## Arrays II

#### Overview

```
- more methods
- nested arrays
*/
```



## .splice method

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

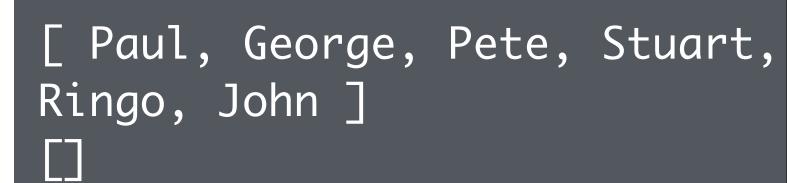
## splice method

```
let names = ['Paul', 'George', 'John', 'Ringo'];
let acrossTheUniverse = names.splice(1, 2);
console.log(names);
console.log(acrossTheUniverse);
```



## splice method

```
/* you can optionally add new elements at the given index */
   let names = ['Paul', 'George', 'Pete', 'John'];
   let formerMembers = names.splice(2, 1, 'Ringo');
   console.log(names);
   console.log(formerMembers);
10
```



## .splice method

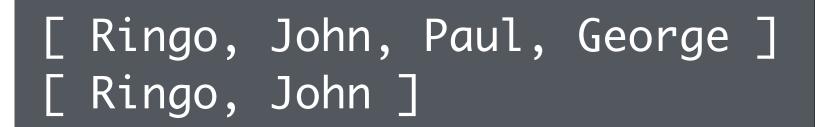
```
/* you don't have to remove any elements! */
   let names = ['Paul', 'George', 'Ringo', 'John'];
   let removedElements = names.splice(2, 0, 'Pete', 'Stuart');
   console.log(names);
   console.log(removedElements);
10
```

## .join method

```
/* .join concatenates the elements of an array into a string; the original
      array is not changed */
   let names = ['Paul', 'George', 'John', 'Ringo'];
   let joinedString = names.join();
   console.log(typeof joinedString);
   console.log(joinedString);
10
```

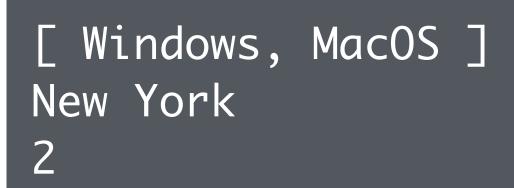
## .join method

```
let names = ['Paul', 'George', 'John', 'Ringo'];
   let joinedString = names.join(' and '); // add a separator
   console.log(joinedString);
10
```



#### .concat method

```
/* concat merges two or more arrays into one */
   /* it returns a new array and doesn't change the array on which is was
      called */
   let older = ['Ringo', 'John'];
   let younger = ['Paul', 'George'];
   let allTogetherNow = older.concat(younger);
10
   console.log(allTogetherNow);
   console.log(older);
```



## Nested arrays

```
/* arrays can contain any type of value, including other arrays */
let relatedThings = [['Windows', 'MacOS'], ['New York', 'Chicago']];
console.log(relatedThings[0]);
console.log(relatedThings[1][0]);
console.log(relatedThings.length);
```

# Jane Mel Jack Rohan David Meg

## Nested arrays: looping

```
let rsvpGroups = [['Jane', 'Mel'], 'Jack', ['Rohan', 'David', 'Meg']];
for (let i = 0; i < rsvpGroups.length; i++) {
  let element = rsvpGroups[i]; // not sure if this is a string or an array
  if (Array.isArray(element)) {
    for (let j = 0; j < element.length; <math>j++) {
      let name = element[j];
      console.log(name);
  else {
    console.log(element)
```

## Nested arrays: grid

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

## Recap

```
- more methods
- nested arrays
*/
```

## Sudoku Project

- 100% optional
- Good practice building a slightly larger program
- Instructions available in extra workshop on LearnDot
- Don't forget to write tidy code!
- Send your solution to the instructors by the end of class next Thursday if you want feedback!