First Day  
**hello.js**  
var http = require('http');

http.createServer(function(req,resp){

resp.writeHead(200);

resp.write("Hello,this is dog.");

resp.end();

}).listen(41180);

console.log('Listening on port 41180');

**\_or\_**

var http = require('http');

var server = http.createServer();

server.on('request',function(request,response){

response.writeHead(200);

response.write('It is car');

response.end();

}).listen(4444);  
  
 **\_or\_**  
var http = require('http');

http.createServer(function(request,response){

response.writeHead(200);

request.on('data',function(chunk){

response.write(chunk);

});

request.on('end',function() {

response.end();

});

}).listen(3333);

**\_or\_**  
http.createServer(function(request,response){

response.writeHead(200);

request.pipe(response);

}).listen(3333);

//curl -d 'hello' http://localhost:3333

Conditions

var x = "true"

/\*

if(x=="true"){

console.log("hello");

}else{

console.log("world");

}

\*/

/\*var x = true

switch(!x){

case true:

console.log("Hello");

break;

case false:

console.log("world");

break;

default:

console.log("bye");

}

\*/

ReadLine

var readline = require('readline'),

rl = readline.createInterface(process.stdin,process.stdout),prefix = 'OHAI> ';

rl.on('line',function(line) {

switch(line.trim()){

case 'hello':

console.log("world");

break;

default:

console.log("Say what? I might have heard"+line.trim() + ' ');

break;

}

rl.setPrompt(prefix,prefix.length);

rl.prompt();

}).on('close',function() {

console.log('Have a great day!');

process.exit(0);

});

Stream

**Stream1.js**  
var fs = require('fs');

var file = fs.createReadStream('readme.md');

var newFile = fs.createWriteStream('readme\_copy.md');

file.pipe(newFile);

**stream2.js**

var fs = require('fs');

var http = require('http');

http.createServer(function(request,response){

var newFile = fs.createWriteStream("readme\_copy.md');

request.pipe(newFile);

request.on('end',function() {

response.end('uploaded!');

});

}).listen(3333);

//curl --upload-file readme.md http://localhost:3333

**stream3.js**var fs = require('fs');

var http = require('http');

http.createServer(function(request,response) {

var newFile = fs.createWriteStream('readme\_copy.md');

request.pip(newFile);

request.on('end',function() {

response.end('uploaded!');

});

}).listen(3333);

**Stream4.js**  
var fs = require('fs');

var http = require('http');

http.createServer(function(request,response) {

var newFile = fs.createWriteStream('readme\_copy.md');

var fileBytes = reques.headers['content-length'];

var uploadedBytes = 0;

request.pipe(newFile);

request.on('data',function(chunk) {

uploadedBytes +=chunk.length;

var progress = (uploadedBytes / fileBytes)\*100;

response.write('progress: ' + parseInt(progress,10) + "%\n");

});

//...

}).listen(3434);

Module

**User.js**var User = function(name, email) {

this.name = name;

this.email = email;

};

module.exports = User;

**CallBack.js**var fs = require('fs');

var contents = fs.readFileSync('./user.js');

var callback = function(err,contents){

console.log(contents);

}

fs.readFile('./user.js',callback);

**Character.js**var Character = function() {

\_this = this;

this.name = 'dustin';

return {

getName: function(){

return \_this.name;

}

}

};

module.exports = Character;

//require('./character')()["getName"]();

Timeout

Timeout.js  
var http = require('http');

http.createServer(function(req,resp){

resp.writeHead(200);

resp.write("Dog is running.");

setTimeout(function(){

resp.write("Dog is done.");

resp.end();

},5000);//5000ms = 5 seconds

}).listen(4080);

EventMitter

**eventcustom1.js**  
var EventEmitter = require('events').EventEmitter;

var logger = new EventEmitter();

logger.on('error',function(message){

console.log('ERR:'+message);

});

logger.emit('error','Spilled Milk');

logger.emit('error','Eggs Cracked');

logger.on('info',function(message){

console.log('INFO:'+message);

});

logger.emit('info','New Message');