Sorting Strings

This lesson will introduce you to the localCompare method which helps us to sort strings.

Often in software development, we have to sort strings. There are a few problems with string sorting:

- Uppercase and lowercase letters are sorted differently: `'a' > 'B'`,
- accented characters are completely out of sequence: `'á' > 'b'`.

The localCompare method

The localCompare string method compares two string in the current locale which is based on the language settings of your browser. It helps solves both the above problems:

```
console.log('á'.localeCompare( 'b' ));
console.log('á'.localeCompare( 'a' ));
console.log('a'.localeCompare( 'A' ));
console.log('b'.localeCompare( 'A' ));
console.log('A'.localeCompare( 'b' ));
console.log('B'.localeCompare( 'b' ));
```

Sorting an array of strings in place works as follows:

```
const words = [ 'Practice', 'ES6', 'in', 'á' ];
const sorter = function( a, b ) {
    return a.localeCompare( b );
}

words.sort( sorter );
console.log(words);
```

This sort method of arrays expects a helper function such as `sorter`. This

This sort method of arrays expects a helper function such as `sorter`. This helper function expects two arguments, `a` and `b`. The helper function should

be written in such a way that it should return a positive value whenever `a > b`, a negative value whenever `a < b`, and zero if `a` and `b` are equal.

The `sort` JavaScript array method sorts its contents in place. This means the order of the elements change inside the array.