Ordered Solutions

Hello World

```
console.log("HELLO WORLD")
```

Baby Steps

```
var result = 0
for (var i = 2; i < process.argv.length; i++)
  result += Number(process.argv[i])
console.log(result)
```

My First I/O

```
var fs = require('fs')
var contents = fs.readFileSync(process.argv[2])
var lines = contents.toString().split('\n').length - 1
console.log(lines)
```

My First ASYNC I/O

```
var fs = require('fs')
var file = process.argv[2]
fs.readFile(file, function (err, contents) {
  // fs.readFile(file, 'utf8', callback) can also be used
  var lines = contents.toString().split('\n').length - 1
  console.log(lines)
})
```

Filtered LS

```
var fs = require('fs')
var path = require('path')
fs.readdir(process.argv[2], function (err, list) {
  list.forEach(function (file) {
    if (path.extname(file) === '.' + process.argv[3])
      console.log(file)
  })
})
```

Make it Modular

```
var filterFn = require('./solution_filter.js')
var dir = process.argv[2]
var filterStr = process.argv[3]
filterFn(dir, filterStr, function (err, list) {
  if (err)
    return console.error('There was an error:', err)
  list.forEach(function (file) {
    console.log(file)
  })
})
```

filter.js

```
var fs = require('fs')
var path = require('path')
module.exports = function (dir, filterStr, callback) {
  fs.readdir(dir, function (err, list) {
    if (err)
      return callback(err)
    list = list.filter(function (file) {
      return path.extname(file) === '.' + filterStr
    })
    callback(null, list)
  })
}
```

HTTP Client

```
var http = require('http')
http.get(process.argv[2], function (response) {
  response.setEncoding('utf8')
  response.on('data', console.log)
  response.on('error', console.error)
})
```

HTTP Collect

```
var http = require('http')
var bl = require('bl')
http.get(process.argv[2], function (response) {
  response.pipe(bl(function (err, data) {
    if (err)
      return console.error(err)
    data = data.toString()
    console.log(data.length)
    console.log(data)
  }))
})
```

Juggling Async

```
var http = require('http')
var bl = require('bl')
var results = []
var count = 0

function printResults () {
  for (var i = 0; i < 3; i++)
    console.log(results[i])
}
function httpGet (index) {
  http.get(process.argv[2 + index], function (response) {
    response.pipe(bl(function (err, data) {
      if (err)
        return console.error(data)
      results[index] = data.toString()
      count++
      if (count == 3) // yay! we are the last one!
        printResults()
    }))
  })
}
for (var i = 0; i < 3; i++)
  httpGet(i)
```

Time Server

```javascript
var net = require('net')

function zeroFill(i) {

  return (i < 10 ? '0' : '') + i

}

function now () {

  var d = new Date()

  return d.getFullYear() + '-'

    + zeroFill(d.getMonth() + 1) + '-'

    + zeroFill(d.getDate()) + ' '

    + zeroFill(d.getHours()) + ':'

    + zeroFill(d.getMinutes())

}

var server = net.createServer(function (socket) {

  socket.end(now() + '\n')

})

server.listen(Number(process.argv[2]))
```

HTTP File Server

```javascript
var http = require('http')

var fs = require('fs')

var server = http.createServer(function (req, res) {

  res.writeHead(200, { 'content-type': 'text/plain' })

  fs.createReadStream(process.argv[3]).pipe(res)

})

server.listen(Number(process.argv[2]))
```

HTTP Uppercase

```javascript
var http = require('http')

var map = require('through2-map')

var server = http.createServer(function (req, res) {

  if (req.method != 'POST')

    return res.end('send me a POST\n')

  req.pipe(map(function (chunk) {

    return chunk.toString().toUpperCase()

  })).pipe(res)

})

server.listen(Number(process.argv[2]))
```

HTTP JSON API Server

```
var http = require('http')
var url = require('url')
function parsetime (time) {
  return {
    hour: time.getHours(),
    minute: time.getMinutes(),
    second: time.getSeconds()
  }
}
function unixtime (time) {
  return { unixtime : time.getTime() }
}
var server = http.createServer(function (req, res) {
  var parsedUrl = url.parse(req.url, true)
  var time = new Date(parsedUrl.query.iso)
  var result
  if (/^\/api\/parsetime/.test(req.url))
    result = parsetime(time)
  else if (/^\/api\/unixtime/.test(req.url))
    result = unixtime(time)
  if (result) {
    res.writeHead(200, { 'Content-Type': 'application/json' })
    res.end(JSON.stringify(result))
  } else {
    res.writeHead(404)
    res.end()
  }
})
server.listen(Number(process.argv[2]))
```