Exercise on String Slicing

Practice your slice function with this exercise.

Sort the rows in the results of Exercise 3 based on their titles. Make sure you exclude the header row.

Hint: the `slice` method works in the same way on arrays as on strings.

Solution

First we get rid of the first row:

```
const result = data.trim().split('\n').map( row => row.split(',') );

const dataRows = result.slice( 1 );
console.log(dataRows)
```

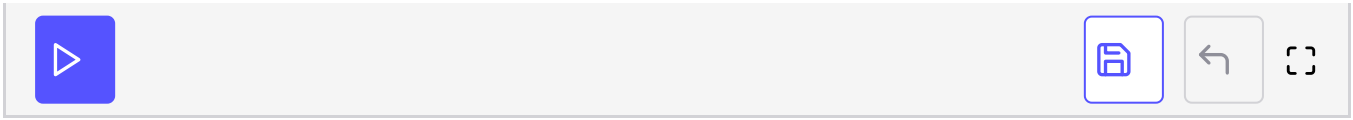
We have to sort the data rows using the array sort method. This method expects a helper function that expects two arguments, `a` and `b`. This function should return a positive value whenever `a` > `b`, a negative value whenever `a` < `b`, and zero if `a` and `b` are equal.

The helper function uses the `localCompare` string method, as it produces a return value in a similar fashion:

```
const result = data.trim().split('\n').map( row => row.split(',') );
const dataRows = result.slice( 1 );
const compareTitles = function( row1, row2 ) {
    return row1[0].localeCompare( row2[0] );
}
dataRows.sort( compareTitles );
console.log(dataRows);
```

Finally, we might want to add the header row on top of the `compareTitles` array. There are at least five different ways to do this, but this time, we will stick to a solution that you already know how to implement:

```
const result = data.trim().split('\n').map( row => row.split(',') );
const dataRows = result.slice( 1 );
const compareTitles = function( row1, row2 ) {
    return row1[0].localeCompare( row2[0] );
}
dataRows.sort( compareTitles );
const dataRowsWithHeader = [ result[0] ];
for ( let row of dataRows ) dataRowsWithHeader.push( row );
console.log(dataRowsWithHeader);
```



Later, once you learn array operations and the spread operator, you can use more concise solutions such as

```
const dataRowsWithHeader = [ result[0], ...dataRows ];
```

or

```
dataRows.unshift( result[0] );
```

The former spreads (enumerates) all elements in `dataRows` and places these elements after `result[0]`. The latter places `result[0]` in front of the elements of `dataRows`. Bear in mind that the last solution mutates (changes) the original `dataRows` array.