

## 04.01 Lab Solutions

1. Declare an array called gems and give it three gems of your choice.

```
const gems = ['diamond', 'emerald', 'ruby'];
```

2. Log the entire array and also the first item.

```
console.log(gems); // ['diamond', 'emerald', 'ruby']  
console.log(gems[0]); // diamond
```

3. Add 'sapphire' to the end of the array, using the index of the new item. If sapphire is already in the array, add another gem.

```
gems[3] = 'sapphire';  
console.log(gems); // ['diamond', 'ruby', 'emerald', 'sapphire']
```

4. Using an array method, add 'amethyst' to the end of the array. If amethyst is already in the array, add some other gem.

```
gems.push('amethyst');  
console.log(gems); // ['diamond', 'ruby', 'emerald', 'sapphire',  
  'amethyst']
```

5. Use an array method to put the gems in alphabetical order.

```
gems.sort();  
console.log(gems); // ['amethyst', 'diamond', 'emerald', 'ruby',  
  'sapphire']
```

6. Log the number of items of the array. The result should be 5.

```
console.log(gems.length); // 5
```

7. Get the next-to-last gem, without deleting it from the array or hard-coding the index number. Save it to a variable and log it. HINT: length property

```
let nextToLastGem = gems[gems.length-2];
console.log(nextToLastGem); // ruby
```

8. Use one array method to remove the last gem from the array, and then another array method to put the item right back in at the end. HINT: Save the removed gem to a new variable.

```
let poppedGem = gems.pop();
console.log(poppedGem); // sapphire
console.log(gems); // ['amethyst', 'diamond', 'emerald', 'ruby']
gems.push(poppedGem);
console.log(gems); // ['amethyst', 'diamond', 'emerald', 'ruby', 'sapphire']
```

9. Given the 12 months: Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec.

- Make a nested array called 'months', consisting of 4 items, each an array
- The child arrays have 3 items each, so Jan, Feb, Mar are in the first array.
- Using double-square brackets, log all months that start with 'J' and 'M'.

```
const months = [ ['Jan', 'Feb', 'Mar'],
                  ['Apr', 'May', 'Jun'],
                  ['Jul', 'Aug', 'Sep'],
                  ['Oct', 'Nov', 'Dec']
                ];
console.log(months[0][0]); // Jan
console.log(months[0][2]); // Mar
console.log(months[1][1]); // May
console.log(months[1][2]); // Jun
console.log(months[2][0]); // Jun
```

10. Given this list of numbers: **111, 23, 202, 12, 23, 1000, 2006, 1, 2, 14, 28**. Put them into an array called nums and then sort the array in descending order. HINT: You need to use an array method that takes a callback function.

```
const nums = [111, 23, 202, 12, 23, 1000, 2006, 1, 2, 14, 28];

nums.sort(function(a,b) {
  return b - a;
});

console.log(nums);
```