6
7
8
9
10
11
12
13
14
```




```
[ Ringo, John, Paul, George ]  
[ Ringo, John ]
```

.concat method

```
1  /* concat merges two or more arrays into one */  
2  
3  /* it returns a new array and doesn't change the array on which is was  
4     called */  
5  
6  let older = ['Ringo', 'John'];  
7  let younger = ['Paul', 'George'];  
8  
9  let allTogetherNow = older.concat(younger);  
10  
11 console.log(allTogetherNow);  
12 console.log(older);  
13  
14
```



[Windows, MacOS]
New York
2

Nested arrays

```
1  /* arrays can contain any type of value, including other arrays */
2
3  let relatedThings = [['Windows', 'MacOS'], ['New York', 'Chicago']];
4
5  console.log(relatedThings[0]);
6  console.log(relatedThings[1][0]);
7  console.log(relatedThings.length);
8
9
10
11
12
13
14
```



Jane
Mel
Jack
Rohan
David
Meg

Nested arrays: looping

```
1 let rsvpGroups = [['Jane', 'Mel'], 'Jack', ['Rohan', 'David', 'Meg']];
2
3 for (let i = 0; i < rsvpGroups.length; i++) {
4   let element = rsvpGroups[i]; // not sure if this is a string or an array
5   if (Array.isArray(element)) {
6     for (let j = 0; j < element.length; j++) {
7       let name = element[j];
8       console.log(name);
9     }
10  }
11  else {
12    console.log(element)
13  }
14 }
```



[1, 1, 1]

Nested arrays: grid

```
1  /* use nested arrays to represent a grid in code */
2
3  /* inner arrays represent the rows of the grid */
4
5  /* each index of the rows represents a column */
6
7  let grid = [
8    [1, 2, 3],
9    [1, 2, 3],
10   [1, 2, 3]
11 ];
12
13 let firstColumn = [grid[0][0], grid[1][0], grid[2][0]];
14 console.log(firstColumn);
```



Recap

```
1  /*
2    - more methods
3
4    - nested arrays
5  */
6
7
8
9
10
11
12
13
14
```