

Assignment 4

Due Date: March 27, 2018 at 7:00 PM

Overview

This assignment will be worth **6%** of your total grade. The goal in part 1 is to demonstrate your knowledge of arrays and the corresponding functions and properties as well as synchronous File I/O to parse text files. Part 2 is designed to utilize the asynchronous properties of JavaScript. Look at `readFile` and `readdir` functions.

Specifications

For this assignment, you will need to understand arrays and File I/O in JavaScript.

Part 1a: Array Manipulation

Download the `array.js` starter file from Blackboard and you will see four arrays named “w”, “x”, “y”, and “z”. The objective is to create a single array called “arr” which contains: [‘Baltimore’, ‘Ravens’, ‘stadium’, ‘is’, ‘located’, ‘Downtown’, ‘near’, ‘Morgan State’].

Stipulations:

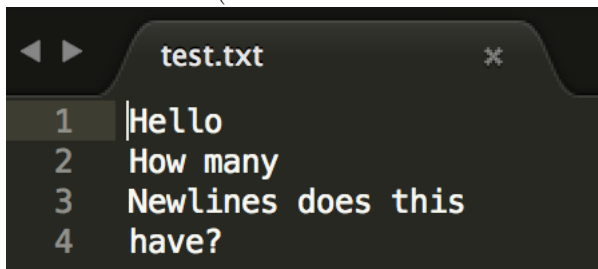
- Can only use *push()*, *pop()*, and *concat()*
- In some way must use all four arrays to form “arr”
- You cannot create any temp variables, only “arr” - the reuse of variables is fine
- You can only use *push()* ONCE
- You cannot *push()* array subscripts or String literals. i.e. both *arr.push(y[1])* and *arr.push(“Downtown”)* are illegal. HINT: Think about what the three functions listed above return

```
Jals-MacBook-Pro:Projects jalirani$ node array.js
[ 'Baltimore',
  'Ravens',
  'stadium',
  'is',
  'located',
  'Downtown',
  'near',
  'Morgan State' ]
```

Part 1b: Count Newlines

Write a program called `newlines.js` that uses a single synchronous filesystem operation to read a file and print the number of newlines it contains to the console (stdout). The file to be read in will come in as the 3rd command line argument. An example of a command would be: `node newlines.js something.txt`

From the `test.txt` (note there is no newline after line 4):



A screenshot of a text editor window titled 'test.txt'. The editor shows four lines of text: 'Hello', 'How many', 'Newlines does this', and 'have?'. The lines are numbered 1 through 4 on the left margin. There is no newline character at the end of line 4.

Example output would be:

```
Jals-MacBook-Pro:Projects jalirani$ node newlines.js test.txt
There are 3 new lines in the test.txt file
```

Part 2a: Looping

Write a program called `loops.js` that accepts one or more numbers as command-line arguments and prints the sum of those numbers to the console (stdout). i.e. the command `node loops.js 1 2 3` would output `6`

Part 2b: Asynchronous Newline Count

Write a program called `asyncNewlines.js` that uses a single asynchronous filesystem operation to read a file and print the number of newlines it contains to the

console (stdout).

You will have a file name as the third argument just like in part 1 of the assignment.

An example run would be: *node asyncNewlines.js test.txt*

Note: This will be run the exact same way as part 1 and have the exact same output from part 1. The only difference is now it will be done asynchronously.

Part 2c: Asynchronous Extension List

Write a program called `asyncListExt.js` that prints a list of files in a given directory, filtered by the extension of the files. The first argument will be the path to the directory we want to filter on (e.g. `‘/path/to/dir/’`) and a file extension to filter by as the second argument.

For example, if you get `‘.txt’` as the second argument then you will need to filter the list to only files that end with `.txt`

Below are some example runs:

An example run would be: *node asyncListExt.js . .js*

Another example run would be: *node asyncListExt.js /Users/jalirani/Desktop .docx*

Submission

Please submit the following on Blackboard:

- `array.js`
- `newlines.js`
- `loops.js`
- `asyncNewlines.js`
- `asyncListExt.js`