GENERAL ASSEMBLY

JAVASCRIPT DEVELOPMENT

Sasha Vodnik, Instructor

HELLO!

1. Pull changes from the `svodnik/JS-SF-7` repo to your computer
2. Open the `starter-code` folder in your code editor

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 _____.

EXIT TICKET FEEDBACK

- “I don't fully understand the concept of closures at this point but I'm fine with that”
- “Today was tough. Concepts felt almost abstract, but I'm going to study more.”

SLACK BOTS

SLACK AND BOTS

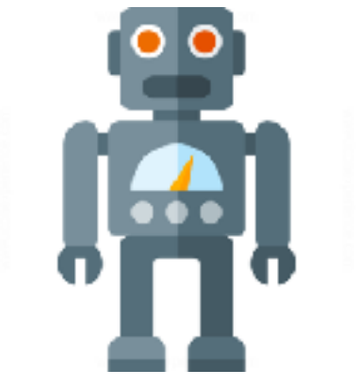
- **Bot**: A script programmed to interact with users as if it's a person
  - Slackbot
  - PlusPlus
- We will use a framework to create our own bots with interactive behaviors that we specify with our code
- These bots will be members of our class Slack organization

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

YEOMAN

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

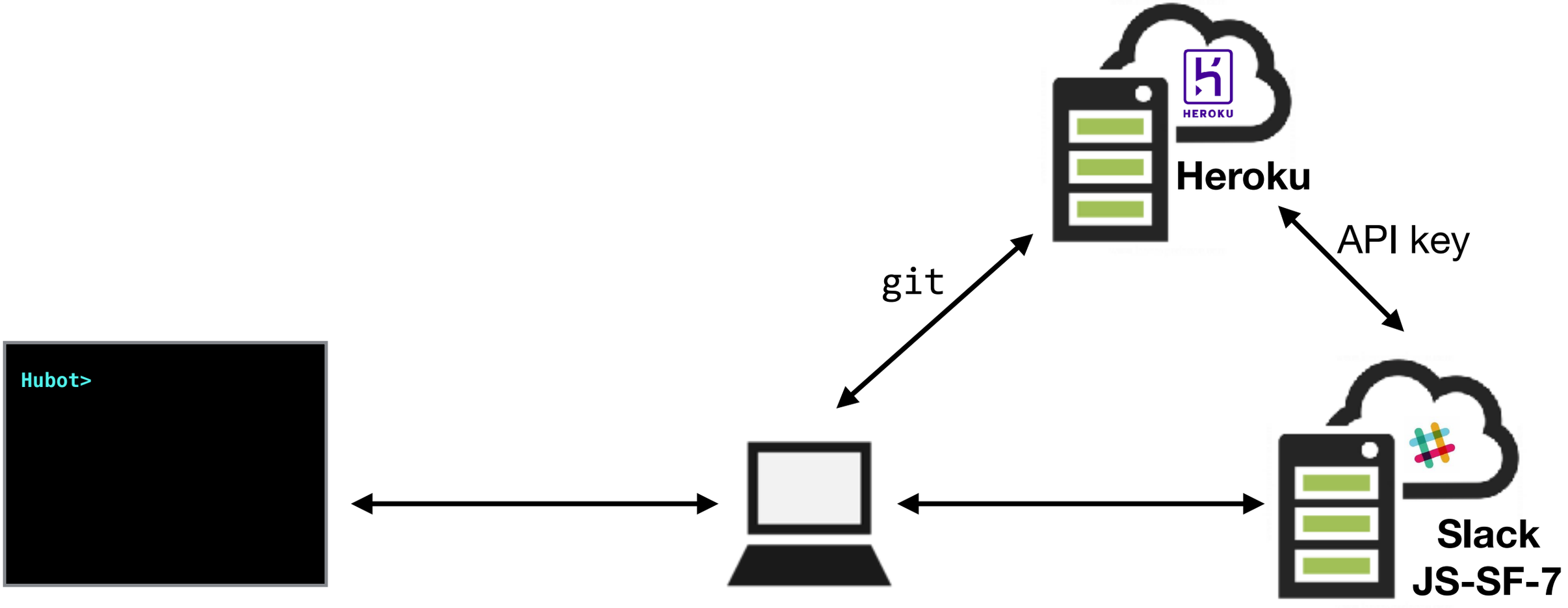
UNDERSTANDING THE HUBOT FRAMEWORK

- 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();
  });
};
```

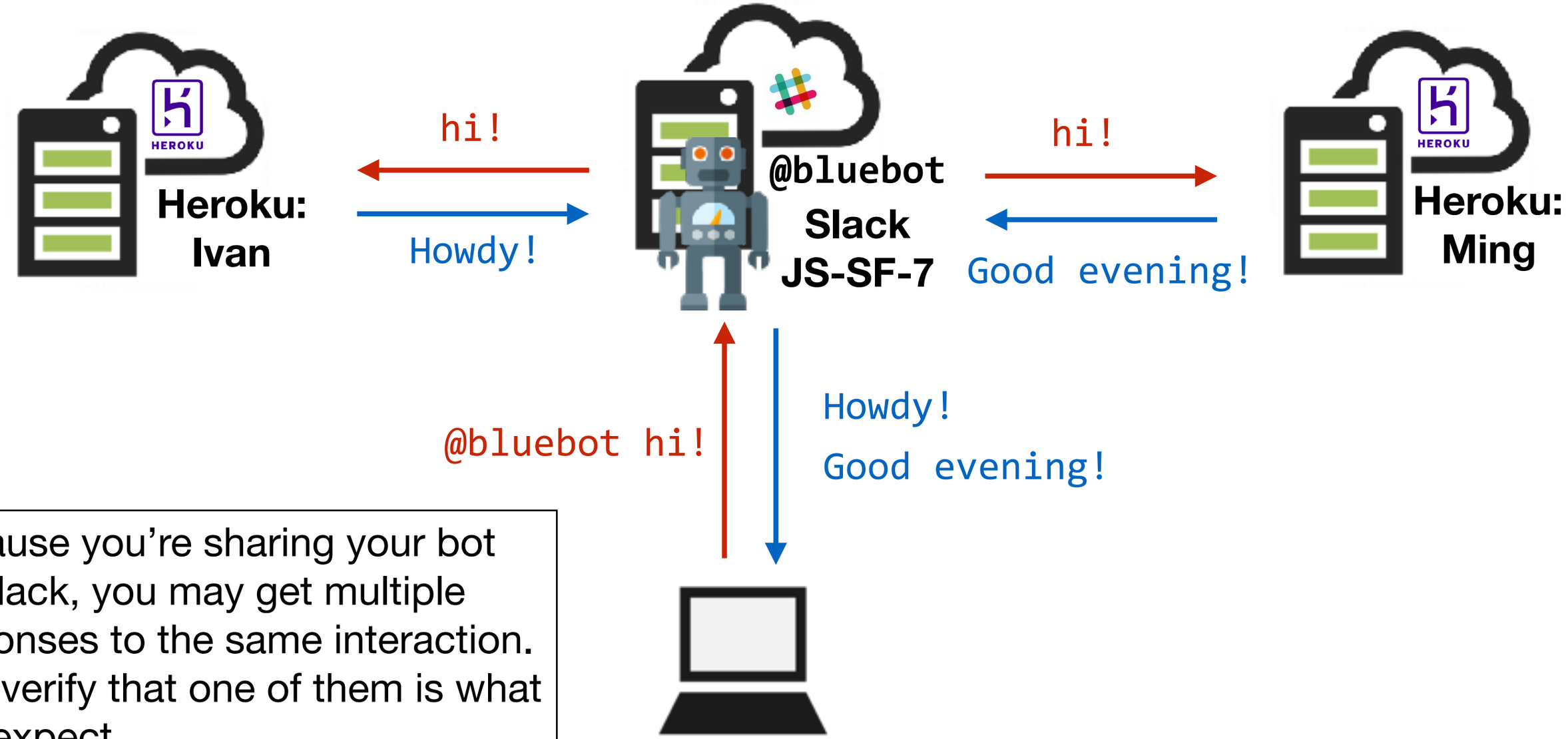
BASIC HUBOT VERBS

- `hear`: called anytime a message's text matches
- `respond`: called for messages immediately preceded by the robot's name or alias



Interacting with your bot at the command line involves local files on your computer only.

Interacting with your bot on the class Slack organization involves the files you published to your Heroku instance.



Because you're sharing your bot on Slack, you may get multiple responses to the same interaction. Just verify that one of them is what you expect.

ACTIVITY — HUBOT CONFIGURATION

KEY OBJECTIVE

- Install and configure all utilities to run a Hubot

LOCATION

- `JS-SF-7-resources > 1-slack-bot-install-guide.md`

EXECUTION

20 min

1. Follow the instructions to install command line utilities for building Hubots.
2. When you finish, start reading and exploring the sample code in `2-slack-bot-code-samples.md`

LAB — BUILD A SLACK BOT

KEY OBJECTIVE

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

LOCATION

- `starter-code > slackbot.js`

EXECUTION

Until 9:20

1. Uncommenting portions of the sample code in slackbot.js to explore how to code in the Hubot framework, and what a bot can do in Slack.
2. Try out some of the code samples in the `2-slack-bot-code-samples.md` file.
3. Create a plan for what you want your bot to be able to do, pseudocode it, and start building it!
4. BONUS: Experiment with advanced commands documented at https://github.com/github/hubot/blob/master/docs/scripting.md

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