**Notes**

**2/25/17 Session 1**

* **development stack**
* **javascript**
  + EMSAScript actual name
* **node.js**
  + v8 compiler
  + chrome compiles
  + allows js on local machine
  + runs on backend server
* **mongoDB**
  + document db
  + JSON
  + store as objects
* **restify.js**
  + js library for writing restful apis
* **lodash.sj**
* **git/github**
  + source control
  + auto backup
  + needed for jobs, put class assignments and projects there
  + every day, commit your work
* **heroku**
  + hosted web service
* db
* store data between sessions and services
* can be a bottleneck if centralized db
* we will learn noSQL
* big table, map reduce
* by google, distributed
* would need to intstall node\_modules in a dir for a sproject
* project structure
  + each project in its own folder
    - this folder becomes git repo
* syntax for cmd lines
  + *command –option argument*
* pwd
  + present working directory
* ~
  + linux root
* usr
  + mac root
* touch
  + make empty file
* delete folder contents recursively
  + rm –r foldername
* \*\*\* warning \*\*\*
  + rm –r /
  + will try to delete everything
* rename folder
  + mv oldname newname
* copy
  + cp
  + need –r for directories
* which
  + tells which command we am using
* man move
  + page though manual
* echo
  + echo what is typed
  + can feed output of one command into another
  + >
    - write to file
  + >>
    - append
* less filename
  + another pager
  + open a file with read only
* .
  + current dir
* ..
  + directory above
* add file into git
  + git init
  + git add index.js
  + git status
* restore a file from previous
  + git checkout –
    - drop changes that have not been committed yet
* create git repo on cloud
* now need to sync local together with cloud version
* mark down
* setup project
  + npm init
* git diff
  + show difference with a previously pushed file with another of the same name that has not been modified
* add lodash to install
* install dependencies
  + npm install lodash –save
* dev trees good to set up
* git public is free
* git private costs a monthly fee
* git pages allows a way to display web pages
* heat map in github under my own profile
  + show my activity for the past 365 days
* be aware of case sensitivity

**Questions on heroku v2 h/w 3/27/17**

* on the setup, under the section “push local changes,” I have followed the steps but at visiting localhost:5000/cool, instead of the funny eyes the following message is displayed on the screen
* “Cannot GET / Cool”

**Study Group 3/28/17**

* copy dir
* mv sourcename/ destname/
* open files/folders
* can drag and into editor
* security on creating new repo on cloud github
* https
* ssh
* tig command
* shows history of git changes
* ls –alh
* checkout a previous git revision
* git log
* git checkout *uniqueID*
* git checkout –b “unique branch name”
* create branch and give name
* git checkout
* is like making a .zip file of the master
* can start heroku server with
* $ heroku local (or)
* npm start run

**4/1/17 Session 2**

JavaScript

* functional language
  + read the line and return something back
* based on lisp
* C++ type of syntax
* open dev tools
  + debug
  + progress
* browsers understand text not code
* REPL
  + read evaluate print loop
  + lisp interpreter ‘>’
* tries to determine variable types
  + sometime it guesses types that are not correct
* truthiness
  + logic tables
  + true || true
  + …
  + true && false
  + …
  + 1 == 2
    - false
  + 1.999999 == 2
    - false
  + ===
    - use most all of the time
* UTF-8
  + same as ascii
* “bob” < “Bob”
  + false
* variable
  + a bucket
* if a variable is undefined then there is space available in memory for it
* if try to add an undefined number to a number will get NaN
* can test code in REPL to see if it works
* semicolons are only required in some places best to use semicolons everywhere
* flow control
  + user of conditional statements
* loops
* github
  + when do locally, do not need to git push
  + push is only for backup to cloud git
* global scoping of variables use up memory
* loops require 3 things
  + initial value
  + test of value
  + add stuff each time
* for loop does all three
* **clone class repo**
  + **for updates**

data structures

* finding last item in array
* array
* .push
* add to array
* .pop
* removes last item on list
* from node REPL
* can see functions that can be performed on arrays
* type array followed by a dot
* they press tab
* multidimensional array

github

* $ git init
* do in each local folder
* $ git add
* git commit
* will not be stored in cloud git repo until do push
* do not have to do a push for a local repo

noSQL

* is based on how data is stored in objects
* MySQL is not done this way

objects

* what arrays are stored as
* one way to determine the construct of a new programming language is how it defines arrays. are they defined as objects?
* def
* a named array

functions

* can store functions inside of objects
* more logical
* scope is global unless it is in an object
* e.g., tom.spanish
* a way to project from using up too much memory
* there are top level objects
* should not have to create a class
* what is returned is the last thing that happened
* return ends the execution

**homework**

* create local node app that lets us populate a to-do list, lets us see what is in the list and ability to mark off. This can run on the command line.
* array
* present to user
* what do you want to do next
* does not have to keep the data
* can use library we have already created