Backend (Node.js) LIJIA XU (1/)

Install Hyper (hyper.is) | fun: hachertyper.com

git-scm.com, use Hyper configer file

Terminal → Bash Shell (Bourne Again Shell) | UNIX

```
ls => list all the items
cd => change directory (autofill with Tab)
cd ~ //back to root
```
cd Documents/learn

//use ↑ and ↓ to go over previous commands

```
cd .. //go up one level
```
(space)

clear //clean screen

Hold alt + move cursor // change location

```
control + A //begining of line
control + Z//end of line
control + U// clean current line
```

terminal-mac-cheat sheet

mac ⇒ open
windows ⇒ start

```
mkdir ⇒ make a new folder
touch Text.txt ⇒ create new file
open Text.txt ⇒ open the file
open -a Atom Text.~~txt~~ ⇒ use Atom open file
rm * ⇒ remove every file in current directory
rm -r Documents/ ⇒ remove the folder
```

rm Text.txt ⇒ delete the file

Mac bonus ⇒ emacs -batch -l dunnet

Front End: HTML CSS JS, botstrap jQuery
server: cakephp, php
Application: Ruby on Rails
Database: Java sprint

```
node --version
```
node JS (Runs on JS) express

```
pwd ⇒ print working directory
```

REPL
Read Evaluation Print Loops

```
.exit or control + C (twice) ⇒ exit the REPL
clear ⇒ clear the previous commands
```

Node build in module File system

```
const fs = require("fs"); //use interinal module
//jshint esversion:6 ⇒ (prevent Atom hints)
fs.copyFileSync("file1.txt", "file2.txt");
```

NPM
node package manager

```
npm init
npm install superheroes
```

```
ex. var sh = require("superheroes");
var Mysh = sh.random();
console.log(Mysh);
```

Express, node framework

atom. ⇒ quickly open project

expressjs.com | npm install express

```
const express = require("express");
const app = express();
app.listen(3000);
//use control + C to stop
```

```
app.listen(3000, function(){
console.log("Hi");
});
```

use chrome: localhost:3000

```
app.get("/", function(request, response){
  response.send("<h1>Hi!</h1>");
});
```

It get error: npm i pstree.remy@1.1.0 -D

```
npm install -g nodemon
```
or add sudo front

nodemon ⇒ auto restart the server

```
npm install -g nodemon
nodemon server.js
```

atom. ⇒ use atom open folder

```
app.get("/", function(req, res){
  res.sendFile(__dirname + "/index.html");
});
```

grab and display the current file location

```
<form action="/" method = "POST">
  <input type="text" name="num1" value="First">
  <input type="text" name="num2" value="Second">
  <button type="submit" name="submit">Cal</button>
</form>
```

1** hold on 2** Here you go 3** Go away 4** You F up 5** I F up

body-parser

```
npm install body-parser //read post
const express = require("express");
const bodyParser = require("body-Parser");
const app = express();
app.use(bodyParser.urlencoded({extended:true}));

app.post("/", function(req, res){
  var num1 = Number(req.body.num1);
  var num2 = Number(req.body.num2);
  var result = num1 + num2;
  res.send("the result" + result);
});
```

parseFloat();

bodyParser enables req.body.name

Git and Github | git init | ls -a //show hidden

```
git status
git add charpte1.txt
```
git add . //add all in current dir

```
git commit -m "complete char1" //use present
git log
```
message

git rm --cached -r //clean all staging area

working Directory —git add→ staging Area —git commit→ Local Repository
git checkout

```
git diff chapter1.txt // red old green new
git checkout chapter1.txt
```

Creating remote

```
git remote add origin
```
//can be other names but prefer 'origin'

https: ___ ___

```
git push -u origin master
```
remote name; < Branch name

```
touch .gitignore
```
→ #comments
*.txt // all files with txt ignored
→ file names

open-source-ios-apps

```
git clone https ____
git branch alien-plot
```
create a branch called alien-plot

```
git checkout alien-plot //switch branch
git branch // show the current branch
git merge alien-plot // run the command from master branch
:q! //save and quit, lear how to use VIM
```

APIs Application Programming Interfaces
create software or interact with external system

cURL ⇒ make http request with command line

man curl ⇒ provide manual | q ⇒ exit

ctrl Alt B ⇒ Beautify the code

JSON

unixtimestamp.com

Iterface ⇒ deal with the requests (require reqs)

Get and Post

Any-API.com

httpstatus.com

```
touch index.js index.html
npm init
npm install express body-Parser request

//jshint esversion:6
const request= require("request");
const express = require("express");
const bodyParser = require("body-parser");
const app = express();
app.use(bodyParser.urlencoded({extended:
                    true}));

app.listen(3000, function(){
   console.log("running on 3000");
});
app.get("/", function(req, res){
   res.sendFile(__dirname + "/index.html");
});
```

in chrome ⇒ shift + enter new line

index.html:

```
<form action="/" method="post">
  <select name="crypto">
    <option value="BTC">Bitcon</option>
    <option value="ETH">Etherum</option>
  </select>
<button type="submit" name="button">Check</button>
</form>
```

```
app.post("/"), function(req, res){
   //req.body.crypto
   request("address ....", function(error, res, body){
     // ___
   });
});
```

JSON ⇒ STD interchange form
JavaScript Object Notation

```
{
  "ask": 36,
  "open":{
     "hour": 85
  }
}
```

```
var some = {
  door: 2,
  color: "red"
}
```

Convert JS object to JSON

```
JSON.stringify(some);
```

convert JSON to JS object

```
var jsonD = JSON.stringify(some);
JSON.parse(jsonD);
```

JSON viewer Awesome → chrome extension

```
var data = JSON.parse(body);
var price = data.last;
```