* REPL is command line utility for Nodejs
* To start a web server:  
  var http = require('http');

var s = http.createServer(request\_handler);

s.listen(8080);

* In javascript, Infinity, -Infinity and NaN are valid values. (1.0 – 0.3 + 0.1 == 0.8 🡺 false). 0,””, NaN, null, undefined are converted as false.
* Global and process are 2 important Nodejs variables.
* prototype is always present for object in js:

function Square(){

}

Square.prototype = new Shape();

Square.prototype.\_\_proto\_\_ = Shape.protootype;

Square.prototype.area = function(){}

* ‘this’ context does not work with JS asynchronous call. To avoid them, we can assign them to var self = this; before calling asynch function.
* In Nodejs, all the process occurs in same thread. So if one function is taking a lot of time, other operation will need to wait for it. Node has 2 method setImmediate() which is useful to call recursive methods and process.nextTick() which is useful for sequential process and faster compare to other.
* While creating server using Nodejs, GET request parameters can be handle by slicing the URL string but for POST request, we need to handle 2 events (req.on) readable and end. In readable, we need to keep append stream to make actual request parameter. ‘querystring’ module is required to parse POST form parameters.
* Use ‘exports’ to expose any method from any module functions. Class can be exposed in 2 ways: 1) Exports creator method in module called factory method. 2) Export class with ‘module.exports’ and initialize with new keyword.
* If there are 2 different version of same module available in package, node makes sure that each module will receive respective version so that they works fine.
* The Mustache framework can be used to combine json with html (mustache exp).
* Usually, for web applications, Node server provides basic html page which load bootstrap script (JQuery) which call server to fetch data and fetch html from file server and then combine them.
* In Expressjs, we can use wildcard to handle multiple urls likes user[s]?/:username will handle user as well as users
* Express uses middleware to handle body, cookies and session with req object. Like:

app.use(express.bodyParser());

app.use(express.cookieParser());

app.use(expression.session({

store: new express.session.MemoryStore()

}));

* ‘process.stdin’ is used to interact with terminal and its always in pause state. So you need to call ‘process.stdin.resume()’ to enable it.