GENERAL ASSEMBLY

JAVASCRIPT DEVELOPMENT

Sasha Vodnik, Instructor

HELLO!

1. Pull changes from the `JS-SF-14-resources` repo to your computer
2. DO NOT open the `05-slackbot-lab/starter-code` folder in your code editor! (We will copy the files to another location during class and work with them from there.)

JAVASCRIPT DEVELOPMENT

SLACK BOT LAB

LEARNING OBJECTIVES

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

- Install and configure all utilities needed to build a bot using the Hubot framework
- Write scripts that allow your bot to interact with users of the class Slack workspace

AGENDA

- Install and configure Slack bot 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

SLACK BOT LAB

WEEKLY OVERVIEW

WEEK 4	Slack Bot Lab / JSON & Intro to DOM
WEEK 5	Intro to jQuery / Events & jQuery
WEEK 6	Ajax & APIs / Asynchronous JS & callbacks

EXIT TICKET QUESTIONS

1. How to create the 3 monkeys as a single class and not 3 separate objects with repeated attributes and methods.
2. I'm interested to see how objects can be coded to be useful in real world scenarios, outside of the console.

HOMEWORK REVIEW

HOMEWORK — GROUP DISCUSSION

EXERCISE

TYPE OF EXERCISE

- Pairs

TIMING

6 min

1. Share 1 thing you’re excited about being able to accomplish.
2. Have each person in the group note 1 thing they found challenging for the exercises and make note. Discuss as a group how you think you could solve that problem.
3. Did you complete either of the bonus exercises? Demonstrate it and show your group how you did it!

REVIEW – CATCH PHRASE!

EXERCISE

TYPE OF EXERCISE

- Pairs

TIMING

3 min

1. Get your partner to guess the word on each piece of paper by giving clues describing it.
2. Take turns giving clues and guessing words.

SLACK BOTS

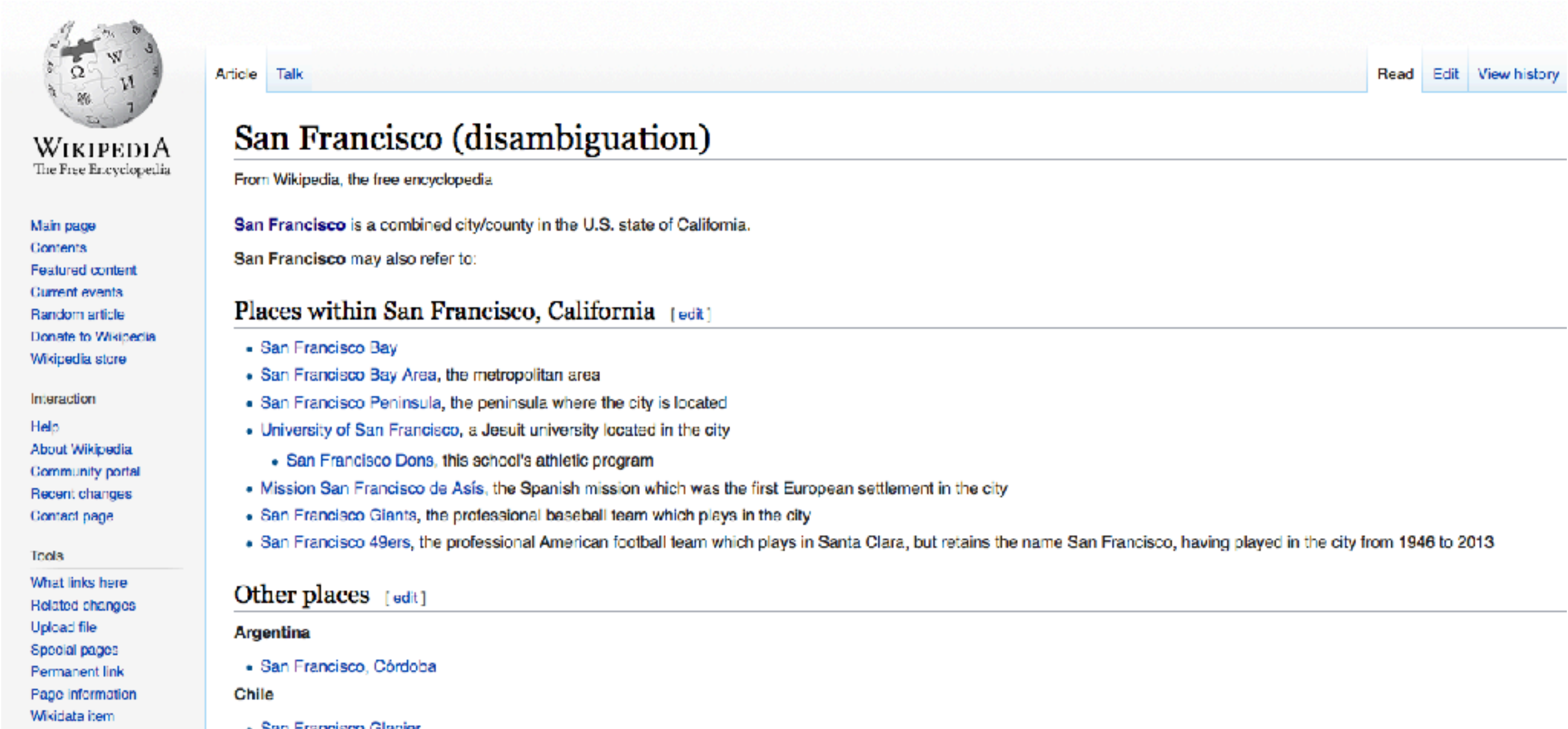
SLACK AND BOTS

- **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

DISAMBIGUATION



San Francisco (disambiguation)

From Wikipedia, the free encyclopedia

San Francisco is a combined city/county in the U.S. state of California.

San Francisco may also refer to:

Places within San Francisco, California [edit]

- San Francisco Bay
- San Francisco Bay Area, the metropolitan area
- San Francisco Peninsula, the peninsula where the city is located
- University of San Francisco, a Jesuit university located in the city
  - San Francisco Dons, this school's athletic program
- Mission San Francisco de Asís, the Spanish mission which was the first European settlement in the city
- San Francisco Giants, the professional baseball team which plays in the city
- San Francisco 49ers, the professional American football team which plays in Santa Clara, but retains the name San Francisco, having played in the city from 1946 to 2013

Other places [edit]

Argentina

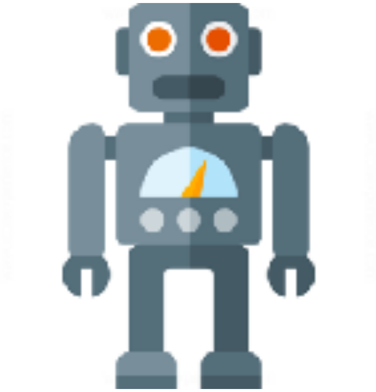
- San Francisco, Córdoba

Chile

- San Francisco Glacier

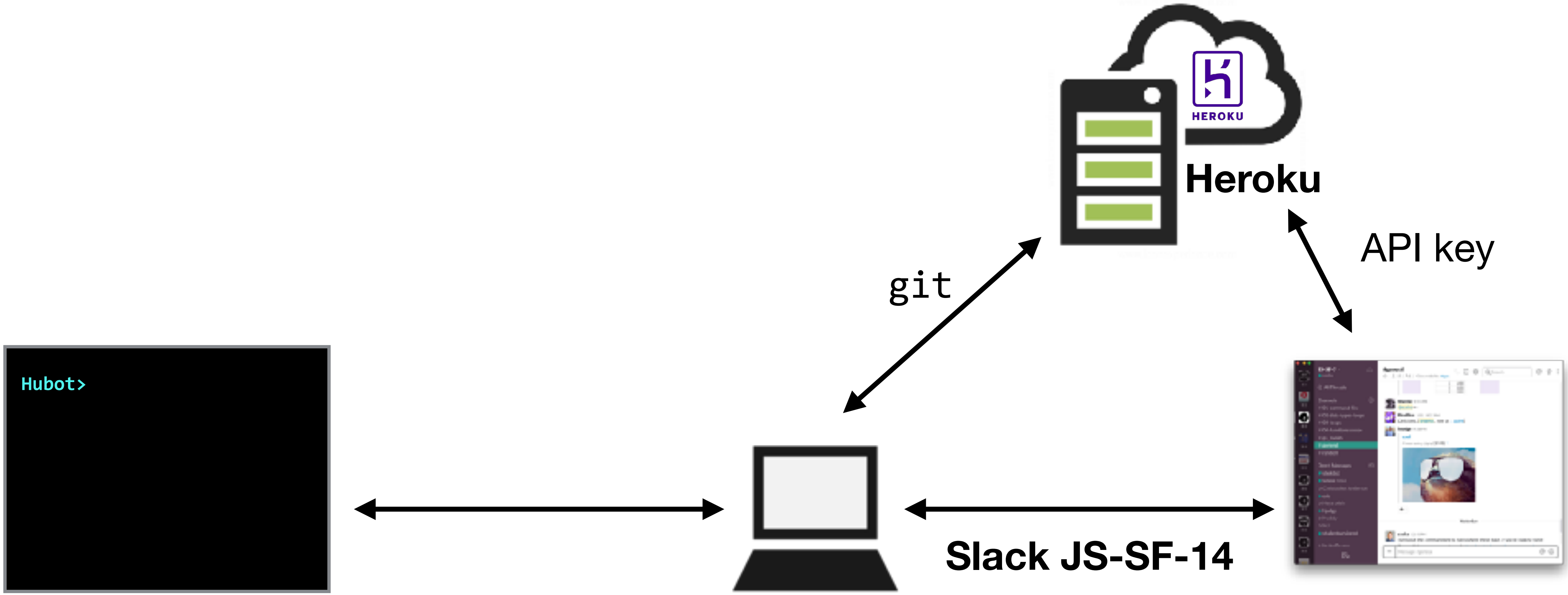
HUBOT vs SLACK BOT vs SLACKBOT

- 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



```
Hubot>
```

Heroku

HEROKU

git

API key

Slack JS-SF-14

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.

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

MARKDOWN

- **Markdown**: A markup language used for creating formatted text documents.
- Easier to use than HTML for basic tasks
- Comes in different flavors; GitHub has its own
- Used to create README files that document projects in GitHub repos
- You will use Markdown to create a README file explaining what your bot does and how to use it

ACTIVITY — HUBOT CONFIGURATION

ACTIVITY

KEY OBJECTIVE

- Install and configure all utilities to run a Hubot

LOCATION

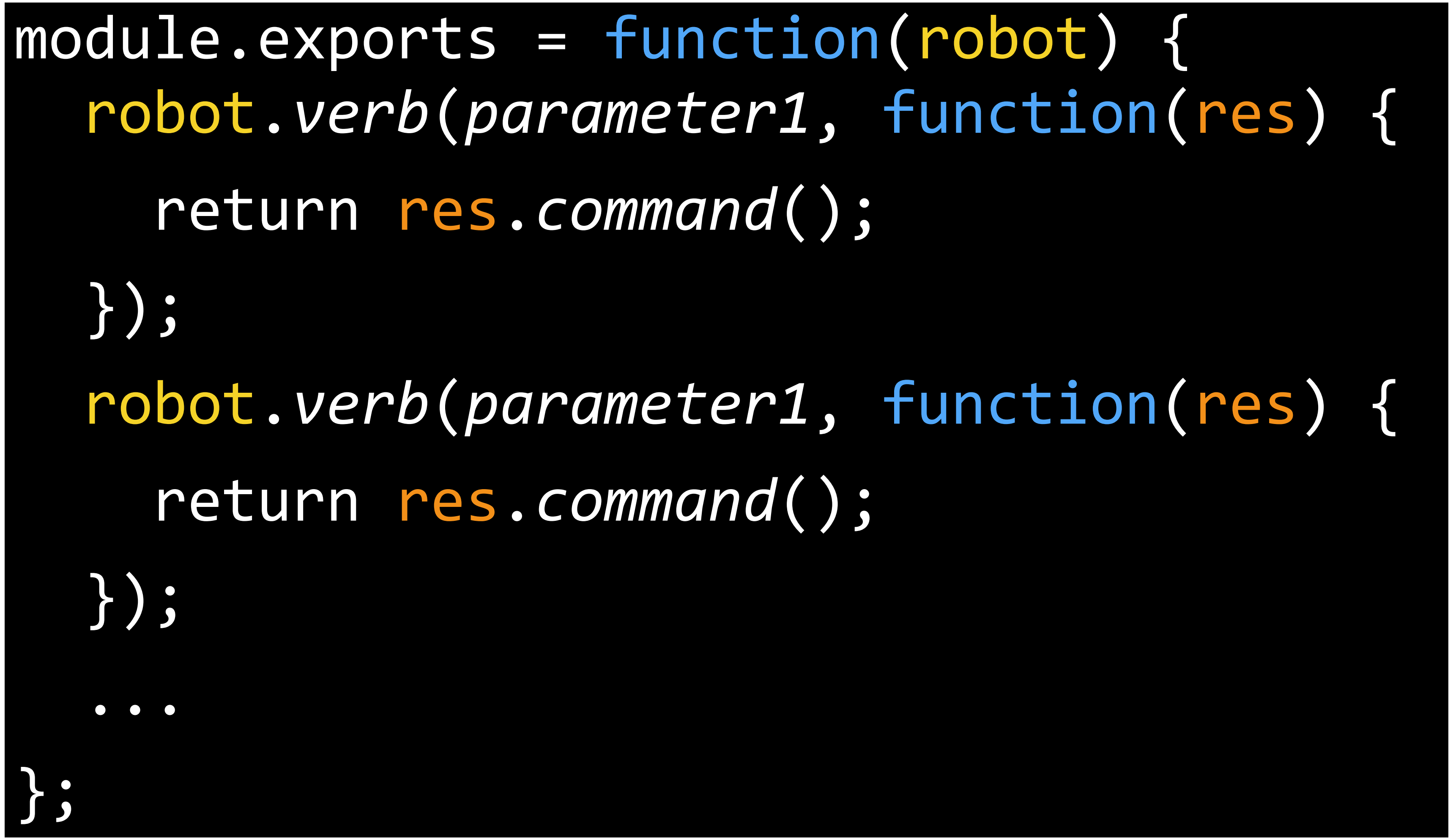
- Slack Bot Lab - Install Guide
(first link in Resources on website for today’s class)

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 Slack Bot Lab - Sample Code (second link in Resources on website for today’s class)

UNDERSTANDING THE HUBOT FRAMEWORK

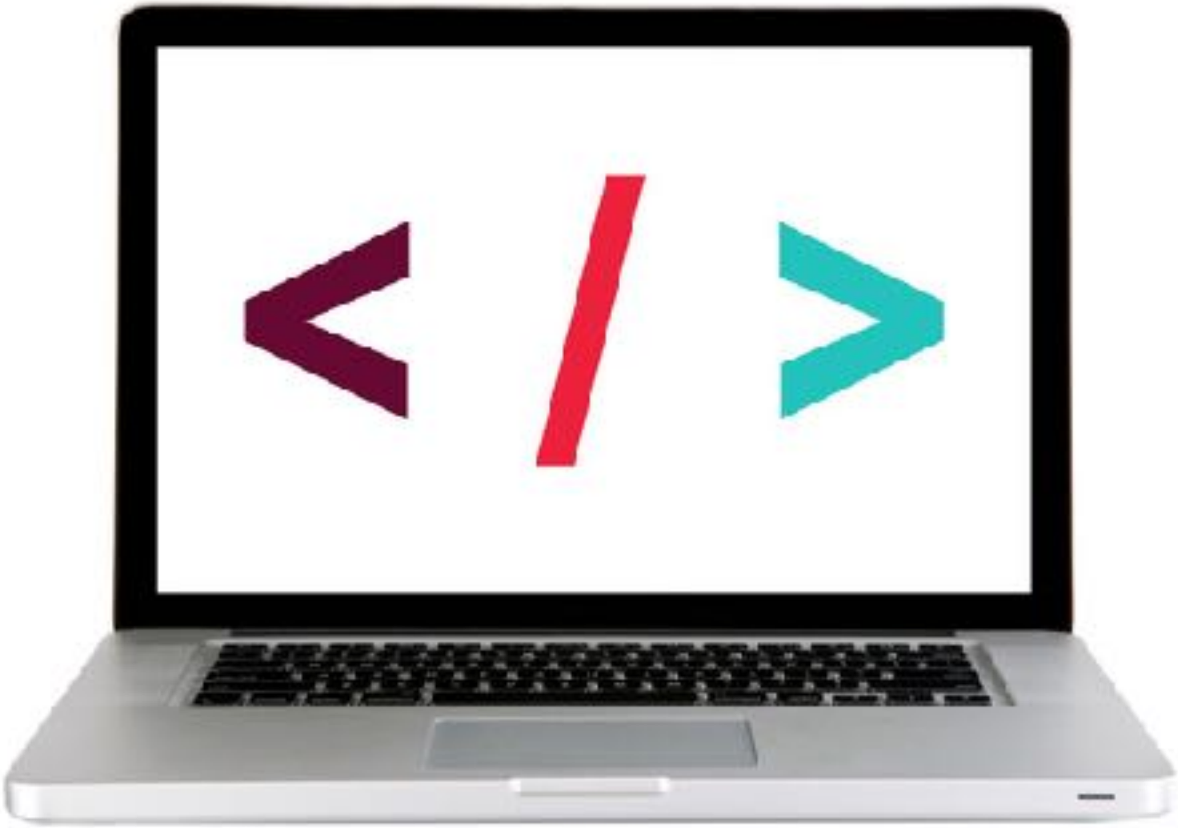


```
module.exports = function(robot) {
  robot.verb(parameter1, function(res) {
    return res.command();
  });
  robot.verb(parameter1, function(res) {
    return res.command();
  });
  ...
};
```

BASIC HUBOT VERBS

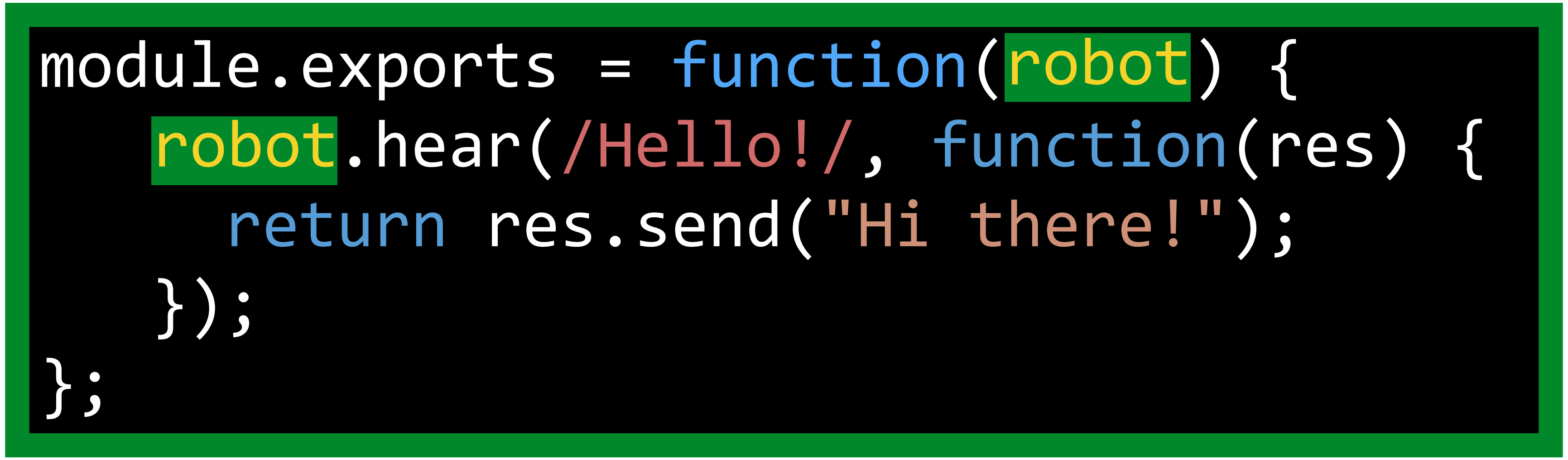
- `hear`: called anytime a message's text matches
- `respond`: called for messages immediately preceded by the robot's name or alias

LET'S TAKE A CLOSER LOOK

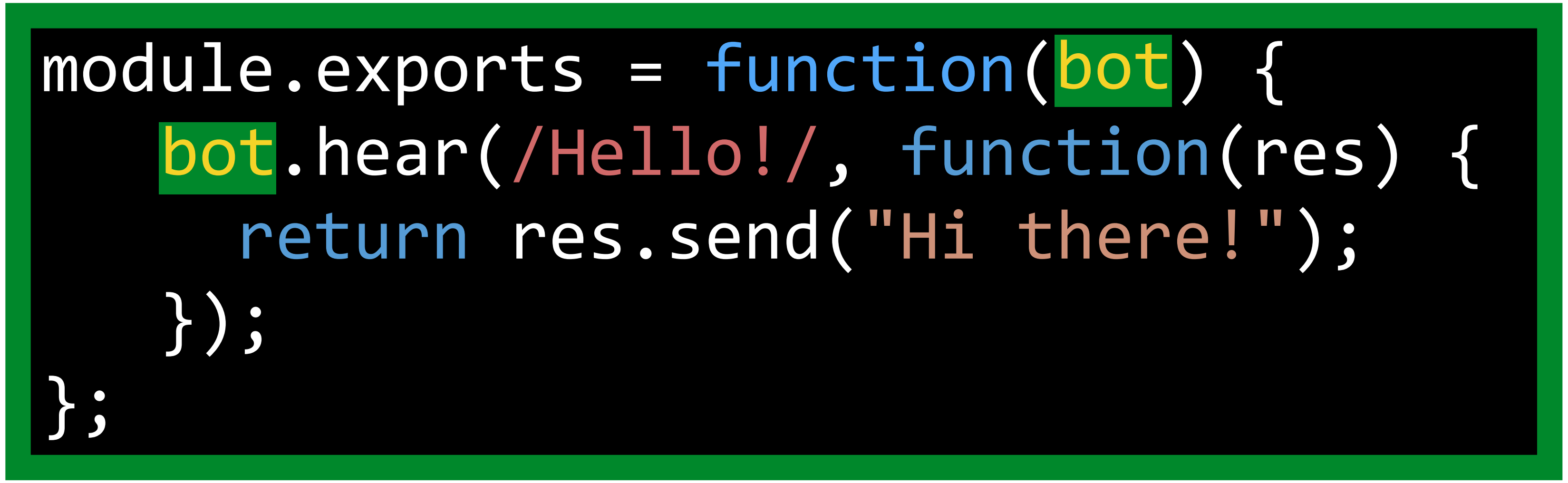


COMMON GOTCHAS

```
module.exports = function(robot) {
   bot.hear(/Hello!/, function(res) {
     return res.send("Hi there!");
   });
};
```



```
module.exports = function(robot) {
   robot.hear(/Hello!/, function(res) {
     return res.send("Hi there!");
   });
};
```



```
module.exports = function(bot) {
   bot.hear(/Hello!/, function(res) {
     return res.send("Hi there!");
   });
};
```

```
module.exports = function(bot) {
   bot.respond(/Hello!/, function(res) {
     return res.send("Hi there!");
   });
};
```

```
thunderbot> Hello!
thunderbot>
```

```
module.exports = function(bot) {
   bot.hear(/Hello!/, function(res) {
     return res.send("Hi there!");
   });
};
```

```
thunderbot> Hello!
thunderbot> Hi there!
```

```
module.exports = function(bot) {
   bot.respond(/Hello!/, function(res) {
     return res.send("Hi there!");
   });
};
```

```
thunderbot> @thunderbot Hello!
thunderbot> Hi there!
```

```
module.exports = function(bot) {
   bot.hear(/Hello!/, function(res) {
     return res.send("Hi there!");
   });
};
module.exports = function(bot) {
   bot.hear(/Yo/, function(res) {
     return res.send("Heya");
   });
};
```

```
thunderbot> Hello!
thunderbot>
thunderbot> Yo
thunderbot> Heya
```

```
module.exports = function(bot) {
    bot.hear(/Hello!/, function(res) {
      return res.send("Hi there!");
    });
    bot.hear(/Yo/, function(res) {
      return res.send("Heya");
    });
};
```

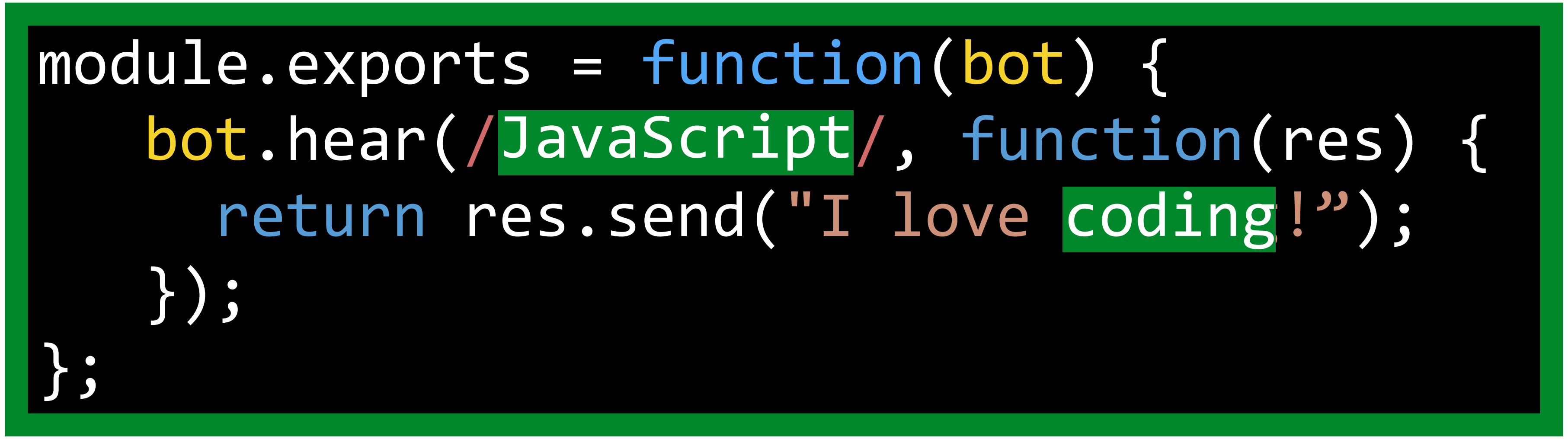
```
thunderbot> Hello!
thunderbot> Hi there!
thunderbot> Yo
thunderbot> Heya
```

SLACK BOT LAB

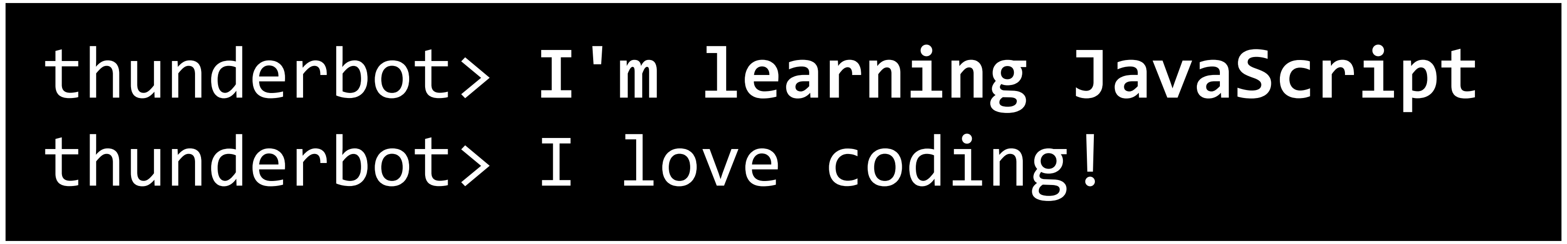
```
module.exports = function(bot) {
   bot.hear(/JavaScript/, function(res) {
     return res.send("I love JavaScript!");
   });
};
```

```
thunderbot> I'm learning JavaScript
thunderbot> I love JavaScript!
thunderbot> I love JavaScript!
thunderbot> I love JavaScript!
thunderbot> I love JavaScript!
```

```
thunderbot> I love JavaScr
thunderbot> I love JavaScr
thunderbot> I love JavaScr
thunderbot> I love JavaScr
bot> I love JavaScr
bot> I love JavaScr
bot> I love JavaScr
bot> I love JavaScr
thunderbot> I love JavaScr
thunderbot> I love JavaScr
thunderbot> I love JavaScr
thunderbot> I love JavaScr
thunderbot> I love JavaScr
thunderbot> I love JavaScr
```



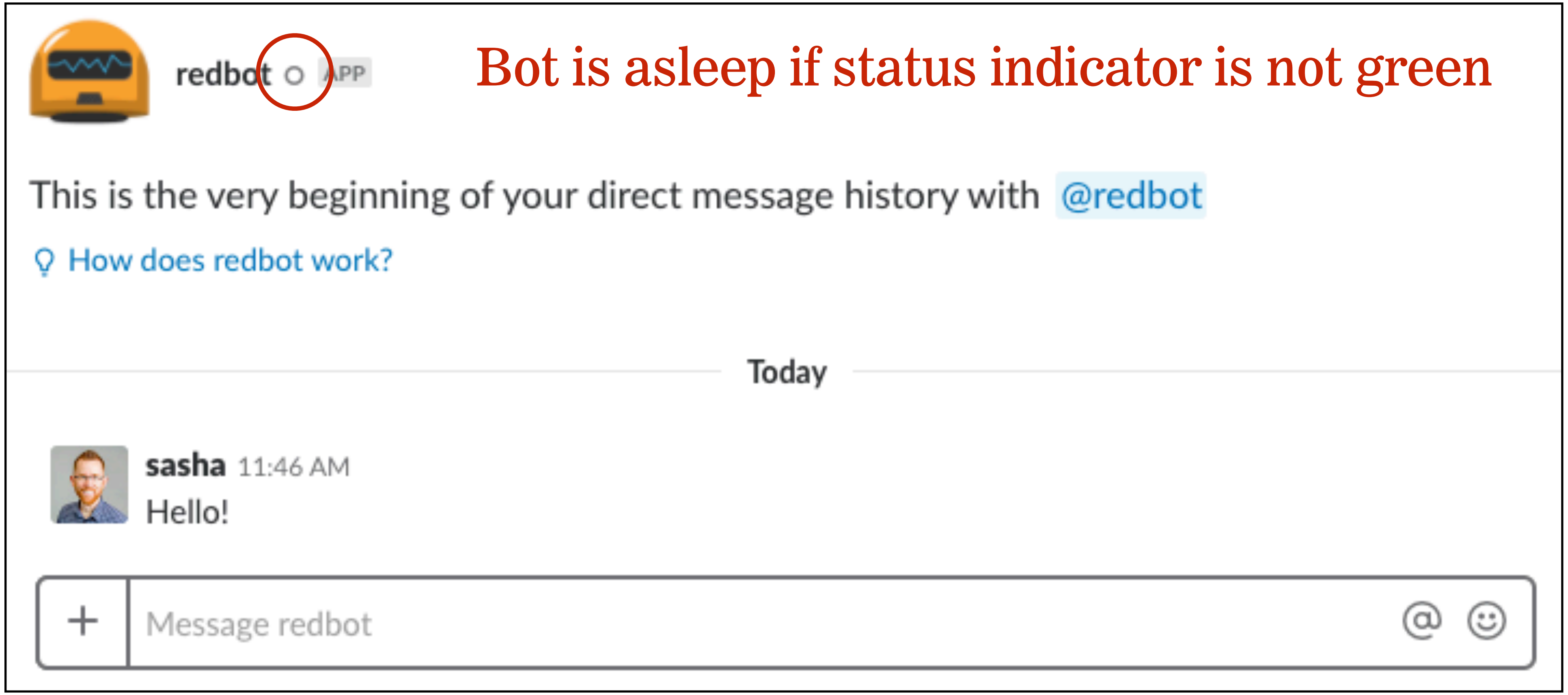
```
module.exports = function(bot) {
   bot.hear(/JavaScript/, function(res) {
     return res.send("I love coding!”);
   });
};
```



```
thunderbot> I'm learning JavaScript
thunderbot> I love coding!
```

```
module.exports = function(bot) {
   bot.respond(/JavaScript/, function(res) {
     return res.send("I love JavaScript!");
   });
};
```

```
thunderbot> @thunderbot I’m learning JavaScript
thunderbot> I love JavaScript!
```



redbot ○ APP

Bot is asleep if status indicator is not green

This is the very beginning of your direct message history with @redbot

How does redbot work?

Today

sasha 11:46 AM
Hello!

```
$ heroku ps:restart
```

DO NOT ADD YOUR API TOKEN TO ANY OF YOUR FILES AS A COMMENT!

LAB — BUILD A SLACK BOT

KEY OBJECTIVE

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

LOCATION

- `JSD > myhubot > scripts > script.js`

EXECUTION

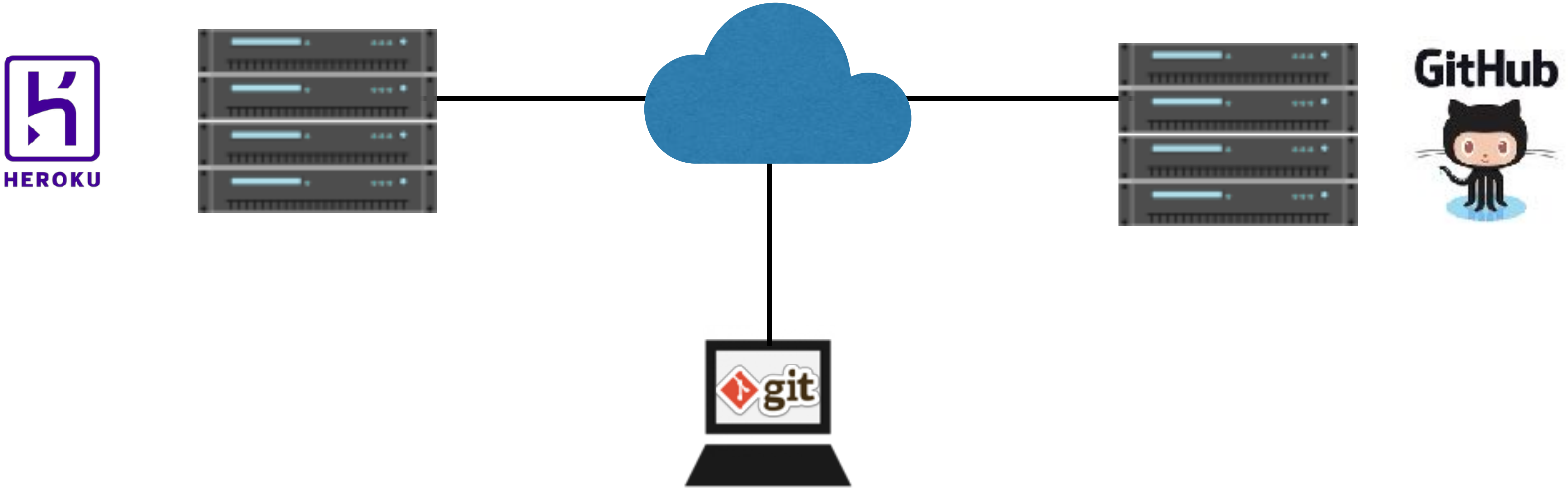
Until 9:20

1. Uncommenting portions of the sample code in `script.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 today’s start code files.
3. Create a plan for what you want your bot to be able to do, pseudocode it, and start building it!
4. Test using the steps in Slack bot lab - Testing & Troubleshooting (third link on class resources on website)
5. BONUS: Experiment with advanced commands documented at https://github.com/github/hubot/blob/master/docs/scripting.md

CREATING A GITHUB ENTERPRISE REPO

Heroku vs GitHub

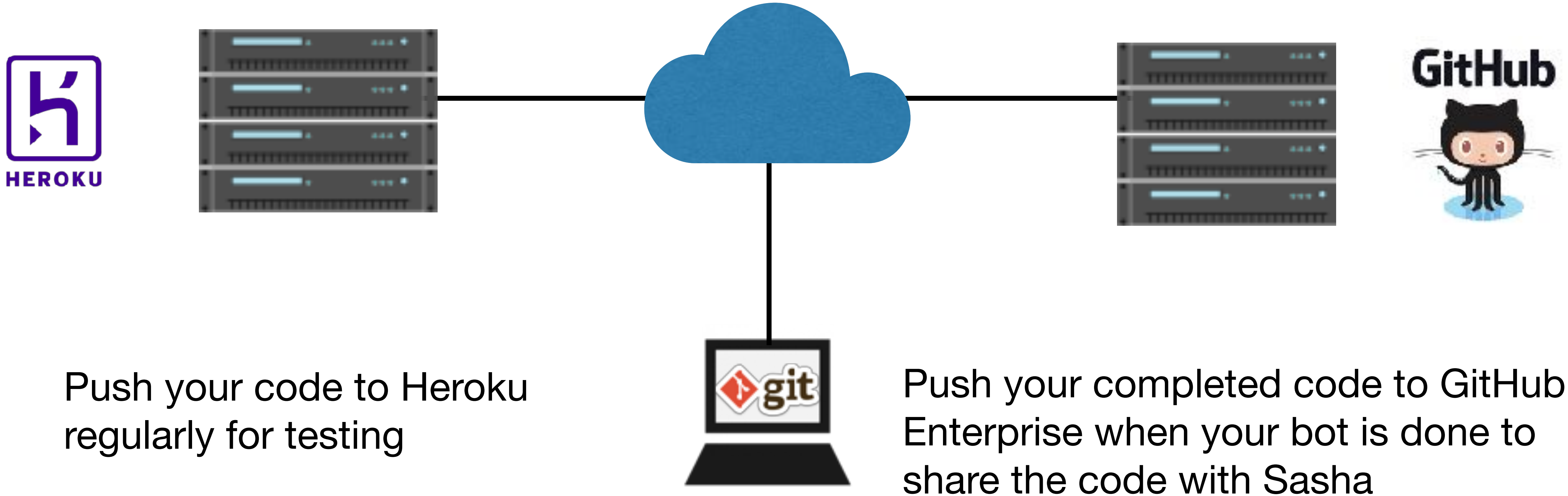
- **Heroku** is running your bot code in the cloud
- **GitHub** is making your raw code available to Sasha



Heroku vs GitHub

```
git push heroku master
```

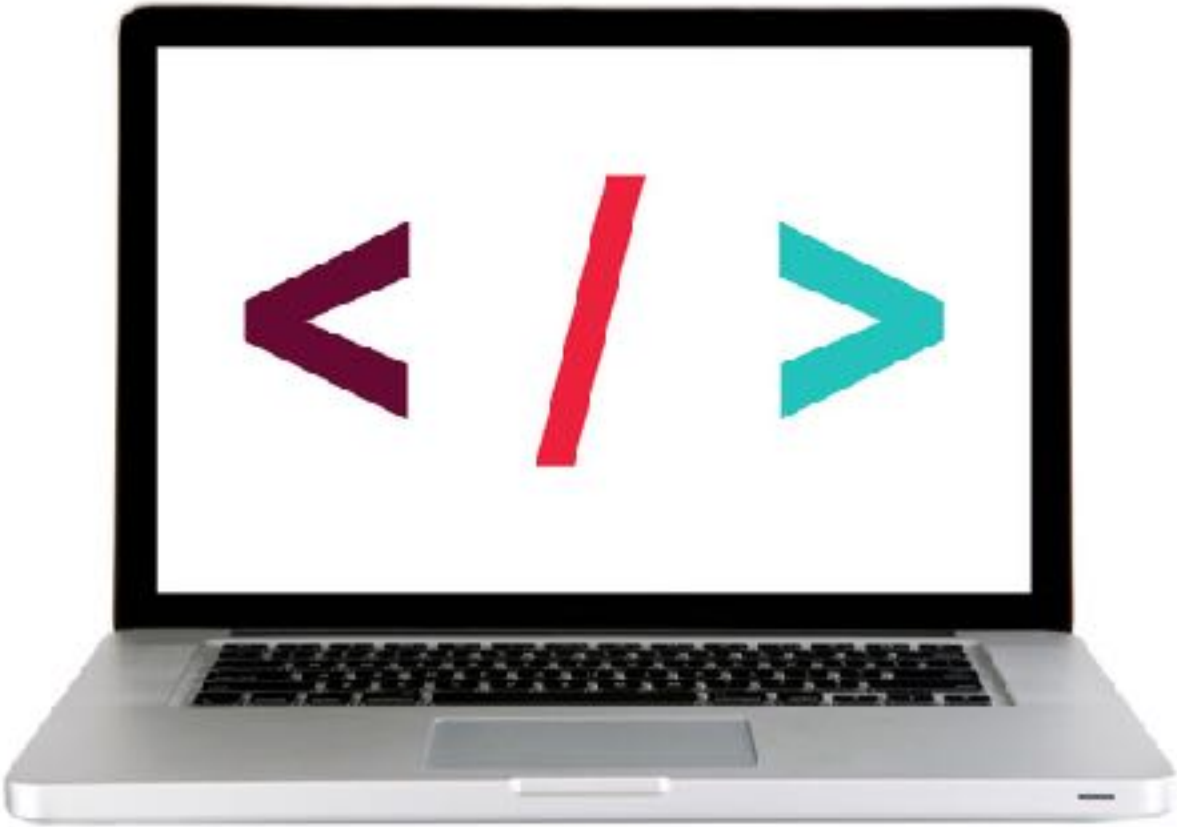
```
git push origin master
```



Push your code to Heroku regularly for testing

Push your completed code to GitHub Enterprise when your bot is done to share the code with Sasha

LET'S TAKE A CLOSER LOOK



Exit Tickets!

(Class #5)

LEARNING OBJECTIVES - REVIEW

- Install and configure all utilities needed to build a bot using the Hubot framework
- Write scripts that allow your bot to interact with users of the class Slack organization

NEXT CLASS PREVIEW

JSON & Intro to the DOM

- Implement and interface with JSON data
- Identify differences between the DOM and HTML.
- Explain the methods and use the DOM in JavaScript.

Q&A