Title: Must know array methods

Answer:

filter()

Creates a new array with all elements that pass the test implemented by the provided function.

```
const inventors = [
  { first: 'Marie', last: 'Curie', year: 1867, passed: 1934 },
  { first: 'Albert', last: 'Einstein', year: 1879, passed: 1955 },
  { first: 'Galileo', last: 'Galilei', year: 1564, passed: 1642 },
  { first: 'Isaac', last: 'Newton', year: 1643, passed: 1727 },
  { first: 'Johannes', last: 'Kepler', year: 1571, passed: 1630 }
];

// Filter the list of inventors for those who were born after 1600
let filter = inventors.filter(inventor => inventor.year > 1600);

console.log(filter);
>>
{
  { first: "Albert", last: "Einstein", year: 1879, passed: 1955 },
  { first: "Isaac", last: "Newton", year: 1643, passed: 1727 },
  { first: 'Marie', last: 'Curie', year: 1867, passed: 1934 },
}
```

map()

Creates a new array with the results of calling a provided function on every element in the calling array.

```
// Give us an array of the inventors' first and last names
let names = inventors.map(inventor => `${inventor.first} ${inventor.last}`);

console.log(names);
>> ["Albert Einstein", "Isaac Newton", "Galileo Galilei", "Johannes Kepler",
"Nicolaus Copernicus"]
```

sort()

Sorts the elements of an array in place and returns the array. The sort is not necessarily stable. The default sort order is according to string Unicode code points.

```
// Sort the inventors by birthdate, oldest to youngest
let sortByBirthday = inventors.sort((a, b) => a.year - b.year);

console.table(sortByBirthday);

{ first: "Nicolaus", last: "Copernicus", year: 1473, passed: 1543 },
{ first: "Galileo", last: "Galilei", year: 1564, passed: 1642 },
{ first: "Johannes", last: "Kepler", year: 1571, passed: 1630 },
{ first: "Isaac", last: "Newton", year: 1643, passed: 1727 },
{ first: "Albert", last: "Einstein", year: 1879, passed: 195 5}
```

reduce()

Applies a function against an accumulator and each element in the array (from left to right) to reduce it to a single value.

```
// How many years did all the inventors live?
let sumAge = inventors.reduce((acc, currVal) => acc + currVal.passed - currV
al.year, 0);

console.log('Sum:', sumAge);
>> Sum: 861
```

Array.from()

Creates a new Array instance from an array-like or iterable object.

```
<!-- Create a list of Boulevards in Paris that contain 'of' anywhere in the
name -->
<div class="links">
  <a href="/wiki/#1">Boulevards of Paris</a>
  <a href="/wiki/#2">Boulevard Saint-Michel</a>
  <a href="/wiki/#3">City Gates of Paris</a>
  <a href="/wiki/#4">Boulevard des Capucines</a>
</div>
```

```
let boulevards = Array.from(document.querySelectorAll('.links a'));
boulevardsWithOf = boulevards.filter(boulevard => boulevard.textContent.incl
udes('of'));

console.log(boulevardsWithOf);
>> [Boulevards of Paris, City Gates of Paris]
```

some()

Tests whether at least one element in the array passes the test implemented by the provided function.

```
const people = [
  { name: 'Wes', year: 1988 },
  { name: 'Kait', year: 1986 },
  { name: 'Irv', year: 1970 },
  { name: 'Lux', year: 2015 }
];

// Array.prototype.some() // is at least one person 19 or older?
let anyOlderThan19 = people.some(person => {
  let age = new Date().getFullYear() - person.year;
  return age > 18;
});

console.log(anyOlderThan19);
>> true
```

every()

Tests whether all elements in the array pass the test implemented by the provided function.

```
// Array.prototype.every() // is everyone 19 or older?
let allOlderThan19 = people.every(person => {
  let age = new Date().getFullYear() - person.year;
  return age > 18;
});

console.log(allOlderThan19);
>> false
```

find()

Returns the value of the first element in the array that satisfies the provided testing function. Otherwise undefined is returned.

```
const comments = [
  { text: 'Love this!', id: 523423 },
  { text: 'Super good', id: 823423 },
  { text: 'You are the best', id: 2039842 },
  { text: 'Ramen is my fav food ever', id: 123523 },
  { text: 'Nice Nice Nice!', id: 542328 }
];

// find the comment with the ID of 823423
let comment = comments.find(comment => comment.id === 823423);
console.log(comment);
>> { id: 823423, text: "Super good" }
```

findIndex()

Returns the index of the first element in the array that satisfies the provided testing function. Otherwise -1 is returned.

```
// Find the comment with this ID
// delete the comment with the ID of 823423

let commentIndex = comments.findIndex(comment => comment.id === 823423);
comments.splice(commentIndex, 1);

console.log(comments);
[
  { text: "Love this!", id: 523423 },
  { text: "You are the best", id: 2039842 },
  { text: "Ramen is my fav food ever", id: 123523 },
  { text: "Nice Nice Nice!", id: 542328}
]
```

Tags: javascript, arrays