Can develop using Nodemon – tool which automatically restarts node based on source file changes

Recommend targeting ES5 – but node can support ES6 via harmony mode

Good book – “Human JavaScript” (<http://humanjavascript.com/> )

setInterval – perform task after specific period; use unref to stop timer preventing program exit

Useful for debugging sometimes - util.print can handle longer printing text than stdout; be aware that it is a blocking call though

Use object.keys and array functions rather than hasOwnProperty when checking to see if something belongs to directly to the object

Can freeze objects to only allow read only access… for example you might use this for configs. Good practice to export a frozen object as a module to stop accidental changes

Global objects in node: <http://nodejs.org/api/globals.html>

Don’t add anything to global object – equivalent to window object in browser

Process object gives access to users etc, plus environment variables.

* Environment variables can also be set in node, e.g. “process.env.name = boris”
* Can read/change working directory chdir
* Access to arguments (process.argv – arguments passed to script begin at 2)

Commander and optimist are two good libraries for parsing command line arguments

console.log is a blocking process… better to do logging via a stream instead

strongloop blog is useful for performance help (<http://strongloop.com/strongblog/> )

node weekly newsletter (<http://nodeweekly.com/> )

heroku can use npm via npm start // need to understand this more… think it just needs a start script adding in the package.json file

“Require” on a package looks for “main” in package.json

Npm version patch / minor/ major works well with git and tags the code with the release number

Use ~ or ^ or x in version numbers to control automatic versioning

Good practice to lock dependencies down to ensure deployments have exactly the same setup as the development environment

Need to understand inheritance of EventEmitter as a way to provide a clean API for event-raising modules

Stream-handbook is a good resource for understanding stream handling (<https://github.com/substack/stream-handbook> )

Promises – make sure any framework follows A+ spec

<https://www.promisejs.org/>

<https://github.com/tildeio/rsvp.js/>

<http://howtonode.org/promises>

Html5rocks promises

<http://www.html5rocks.com/en/tutorials/es6/promises/>

Debugging – run node-inspector in one window; run “node --debug <script>” in another to debug

Tracegl is a useful tracing library but seems difficult to track down

<http://remysharp.com/2013/05/13/tracegl-for-javascript-debugging/>