## arrays.js (Page 1 of 1)

```
//-----------------------------------------------------------------------
// arrays.js
// Author: Bob Dondero
//-----------------------------------------------------------------------

'use strict';

function main() {
   // Creating an array
   let a = ['Ruth', 'Gehrig', 'Jeter'];
   console.log(a);  // [ 'Ruth', 'Gehrig', 'Jeter' ]
   console.log(a.length);  // 3
   console.log('-----------------------------------------------');

   // Accessing an element
   let player = a[1];  // 'Gehrig'
   console.log(a);  // [ 'Ruth', 'Gehrig', 'Jeter' ]
   console.log(a.length);  // 3
   console.log('-----------------------------------------------');

   // Changing an element
   a[1] = 'Mantle';
   console.log(a);  // [ 'Ruth', 'Mantle', 'Jeter' ]
   console.log(a.length);  // 3
   console.log('-----------------------------------------------');

   // Adding an element to the end
   a.push('Berra');
   console.log(a);  // [ 'Ruth', 'Mantle', 'Jeter', 'Berra' ]
   console.log(a.length);  // 4
   console.log('-----------------------------------------------');

   // Removing an element from the end
   let element = a.pop();  // 'Berra'
   console.log(a);  // [ 'Ruth', 'Mantle', 'Jeter' ]
   console.log(a.length);  // 3
   console.log('-----------------------------------------------');

   // Iterating over an array (version 1)
   for (let i = 0; i < a.length; i++)
      console.log(a[i]);
      // Ruth
      // Mantle
      // Jeter
   console.log('-----------------------------------------------');

   // Iterating over an array (version 2)
   for (let element of a)
      console.log(element);
      // Ruth
      // Mantle
      // Jeter
   console.log('-----------------------------------------------');
}

if (require.main === module)
   main();
```

## linesort.js (Page 1 of 1)

```
//-----------------------------------------------------------------------
// linesort.js
// Author: Bob Dondero
//-----------------------------------------------------------------------

'use strict';
const fs = require('fs');
const path = require('path');
const pgmName = path.basename(process.argv[1]);

//-----------------------------------------------------------------------

function splitIntoLines(str) {
   let lines = [];
   let line = '';
   for (let c of str) {
      if (c === '\n') { lines.push(line); line = ''; }
      else line += c;
   }
   if (line !== '') lines.push(line);
   return lines;
}

//-----------------------------------------------------------------------

function main() {

   if (process.argv.length !== 3) {
      process.stderr.write(`usage: node ${pgmName} infile\n`);
      process.exit(1);
   }

   let fileName = process.argv[2];

   try {
       let data = fs.readFileSync(fileName, 'UTF-8');
       let lines = splitIntoLines(data);
       lines.sort();
       for (let line of lines)
          process.stdout.write(line + '\n');
   }
   catch (err) {
       process.stderr.write(`${pgmName}: ${err.message}\n`);
   }
}

if (require.main === module)
   main();
```

## assocarrays1.js (Page 1 of 1)

```
//-----------------------------------------------------------------------
// assocarrays1.js
// Author: Bob Dondero
//-----------------------------------------------------------------------

'use strict';

function main() {
   // Creating an associative array of 3 key-value bindings
   let aa = {'Ruth': 'RF', 'Gehrig': '1B', 'Jeter': 'SS'};
   console.log(aa);
      // { Ruth: 'RF', Gehrig: '1B', Jeter: 'SS' }
   console.log('-----------------------------------------------');

   // Accessing a value for a given key
   let position = aa['Gehrig'];     // '1B'
   console.log(aa);
      // { Ruth: 'RF', Gehrig: '1B', Jeter: 'SS' }
   console.log('-----------------------------------------------');

   // Changing a value for a given key
   aa['Ruth'] = 'P';
   console.log(aa);
      // { Ruth: 'P', Gehrig: '1B', Jeter: 'SS' }
   console.log('-----------------------------------------------');

   // Adding a binding
   aa['Maris'] = 'RF';
   console.log(aa);
      // { Ruth: 'P', Gehrig: '1B', Jeter: 'SS', Maris: 'RF' }
   console.log('-----------------------------------------------');

   // Deleting a binding
   delete aa['Maris'];
   console.log(aa);
      // { Ruth: 'P', Gehrig: '1B', Jeter: 'SS' }
   console.log('-----------------------------------------------');

   // Iterating over bindings
   for (let key in aa)
      console.log(key + ': ' + aa[key]);
      // Ruth: P
      // Gehrig: 1B
      // Jeter: SS
}

if (require.main === module)
   main();
```

## assocarrays2.js (Page 1 of 1)

```
//-----------------------------------------------------------------------
// assocarrays2.js
// Author: Bob Dondero
//-----------------------------------------------------------------------

'use strict';

function main() {
   // Creating an associative array of 3 key-value bindings
   let aa = {Ruth: 'RF', Gehrig: '1B', Jeter: 'SS'};
   console.log(aa);
      // { Ruth: 'RF', Gehrig: '1B', Jeter: 'SS' }
   console.log('-----------------------------------------------');

   // Accessing a value for a given key
   let position = aa.Gehrig;     // '1B'
   console.log(aa);
      // { Ruth: 'RF', Gehrig: '1B', Jeter: 'SS' }
   console.log('-----------------------------------------------');

   // Changing a value for a given key
   aa.Ruth = 'P';
   console.log(aa);
      // { Ruth: 'P', Gehrig: '1B', Jeter: 'SS' }
   console.log('-----------------------------------------------');

   // Adding a binding
   aa.Maris = 'RF';
   console.log(aa);
      // { Ruth: 'P', Gehrig: '1B', Jeter: 'SS', Maris: 'RF' }
   console.log('-----------------------------------------------');

   // Deleting a binding
   delete aa.Maris;
   console.log(aa);
      // { Ruth: 'P', Gehrig: '1B', Jeter: 'SS' }
   console.log('-----------------------------------------------');

   // Iterating over bindings
   for (let key in aa)
      console.log(key + ': ' + aa[key]);
      // Ruth: P
      // Gehrig: 1B
      // Jeter: SS
}

if (require.main === module)
   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();
```

## blank (Page 1 of 1)

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

## linesortcallback1.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 splitIntoLines(str) {
 14:    let lines = [];
 15:    let line = '';
 16:    for (let c of str) {
 17:       if (c === '\n') { lines.push(line); line = ''; }
 18:       else line += c;
 19:    }
 20:    if (line !== '') lines.push(line);
 21:    return lines;
 22: }
 23:
 24: //-------------------------------------------------------------------------
 25:
 26: function sortWriteLines(err, data) {
 27:    if (err)
 28:       process.stderr.write(`${pgmName}: ${err.message}\n`);
 29:    else {
 30:       let lines = splitIntoLines(data);
 31:       lines.sort();
 32:       for (let line of lines)
 33:          process.stdout.write(line + '\n');
 34:    }
 35: }
 36:
 37: //-------------------------------------------------------------------------
 38:
 39: function main() {
 40:    if (process.argv.length !== 3) {
 41:       process.stderr.write(`usage: node ${pgmName} infile\n`);
 42:       process.exit(1);
 43:    }
 44:    let fileName = process.argv[2];
 45:    process.stdout.write('Doing some work\n');
 46:    fs.readFile(fileName, 'UTF-8', sortWriteLines);
 47:    process.stdout.write('Doing more work\n');
 48: }
 49:
 50: if (require.main === module)
 51:    main();
```

## linesortcallback2.js (Page 1 of 1)

```
  1: //-------------------------------------------------------------------------
  2: // linesortcallback2.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 splitIntoLines(str) {
 14:    let lines = [];
 15:    let line = '';
 16:    for (let c of str) {
 17:       if (c === '\n') { lines.push(line); line = ''; }
 18:       else line += c;
 19:    }
 20:    if (line !== '') lines.push(line);
 21:    return lines;
 22: }
 23:
 24: //-------------------------------------------------------------------------
 25:
 26: function main() {
 27:    if (process.argv.length !== 3) {
 28:       process.stderr.write(`usage: node ${pgmName} infile\n`);
 29:       process.exit(1);
 30:    }
 31:
 32:    let fileName = process.argv[2];
 33:    process.stdout.write('Doing some work\n');
 34:
 35:    fs.readFile(fileName, 'UTF-8',
 36:       function(err, data) {
 37:          if (err)
 38:             process.stderr.write(`${pgmName}: ${err.message}\n`);
 39:          else {
 40:             let lines = splitIntoLines(data);
 41:             lines.sort();
 42:             for (let line of lines)
 43:                process.stdout.write(line + '\n');
 44:          }
 45:       }
 46:    );
 47:
 48:    process.stdout.write('Doing more work\n');
 49: }
 50:
 51: if (require.main === module)
 52:    main();
```

## linesortpromises1.js (Page 1 of 1)

```
1: //-------------------------------------------------------------------------
2: // linesortpromises1.js
3: // Author: Bob Dondero
4: //-------------------------------------------------------------------------
5: 'use strict';
6: const fs = require('fs');
7: const path = require('path');
8: const pgmName = path.basename(process.argv[1]);
9:
10: //-------------------------------------------------------------------------
11: function readFileP(fileName, encoding) {
12:    return new Promise(
13:       function(resolve, reject) {
14:          fs.readFile(fileName, encoding,
15:             function (err, data) {
16:                if (err)
17:                   reject(err);
18:                else
19:                   resolve(data);
20:             }
21:          );
22:       }
23:    );
24: }
25:
26: //-------------------------------------------------------------------------
27: function splitIntoLines(str) {
28:    let lines = [];
29:    let line = '';
30:    for (let c of str) {
31:       if (c === '\n') { lines.push(line); line = ''; }
32:       else line += c;
33:    }
34:    if (line !== '') lines.push(line);
35:    return lines;
36: }
37:
38: //-------------------------------------------------------------------------
39: function reportError(err) {
40:    process.stderr.write(`${pgmName}: ${err.message}\n`);
41: }
42:
43: //-------------------------------------------------------------------------
44: function sortWriteLines(data) {
45:    let lines = splitIntoLines(data);
46:    lines.sort();
47:    for (let line of lines)
48:       process.stdout.write(line + '\n');
49: }
50:
51: //-------------------------------------------------------------------------
52: function main() {
53:    if (process.argv.length !== 3) {
54:       process.stderr.write(`usage: node ${pgmName} infile\n`);
55:       process.exit(1);
56:    }
57:    let fileName = process.argv[2];
58:    process.stdout.write('Doing some work\n');
59:    readFileP(fileName, 'UTF-8')
60:       .then(sortWriteLines)
61:       .catch(reportError);
62:    process.stdout.write('Doing more work\n');
63: }
64: if (require.main === module)
65:    main();
```

## linesortpromises2.js (Page 1 of 1)

```
1: //-------------------------------------------------------------------------
2: // linesortpromises2.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 splitIntoLines(str) {
14:    let lines = [];
15:    let line = '';
16:    for (let c of str) {
17:       if (c === '\n') { lines.push(line); line = ''; }
18:       else line += c;
19:    }
20:    if (line !== '') lines.push(line);
21:    return lines;
22: }
23:
24: //-------------------------------------------------------------------------
25:
26: function reportError(err) {
27:    process.stderr.write(`${pgmName}: ${err.message}\n`);
28: }
29:
30: //-------------------------------------------------------------------------
31:
32: function sortWriteLines(data) {
33:    let lines = splitIntoLines(data);
34:    lines.sort();
35:    for (let line of lines)
36:       process.stdout.write(line + '\n');
37: }
38:
39: //-------------------------------------------------------------------------
40:
41: function main() {
42:    if (process.argv.length !== 3) {
43:       process.stderr.write(`usage: node ${pgmName} infile\n`);
44:       process.exit(1);
45:    }
46:    let fileName = process.argv[2];
47:
48:    process.stdout.write('Doing some work\n');
49:
50:    fs.promises.readFile(fileName, 'UTF-8')
51:       .then(sortWriteLines)
52:       .catch(reportError);
53:
54:    process.stdout.write('Doing more work\n');
55: }
56:
57: if (require.main === module)
58:    main();
```

## linesortawait1.js (Page 1 of 1)

```
//---------------------------------------------------------------------------
// linesortawait1.js
// Author: Bob Dondero
//---------------------------------------------------------------------------

'use strict';
const fs = require('fs');
const path = require('path');
const pgmName = path.basename(process.argv[1]);

//---------------------------------------------------------------------------
function readFileP(fileName, encoding) {
   return new Promise(
      function(resolve, reject) {
         fs.readFile(fileName, encoding,
            function (err, data) {
               if (err)
                  reject(err);
               else
                  resolve(data);
            }
         );
      }
   );
}

//---------------------------------------------------------------------------
function splitIntoLines(str) {
   let lines = [];
   let line = '';
   for (let c of str) {
      if (c === '\n') { lines.push(line); line = ''; }
      else line += c;
   }
   if (line !== '') lines.push(line);
   return lines;
}

//---------------------------------------------------------------------------
async function handleFile(fileName) {
   process.stdout.write('Doing some work\n');
   try {
      let data = await readFileP(fileName, 'UTF-8');
      let lines = splitIntoLines(data);
      lines.sort();
      for (let line of lines)
         process.stdout.write(line + '\n');   }
   catch (err) {
      process.stderr.write(`${pgmName}: ${err.message}\n`);
   }
}

//---------------------------------------------------------------------------
function main() {
   if (process.argv.length !== 3) {
      process.stderr.write(`usage: node ${pgmName} infile\n`);
      process.exit(1);
   }
   let fileName = process.argv[2];
   handleFile(fileName);
   process.stdout.write('Doing more work\n');
}

if (require.main === module)
   main();
```

## linesortawait2.js (Page 1 of 1)

```
//---------------------------------------------------------------------------
// linesortawait.js
// Author: Bob Dondero
//---------------------------------------------------------------------------

'use strict';
const fs = require('fs');
const path = require('path');
const pgmName = path.basename(process.argv[1]);

//---------------------------------------------------------------------------

function splitIntoLines(str) {
   let lines = [];
   let line = '';
   for (let c of str) {
      if (c === '\n') { lines.push(line); line = ''; }
      else line += c;
   }
   if (line !== '') lines.push(line);
   return lines;
}

//---------------------------------------------------------------------------

async function handleFile(fileName) {
   process.stdout.write('Doing some work\n');
   try {
      let data = await fs.promises.readFile(fileName, 'UTF-8');
      let lines = splitIntoLines(data);
      lines.sort();
      for (let line of lines)
         process.stdout.write(line + '\n');   }
   catch (err) {
      process.stderr.write(`${pgmName}: ${err.message}\n`);
   }
}

//---------------------------------------------------------------------------

function main() {
   if (process.argv.length !== 3) {
      process.stderr.write(`usage: node ${pgmName} infile\n`);
      process.exit(1);
   }
   let fileName = process.argv[2];
   handleFile(fileName);
   process.stdout.write('Doing more work\n');
}

if (require.main === module)
   main();
```