



# WARNING

ZOMBIE ATTACK  
IMMINENT

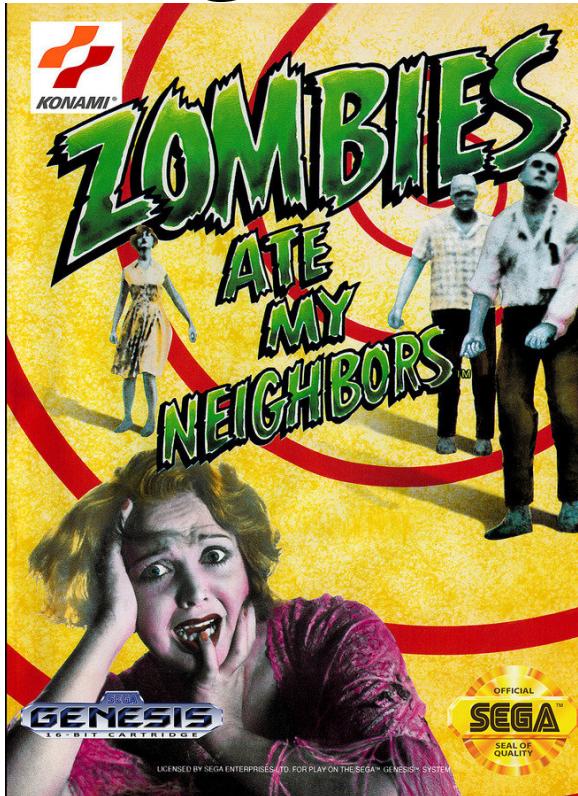
# HISTORY TIME

Get pumped!

**HISTORY TIME**  
Get pumped!

1993

# Zombies Ate My Neighbors





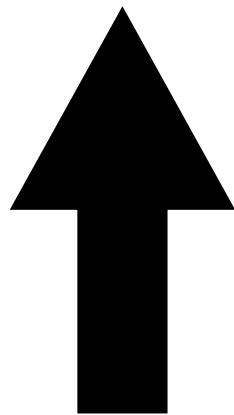
@benspoon

1995

~~Live~~Script JavaScript was born

# HISTORY TIME

Get pumped!



1999

ECMAScript third edition

- powerful regular expressions
- better string handling
- new control statements
- try/catch exception handling
- tighter definition of errors
- formatting for numeric output



*“ Continuing lack of an IDE, debugger, and other standard development tools are still considered insurmountable problems by many programmers, who defend their Windows-only applications with the not unreasonable excuse that debugging is too hard on a platform that lacks a debugger.*

**- Steve Champeon,**  
**2001**



*“ Most of the people writing in  
JavaScript are not programmers. They  
lack the training and discipline to  
write good programs... ”*

- **Douglas Crockford,**  
2001

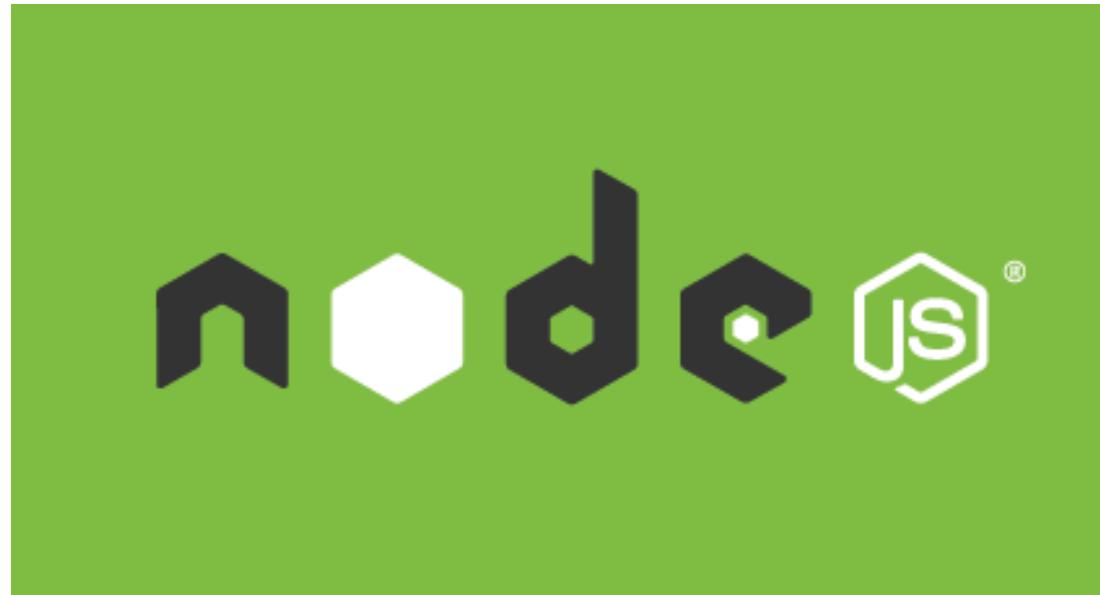
“ ...*This has given JavaScript a reputation of being strictly for the amateurs, that it is not suitable for professional programming. This is simply not the case.*

- Douglas Crockford,  
2001

# **But it isn't 2001**

# JavaScript Developers

We are kind of a big deal.



@benspoon

# Web Servers

Games

Phone Calls

Text Messages

Web Sockets

# SMTP Servers

It's pretty amazing.

# Node.js Serial Port

[github.com/voodootikigod/node-serialport](https://github.com/voodootikigod/node-serialport)



**Chris Williams**

@voodootikigod FOLLOWS YOU

Cofounder @etechadvisor 'the Phil Collins of Node & Hardware' Maker of Improbable Things: VoodooSpark, node-serialport, nodebots, nodecopter, JSConf, RobotsConf

📍 Reston, VA

🔗 [voodootikigod.com](http://voodootikigod.com)

# Serial Port Access

Johnny-Five

Cylon.js

node-l8smartlight

firmata

tmpad source

duino

Arduino Drinking Game Extravaganza

Arduino controlling popcorn.js

Robotic JavaScript

devicestack

reflecta

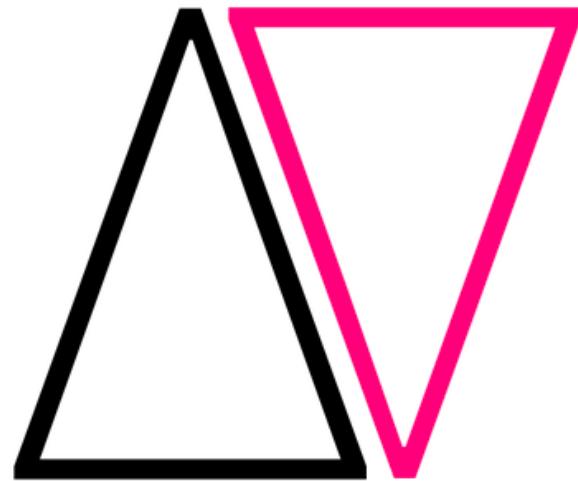
rc4pt-node



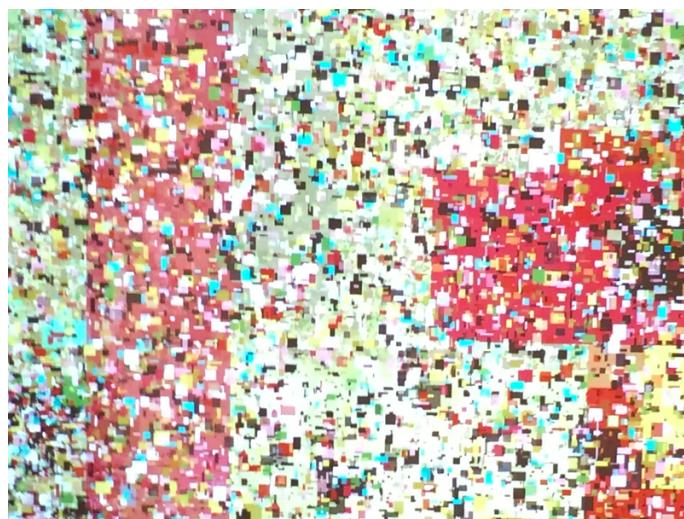
 @benspoon

 AWeber

**PhilaU**  
PHILADELPHIA UNIVERSITY



ART by ALGORITHM



@benspoon



SIGN IN

WHAT

WHY

ARTISTS

BLOG

Username

\*

Password

\*

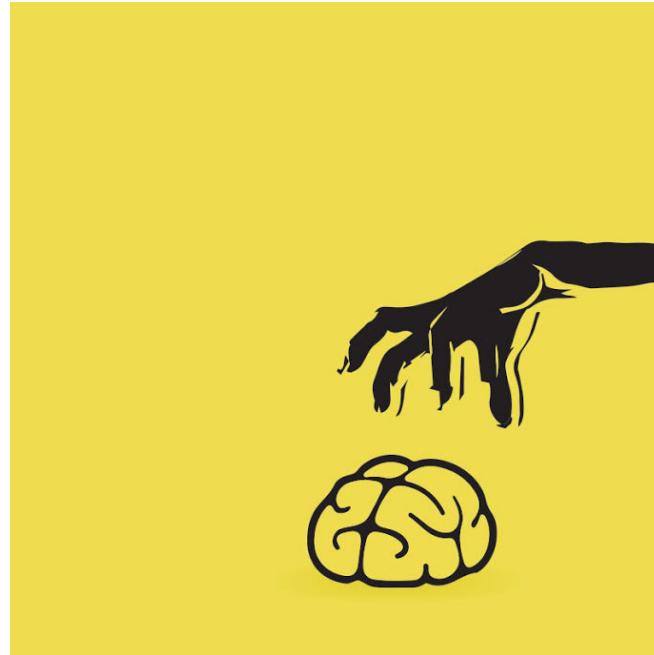
SIGN IN

ARTIST SIGN IN

FORGET SOMETHING?

# Protect Yourself Against Zombies

With JavaScript



(yes, JavaScript)

An Exploration of JavaScript Touching on The  
Acceptance and Diversity of the Language Within  
Real World Applications.

**Sounded way too boring**

# Zombies



# In our scenario

- The Zombies didn't kill all the electricity
- Or the internet
- Or are very intelligent
- Or move very fast
- And may or may not be vulnerable to water



# **And Luckily:**

- You know how to program in JavaScript
- You own a few drones
- You own a bunch of Arduino boards and parts
- You own a bunch of computers

# **Getting Started:**

Protecting the perimeter

## **Getting Started:**

Protecting the perimeter

# Detect

# Alert

# Take Action

## **Getting Started:**

Protecting the perimeter

# Detect

## Getting Started:

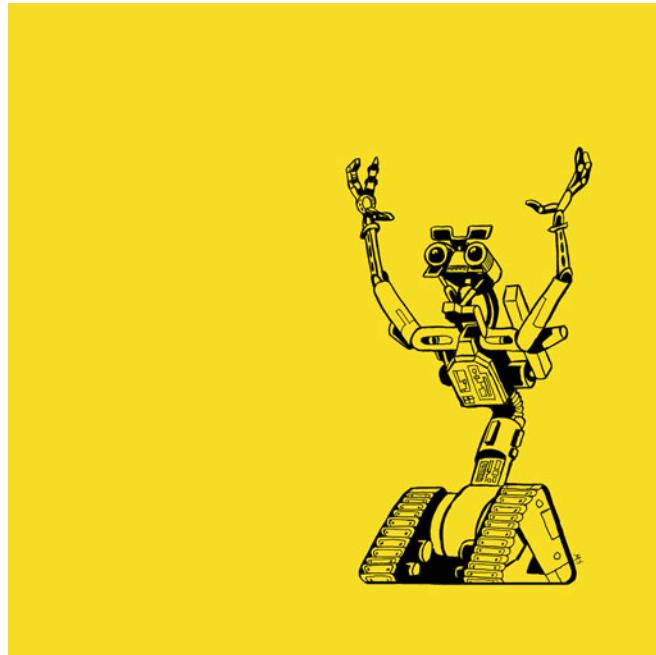
Protecting the perimeter



## **Getting Started:**

Protecting the perimeter

# So let's talk about robots



## Getting Started:

Protecting the perimeter

# Johnny-Five



**rick waldron**  
@rwaldron

Champion for all NodeBots. Creator of Johnny-Five, the JavaScript Robotics Programming framework for Node.js  
[johnny-five.io](http://johnny-five.io)

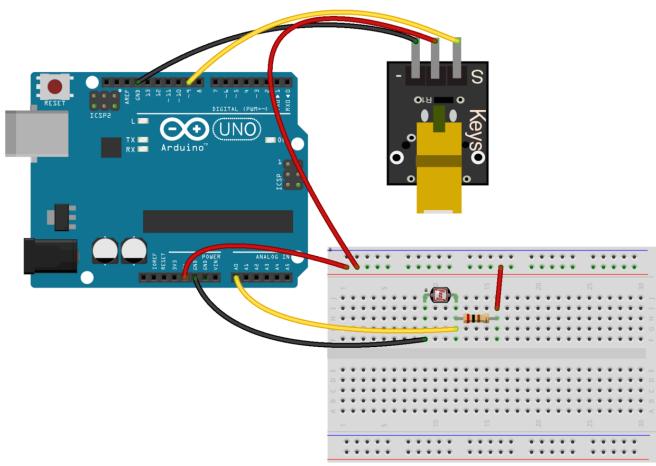
📍 Brooklyn, NY  
🔗 [github.com/rwaldron](https://github.com/rwaldron)  
⌚ Joined September 2008

<https://github.com/rwaldron/johnny-five>

## Getting Started:

Protecting the perimeter

# Laser Trip Wire



```
var five = require("johnny-five");
var board = new five.Board();

board.on("ready", function() {
  var laser = new five.Led(9);
  var detection = new five.Sensor("A0");
  var isSecure = false;

  laser.on();

  detection.scale(0, 1).on("change", function() {
    // Note that my sensor isn't as good as
    // the one used in the example (I guess)
    // so I had to change this next line to over
    // 0.9 instead of 0 | 1.
    var reading = !(this.value > 0.9);

    if (isSecure !== reading) {
      isSecure = reading;

      if (!isSecure) {
        console.log("Intruder");
      }
    }
  });
});
```

<https://github.com/rwaldron/johnny-five/blob/master/docs/laser-trip-wire.md>

@benspoon

## **Getting Started:**

Protecting the perimeter

Zombies don't care about  
console.log()

## **Getting Started:**

Protecting the perimeter

# Alert

## Getting Started:

Protecting the perimeter

# Express

<http://expressjs.com/>

```
var express = require('express'),  
    app = express();  
  
var port = process.env.PORT || 8080;  
  
// test route to make sure everything is working (accessed at GET http://localhost:8080/api)  
app.get('/', function(req, res) {  
  res.send('Zombie protection service is up and running!');  
  // optionally we could use json in our response:  
  // res.json(message: 'Zombie protection service is up and running!');  
});  
  
//start server  
app.listen(port);  
console.log('Starting Zombie protection service on port ' + port);
```

```
$ node example/scripts server.js  
Running "execute:server" (execute) task  
-> executing /Users/spoon/git/zombies/scripts/server.js  
Starting Zombie protection system on port 8080
```

## Getting Started:

Protecting the perimeter

# Request

<https://github.com/request/request>

```
var request = require('request');

request('http://localhost:8080/', function (error, response, body) {
  if (!error && response.statusCode == 200) {
    console.log(body)
  } else {
    console.log('Some sort of error!');
  }
})
```

```
$ node scripts/request.js
Zombie protection service is up and running!
```

## Getting Started:

Protecting the perimeter

# Database time



<https://www.mongodb.org/>

# mongoose

<http://mongoosejs.com/>

## Getting Started:

Protecting the perimeter

```
var express = require("express");
var mongoose = require("mongoose");

mongoose.connect('mongodb://localhost/zombies');

var sensorListSchema = {
    unit_id: Number,
    location: String,
    type: String
}

var SensorList = mongoose.model('SensorList', sensorListSchema, 'sensors');

var app = express();

// List all the sensors
app.get('/listSensors', function (req, res) {
    SensorList.find(function (err, doc) {
        res.send(doc);
    });
});

app.listen(8080);
```

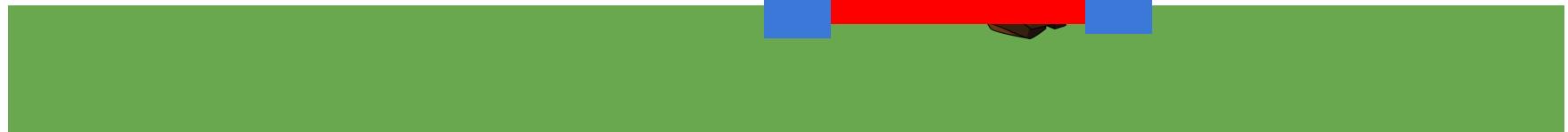
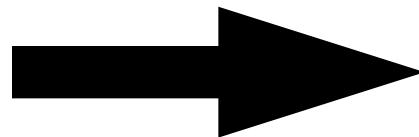
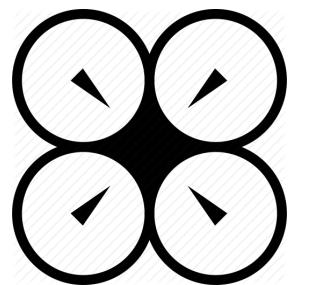
## **Getting Started:**

Protecting the perimeter

# Take Action

## Getting Started:

Protecting the perimeter



## Getting Started:

Protecting the perimeter

# Parrot Rolling-Spider



## **Getting Started:**

Protecting the perimeter

# Parrot AR.Drone 2.0



<https://github.com/felixge/node-ar-drone>  
<http://www.nodecopter.com/>

## Getting Started:

Protecting the perimeter

# Parrot Rolling Spider

<https://github.com/voodootikigod/node-rolling-spider>

```
var RollingSpider = require("rolling-spider"),
    temporal = require("temporal"),
    drone = new RollingSpider();

drone.connect(function() {
  drone.setup(function() {
    temporal.queue([
      {
        delay: 0,
        task: function () {
          drone.flatTrim();
          drone.startPing();
          console.log('connected!');
          drone.takeOff();
        }
      },
      {
        delay: 500,
        task: function() {
          console.log('landing!');
          drone.land();
        }
      }
    ]);
  });
});
```

@benspoon

## **Getting Started:**

Protecting the perimeter

# Sending Email (SMTP)

## Getting Started:

Protecting the perimeter

<https://github.com/andris9/Nodemailer>

```
var nodemailer = require('nodemailer'),
  fs = require('fs'),
  data = fs.readFileSync('./smtp-config.json'),
  smtpConfig;

smtpConfig = JSON.parse(data);

// create reusable transporter object using SMTP transport
var transporter = nodemailer.createTransport({
  service: 'Gmail',
  auth: {
    user: smtpConfig.username,
    pass: smtpConfig.password
  }
});
```

```
{
  "username" : "youremail@gmail.com",
  "password" : "yourPassword"
}
```

## Getting Started:

Protecting the perimeter

<https://github.com/andris9/Nodemailer>

```
// setup e-mail data with unicode symbols
var mailOptions = {
  from: 'Ben Spoon <ben@benspoon.com>', // sender address
  to: 'ben@benspoon.com', // list of receivers
  subject: 'ZOMBIES ARE HERE', // Subject line
  text: 'Dude, there are zombies. You should do something', // plaintext body
  html: '<b>Dude, there are zombies. You should do something</b>' // html body
};

// send mail with defined transport object
transporter.sendMail(mailOptions, function(error, info){
  if(error){
    return console.log(error);
  }
  console.log('Message sent: ' + info.response);

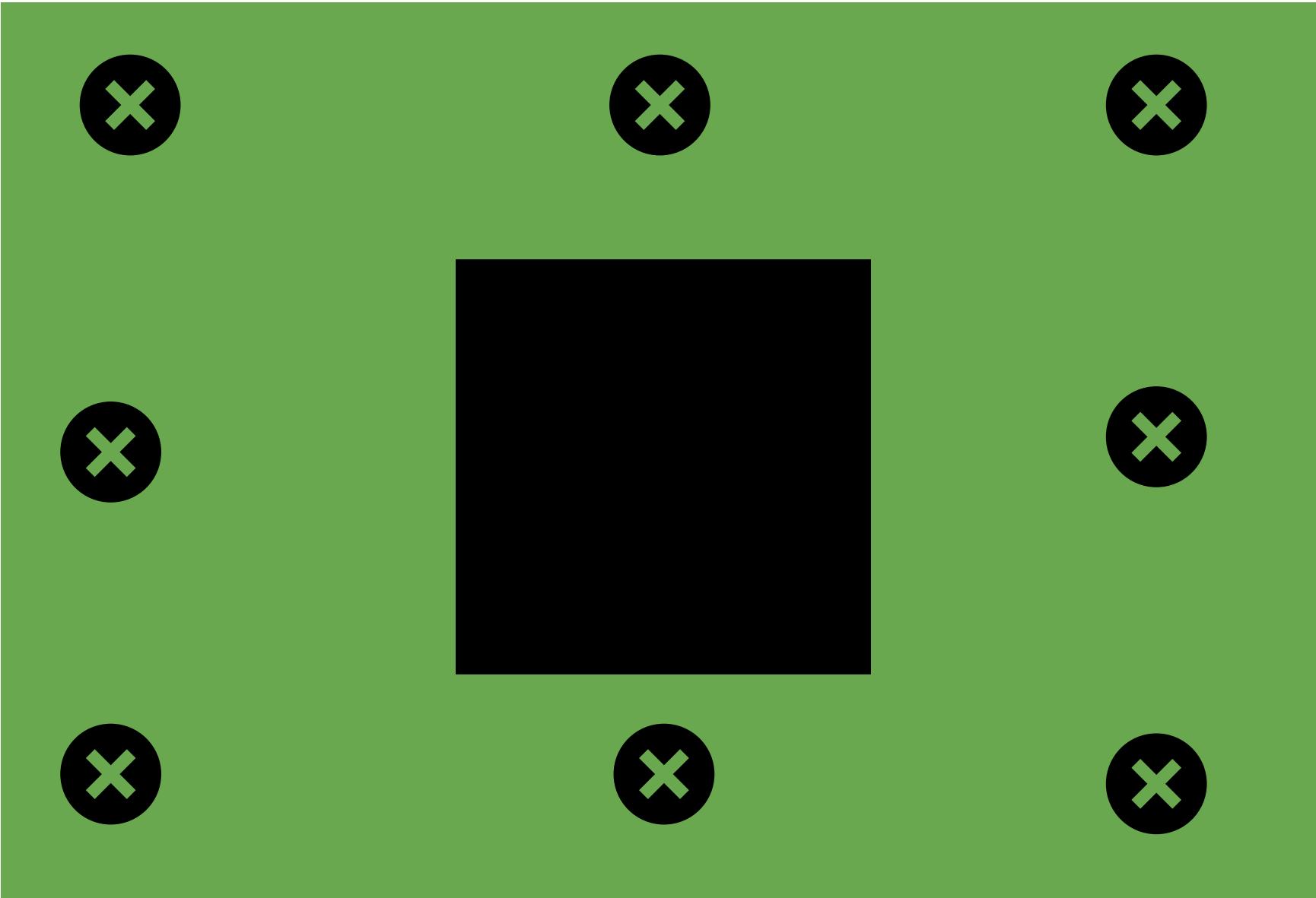
});
```

# Putting it all together

Putting it all together

- Laster Trip Wire - Johnny-Five
- Web Server - Express
- Sending a request - Request
- MongoDB & Mongoose
- Sending email - Nodemailer
- Launching a drone - rolling-spider

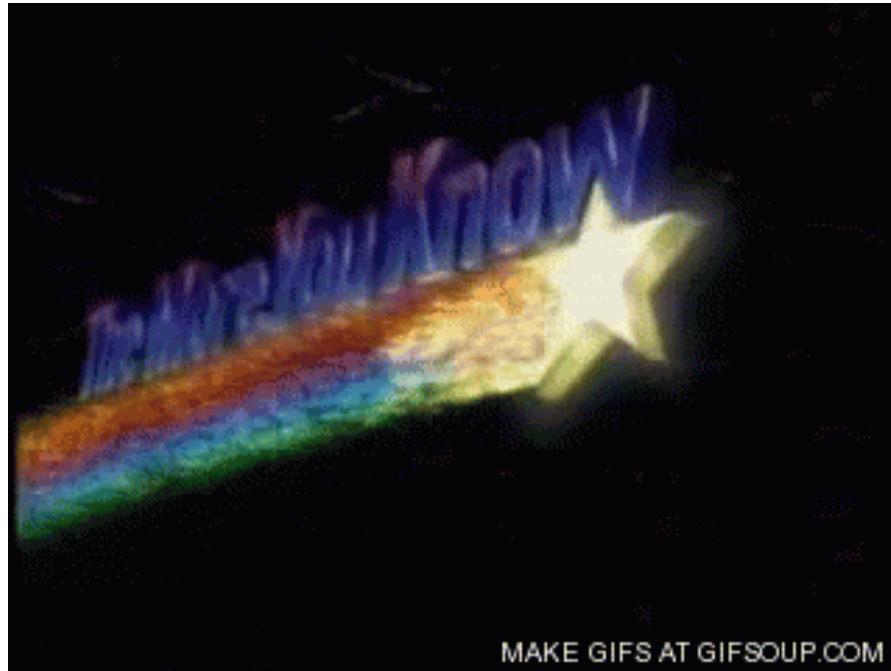
LETS DO IT



"Ok cool story"



"but why?"



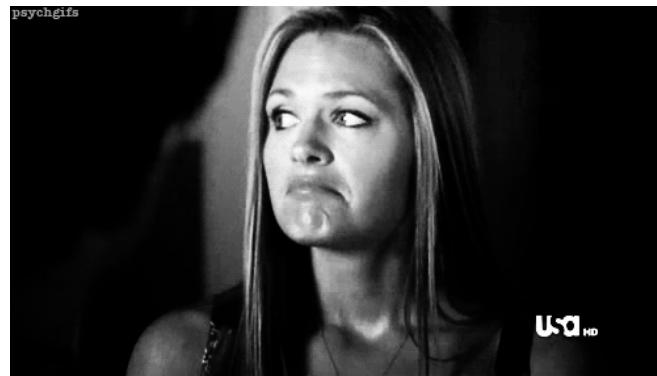
MAKE GIFS AT GFSOUP.COM

@benspoon

Maybe not



But hopefully



#notBad

@benspoon

<http://github.com/spoonben/zombies>



@benspoon

Thanks.