arrays.js (Page 1 of 1)

```
 1: //-----------------------------------------------------------------------
 2: // arrays.js
 3: // Author: Bob Dondero
 4: //-----------------------------------------------------------------------
 5:
 6: 'use strict';
 7:
 8: function main() {
 9:    // Creating an array
10:    let a = ['Ruth', 'Gehrig', 'Jeter'];
11:    console.log(a);  // [ 'Ruth', 'Gehrig', 'Jeter' ]
12:    console.log(a.length);  // 3
13:    console.log('-----------------------------------------------');
14:
15:    // Accessing an element
16:    let player = a[1];  // 'Gehrig'
17:    console.log(a);  // [ 'Ruth', 'Gehrig', 'Jeter' ]
18:    console.log(a.length);  // 3
19:    console.log('-----------------------------------------------');
20:
21:    // Changing an element
22:    a[1] = 'Mantle';
23:    console.log(a);  // [ 'Ruth', 'Mantle', 'Jeter' ]
24:    console.log(a.length);  // 3
25:    console.log('-----------------------------------------------');
26:
27:    // Adding an element to the end
28:    a.push('Berra');
29:    console.log(a);  // [ 'Ruth', 'Mantle', 'Jeter', 'Berra' ]
30:    console.log(a.length);  // 4
31:    console.log('-----------------------------------------------');
32:
33:    // Removing an element from the end
34:    let element = a.pop();  // 'Berra'
35:    console.log(a);  // [ 'Ruth', 'Mantle', 'Jeter' ]
36:    console.log(a.length);  // 3
37:    console.log('-----------------------------------------------');
38:
39:    // Iterating over an array (version 1)
40:    for (let i = 0; i < a.length; i++)
41:       console.log(a[i]);
42:       // Ruth
43:       // Mantle
44:       // Jeter
45:    console.log('-----------------------------------------------');
46:
47:    // Iterating over an array (version 2)
48:    for (let element of a)
49:       console.log(element);
50:       // Ruth
51:       // Mantle
52:       // Jeter
53:    console.log('-----------------------------------------------');
54: }
55:
56: if (require.main === module)
57:    main();
```

linesort.js (Page 1 of 1)

```
 1: //-----------------------------------------------------------------------
 2: // linesort.js
 3: // Author: Bob Dondero
 4: //-----------------------------------------------------------------------
 5:
 6: 'use strict';
 7: const fs = require('fs');
 8:
 9: //-----------------------------------------------------------------------
10:
11: function splitIntoLines(str) {
12:    let lines = [];
13:    let line = '';
14:    for (let c of str) {
15:       if (c === '\n') { lines.push(line); line = ''; }
16:       else line += c;
17:    }
18:    if (line !== '') lines.push(line);
19:    return lines;
20: }
21:
22: //-----------------------------------------------------------------------
23:
24: function main() {
25:
26:    if (process.argv.length !== 3) {
27:       process.stderr.write('Usage: ' +
28:          process.argv[0] + ' ' + process.argv[1] + ' infile\n');
29:       process.exit(1);
30:    }
31:
32:    let fileName = process.argv[2];
33:
34:    try {
35:        let data = fs.readFileSync(fileName, 'UTF-8');
36:        let lines = splitIntoLines(data);
37:        lines.sort();
38:        for (let line of lines)
39:           process.stdout.write(line + '\n');
40:    }
41:    catch (err) {
42:        process.stderr.write(err + '\n');
43:    }
44: }
45:
46: if (require.main === module)
47:    main();
```

assocarrays1.js (Page 1 of 1)

```
1: //-------------------------------------------------------------------------
2: // assocarrays1.js
3: // Author: Bob Dondero
4: //-------------------------------------------------------------------------
5:
6: 'use strict';
7:
8: function main() {
9:    // Creating an associative array of 3 key-value bindings
10:   let aa = {'Ruth': 'RF', 'Gehrig': '1B', 'Jeter': 'SS'};
11:   console.log(aa);
12:      // { Ruth: 'RF', Gehrig: '1B', Jeter: 'SS' }
13:   console.log('-----------------------------------------------');
14:
15:   // Accessing a value for a given key
16:   let position = aa['Gehrig'];     // '1B'
17:   console.log(aa);
18:      // { Ruth: 'RF', Gehrig: '1B', Jeter: 'SS' }
19:   console.log('-----------------------------------------------');
20:
21:   // Changing a value for a given key
22:   aa['Ruth'] = 'P';
23:   console.log(aa);
24:      // { Ruth: 'P', Gehrig: '1B', Jeter: 'SS' }
25:   console.log('-----------------------------------------------');
26:
27:   // Adding a binding
28:   aa['Maris'] = 'RF';
29:   console.log(aa);
30:      // { Ruth: 'P', Gehrig: '1B', Jeter: 'SS', Maris: 'RF' }
31:   console.log('-----------------------------------------------');
32:
33:   // Deleting a binding
34:   delete aa['Maris'];
35:   console.log(aa);
36:      // { Ruth: 'P', Gehrig: '1B', Jeter: 'SS' }
37:   console.log('-----------------------------------------------');
38:
39:   // Iterating over bindings
40:   for (let key in aa)
41:      console.log(key + ': ' + aa[key]);
42:      // Ruth: P
43:      // Gehrig: 1B
44:      // Jeter: SS
45: }
46:
47: if (require.main === module)
48:    main();
```

assocarrays2.js (Page 1 of 1)

```
1: //-------------------------------------------------------------------------
2: // assocarrays2.js
3: // Author: Bob Dondero
4: //-------------------------------------------------------------------------
5:
6: 'use strict';
7:
8: function main() {
9:    // Creating an associative array of 3 key-value bindings
10:   let aa = {Ruth: 'RF', Gehrig: '1B', Jeter: 'SS'};
11:   console.log(aa);
12:      // { Ruth: 'RF', Gehrig: '1B', Jeter: 'SS' }
13:   console.log('-----------------------------------------------');
14:
15:   // Accessing a value for a given key
16:   let position = aa.Gehrig;     // '1B'
17:   console.log(aa);
18:      // { Ruth: 'RF', Gehrig: '1B', Jeter: 'SS' }
19:   console.log('-----------------------------------------------');
20:
21:   // Changing a value for a given key
22:   aa.Ruth = 'P';
23:   console.log(aa);
24:      // { Ruth: 'P', Gehrig: '1B', Jeter: 'SS' }
25:   console.log('-----------------------------------------------');
26:
27:   // Adding a binding
28:   aa.Maris = 'RF';
29:   console.log(aa);
30:      // { Ruth: 'P', Gehrig: '1B', Jeter: 'SS', Maris: 'RF' }
31:   console.log('-----------------------------------------------');
32:
33:   // Deleting a binding
34:   delete aa.Maris;
35:   console.log(aa);
36:      // { Ruth: 'P', Gehrig: '1B', Jeter: 'SS' }
37:   console.log('-----------------------------------------------');
38:
39:   // Iterating over bindings
40:   for (let key in aa)
41:      console.log(key + ': ' + aa[key]);
42:      // Ruth: P
43:      // Gehrig: 1B
44:      // Jeter: SS
45: }
46:
47: if (require.main === module)
48:    main();
```

concord.js (Page 1 of 1)

```
 1: //---------------------------------------------------------------------
 2: // concord.js
 3: // Author: Bob Dondero
 4: //---------------------------------------------------------------------
 5:
 6: 'use strict';
 7:
 8: const fs = require('fs');
 9:
10: //---------------------------------------------------------------------
11:
12: function createConcordance(data) {
13:    let lowercaseData = data.toLowerCase();
14:    let words = lowercaseData.match(/[a-z]+/g);
15:    if (words === null)
16:       words = [];
17:
18:    let concordance = {};
19:    for (let word of words)
20:       if (word in concordance)
21:          concordance[word]++;
22:       else
23:          concordance[word] = 1;
24:    return concordance;
25: }
26:
27: //---------------------------------------------------------------------
28:
29: function writeConcordance(concordance) {
30:    for (let word in concordance)
31:       process.stdout.write(word + ': ' + concordance[word] + '\n');
32: }
33:
34: //---------------------------------------------------------------------
35:
36: function main() {
37:    if (process.argv.length !== 3) {
38:       process.stderr.write('Usage: ' + process.argv[0] + ' ' +
39:          process.argv[1] + ' infile\n');
40:       process.exit(1);
41:    }
42:
43:    let fileName = process.argv[2];
44:
45:    try {
46:       let data = fs.readFileSync(fileName, 'UTF-8');
47:       let concordance = createConcordance(data);
48:       writeConcordance(concordance);
49:    }
50:    catch (err) {
51:        process.stderr.write(err + '\n');
52:    }
53: }
54:
55: if (require.main === module)
56:    main();
```

linesortcallback.js (Page 1 of 1)

```
 1: //---------------------------------------------------------------------
 2: // linesortcallback.js
 3: // Author: Bob Dondero
 4: //---------------------------------------------------------------------
 5:
 6: 'use strict';
 7: const fs = require('fs');
 8:
 9: //---------------------------------------------------------------------
10:
11: function splitIntoLines(str) {
12:    let lines = [];
13:    let line = '';
14:    for (let c of str) {
15:       if (c === '\n') { lines.push(line); line = ''; }
16:       else line += c;
17:    }
18:    if (line !== '') lines.push(line);
19:    return lines;
20: }
21:
22: //---------------------------------------------------------------------
23:
24: function reportError(err) {
25:    process.stderr.write(err.message + '\n');
26: }
27:
28: //---------------------------------------------------------------------
29:
30: function writeLines(lines) {
31:    for (let line of lines)
32:       process.stdout.write(line + '\n');
33: }
34:
35: //---------------------------------------------------------------------
36:
37: function sortWriteLines(err, data) {
38:    if (err)
39:       reportError(err);
40:    else {
41:       let lines = splitIntoLines(data);
42:       lines.sort();
43:       writeLines(lines);
44:    }
45: }
46:
47: //---------------------------------------------------------------------
48:
49: function main() {
50:    if (process.argv.length !== 3) {
51:       process.stderr.write('Usage: ' + process.argv[0] + ' ' +
52:          process.argv[1] + ' infile\n');
53:       process.exit(1);
54:    }
55:    let fileName = process.argv[2];
56:    fs.readFile(fileName, 'UTF-8', sortWriteLines);
57:    process.stderr.write('Doing other work\n');
58: }
59:
60: if (require.main === module)
61:    main();
```

linesortpromises.js (Page 1 of 1)

```
  1: //-------------------------------------------------------------------------
  2: // linesortpromises.js
  3: // Author: Bob Dondero
  4: //-------------------------------------------------------------------------
  5:
  6: 'use strict';
  7: const fs = require('fs');
  8:
  9: //-------------------------------------------------------------------------
 10: function splitIntoLines(str) {
 11:    let lines = [];
 12:    let line = '';
 13:    for (let c of str) {
 14:       if (c === '\n') { lines.push(line); line = ''; }
 15:       else line += c;
 16:    }
 17:    if (line !== '') lines.push(line);
 18:    return lines;
 19: }
 20:
 21: //-------------------------------------------------------------------------
 22: function reportError(err) {
 23:    process.stderr.write(err.message + '\n');
 24: }
 25:
 26: //-------------------------------------------------------------------------
 27: function writeLines(lines) {
 28:    for (let line of lines)
 29:       process.stdout.write(line + '\n');
 30: }
 31:
 32: //-------------------------------------------------------------------------
 33: function sortLines(data) {
 34:    let lines = splitIntoLines(data);
 35:    lines.sort();
 36:    return lines;
 37: }
 38:
 39: //-------------------------------------------------------------------------
 40: function main(argv) {
 41:    if (argv.length !== 3) {
 42:       process.stderr.write(
 43:          'Usage: ' + process.argv[0] + ' ' + process.argv[1] +
 44:          ' infile\n');
 45:       process.exit(1);
 46:    }
 47:    let fileName = process.argv[2];
 48:
 49:    // let promise1 = fs.promises.readFile(fileName, 'UTF-8');
 50:    // let promise2 = promise1.then(sortLines);
 51:    // let promise3 = promise2.then(writeLines);
 52:    // promise3.catch(reportError);
 53:
 54:    fs.promises.readFile(fileName, 'UTF-8')
 55:       .then(sortLines)
 56:       .then(writeLines)
 57:       .catch(reportError);
 58:
 59:    process.stderr.write('Doing other work\n');
 60: }
 61:
 62: if (require.main === module)
 63:    main(process.argv);
```

linesortawait.js (Page 1 of 1)

```
  1: //-------------------------------------------------------------------------
  2: // linesortawait.js
  3: // Author: Bob Dondero
  4: //-------------------------------------------------------------------------
  5:
  6: 'use strict';
  7: const fs = require('fs');
  8:
  9: //-------------------------------------------------------------------------
 10: function splitIntoLines(str) {
 11:    let lines = [];
 12:    let line = '';
 13:    for (let c of str) {
 14:       if (c === '\n') { lines.push(line); line = ''; }
 15:       else line += c;
 16:    }
 17:    if (line !== '') lines.push(line);
 18:    return lines;
 19: }
 20:
 21: //-------------------------------------------------------------------------
 22: function reportError(err) {
 23:    process.stderr.write(err.message + '\n');
 24: }
 25:
 26: //-------------------------------------------------------------------------
 27: function writeLines(lines) {
 28:    for (let line of lines)
 29:       process.stdout.write(line + '\n');
 30: }
 31:
 32: //-------------------------------------------------------------------------
 33: function sortLines(data) {
 34:    let lines = splitIntoLines(data);
 35:    lines.sort();
 36:    return lines;
 37: }
 38:
 39: //-------------------------------------------------------------------------
 40: async function handleFile(fileName) {
 41:    try {
 42:       let data = await fs.promises.readFile(fileName, 'UTF-8');
 43:       let lines = sortLines(data);
 44:       writeLines(lines);
 45:    }
 46:    catch (err) {
 47:       reportError(err);
 48:    }
 49: }
 50:
 51: //-------------------------------------------------------------------------
 52: function main() {
 53:    if (process.argv.length !== 3) {
 54:       process.stderr.write('Usage: ' + process.argv[0] + ' ' +
 55:          process.argv[1] + ' infile\n');
 56:       process.exit(1);
 57:    }
 58:    let fileName = process.argv[2];
 59:    handleFile(fileName);
 60:    process.stderr.write('Doing other work\n');
 61: }
 62:
 63: if (require.main === module)
 64:    main();
```