GENERAL ASSEMBLY

JAVASCRIPT DEVELOPMENT

Sasha Vodnik, Instructor

JAVASCRIPT DEVELOPMENT

THE COMMAND LINE & DATA TYPES

WEEKLY OVERVIEW

WEEK 1	Installfest / The Command Line & Data Types
WEEK 2	Loops & Arrays / Conditionals & Functions
WEEK 3	Scope & Objects / Slack bot lab

LEARNING OBJECTIVES

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

- Write pseudocode and explain how it relates to programmatic thinking.
- Work with files/directories via the terminal window
- Create a Git repository and push/pull changes
- Run basic JavaScript code on the command line
- Describe the concept of a "data type" and how it relates to variables.

AGENDA

- Pseudocoding
- JS and web technology
- The terminal
- Git and GitHub
- Command line JS
- Data type

EXIT TICKET QUESTIONS

1. Is my git config correct?
2. Why did we use Homebrew instead of NPM to install Git?
3. Node.js…. what are you? I am sure we will learn
4. What applications (git, node.js, npm etc.) need to be running when we are writing code?
5. Why is npm necessary? Can we just install packages from invidiual websites?
6. How to understand the command line

Think about last class:

- We installed software from the command line by typing commands
- We also installed software by downloading an installer, double-clicking it, and following the prompts

ACTIVITY

EXERCISE

KEY OBJECTIVE

- Use the most common commands to navigate and modify files / directories via the terminal window.

TYPE OF EXERCISE

- Turn and Talk

TIMING

2 min

1. List at least 2 advantages to using the command line.
2. List at least 2 disadvantages to using the command line.

PSEUDOCODE

THINKING LIKE A PROGRAMMER

- **What is a program?**
  - A program is a set of instructions that tells a computer how to carry out a task
- **What is programming?**
  - Programming is the task of writing those instructions in a language that a computer can understand
- **What's the first step in becoming a programmer?**
  - Not learning a particular language, but learning how to think like a computer

PSEUDOCODE

- An outline of a program that can be converted into code
- The process of writing pseudocode helps you through a program, step-by-step, without actually writing a line of code
- Allows a programmer to focus on problem solving, not the precise layout of the code and its syntax
- Don’t need to know how to code to write pseudocode

PSEUDO CODE

- When we write a program, we need to figure out a way to translate the ideas that are in our heads into code
- Pseudo code is a way to 'plan out' your program before coding it
- **Pseudo code** is a *detailed yet readable description* of what a computer program must do
- Expressed in plain English rather than in a programming language

PSEUDOCODE — THE IMPORTANCE OF PLANNING

PSEUDOCODE — HEIGHT COMPARISON

</ />

CODE ALONG

PSEUDOCODE — PASSING SCORE

CODE ALONG

LAB — PSEUDOCODE

EXERCISE

KEY OBJECTIVE

- Write pseudocode and explain how it relates to programmatic thinking.

TYPE OF EXERCISE

- Pairs

TIMING

5 min	1. Create pseudocode for a program that calculates the number of miles a user travels between home and work (or another destination) per year. 2. Take into account distance between home and destination, times per day the user makes that trip (probably 2), and working days per year.

ACTIVITY

EXERCISE

KEY OBJECTIVE

- Explain how pseudocode relates to programmatic thinking.

TYPE OF EXERCISE

- Turn and Talk

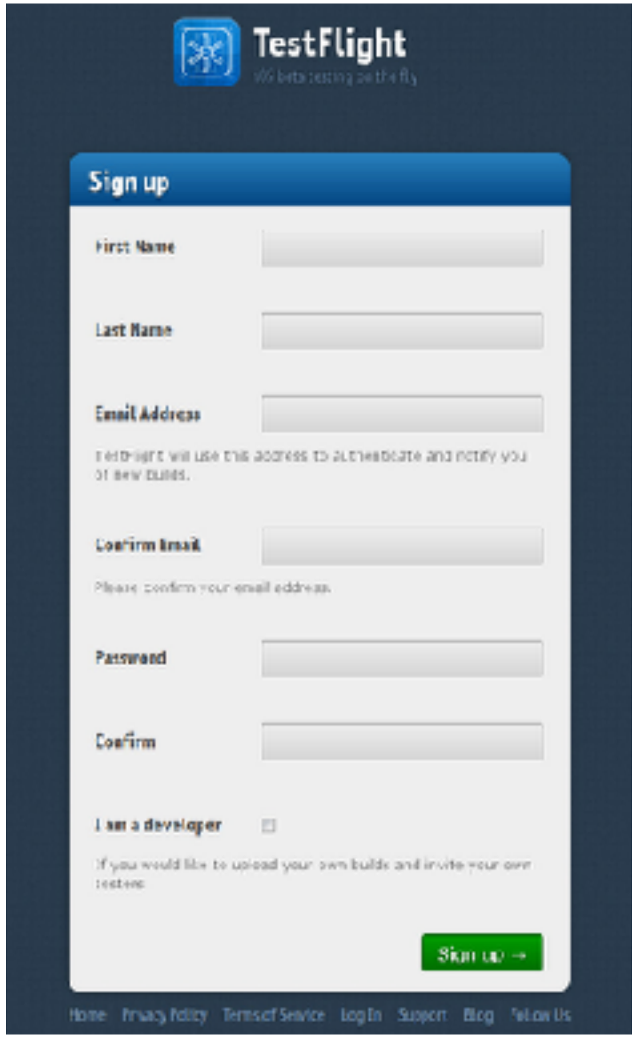
TIMING

4 min

1. Describe pseudocode in your own words.
2. Explain what programmatic thinking is, and how it relates to pseudocode.

JavaScript & Web Technology

WHAT CAN JAVASCRIPT DO?



TestFlight
iOS beta testing on the fly

Sign up

First Name

Last Name

Email Address

TestFlight will use this address to authenticate and notify you of new builds.

Confirm Email

Please confirm your email address.

Password

Confirm

I am a developer ☐

If you would like to upload your own builds and invite your own testers

Sign up →

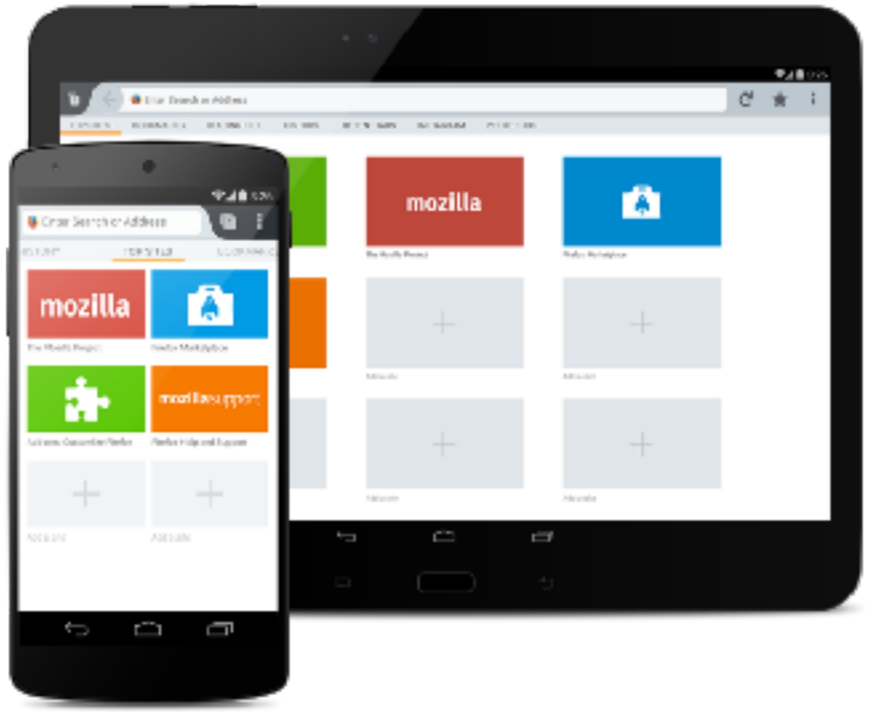
Home Privacy Policy Terms of Service Log In Support Blog Follow Us

front end tasks (animations, buttons, forms)

JS

APIs, databases, back end tasks

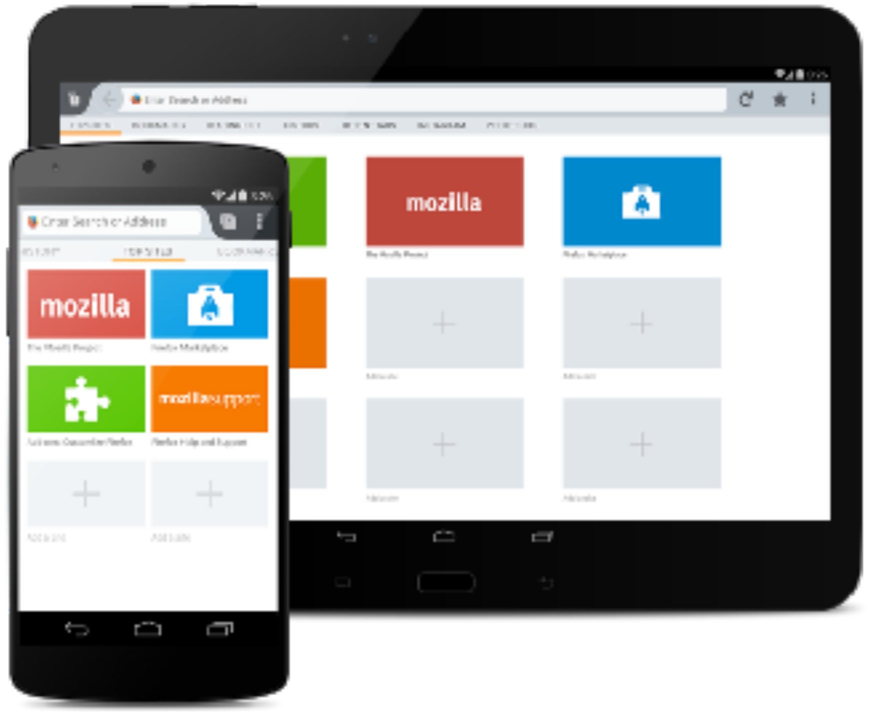
VERY FEW STEPS TO RUN



runs directly in browser within an HTML file

also runs in node.js

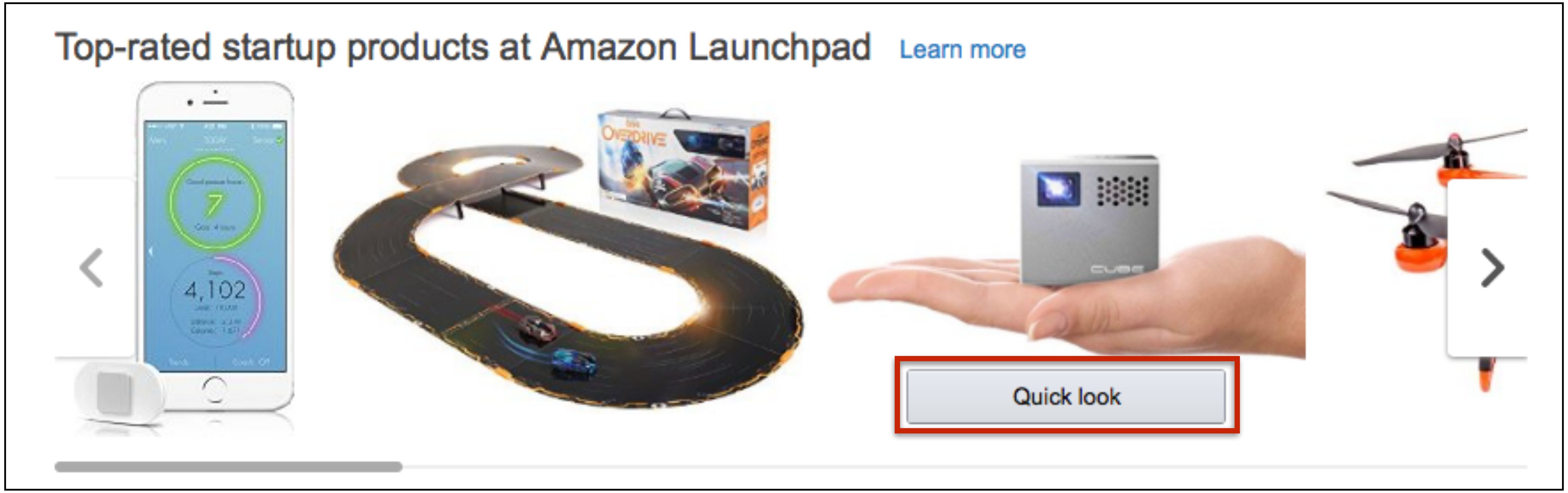
AND WORKS EVEN WHEN COMPUTERS ARE OFFLINE



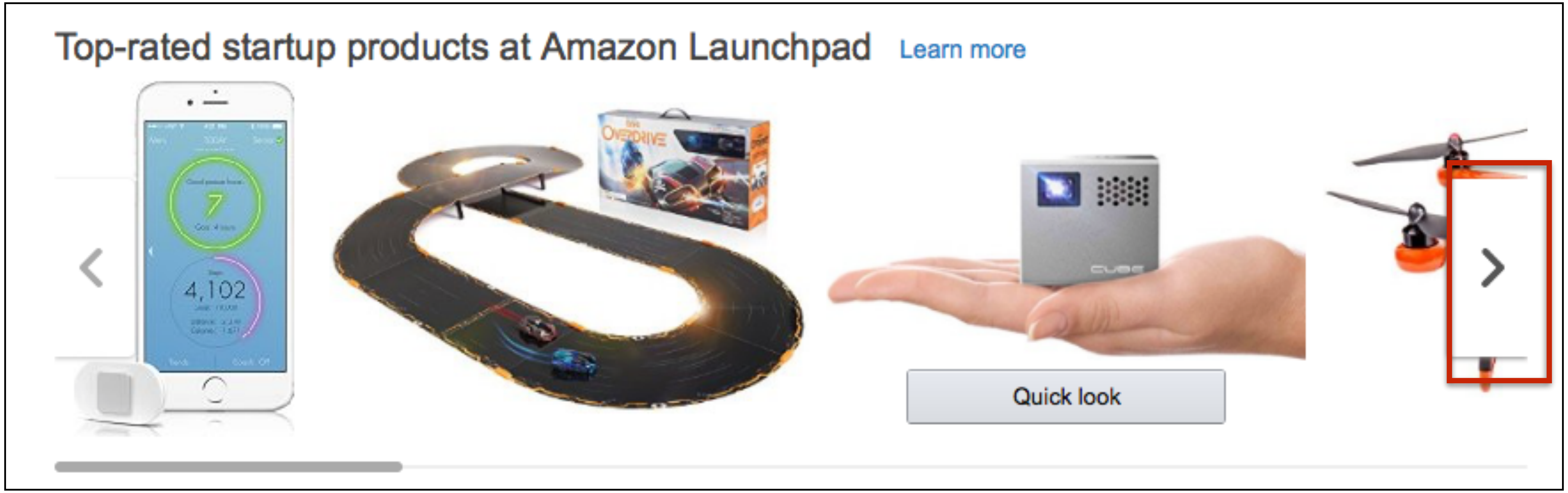
runs directly in browser within an HTML file

also runs in node.js

HIGHLY RESPONSIVE INTERFACES



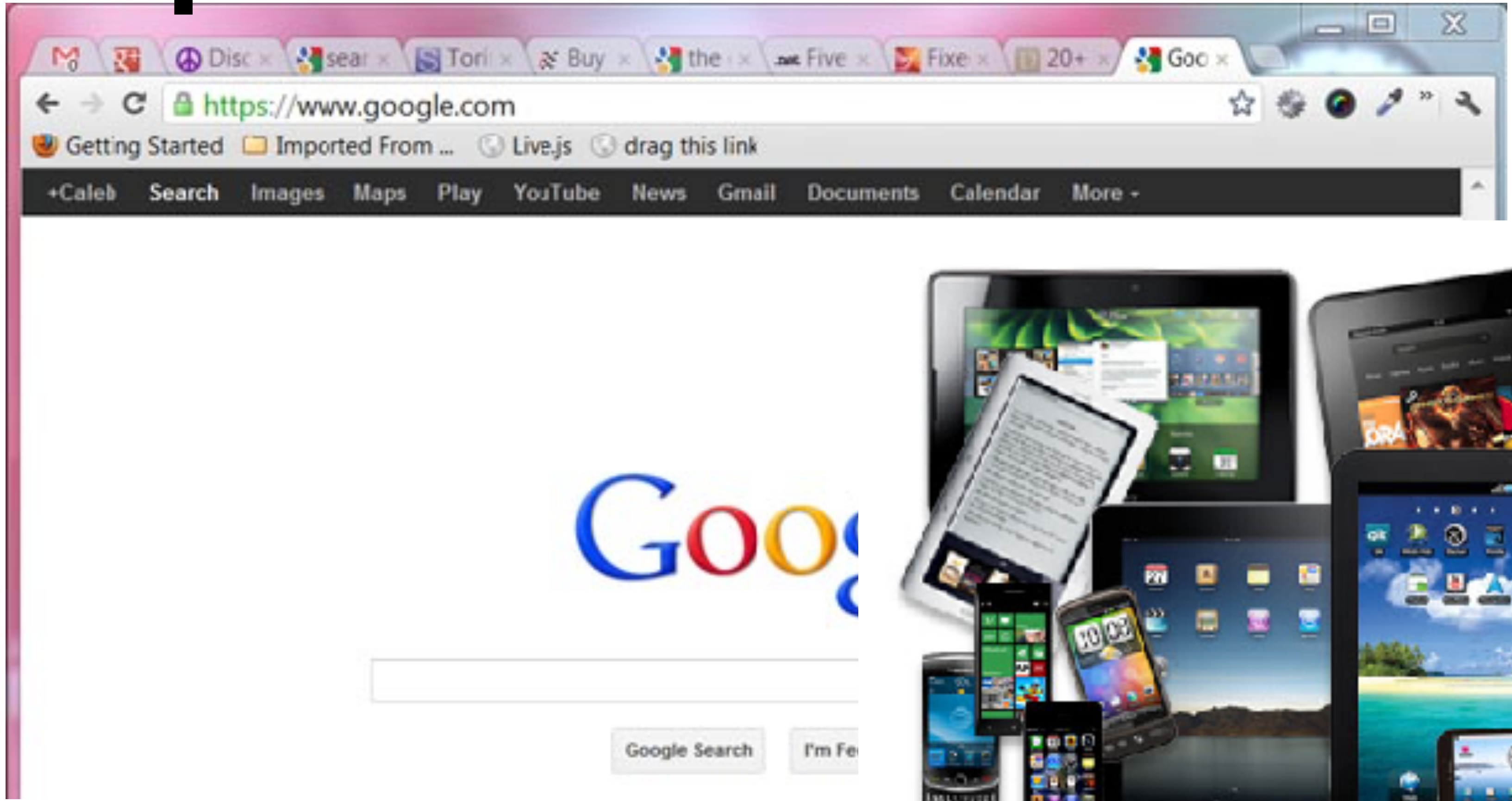
LOAD ADDITIONAL CONTENT WHEN USER NEEDS IT (AJAX)



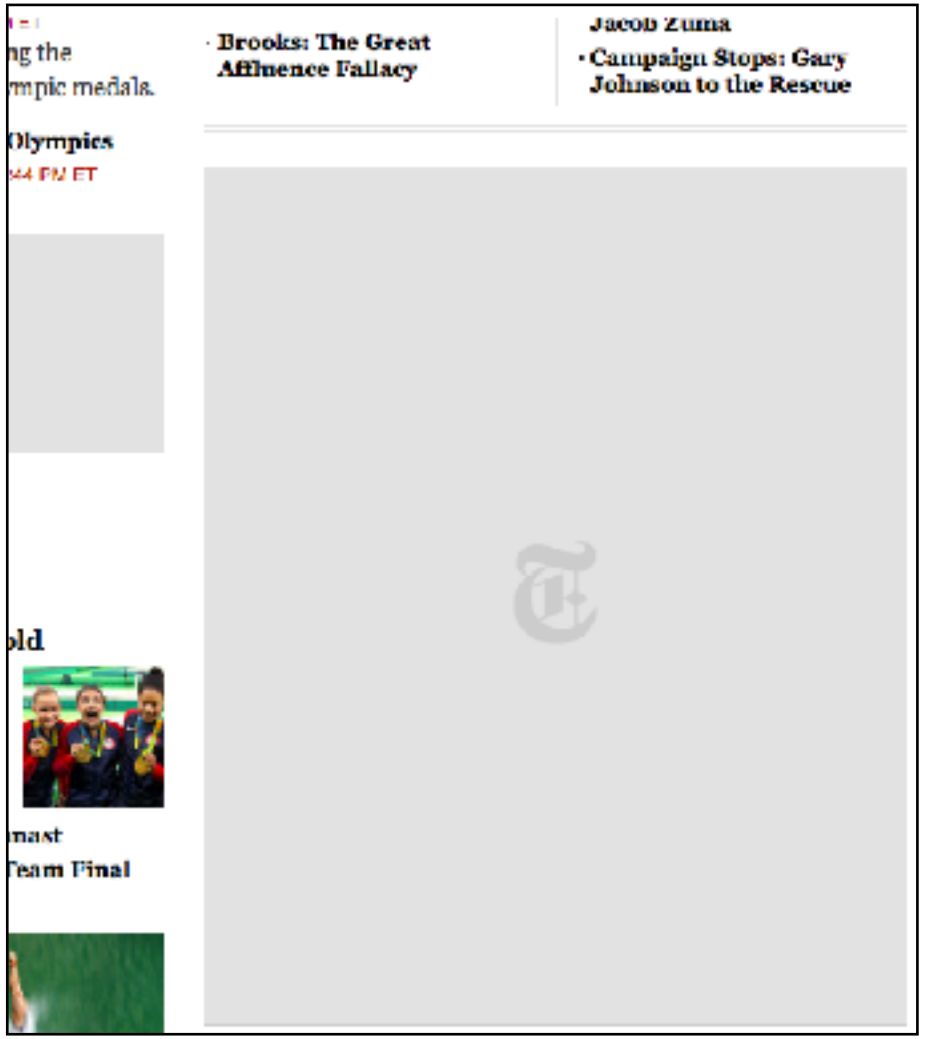
WHAT ELSE CAN JAVASCRIPT DO?

- Determine your browser functional limitations and react accordingly (progressive enhancement)
- Power website backends and physical devices (node.js)

DRAWBACK: The environment in which JavaScript operates is unknown



DRAWBACK: JavaScript can be disabled


ng the
mpic medals.

Olympics
44 PM ET

old

nast
Team Final

· Brooks: The Great Affluence Fallacy

Jacob Zuma

· Campaign Stops: Gary Johnson to the Rescue

Node.js

Node.js

- A definition (from Wikipedia):
    - In software development, Node.js is an open-source, cross-platform runtime environment for developing server-side Web applications.
- Enables JavaScript on the server (the backend)
- Written in C, C++, and JS (so, not a JS framework)
- Interprets JS using Chrome's V8 engine
- Module driven; see Node Package Manager (npm)
- All about non-blocking, asychronous input/output

Node.js

- We will not be using Node.js as a web server (backend) - see Firebase
- We will be taking advantage of Node's command line interface
- Allows us to run JavaScript from our terminal applications
- More at the end of class...

JavaScript Frameworks & Libraries

A Library

- Set of predefined functions that your code calls
- Each call performs work and returns a result (and control) to your code
- Specific, well-defined operations
- Example: jQuery

A Framework

- Opinionated architecture for building software
- Control-flow exists, you fill in with your code
- Calls your code; is always in control
- Examples: React, Angular, Vue, Ember

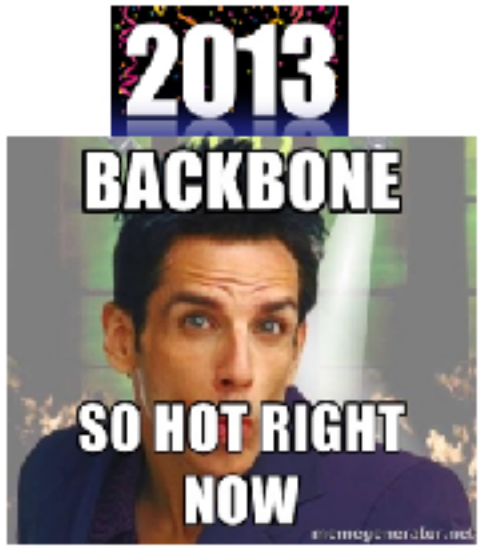
Libraries vs Frameworks

- The primary difference (source):
  - You call library
  - Framework calls you
- Please Note:
  - JSD focuses on the foundations of JavaScript as a programming language
  - We will be using the jQuery library
  - Opportunity towards class end for a framework intro

2007
GWT
SO HOT RIGHT NOW
memegenerator.net

2009
EXTJS
SO HOT RIGHT NOW
memegenerator.net

2012
KNOCKOUT
SO HOT RIGHT NOW
memegenerator.net


2013
BACKBONE
SO HOT RIGHT NOW
memegenerator.net

2014
ANGULAR
SO HOT RIGHT NOW
memegenerator.net

2015
REACT
SO HOT RIGHT NOW
memegenerator.net

The Terminal

INTRODUCTION TO THE TERMINAL

- Terminal allows you to interact with your computer faster
- Terminal === Command Line === Console

UNIX

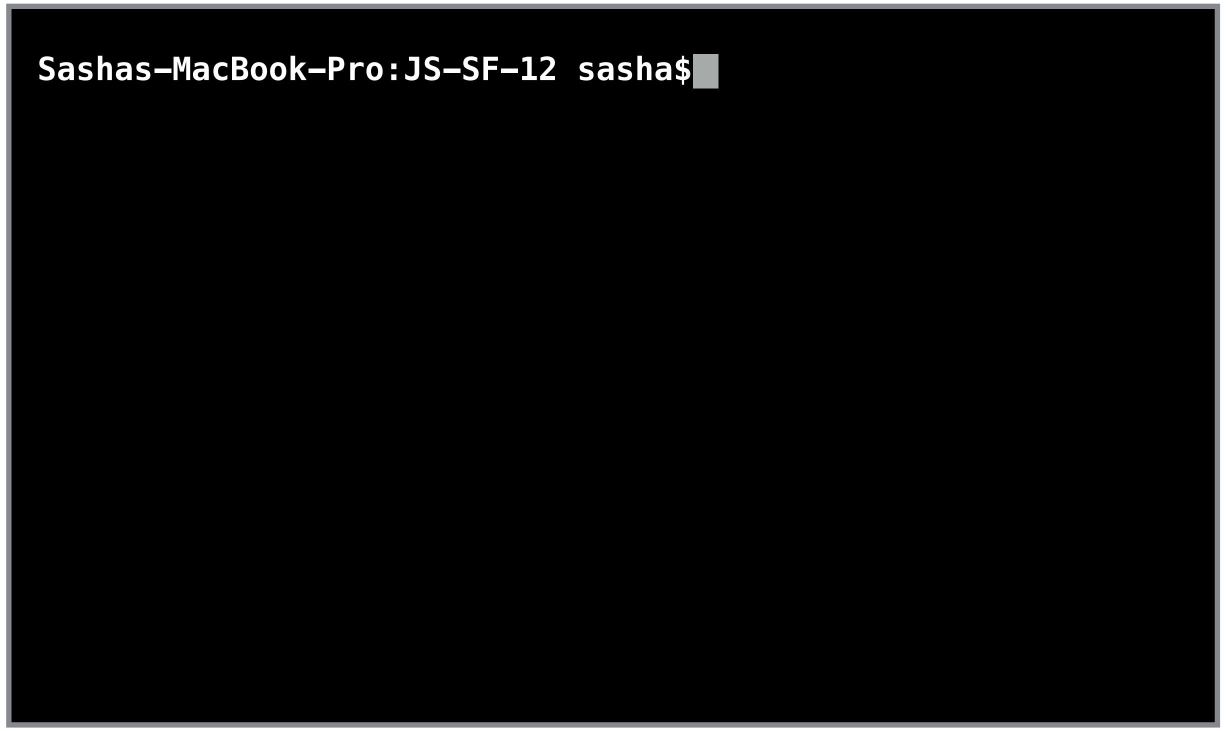
UNIX

- Family of operating systems, including all Linux systems and OS X/macOS

SHELL

- A generic name for the primary program that runs inside a terminal

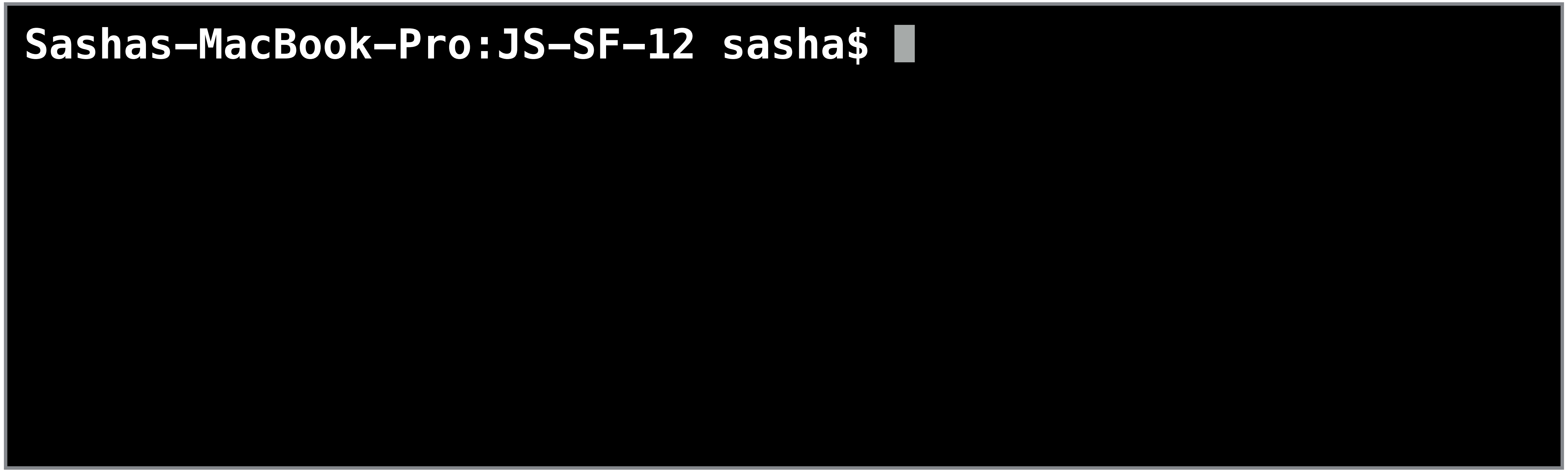
BASH



```
Sashas-MacBook-Pro:JS-SF-12 sasha$
```

- Bourne-Again Shell: a specific shell program

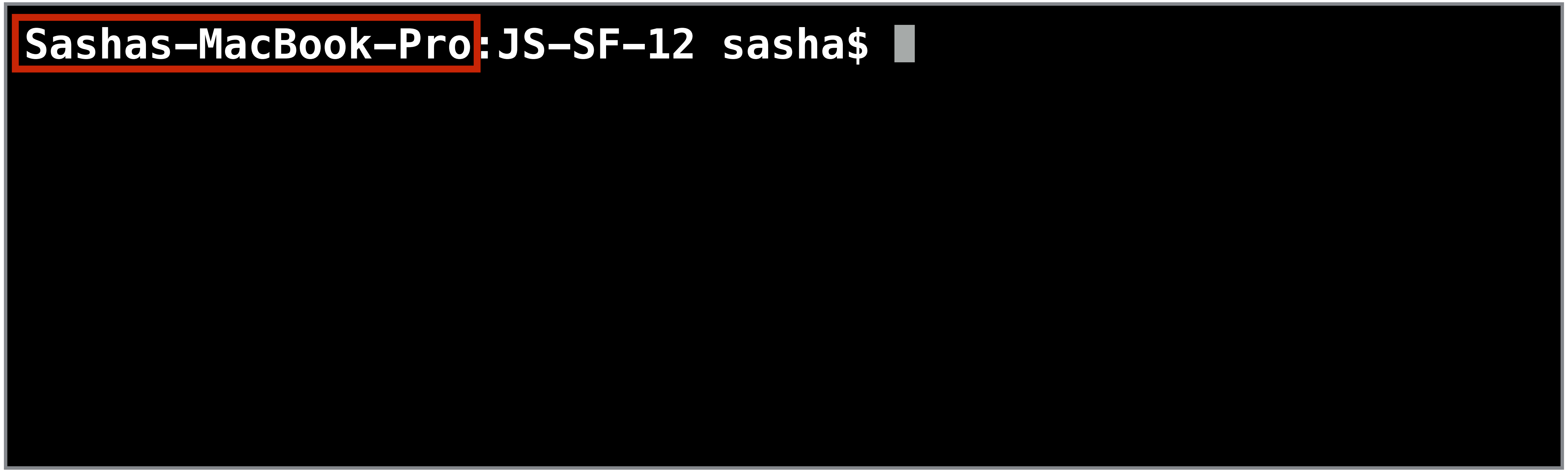
ANATOMY OF THE TERMINAL



```
Sashas-MacBook-Pro:JS-SF-12 sasha$
```

ANATOMY OF THE TERMINAL

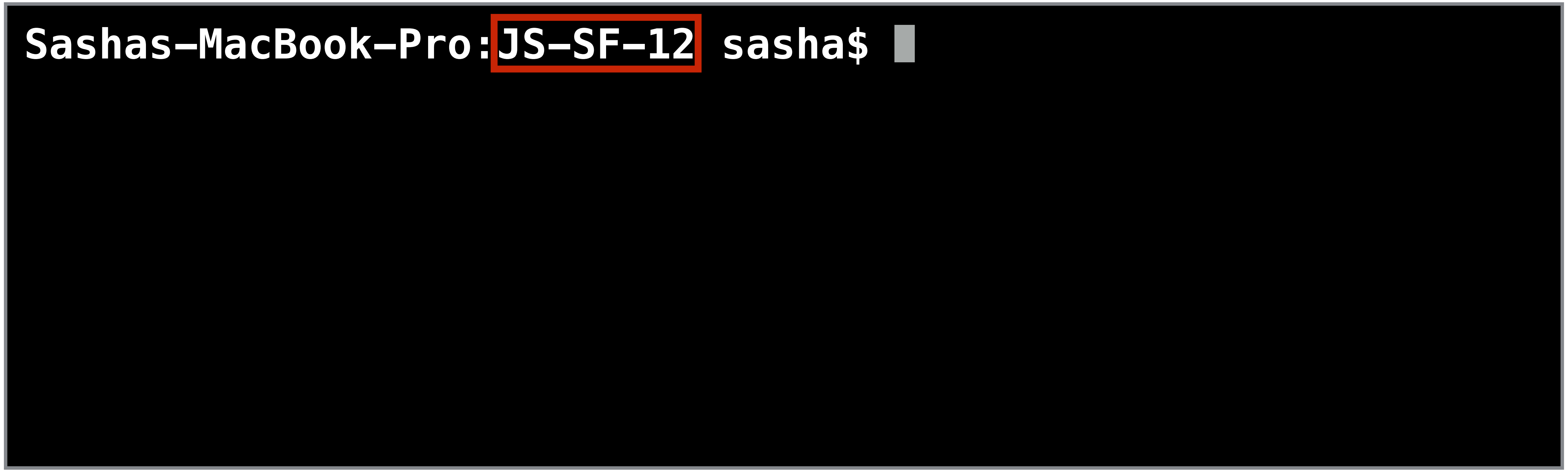
Host (computer) name



```
Sashas-MacBook-Pro:JS-SF-12 sasha$
```

ANATOMY OF THE TERMINAL

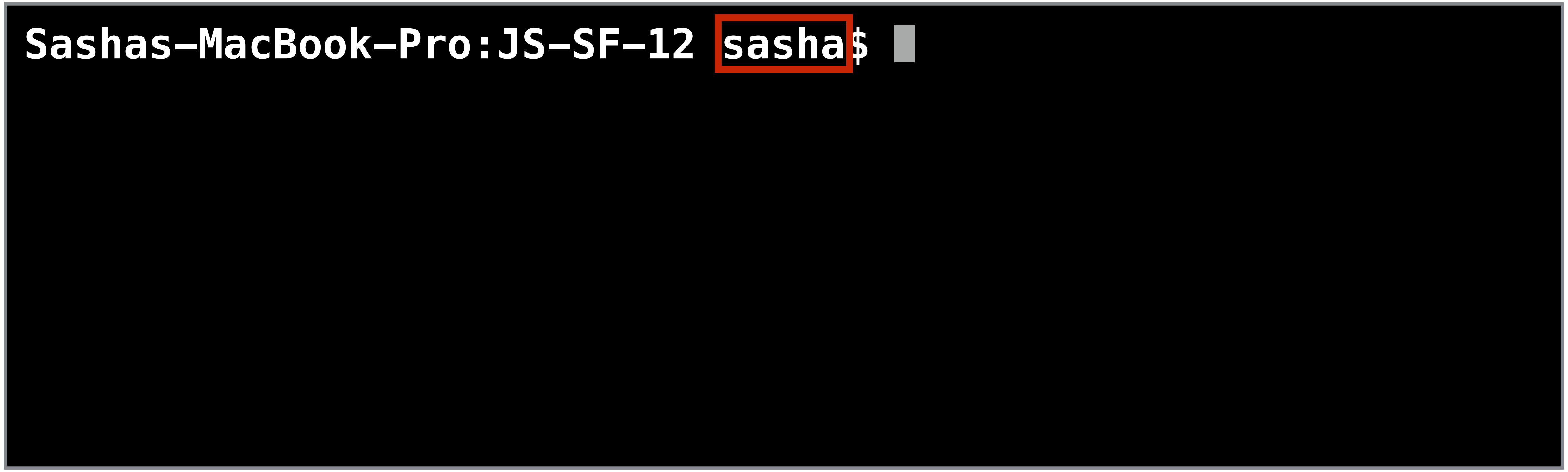
Working directory (current folder)



```
Sashas-MacBook-Pro:JS-SF-12 sasha$ 
```

ANATOMY OF THE TERMINAL

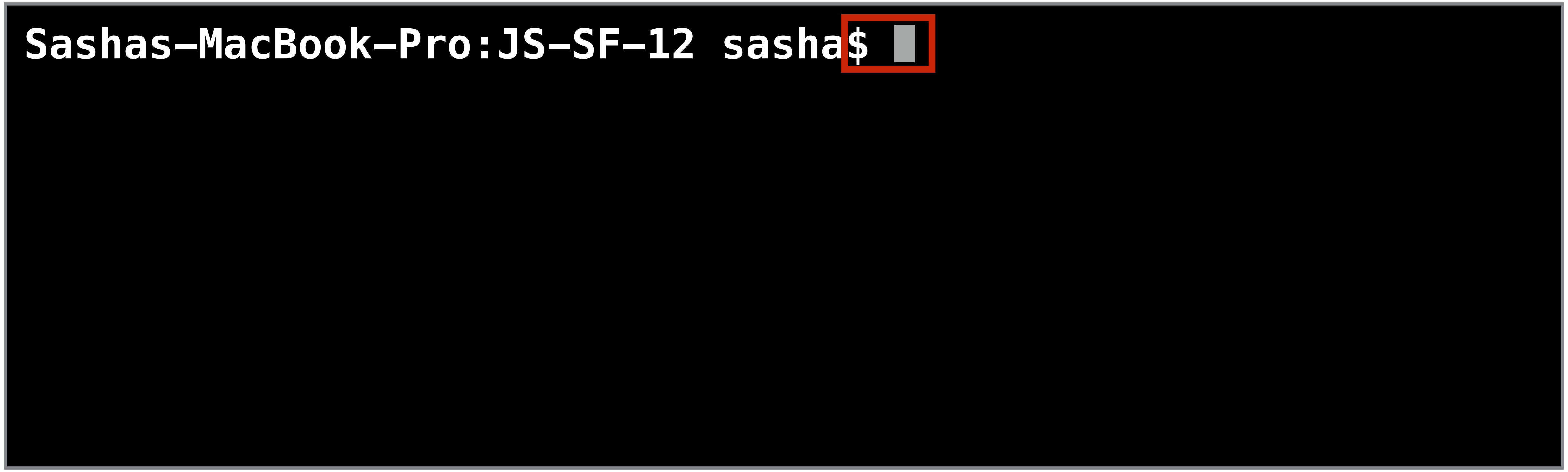
Username



```
Sashas-MacBook-Pro:JS-SF-12 sasha$
```

ANATOMY OF THE TERMINAL

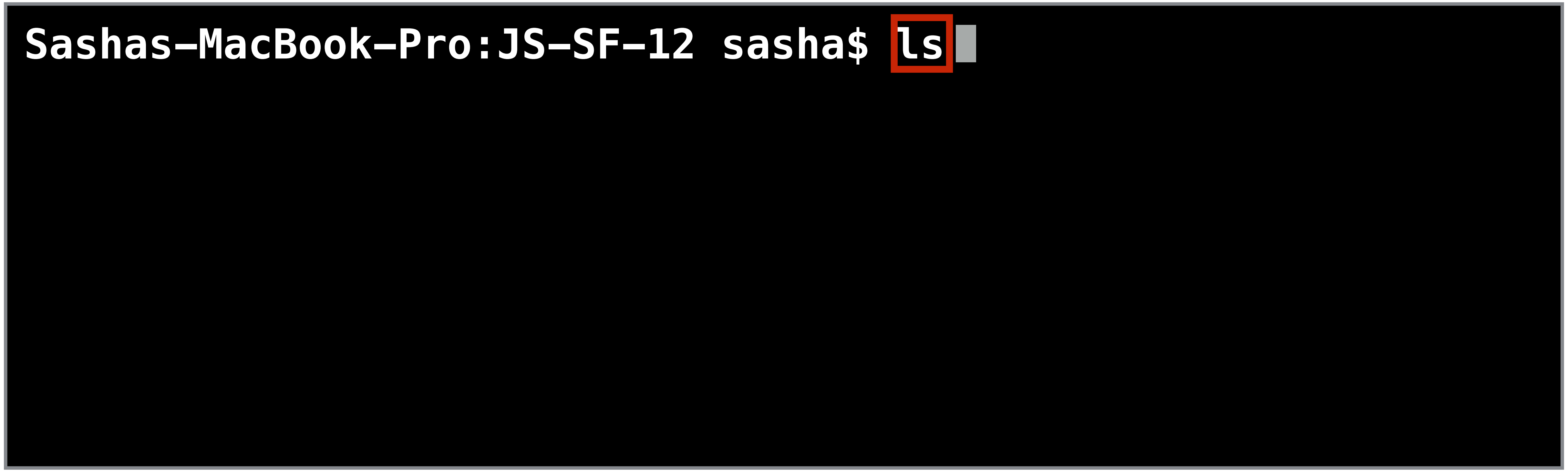
Bash prompt



```
Sashas-MacBook-Pro:JS-SF-12 sasha$ 
```

ANATOMY OF THE TERMINAL

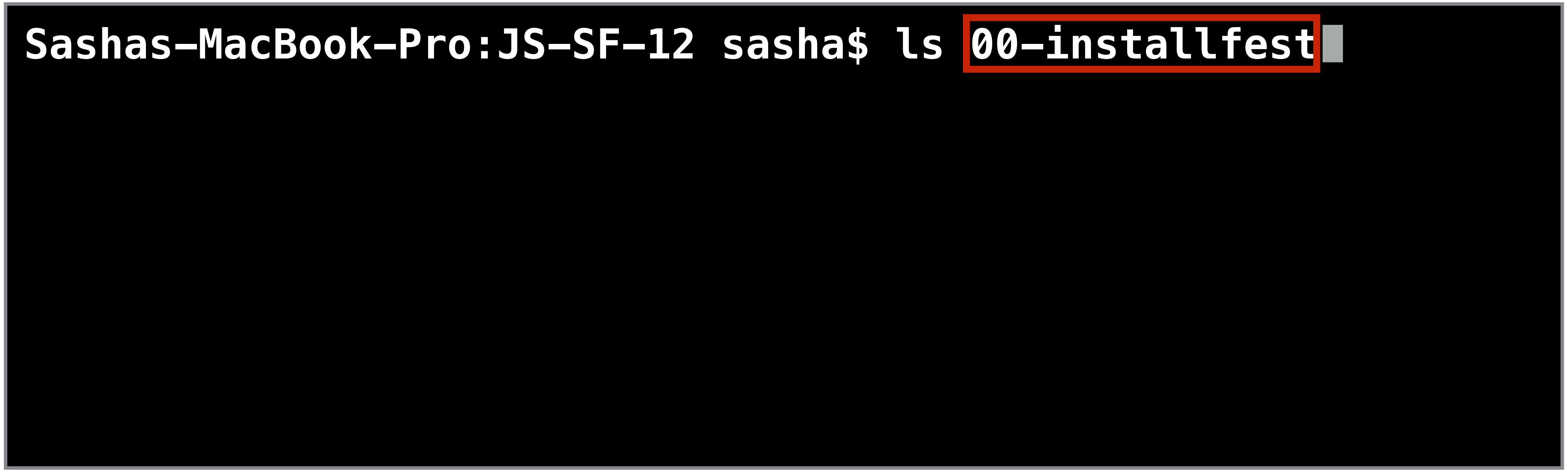
Command (program)



```
Sashas-MacBook-Pro:JS-SF-12 sasha$ ls
```

ANATOMY OF THE TERMINAL

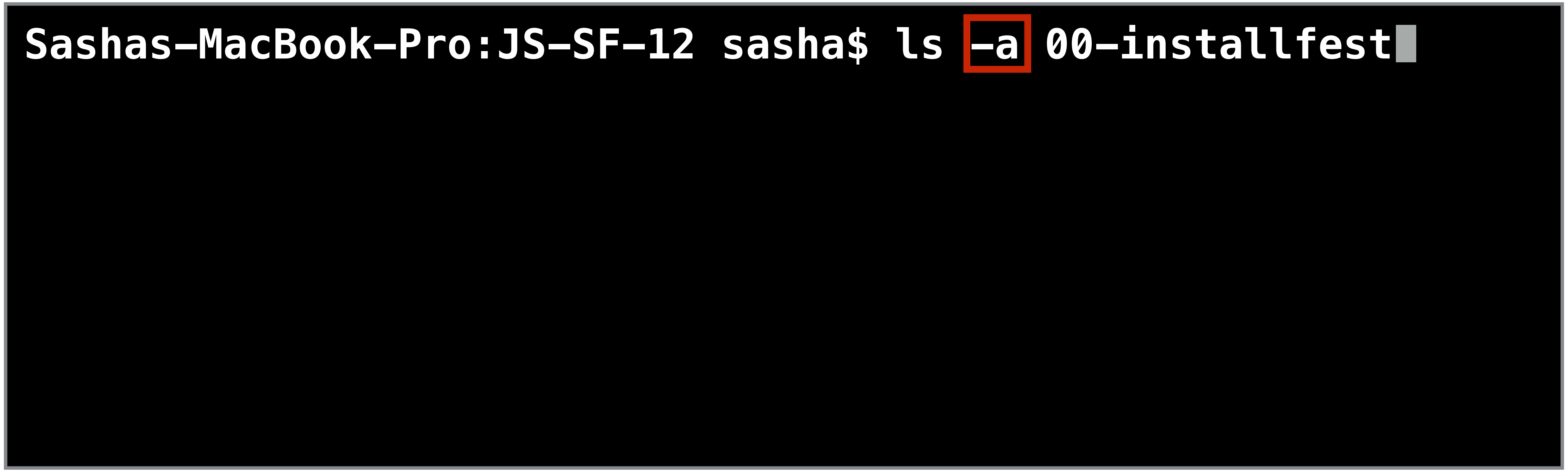
Argument (input)



```
Sashas-MacBook-Pro:JS-SF-12 sasha$ ls 00-installfest
```

ANATOMY OF THE TERMINAL

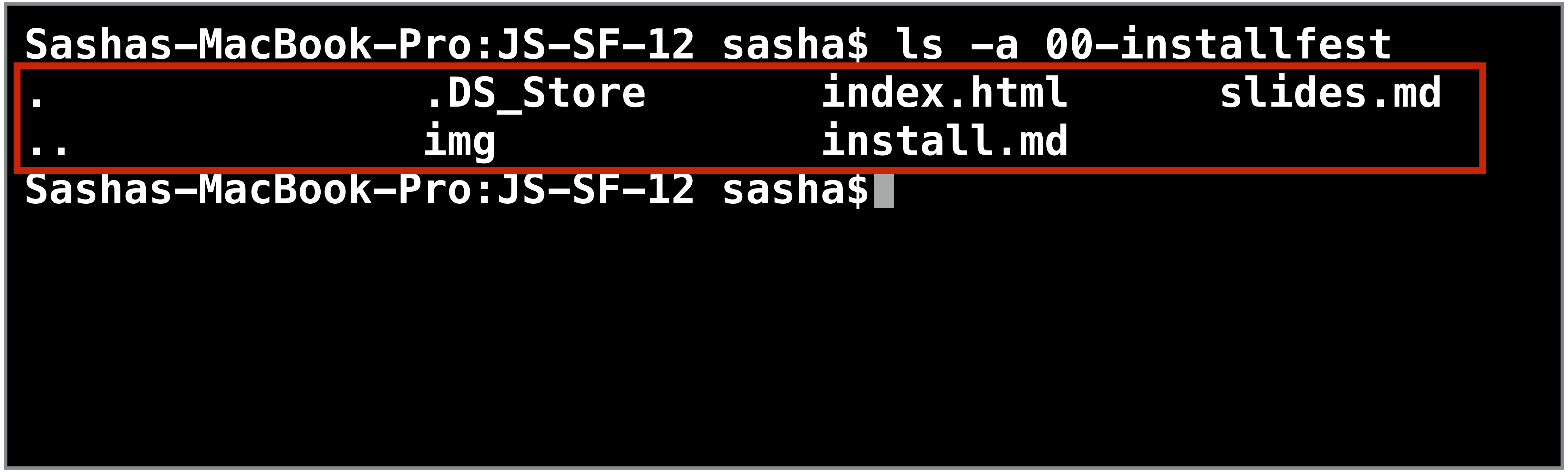
Option



```
Sashas-MacBook-Pro:JS-SF-12 sasha$ ls -a 00-installfest
```

ANATOMY OF THE TERMINAL

Output



```
Sashas-MacBook-Pro:JS-SF-12 sasha$ ls -a 00-installfest
.                .DS_Store        index.html       slides.md
..               img              install.md
Sashas-MacBook-Pro:JS-SF-12 sasha$
```

THE COMMAND LINE & DATA TYPES

Command line codealong

For Mac

Open the Terminal app (Applications > Utilities > Terminal)

For Windows

Open the PowerShell application

LAB — COMMAND LINE

EXERCISE

KEY OBJECTIVE

- Use the most common commands to navigate and modify files / directories via the terminal window.

TYPE OF EXERCISE

- Individual/Pairs

TIMING

10 min	Follow the instructions posted on the class website to navigate and modify files and directories using the command line.

EXERCISE — COMMAND LINE

EXERCISE

KEY OBJECTIVE

- Use the most common commands to navigate and modify files / directories via the terminal window.

TYPE OF EXERCISE

- Whole class brainstorm

TIMING

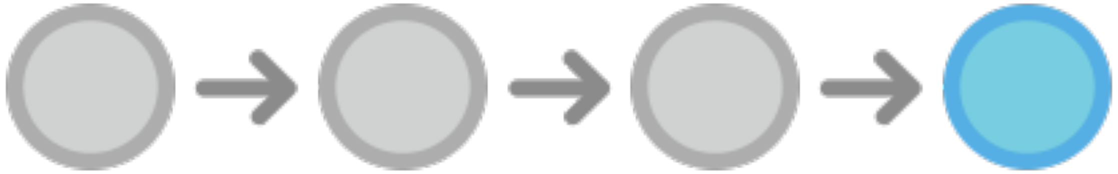
2 min

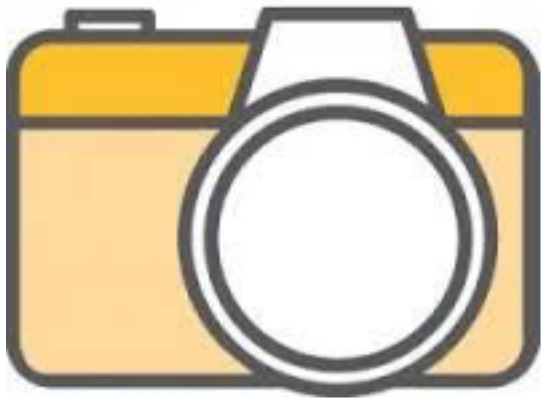
1. Name a command line command and explain what it does. Let’s hear from everyone at least once!

Introduction to Git/GitHub

GIT

- A **version control** program that saves the state of your project's files and folders
- Basically, it takes a "snapshot" of what all your files look like at a moment and stores a reference to that "snapshot"





GITHUB

- A **web app/platform** that makes it easy to manage git repositories.
- Similar to Dropbox or Google Drive, but for code.
- Stores a history of files and the changes that happen within each changed document.
- Hosts files on the cloud so you can share the finished product with other people.
- **Git** - the technology that Github is based on top of - was designed to allow for multiple engineers to work on the same project.

Why use GitHub?

HISTORY

- Since GitHub stores a history of the code, it allows developers to go back in time if something breaks.

COLLABORATION

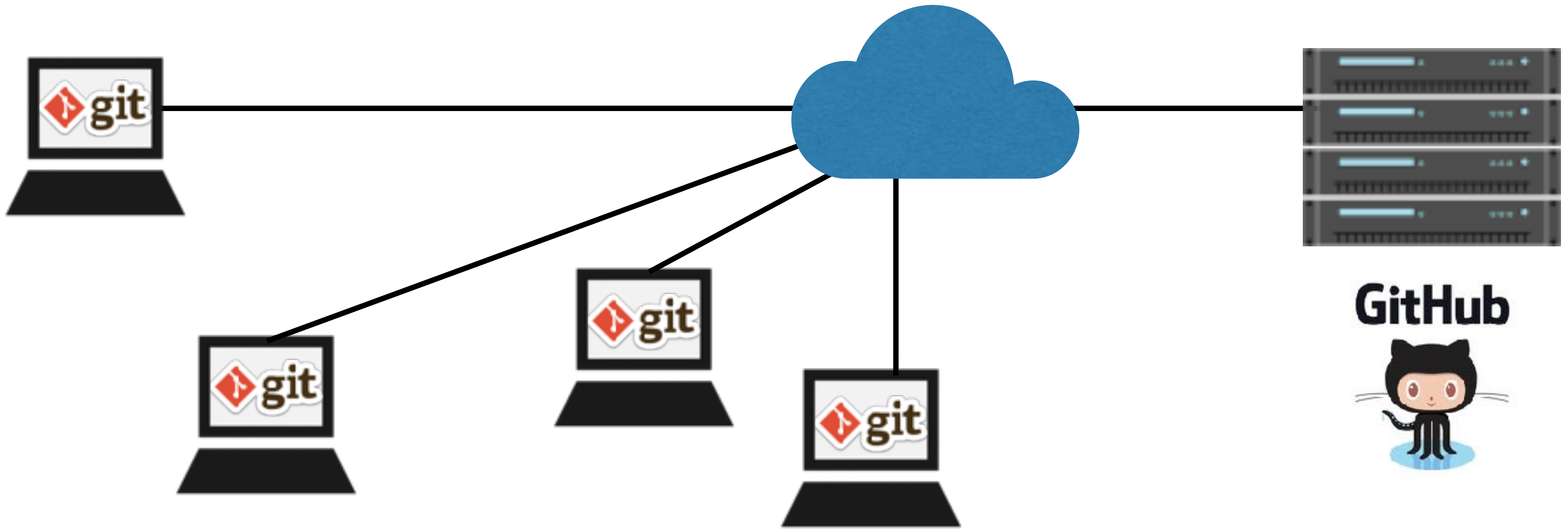
- Allows multiple developers to work on the same project. Much like Google Drive lets multiple people collaborate on the same document, GitHub allows this for code.
- You can see who worked on what.

FEEDBACK

- GitHub allows for feedback to be given on the code which, hopefully, increases code quality.

Git vs GitHub

- **Git** is version control software
- **GitHub** is a website and platform for utilizing Git in a collaborative way



Git/GitHub Vocabulary

- Repository
- Clone
- Commit
- Push
- Pull

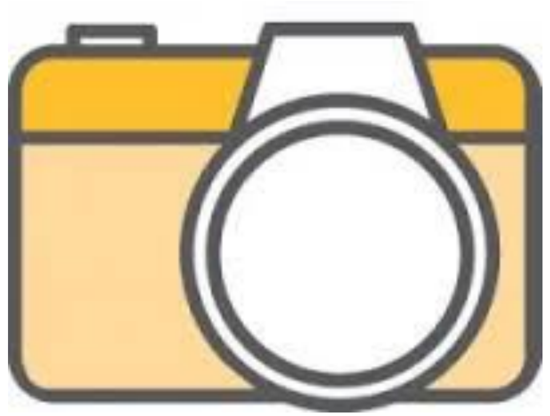
What is a repository (repo)?

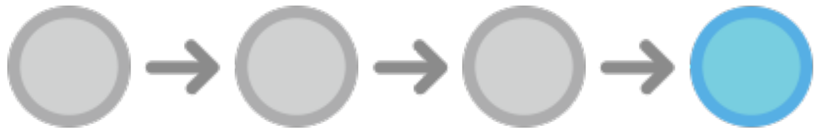
- Basic element of GitHub
- Contains all of a project's files (all the code)
- One or more users can contribute to a single repository
- Repositories are either public or private
- By the end of class today, you will create your own repo

`clone`

- Git command that copies/clones a **remote** repo to your machine
- This copy/clone is called a **local** repo
- Changes to the **local** repo will not affect the **remote**

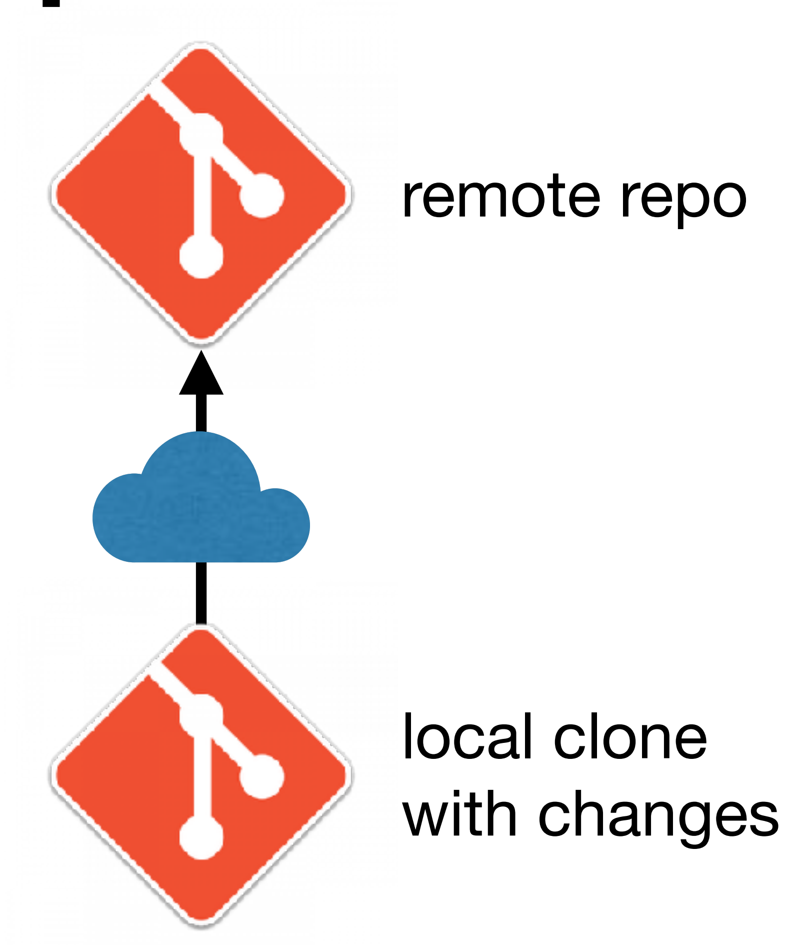
`commit`





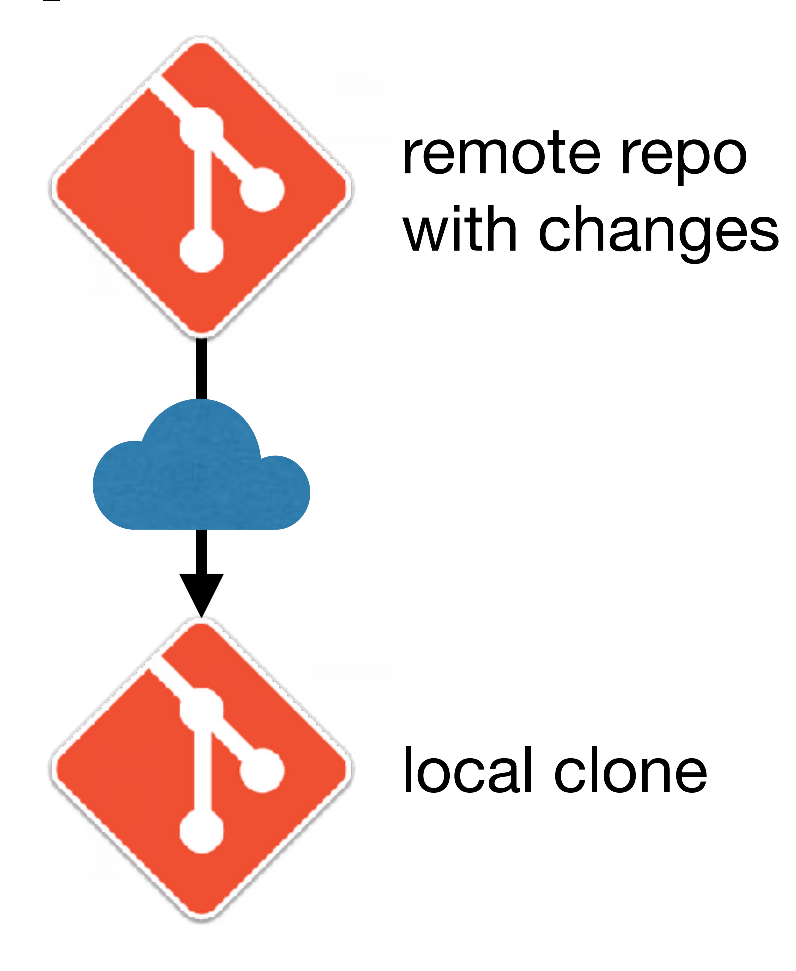
- Git command that creates a snapshot of changes to a repo
- Think of it as saving your changes with a timestamp
- Contains a message describing the changes made

push



- Git command that sends your commits (saved changes) to a **remote** repository
- Allows other developers to see your changes and copy ("pull") them to their own local repos

pull



- Git command that copies (pulls) changes by other developers from a remote repository to your local clone
- Allows you to see changes made by other developers and incorporate them into your local clone

How will we use GitHub in JSD12?

`JS-SF-12-resources`

- contains start and solution files
- you will pull changes at the start of each class

`JS-SF-12-homework`

- currently empty
- you will push your completed homework and receive feedback here

You will create your own additional repos for the 3 projects during this course.

GIT COMMANDS

THE COMMAND LINE & DATA TYPES

EXERCISE — GIT/GITHUB

EXERCISE

KEY OBJECTIVE

- Understand how to initialize a local Git repository and push/pull changes to a remote Git repository.

TYPE OF EXERCISE

- Pairs

TIMING

2 min

1. What command do you use to initialize a local Git repository? (Hint: Check the handout.) What does initializing do?
2. What command do you use to push changes to a remote Git repository? What does pushing do?
3. What command do you use to pull changes from a remote Git repository? What does pulling do?
4. BONUS: Draw a diagram illustrating all 3 commands

Intro to Node.js and command line JS

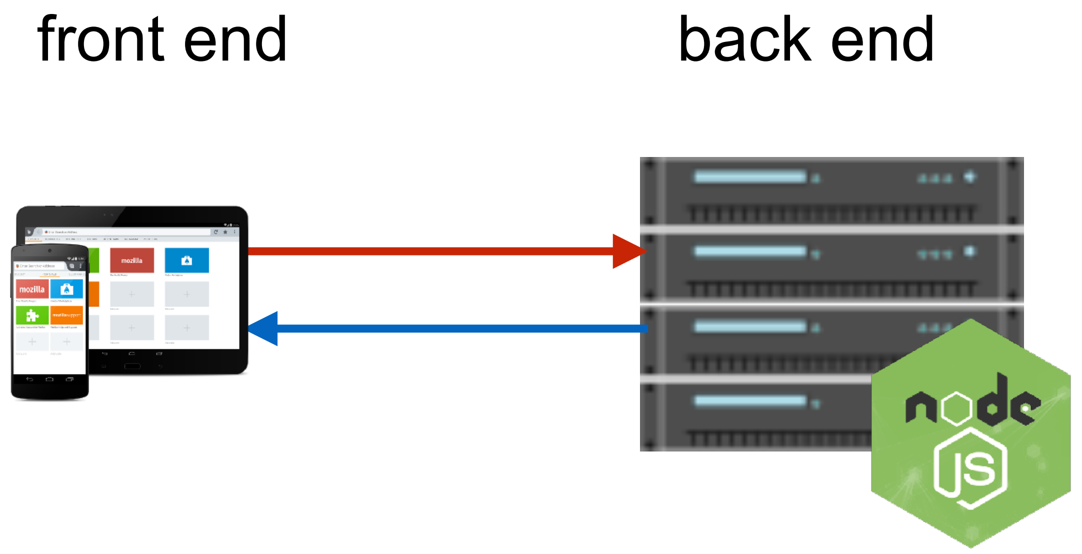
WENGER

How is Node different from JS in the browser?

- No browser-specific functionality
- Same JS engine as Chrome

What is Node good for?

- Creating a backend server for a web application
- Running a script to do data analysis
- File management
- Making command line programs



Ways to run commands in Node

Interactive command line

Your command
Node's response

```
> 5 + 2
< 7
```

Run a file

You

```
> script.js
```

Node loads the file script.js and executes its contents

Node

```
< 7
```

Executing JavaScript code

Let's write some JavaScript!

Variables

- Containers that allow us to store values
- Let us tell our program to remember values for us to use later on
- The action of saving a value to a variable is called **assignment**

Declaring a variable

```
let age;
```

Assigning a value to a variable

```
age = 29;
```

Declaring and assigning in a single statement

```
let age = 29;
```

Printing things out for our own inspection

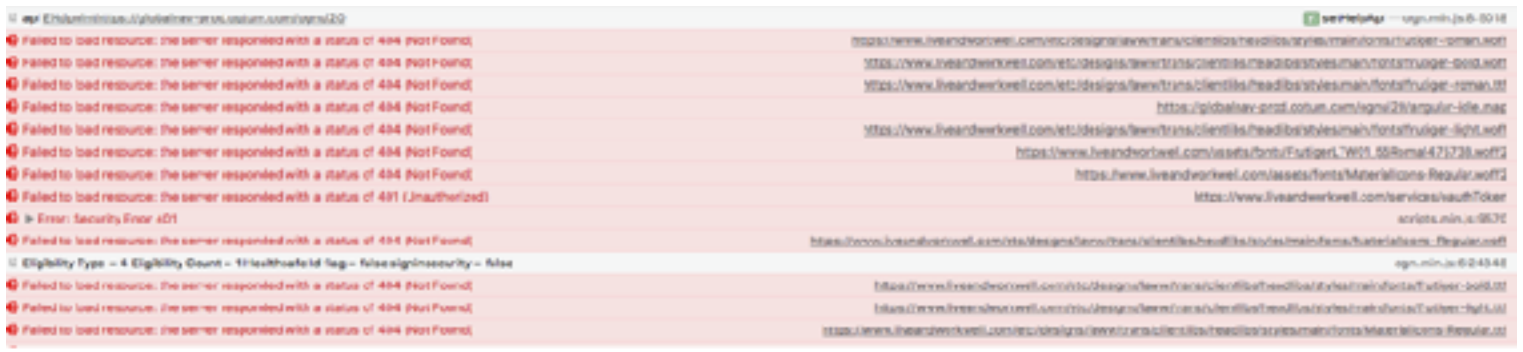
```
console.log("Hello!");
```

Printing a variable value out for our own inspection

```
console.log(age);
```

When do you use console.log?

- When you are developing a program and need help figuring out what’s going on (aka debugging)
- When you want to print things to the command line



```
[illegible]                                                          [illegible]
Failed to load resource: the server responded with a status of 404 (Not Found)   [illegible]
Failed to load resource: the server responded with a status of 404 (Not Found)   https://www.liveandworkwell.com/etc/designs/[illegible]fruiger-bold.woff
Failed to load resource: the server responded with a status of 404 (Not Found)   https://www.liveandworkwell.com/etc/designs/[illegible]fruiger-roman.ttf
Failed to load resource: the server responded with a status of 404 (Not Found)   https://globalnav-prod.[illegible]/angular-idle.map
Failed to load resource: the server responded with a status of 404 (Not Found)   https://www.liveandworkwell.com/etc/designs/[illegible]fruiger-light.woff
Failed to load resource: the server responded with a status of 404 (Not Found)   https://www.liveandworkwell.com/assets/fonts/FutigerLTW01_55Roman[illegible].woff2
Failed to load resource: the server responded with a status of 404 (Not Found)   https://www.liveandworkwell.com/assets/fonts/MaterialIcons-Regular.woff2
Failed to load resource: the server responded with a status of 401 (Unauthorized)   https://www.liveandworkwell.com/services/oauthToken
▶ Error: Security Error 401                                                     [illegible]
Failed to load resource: the server responded with a status of 404 (Not Found)   [illegible]
Eligibility Type – 4 Eligibility Count – 1 [illegible] – false [illegible] – false   [illegible]
Failed to load resource: the server responded with a status of 404 (Not Found)   [illegible]
Failed to load resource: the server responded with a status of 404 (Not Found)   [illegible]
Failed to load resource: the server responded with a status of 404 (Not Found)   [illegible]
```

browser developer tools

command line

THE COMMAND LINE & DATA TYPES

CODE ALONG

Exit the Node console

Node prompt

```
>
```

`control` + `C` twice

BASH prompt

```
$
```

EXERCISE — NODE

EXERCISE

KEY OBJECTIVE

- Run basic JavaScript code on the command line using Node.

TYPE OF EXERCISE

- Turn and talk

TIMING

2 min

1. What is Node?
2. What did we use it for today?
3. BONUS: How else can it be used?

DATA TYPES

THE DATA TYPE IDENTIFIES THE KIND OF DATA

"I just pushed my changes to the repo."

`string`

"red", "orange", "yellow", "green", "blue", "violet"

`array`

42

`number`

STRINGS

"a"

"satisfied"

"none of the above"

"Touch my hair. It’s real. (Donald Trump, June 18, 2015)"

NUMBERS

1.5

3.1415926535

27.36

45

525600

SOME LANGUAGES TREAT INTEGERS AND FLOATS AS SEPARATE TYPES, BUT NOT JAVASCRIPT

~~integers~~	~~floats~~
0	1.5
45	3.1415926535
2016	27.36
525600	

numbers

WORKING WITH DATA IN JAVASCRIPT

LIBRARY OF OBJECTS

```
Array()
Date()
Math()
…
```

LANGUAGE ELEMENTS

Operators (`+ - * / %` …)

Statements

```
  for
  function
  return
  …
```

DOM MANIPULATION

- create elements
- place elements in the browser window
- change properties of elements in the browser window
- respond to user events

IDENTIFYING DATA TYPE

- `typeof()` function
- Returns a string naming the data type of the data you pass to it
- Syntax:
  - `typeof(`*`data`*`)`, where `data` is a number, string, or other data

```
typeof(5);                         "number"
typeof('Chill');                   "string"
typeof(['red','green','blue']);    "object"
```

JS treats an array as a type of object, rather than a separate data type

ARITHMETIC OPERATORS

+	add (also concatenates strings)
-	subtract
*	multiply
/	divide
%	modulus (remainder)

SPECIAL NUMBER OPERATORS

The `Math` object provides methods for additional operations

`Math.pow(m,n)`	Returns `m` to the power of `n`
`Math.sqrt(n)`	Returns the square root of `n`
`Math.random()`	Returns a random number between 0 (inclusive) and 1 (exclusive)
`Math.floor(n)`	Returns largest integer less than or equal to `n`
`Math.ceil(n)`	Returns smallest integer greater than or equal to `n`

Exit Tickets!

(Class #1)

LEARNING OBJECTIVES – REVIEW

- Work with files/directories via the terminal window
- Create a Git repository and push/pull changes
- Run basic JavaScript code on the command line
- Describe the concept of a "data type" and how it relates to variables.

Next class preview: Arrays & Loops

- Declare, assign to, and manipulate data stored in a variable.
- Create arrays and access values in them.
- Iterate over and manipulate values in an array.
- Build iterative loops using `for` statements.

Q&A