NodeSchool International Day and Hebocon (crappy robot contest) extravaganza!

Luiza felty, Monday, reboot grant 503 612 4331

Service agreement ends? Jan 22 2016 termination fees

Object.keys()

Npm start initiates server

<http://localhost:3000/>

outer html: another string

node: text between tags, generic thing in the tree, not all nodes are elements, some nodes are only text

element: special kind of node which represents something in a tag

first child: first leftmost child which may or may not be an element

first element child: first thing that’s an element

traverse the tree or dom:

document.body.children[1]

move down dom: children

move up dom: parentElement

parentNode and parentElement is basically the same thing

an element is a node with hmtl tags

getting/setting html

element.innerHTML

traversal: element.children, element.parent, element.next, element.previous

retrieval: document.getElementByID, or className, Name

moving/restructuring things:

restructuring: appendChild

h1.appendChild(p1)

create a variable for something in the dom: var variable = document.body

decorating: element.className

class name: 1 single string

class list: all words in string broken into an array

classList.add(‘class’)

classList.remove

classList.toggle

express.js: framework for serving webpages, library for node that builds websites

routing

rendering

error handling

jade: templating language for HTML

significant whitespace, it’s meaningful ot the software that interprets your template

compiles to html

JavaScript uses indendation as a convention (makes it easier for the user to read and understand)

More archane than html but more versitile

single app that looks at urls and does some logic based on them that isn’t just look at the file system

look at the list of things the server knows how to handle

extends layout : before you start putting out html, it get’s layout.jade

cmd shift p, install packages, jade, look at bottom right, change plain text to jade

routing:

get: client requests something from a server

ajax and rest

asyncronous calls with javascript and json (not xml)

async: lets the whole page get loaded, then some more content gets added on when it’s ready

npm install –g express-generator

express json-play

cd json-play

git init

git add/commit “initial commit”

npm install

cmd option j brings up javascript console

rest: way to share documents.

Vague term

Nowns: something on the server

Verb: here’s what I want to do, view a users profile, send a new message

Get: read

Post: create

Put: update

Delete: delete

Rest provides a context/shared notion of how thigns will work around an api (application programming interface)

Api: url’s you can call, methods you can use

Documentation: human readable text to describe what you can do with the api

Think of the url you’re fetching and ….

Crud: post, read, update, delete

Format: underlying how I want my browser to encode it so the server can understand it

Verbs:

gitHub api