JAVASCRIPT DEVELOPMENT

Sasha Vodnik, Instructor

HELLO!

Pull changes from the `svodnik/jsd6` repo to your computer

SLACK BOT LAB

LEARNING OBJECTIVES

At the end of this class, you will be able to

‣ Install and configure all utilities needed to run a Hubot

‣ Write scripts that allow your Hubot to interact with users of the class Slack organization

AGENDA

‣ Install and configure utilities and accounts

‣ Explore sample code for bots

‣ Plan what you'd like your bot to do

‣ Create a basic bot to verify that your setup works

‣ Expand on your basic code to add your planned functionality

Checkin and questions

‣ The **most significant thing I learned** about Scope and Closures is _____.

‣ My **biggest outstanding question** about Scope and Closures is _____.

SLACK BOTS

SLACK AND BOTS

‣ **Bot**: A script programmed to interact with users as if it's a person

  ‣ Slackbot

  ‣ PlusPlus

‣ We will use a framework to create our own bots with interactive behaviors that we specify with our code

‣ These bots will be members of our class Slack organization

HUBOT

‣ **Hubot**: A framework meant to speed the process of developing bots for a variety of platforms, including Slack

‣ Includes built-in functionality for performing common bot tasks, such as posting images.

‣ We will use the Hubot framework to create our bots

HUBOT VS SLACK BOT VS SLACKBOT

‣ Hubot is the framework we're using

‣ Each of us will be building a bot for Slack === a Slack bot

‣ Slackbot is the name of a specific bot already installed in our Slack organization; it answers questions about how to use Slack

HEROKU

‣ **Heroku**: A platform for hosting and running apps in the cloud.

‣ We will create our code on our computers, then push it to Heroku so it can run even when our computers are sleeping or shut down

YEOMAN

‣ **Yeoman**: A set of tools that provides a scaffolding (basic structure) for getting web apps up and running quickly

‣ We'll use a Yeoman tool called yo, which automates a lot of behind-the-scenes work

COFFEESCRIPT

‣ **CoffeeScript**: A variant of JavaScript, intended to be more readable and faster to type.

‣ Only JavaScript can run in browsers

  ‣ Before being used, CoffeeScript code must be compiled, which is a process that translates it into JavaScript

‣ Many Hubot examples are written in CoffeeScript, but you can write Hubot code in vanilla JavaScript without any problem

UNDERSTANDING THE HUBOT FRAMEWORK

‣ As a framework, Hubot has its own way of doing things
‣ The code for your bot behaviors is structured as follows:

```
module.exports = function(robot) {
  robot.verb(parameter1, function(res) {
    return res.command();
  });
};
```

BASIC HUBOT VERBS

‣ `hear`: called anytime a message's text matches

‣ `respond`: called for messages immediately preceded by the robot's name or alias

LEARNING OBJECTIVES – REVIEW

‣ Install and configure all utilities needed to run a Hubot

‣ Write scripts that allow your Hubot to interact with users of the class Slack organization

NEXT CLASS PREVIEW

Objects and JSON

‣ Identify likely objects, attributes, and methods in real-world scenarios

‣ Create JavaScript objects using object literal notation

‣ Implement and interface with JSON data

Exit Tickets!

Q&A