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: const path = require('path');
 9: const pgmName = path.basename(process.argv[1]);
10:
11: //-----------------------------------------------------------------------
12:
13: function main() {
14:    if (process.argv.length !== 3) {
15:       process.stderr.write(`usage: node ${pgmName} infile\n`);
16:       process.exit(1);
17:    }
18:
19:    let fileName = process.argv[2];
20:
21:    try {
22:       let data = fs.readFileSync(fileName, 'UTF-8');
23:       let lines = data.trimEnd().split('\n');
24:       lines.sort();
25:       for (let line of lines)
26:          process.stdout.write(line + '\n');
27:    }
28:    catch (err) {
29:       process.stderr.write(`${pgmName}: ${err.message}\n`);
30:    }
31: }
32:
33: //-----------------------------------------------------------------------
34:
35: if (require.main === module)
36:    main();
```

assocarrays.js (Page 1 of 1)

```
1: //---------------------------------------------------------------------
2: // assocarrays.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:   // let aa = {'Ruth': 'RF', 'Gehrig': '1B', 'Jeter': 'SS'};
12:   console.log(aa);
13:      // { Ruth: 'RF', Gehrig: '1B', Jeter: 'SS' }
14:   console.log('-----------------------------------------------');
15:
16:   // Accessing a value for a given key
17:   let position = aa.Gehrig;     // '1B'
18:   // let position = aa['Gehrig'];     // '1B'
19:   console.log(aa);
20:      // { Ruth: 'RF', Gehrig: '1B', Jeter: 'SS' }
21:   console.log('-----------------------------------------------');
22:
23:   // Changing a value for a given key
24:   aa.Ruth = 'P';
25:   // aa['Ruth'] = 'P';
26:   console.log(aa);
27:      // { Ruth: 'P', Gehrig: '1B', Jeter: 'SS' }
28:   console.log('-----------------------------------------------');
29:
30:   // Adding a binding
31:   aa.Maris = 'RF';
32:   // aa['Maris'] = 'RF';
33:   console.log(aa);
34:      // { Ruth: 'P', Gehrig: '1B', Jeter: 'SS', Maris: 'RF' }
35:   console.log('-----------------------------------------------');
36:
37:   // Deleting a binding
38:   delete aa.Maris;
39:   // delete aa['Maris'];
40:   console.log(aa);
41:      // { Ruth: 'P', Gehrig: '1B', Jeter: 'SS' }
42:   console.log('-----------------------------------------------');
43:
44:   // Iterating over bindings
45:   for (let key in aa)
46:      console.log(key + ': ' + aa[key]);
47:      // Ruth: P
48:      // Gehrig: 1B
49:      // Jeter: SS
50: }
51:
52: if (require.main === module)
53:    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: const path = require('path');
10: const pgmName = path.basename(process.argv[1]);
11:
12: //---------------------------------------------------------------------
13:
14: function createConcordance(data) {
15:    let lowercaseData = data.toLowerCase();
16:    let words = lowercaseData.match(/[a-z]+/g);
17:    if (words === null)
18:       words = [];
19:
20:    let concordance = {};
21:    for (let word of words)
22:       if (word in concordance)
23:          concordance[word]++;
24:       else
25:          concordance[word] = 1;
26:    return concordance;
27: }
28:
29: //---------------------------------------------------------------------
30:
31: function writeConcordance(concordance) {
32:    for (let word in concordance)
33:       process.stdout.write(`${word}: ${concordance[word]}\n`);
34: }
35:
36: //---------------------------------------------------------------------
37:
38: function main() {
39:    if (process.argv.length !== 3) {
40:       process.stderr.write(`usage: node ${pgmName} infile\n`);
41:       process.exit(1);
42:    }
43:
44:    let fileName = process.argv[2];
45:
46:    try {
47:       let data = fs.readFileSync(fileName, 'UTF-8');
48:       let concordance = createConcordance(data);
49:       writeConcordance(concordance);
50:    }
51:    catch (err) {
52:        process.stderr.write(`${pgmName}: ${err.message}\n`);
53:    }
54: }
55:
56: if (require.main === module)
57:    main();
```

linesortcallback.js (Page 1 of 1)

```
 1: //-----------------------------------------------------------------------
 2: // linesortcallback1.js
 3: // Author: Bob Dondero
 4: //-----------------------------------------------------------------------
 5:
 6: 'use strict';
 7: const fs = require('fs');
 8: const path = require('path');
 9: const pgmName = path.basename(process.argv[1]);
10:
11: //-----------------------------------------------------------------------
12:
13: function sortWriteLines(err, data) {
14:    if (err)
15:       process.stderr.write(`${pgmName}: ${err.message}\n`);
16:    else {
17:       let lines = data.trimEnd().split('\n');
18:       lines.sort();
19:       for (let line of lines)
20:          process.stdout.write(line + '\n');
21:    }
22: }
23:
24: //-----------------------------------------------------------------------
25:
26: function main() {
27:    if (process.argv.length !== 3) {
28:       process.stderr.write(`usage: node ${pgmName} infile\n`);
29:       process.exit(1);
30:    }
31:
32:    let fileName = process.argv[2];
33:
34:    fs.readFile(fileName, 'UTF-8', sortWriteLines);
35:
36:    process.stdout.write('Doing more work\n');
37: }
38:
39: //-----------------------------------------------------------------------
40:
41: if (require.main === module)
42:    main();
```

blank (Page 1 of 1)

```
1: This page is intentionally blank.
```

linesorttofile.js (Page 1 of 1)

```
 1: //-----------------------------------------------------------------------
 2: // linesorttofile.js
 3: // Author: Bob Dondero
 4: //-----------------------------------------------------------------------
 5:
 6: 'use strict';
 7: const fs = require('fs');
 8: const path = require('path');
 9: const pgmName = path.basename(process.argv[1]);
10:
11: //-----------------------------------------------------------------------
12:
13: function main() {
14:
15:    if (process.argv.length !== 4) {
16:       process.stderr.write(`usage: node ${pgmName} infile outfile\n`);
17:       process.exit(1);
18:    }
19:
20:    try {
21:       let inData = fs.readFileSync(process.argv[2], 'UTF-8');
22:       let lines = inData.trimEnd().split('\n');
23:       lines.sort();
24:       let outData = lines.join('\n') + '\n';
25:       fs.writeFileSync(process.argv[3], outData, 'UTF-8');
26:       process.stdout.write('Finished\n');
27:    }
28:    catch (err) {
29:       process.stderr.write(`${pgmName}: ${err.message}\n`);
30:    }
31: }
32:
33: //-----------------------------------------------------------------------
34:
35: if (require.main === module)
36:    main();
```

linesorttofilecallback.js (Page 1 of 1)

```
 1: //-----------------------------------------------------------------------
 2: // linesorttofilecallback.js
 3: // Author: Bob Dondero
 4: //-----------------------------------------------------------------------
 5:
 6: 'use strict';
 7: const fs = require('fs');
 8: const path = require('path');
 9: const pgmName = path.basename(process.argv[1]);
10:
11: //-----------------------------------------------------------------------
12:
13: function finish(err) {
14:    if (err)
15:       process.stderr.write(`${pgmName}: ${err.message}\n`);
16:    else
17:       process.stdout.write('Finished\n');
18: }
19:
20: //-----------------------------------------------------------------------
21:
22: function sortWriteFinish(err, inData) {
23:    if (err)
24:       process.stderr.write(`${pgmName}: ${err.message}\n`);
25:    else {
26:       let lines = inData.trimEnd().split('\n');
27:       lines.sort();
28:       let outData = lines.join('\n') + '\n';
29:       fs.writeFile(process.argv[3], outData, 'UTF-8', finish);
30:    }
31: }
32:
33: //-----------------------------------------------------------------------
34:
35: function main() {
36:    if (process.argv.length !== 4) {
37:       process.stderr.write(`usage: node ${pgmName} infile outfile\n`);
38:       process.exit(1);
39:    }
40:
41:    fs.readFile(process.argv[2], 'UTF-8', sortWriteFinish);
42:
43:    process.stdout.write('Doing more work\n');
44: }
45:
46: //-----------------------------------------------------------------------
47:
48: if (require.main === module)
49:    main();
```

linesorttofilepromises.js (Page 1 of 1)

```
  1: //---------------------------------------------------------------------
  2: // linesorttofilepromises.js
  3: // Author: Bob Dondero
  4: //---------------------------------------------------------------------
  5:
  6: 'use strict';
  7: const fs = require('fs');
  8: const path = require('path');
  9: const pgmName = path.basename(process.argv[1]);
 10:
 11: //---------------------------------------------------------------------
 12:
 13: function reportError(err) {
 14:    process.stderr.write(`${pgmName}: ${err.message}\n`);
 15: }
 16:
 17: //---------------------------------------------------------------------
 18:
 19: function finish() {
 20:    process.stdout.write('Finished\n');
 21: }
 22:
 23: //---------------------------------------------------------------------
 24:
 25: function writeFileAdapt(outData) {
 26:    return fs.promises.writeFile(process.argv[3], outData, 'UTF-8');
 27: }
 28:
 29: //---------------------------------------------------------------------
 30:
 31: function sortLines(inData) {
 32:    let lines = inData.trimEnd().split('\n');
 33:    lines.sort();
 34:    let outData = lines.join('\n') + '\n';
 35:    return outData;
 36: }
 37:
 38: //---------------------------------------------------------------------
 39:
 40: function main() {
 41:    if (process.argv.length !== 4) {
 42:       process.stderr.write(`usage: node ${pgmName} infile outfile\n`);
 43:       process.exit(1);
 44:    }
 45:
 46:    fs.promises.readFile(process.argv[2], 'UTF-8')
 47:       .then(sortLines)
 48:       .then(writeFileAdapt)
 49:       .then(finish)
 50:       .catch(reportError);
 51:
 52:    process.stdout.write('Doing more work\n');
 53: }
 54:
 55: //---------------------------------------------------------------------
 56:
 57: if (require.main === module)
 58:    main();
```

linesorttofileawait.js (Page 1 of 1)

```
  1: //---------------------------------------------------------------------
  2: // linesorttofileawait.js
  3: // Author: Bob Dondero
  4: //---------------------------------------------------------------------
  5:
  6: 'use strict';
  7: const fs = require('fs');
  8: const path = require('path');
  9: const pgmName = path.basename(process.argv[1]);
 10:
 11: //---------------------------------------------------------------------
 12:
 13: async function handleFile() {
 14:    try {
 15:       let inData = await fs.promises.readFile(process.argv[2], 'UTF-8');
 16:       let lines = inData.trimEnd().split('\n');
 17:       lines.sort();
 18:       let outData = lines.join('\n') + '\n';
 19:       await fs.promises.writeFile(process.argv[3], outData, 'UTF-8');
 20:       process.stdout.write('Finished\n');
 21:    }
 22:    catch (err) {
 23:       process.stderr.write(`${pgmName}: ${err.message}\n`);
 24:    }
 25: }
 26:
 27: //---------------------------------------------------------------------
 28:
 29: function main() {
 30:    if (process.argv.length !== 4) {
 31:       process.stderr.write(`usage: node ${pgmName} infile outfile\n`);
 32:       process.exit(1);
 33:    }
 34:
 35:    handleFile();
 36:
 37:    process.stdout.write('Doing more work\n');
 38: }
 39:
 40: //---------------------------------------------------------------------
 41:
 42: if (require.main === module)
 43:    main();
```

linesorttofileawait2.js (Page 1 of 2)

```
 1: //---------------------------------------------------------------------
 2: // linesorttofileawait2.js
 3: // Author: Bob Dondero
 4: //---------------------------------------------------------------------
 5:
 6: 'use strict';
 7: const fs = require('fs');
 8: const path = require('path');
 9: const pgmName = path.basename(process.argv[1]);
10:
11: //---------------------------------------------------------------------
12:
13: function readFilePromise(fileName, encoding) {
14:    return new Promise(
15:       function(resolve, reject) {
16:          fs.readFile(fileName, encoding,
17:             function (err, data) {
18:                if (err)
19:                   reject(err);
20:                else
21:                   resolve(data);
22:             }
23:          );
24:       }
25:    );
26: }
27:
28: //---------------------------------------------------------------------
29:
30: function writeFilePromise(fileName, data, encoding) {
31:    return new Promise(
32:       function(resolve, reject) {
33:          fs.writeFile(fileName, data, encoding,
34:             function (err, data) {
35:                if (err)
36:                   reject(err);
37:                else
38:                   resolve(data);
39:             }
40:          );
41:       }
42:    );
43: }
44:
45: //---------------------------------------------------------------------
46:
47: async function handleFile() {
48:    try {
49:       let inData = await readFilePromise(process.argv[2], 'UTF-8');
50:       let lines = inData.trimEnd().split('\n');
51:       lines.sort();
52:       let outData = lines.join('\n') + '\n';
53:       await writeFilePromise(process.argv[3], outData, 'UTF-8');
54:       process.stdout.write('Finished\n');
55:    }
56:    catch (err) {
57:       process.stderr.write(`${pgmName}: ${err.message}\n`);
58:    }
59: }
60:
61: //---------------------------------------------------------------------
62:
63: function main() {
64:    if (process.argv.length !== 4) {
65:       process.stderr.write(`usage: node ${pgmName} infile outfile\n`);
```

linesorttofileawait2.js (Page 2 of 2)

```
66:       process.exit(1);
67:    }
68:
69:    handleFile();
70:
71:    process.stdout.write('Doing more work\n');
72: }
73:
74: //---------------------------------------------------------------------
75:
76: if (require.main === module)
77:    main();
```