Array Cardio Practice Day 1 Explanation

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Array Cardio □</title>
  <link rel="icon" href="https://fav.farm/✅" />
</head>
<body>
  <p><em>Psst: have a look at the JavaScript Console</em> □</p>
  <script>
    // Get your shorts on - this is an array workout!
    // ## Array Cardio Day 1

    // Some data we can work with

    const inventors = [
      { first: 'Albert', last: 'Einstein', year: 1879, passed: 1955 },
      { first: 'Isaac', last: 'Newton', year: 1643, passed: 1727 },
      { first: 'Galileo', last: 'Galilei', year: 1564, passed: 1642 },
      { first: 'Marie', last: 'Curie', year: 1867, passed: 1934 },
      { first: 'Johannes', last: 'Kepler', year: 1571, passed: 1630 },
      { first: 'Nicolaus', last: 'Copernicus', year: 1473, passed: 1543
},
      { first: 'Max', last: 'Planck', year: 1858, passed: 1947 },
      { first: 'Katherine', last: 'Blodgett', year: 1898, passed:
1979 },
      { first: 'Ada', last: 'Lovelace', year: 1815, passed: 1852 },
      { first: 'Sarah E.', last: 'Goode', year: 1855, passed: 1905 },
      { first: 'Lise', last: 'Meitner', year: 1878, passed: 1968 },
      { first: 'Hanna', last: 'Hammarström', year: 1829, passed: 1909 }
    ];

    const people = [
      'Bernhard, Sandra', 'Bethea, Erin', 'Becker, Carl', 'Bentsen,
Lloyd', 'Beckett, Samuel', 'Blake, William', 'Berger, Ric', 'Beddoes,
Mick', 'Beethoven, Ludwig',
      'Belloc, Hilaire', 'Begin, Menachem', 'Bellow, Saul', 'Benchley,
Robert', 'Blair, Robert', 'Benenson, Peter', 'Benjamin, Walter',
'Berlin, Irving',
      'Benn, Tony', 'Benson, Leana', 'Bent, Silas', 'Berle, Milton',
'Berry, Halle', 'Biko, Steve', 'Beck, Glenn', 'Bergman, Ingmar',
'Black, Elk', 'Berio, Luciano',
```

```
      'Berne, Eric', 'Berra, Yogi', 'Berry, Wendell', 'Bevan, Aneurin',
'Ben-Gurion, David', 'Bevel, Ken', 'Biden, Joseph', 'Bennington,
Chester', 'Bierce, Ambrose',
      'Billings, Josh', 'Birrell, Augustine', 'Blair, Tony', 'Beecher,
Henry', 'Biondo, Frank'
    ];

    // Array.prototype.filter()
    // 1. Filter the list of inventors for those who were born in the
1500's
    const fifteen = inventors.filter(inventor => (inventor.year >= 1500
&& inventor.year < 1600));

    console.table(fifteen);

    // Array.prototype.map()
    // 2. Give us an array of the inventor first and last names
    const fullNames = inventors.map(inventor => `${inventor.first} $
{inventor.last}`);
    console.log(fullNames);

    // Array.prototype.sort()
    // 3. Sort the inventors by birthdate, oldest to youngest
    // const ordered = inventors.sort(function(a, b) {
    //   if(a.year > b.year) {
    //     return 1;
    //   } else {
    //     return -1;
    //   }
    // });

    const ordered = inventors.sort((a, b) => a.year > b.year ? 1 : -1);
    console.table(ordered);

    // Array.prototype.reduce()
    // 4. How many years did all the inventors live?
    const totalYears = inventors.reduce((total, inventor) => {
      return total + (inventor.passed - inventor.year);
    }, 0);

    console.log(totalYears);

    // 5. Sort the inventors by years lived
    const oldest = inventors.sort(function(a, b) {
      const lastInventor = a.passed - a.year;
      const nextInventor = b.passed - b.year;
      return lastInventor > nextInventor ? -1 : 1;
    });
    console.table(oldest);
```

```
    // 6. create a list of Boulevards in Paris that contain 'de'
anywhere in the name
    // https://en.wikipedia.org/wiki/Category:Boulevards_in_Paris

    // const category = document.querySelector('.mw-category');
    // const links = Array.from(category.querySelectorAll('a'));
    // const de = links
    //             .map(link => link.textContent)
    //             .filter(streetName => streetName.includes('de'));

    // 7. sort Exercise
    // Sort the people alphabetically by last name
    const alpha = people.sort((lastOne, nextOne) => {
      const [aLast, aFirst] = lastOne.split(', ');
      const [bLast, bFirst] = nextOne.split(', ');
      return aLast > bLast ? 1 : -1;
    });
    console.log(alpha);

    // 8. Reduce Exercise
    // Sum up the instances of each of these
    const data = ['car', 'car', 'truck', 'truck', 'bike', 'walk',
'car', 'van', 'bike', 'walk', 'car', 'van', 'car', 'truck',
'pogostick'];

    const transportation = data.reduce(function(obj, item) {
      if (!obj[item]) {
        obj[item] = 0;
      }
      obj[item]++;
      return obj;
    }, {});

    console.log(transportation);

  </script>
</body>
</html>
```

Below is a brief explanation of how the exercises that I have made for this challenge:

These exercises serve as practice for working with array methods, such as filter(), map(), sort(), and reduce(), and cover various tasks like filtering, transforming, sorting, and counting data.

1. **Filtering Inventors Born in the 1500s:**
   The filter() method is used to create a new array (fifteen) that contains only the inventors born in the 1500s. It checks the year property of each inventor and filters out those whose birth year is not within the specified range.

2. **Extracting Full Names of Inventors:**
   The map() method is used to create a new array (fullName) that contains the full names of the inventors. It combines the first and last properties of each inventor using string interpolation to form a full name.

3. **Sorting Inventors by Birthdate:**
   The sort() method is used to sort the inventors array in ascending order based on the birth year (year property). The arrow function inside sort() compares the birth years of two inventors and returns 1 or -1 to determine the sorting order.

4. **Calculating Total Years Lived by Inventors:**
   The reduce() method is used to calculate the total years lived by all inventors. It iterates over the inventors array and accumulates the difference between the passed and year properties of each inventor. The initial value of the accumulator (total) is 0.

5. **Sorting Inventors by Years Lived:**
   Another sort() method is used to sort the inventors array in descending order based on the number of years lived (passed - year). The callback function compares the years lived by two inventors and returns -1 or 1 to determine the sorting order.

6. **Extracting Boulevard Names in Paris with 'de':**
   This exercise is commented out in the code. It demonstrates how to extract the names of boulevards in Paris that contain the substring 'de'. It utilizes the map() and filter() methods on a list of links to boulevards on a Wikipedia page.

7. **Sorting People by Last Name:**
   The sort() method is used to sort the people array alphabetically by last name. The callback function splits each element of the array into last and first names using the split() method. It then compares the last names and returns 1 or -1 to determine the sorting order.

8. **Counting Instances of Transportation Types:**
   The reduce() method is used to count the number of occurrences of each transportation type in the data array. It initializes an empty object as the accumulator (obj) and increments the count for each transportation type encountered in the array.

What I have learned

From this challenge, I have learned to utilize essential array methods such as filter(), map(), sort(), and reduce() to filter, transform, sort, and accumulate data within arrays.