Overview

- Crude JS equivalent of Unix command-line utility.
- Illustrates multiple JS features including modules and asynchronicity which were not covered in the *Quick Start*.
- To keep simple, code duplication, no real error architecture.
- Command-line server-side program: fully synchronous.
- Browser based program: illustrates asynchronous code.
- Function `jgrep(regex, content)` is core of program and used by both server-side and browser programs.
- `jgrep()` function is pure.

Server Side Program Log

Truncation of long lines indicated by ....

```
$ ./main.mjs
main.mjs REGEX PATH

$ ./main.mjs '[a' main.mjs
bad regex "[a"

$ ./main.mjs 'for' Main.mjs
cannot read Main.mjs: ENOENT: no such file ...

$ ./main.mjs 'for' main.mjs
main.mjs:47:    for (const result of results) {
```

Server Side Program Log Continued

Truncation of long lines indicated by ....

```
$ ./main.mjs '\(\`' main.mjs
main.mjs:27:    if (!regex) error(`bad regex ...
main.mjs:41:      error(`cannot read ${path}:...
main.mjs:49:      console.log(`${path}:${line...
main.mjs:60:    error(`${Path.basename(proces...

$ ./main.mjs '\$\{.+\}' main.mjs
main.mjs:27:    if (!regex) error(`bad regex ...
main.mjs:41:      error(`cannot read ${path}:...
main.mjs:49:      console.log(`${path}:${line...
main.mjs:60:    error(`${Path.basename(proces...
$
```

Server Side Program Implementation

- main.mjs main program.
- jgrep.mjs jgrep core used by both server-side and browser programs.
- common.mjs trivial utilities used by both server-side and browser programs.

Server-Side Main Program Startup

- `#!` shebang line is a Unix feature which allows script execution.
- The portion of the line after the `#!` is the absolute path for an interpreter which the kernel sets up to run with the script file passed in as input to the interpreter.
- Use `/usr/bin/env` as interpreter, so that it can find nodejs which may be in a non-standard location.
- When nodejs starts up, it reads file as program to be interpreted.
- Normally a # character would cause an error in a nodejs file. nodejs ignores it only when it is on the first line of a file.

+‘process.argv‘ contains command-line args: `argv[0]: nodejs path`, argv[1]: script path‘. "Real" script args start at argv[2].

Client Side Implementation

- Illustrates the use of `async` and `await` for fetching remote content.
- Shows the use of anonymous functions for event handlers.
- *Running app*.