Table of Contents

Intro To Javascript with Node.js

Welcome! We are glad you are here.

Welcome to **Codetrotters Academy**'s Intro to Javascript with Node.js course! By signing up for this course, you are taking the first steps towards becoming a **rockstar coder**, and we could not be more excited for you and your coding journey.

This GitBook will be your **life saver** for the duration of this course. When you come online, you'll find your place in the Phases and Modules, and be led through the reading and exercises.

Each Module is broken up into the following sections:

- **In a Nutshell**
    - This section provides a high-level overview of what you will learn in this module
- **Learning Objectives**
    - This section outlines the learning objectives of the module, in question format. You'll know it's time for you to move onto the next module when you can answer each question in this section confidently.
- **Required Reading and Prep-Work**
    - This section contains reading and exercises that are to be completed *before* you meet with your mentor. These articles and exercises are meant for you to get your toes wet with the material in the module, so you can really take a deep dive in the *Required Exercises* section.
- **Required Exercises**
    - This section contains tutorials and exercises that cover the learning objectives for the module. This is where most of your time will be spent. (*The exercises are arranged from easiest to hardest, so make sure you go in order!*) If you run into problems going through the exercises on your own, we suggest scheduling time with your mentor for some pair programming to get you through these exercises and for you to really digest and understand the material.
- **Additional Exercises**
    - If you find yourself with extra time, or if you really want additional practice, you can turn to the exercises in this section. These exercises are designed to be a *little* more challenging than the Required Exercises, so it might be a good idea to

schedule these exercises during a one-on-one session with your mentor!

- **Resources**
    - Here you will find helpful resources to complement your learning during the exercises. We include cheat sheets, helpful blog posts, diagrams, and even additional reading material. If you are getting stuck or frustrated, it will be a good idea to peruse the *Resources* section for additional context, that can help get you unstuck!

Ready to get going?

Remember: You'll move through the Modules in **chronological order**. You'll have **one-on-one support** from your mentor or instructor each week, so that you'll never be stuck for too long. Remember to reach out to the **Codetrotters Slack community** if you have a question that just can't wait until your mentoring hour or scheduled class time. And don't be afraid to **Google your questions**! Part of coding is being okay with not knowing the answer. Being able to ask questions and find the answer for yourself. *Be curious. Be inquisitive. Go investigate.* ***You're going to rock!***

Module 0: Git, Github, and the Command Line

In a Nutshell

This module will dive into Git, Git Desktop, and Command Line Basics. You will gather everything needed for the course, sign up for all required services, and meet your mentor and\/or class. It will also give a glimpse into what the future holds for careers as rockstar coders.

Learning Objectives

- Navigate your computer's file system using the command line
- Understand Git basics, including commits, branches, and merging

---

Required Reading & Prep-Work

Command Line: *(3 hours)*

- https:\/\/www.codecademy.com\/learn\/learn-the-command-line

Git: *(15 minutes)*

- https:\/\/www.codeschool.com\/courses\/try-git

Improve Typing Skills: *(as needed)*

- http:\/\/thetypingcat.com\/

Required Exercises

Dive into Git and start practicing with the following tutorials:

- https:\/\/www.codecademy.com\/learn\/learn-git
- https:\/\/www.learnenough.com\/git-tutorial

Additional Exercises

Get a head start on the next module with this Khan Academy tutorial, introducing you to HTML and CSS:

- https:\/\/www.khanacademy.org\/computing\/computer-programming\/html-css\/html-tags-continued\/p\/html-links

---

Resources

- Github: https:\/\/github.com\/
- Khan Academy: https:\/\/www.khanacademy.org\/
- Slack: http:\/\/codetrotters.herokuapp.com\/

Module 1: Data Structures, Variables, & Operators

In a Nutshell

In this module, you will get familiar with **basic data types** and learn how to interact with them. You will perform operations relying on logic and avoid repeating yourself by using powerful **loops**. All of these lessons are applicable in most of the other programming languages, and dominating them is **crucial for any programmer's foundation**.

Learning Objectives

Javascript as a Language:

- What is JS?
- Why do we use JS?
- Who uses JS?
- Where can I find documentation?
- How do I print to my console?

Variables\/Numbers:

- What is `var` ?
- How do I declare a variable?
- What happens when I declare a variable without a value?
- What is `null` ?
- How do I declare multiple variables in-line?
- How do I change the value of a variable?
- What basic mathematical operators are available to me, and how do I use them effectively? `+, -, *, /, %`
- What is mathematical order of precedence?
- What are shorthand operators, and how do I use them effectively? `i++, i+=1, ++i`

STDIN:

- Install and use https:\/\/www.npmjs.com\/package\/readline-sync

Decision Trees\/Booleans:

- What is a boolean?
- What are comparison operators?
- What are logical operators?
- What is an `If` statement?
- What is an `else` statement?
- What is an `else if` statement?
- Can I have nested decision trees?

Arrays:

- What are they?
- How many properties can it hold?
- How to declare them?
- How to navigate them?
- How to modify its values?
- How to add values?
- How to remove values?

Required Reading & Prep-Work

Complete the following exercises to kickstart your learning about CS Basics with Javascript. These exercises should be completed *prior* to meeting with your mentor or class for this module:

CS Basics with Eloquent JavaScript - *taking your first dive into Javascript:*

- Exercise 00 - What is a JavaScript?
- Exercise 01 - Introduction, Values, Numbers, Arithmetic, Boolean Values, Comparisons, Logical Operators, Undefined Values
- Exercise 02 - Variables, Keywords and Reserved Words, Updating variables succinctly, Conditional execution
- Exercise 04 - Data Sets

The Development Process - *asking the right questions and finding your own answers:*

- How Can I Use Stack Overflow Effectively as a Beginner

Required Exercises

The required exercises for this Module are going to be performed primarily in your *terminal* or *console*. The instructions for the exercises are arranged in this GitHub repository, in the README files. For convenience, we have also listed out each exercise below in a list:

- Exercise 1 - Practicing `console.log`
- Exercise 2 - Pre-Increments and Post-Increments
- Exercise 3 - User Prompts
- Exercise 4 - Determining Even & Odd Numbers Programmatically
- Exercise 5 - Determining Divisibility Programmatically
- Exercise 6 - BAC Calculator - *(particularly for Medalla consumption* ;) *)*

The exercises get increasingly challenging and time consuming, so budget your time accordingly! These exercises altogether should take you between *8 and 12 hours to complete* depending on your speed.

Additional Exercises

If you wizzed through the above exercises, not to worry! We have more challenges coming your way. Head over to the Extras folder of the previous GitHub repository for 3 more exercises:

- Challenge 1 - Ranges
- Challenge 2 - Rectangles
- Challenge 3 - `n/n-1`

Resources

- http:\/\/eloquentjavascript.net\/

Module 2: Loops & Functions

In a Nutshell

In this module, you will get familiar with **basic data types** and learn how to interact with them. You will perform operations relying on logic and avoid repeating yourself by using powerful **loops**. All of these lessons are applicable in most of the other programming languages, and dominating them is **crucial for any programmer's foundation**.

Learning Objectives

Iterations:

- What are the components?
- What is a `while` loop?
- What is a `do while` loop?
- What is a `for` loop?
- What is an infinite loop?
- Can I have nested loops?
- How to use them with arrays?

Functions:

- How do I declare a function?
- How do I execute a function?
- How do I pass in values to a function?
- How do I return values from a function?
- When do I use a function?

Strings:

- What is a string?
- How do I declare a string?
- How do I mutate a string?

- How do I get slices of a string?
- How do I perform a new line?
- How do I perform a tab?
- How do I concatenate strings (using `+`)?

Required Reading & Prep-Work

Complete the following exercises to begin learning about Strings, Loops, and Functions. These exercises should be completed *prior* to meeting with your mentor or class for this module:

CS Basics with Eloquent JavaScript - *taking your first dive into Javascript:*

- Exercise 01 - Strings
- Exercise 02 - Functions, console.log, Return values, While and Do Loops, For Loops, Breaking Out Of a Loop
- Exercise 03 - Intro, Defining a Function, Parameters And Scope, Functions as Values, Declaration Notation
- Exercise 04 - Properties, Methods, Futher Arrayology, Strings and their Properties

Required Exercises

TODO

Additional Exercises

TODO

For the speedy ones out there! Here's more exercises to keep you busy:

More Exercises with *Eloquent JavaScript*:

- Exercise 02 -
- Exercise 02 -
- Exercise 04 -
- Exercise 04 -

Resources

- http:\/\/eloquentjavascript.net\/

Module 3: JSON, OOP, & Maps

In a Nutshell

In this Module you will learn the basics of **Object Oriented Programming**, or **OOP**. OOP will help you to simplify the representation of complex relationships programmatically. You will also learn about **JSON**, which enables the **communication of data across multiple languages** and programs. It is also a tool to represent data. Finally, you will begin learning about **programmatic testing**. Testing is a crucial part of any professional developer's thought process, and will allow you to build strong and scalable codebases.

Learning Objectives

JSON\/Maps:

- What is JSON?
- What is a Map?
- How do I declare a JSON map?
- What properties can it hold?
- How many properties can it hold?
- How do I navigate a JSON map?
- How do I mutate a value?
- How do I add values?
- How do I remove values?
- How do I rename values?
- How do I check if an element is in the Map?
- How do I iterate through a Map?

Classes\/Objects\/Methods:

- What is a Class?
- What is an Object?

- What do they abstract?
- How do I declare a class?
- What are instance variables?
- How do I instantiate a class (resulting in an object)?
- What is the difference between a class and an object?
- How can I modify variables?
- What is `this` ?
- What are methods?
- How can I invoke a method?
- How can I associate a method with a class\/object?
- What is the difference between a method and a static method?
- What are Global Objects?

Required Reading & Prep-Work

Complete the following exercises to begin learning about JSON, Maps, Classes, Objects, and Methods. These exercises should be completed *prior* to meeting with your mentor or class for this module:

The Docs! - *there's no better place to begin than the documentation. Please read the following sections of the documentation for JavaScript:*

- Docs: Introduction to Object-Oriented JavaScript

CS Basics with Eloquent JavaScript - *continuing to dive into Javascript:*

- Exercise 04 - Objects, Objects as Maps, Mutability
- Exercise 05 - JSON, Methods, Prototypes, Constructors, Getters and Setters, The `InstanceOf` Operator

---

Required Exercises

WIP - TODO verify links

Complete the following exercises to truly begin working with JSON, Maps, Classes, Objects, and Methods.

Eloquent JavaScript - *Take a deeper dive into Objects with these exercises*

- Exercise 06a
- Exercise 06b

Creating your own `Human` class - *In this exercise, you will be practicing writing your very own class - a `Human` class. Fork and Clone the following GitHub repo - you'll find instructions on the GitHub README.*

- GITHUB LINK HERE

Additional Exercises

TODO

For the speedy ones out there! Here's more exercises to keep you busy:

More Exercises with *****Eloquent JavaScript*****:**

- Exercise 06 -

Codetrotters GitHub:

- Repo - Find the intersection of two arrays

Resources

TODO

- http:\/\/www.json.org\/

Module 4: Anonymous Functions & Callbacks

In a Nutshell

In this Module you will continue to learn the basics of **Object Oriented Programming**, or **OOP**. You will be introduced to **functions**, which help a programmer encapsulate and re-use logic. Finally, you will begin learning about **programmatic testing**. Testing is a crucial part of any professional developer’s thought process, and will allow you to build strong and scalable codebases.

Learning Objectives

Anonymous Functions

- How are anonymous functions different than named functions?
- How do I declare an anonymous function?
- What are some common use cases of anonymous functions?

Callbacks

- What are callbacks?
- What are some common use cases of callback functions?
- How do I use callback functions?
- When should I use callback functions?
- What is the difference between `a(b(1))` and `a(b)` ?

Required Reading & Prep-Work

WIP - TODO verify links

Complete the following exercises to begin learning about anonymous functions and callbacks. These exercises should be completed *prior* to meeting with your mentor or class for this module:

Some Light Reading... - *below are links to some helpful material on anonymous functions and callbacks. Start by reading these blog posts and articles to familiarize yourself with some of the concepts:*

- What are Anonymous Functions For?
- Anonymous Functions - An Overview
- Callback Functions - An Overview
- More on Callback Functions

Amy Simmons' Github Exercise - *In this exercise, you'll begin practicing writing and using callback functions. Follow the Github link, fork, and clone the repo. Then follow the instructions in the README.*

- Amy Simmons - Callbacks

Functions with Eloquent JavaScript - *continuing to dive into Javascript:*

- Exercise 05 - Higher Order Functions, Filtering an Array, Transforming with Map, Summarizing with Reduce

---

Required Exercises

WIP - TODO verify links

Complete the following exercises to truly begin working with anonymous functions and callbacks.

Eloquent JavaScript - *Take a deeper dive into Objects with these exercises*

- Exercise 05a
- Exercise 05b

Extend your `Human` class - *In this exercise, you will be working off of the `Human` class you built in* ***Module 03****. This time, you will extend the `Human` class using functions. Fork and Clone the following GitHub repo - you'll find instructions on the GitHub README.*

- GITHUB LINK HERE

Additional Exercises

There are no additional exercises in this module! Continue working on extending and adding to your `Human` class if you have free time :)

Resources

- https:\/\/developer.mozilla.org\/en-US\/docs\/Web\/JavaScript\/Introduction_to_Object-Oriented_JavaScript
- http:\/\/www.json.org\/

Module 5: Closures & Testing

In a Nutshell

TODO - elaborate on closures

In this Module you will be introduced to closures. In the later part of the module, you will begin learning about **programmatic testing**. Testing is a crucial part of any professional developer's thought process, and will allow you to build strong and scalable codebases.

Learning Objectives

Closures

- What are closures?
- What are common use cases of closures? *(private variables in a class)*
- How do I use closures?
- When should I use closures?

Testing

- What is testing used for?
- What are common practices?
- When should I test?
- What should I test?
- How often should I test?

Required Reading & Prep-Work

WIP - TODO verify links

Complete the following exercises to begin learning about anonymous functions and callbacks. These exercises should be completed *prior* to meeting with your mentor or class for this module:

Going back to the Docs - *the Docs are always the best place to start. The reading materials for this module begin with the docs on Private Properties. Then you'll read a helpful blog article on the topic.*

- Docs: Private Properties
- Some Reading: Private

Functions with Eloquent JavaScript - *continuing to dive into Javascript:*

- Exercise 03 - Nested Scope, Optinal Arguments, Closure

Required Exercises

WIP - TODO verify links

Complete the following exercises to truly begin working with closures and testing.

Private Variables in your `Human` Class - *In this exercise, you will be working off of the* `Human` *class you built in **Module 03** and extended in **Module 04**. This time, you will extend the* `Human` *class again, this time by adding private variables. Fork and Clone the following GitHub repo - you'll find instructions on the GitHub README.*

- GITHUB LINK HERE

Additional Exercises

There are no additional exercises in this module! But there's a *lot* of prep work for the next module, Module 06, so if you have free time, it might be a good idea to get a head start on that :)

Resources

- https:\/\/developer.mozilla.org\/en-US\/docs\/Web\/JavaScript\/Introduction_to_Object-Oriented_JavaScript

- https:\/\/developer.mozilla.org\/en-US\/Add-ons\/SDK\/Guides\/Contributor_s_Guide\/Private_Properties

- http:\/\/www.json.org\/

Module 6: Node.js Ecosystem & Tooling

In a Nutshell

Take JavaScript to the **server side using Nodejs**. In this phase, you will begin to fully understand the **environment** that we have been using throughout our previous exercises. You will more fully understand the tooling and the powerful module system which **increases a developer's productivity tenfold**.

Learning Objectives

Data Types

- What is `NaN` ? How can I trigger `NaN` ?
- How can I test for `NaN` ?
- What is `undefined` ? How can I trigger `undefined` ?
- How can I test for `undefined` ?

Node.js

- What is Node?
- What is the `repl` ? What’s its usage?
- What is `require` , and how does it affect the way I program?
- Where can I find documentation?
- What do the `[]` around parameters in the signatures of functions found in documentation mean?
- Why do some methods have a `Sync` counterpart?

Modules

- How can I create my own?
- How can I use my own in another file?
- What is `npm` ?
- How can I install modules?

- What is the `node_modules` folder?
- Which are the most common modules in npm.com?
- What is the `package.json` ?
- How can I generate a `package.json` automatically?
- Why should I use the `--save` flag when I install modules?

Required Reading & Prep-Work

Complete the following exercises to begin learning about anonymous functions and callbacks. These exercises should be completed *prior* to meeting with your mentor or class for this module:

Going back to the Docs - *the Docs are always the best place to start. Browse the beginning sections of the Node and NPM documentation.*

- Node Intro
- Node Docs
- NPM Docs

Some Helpful Articles and Blog Posts - *diving into server-side JavaScript is a big move. Make yourself more comfortable by reading the following blog articles.*

- Why the Hell Would I Use Node.js?

Some Helpful Tutorials - *now that you have the background from the articles, take a stab at the following tutorials to start getting your hands dirty.*

WIP - TODO verify links

- Tutorials Point - Node.js
- SitePoint - Module Exports in Node.js
- Packages in Node.js (?)
- SitePoint - Node Package Manager
- RisingStack - NPM Tutorial

http:\/\/www.tutorialspoint.com\/nodejs\/nodejs_repl_terminal.htm

https:\/\/www.sitepoint.com\/understanding-module-exports-exports-node-js\/

http:\/\/browsenpm.org\/package.json

https:\/\/www.sitepoint.com\/beginners-guide-node-package-manager\/

https:\/\/blog.risingstack.com\/node-hero-npm-tutorial\/

Required Exercises

WIP - TODO verify

Complete the following reading exercises to take your first dive into server-side JavaScript.

Absolute Beginner's Guide to Node.js - *Read the following article to make sure you are comfortable with the lingo surrounding Node.js*

- Absolute Beginner's Guide to Node.js

Convert your `Human` Class into a Module - *In this exercise, you will be working with the `Human` class you built in **Module 03** and extended in **Module 04** and Module 05. This time, you will convert the `Human` class from a class into a module. Fork and Clone the following GitHub repo - you'll find instructions on the GitHub README.*

- GITHUB LINK HERE

- Convert Human class into a module. Create a module that creates random Human Objects when invoked.2. Install several modules from NPM and play with them. Look into Chalk.3.

Additional Exercises

For the speedy ones out there:

- AirPair Node Tutorial

Resources

Docs, Docs, Docs...

- Node Intro
- Node Docs
- NPM Docs

Module 7: Advanced Testing, HTTP, and Express

In a Nutshell

Take JavaScript to the **server side using Nodejs**. In this phase, you will begin to fully understand the **environment** that we have been using throughout our previous exercises. You will more fully understand the tooling and the powerful module system which **increases a developer's productivity tenfold**.

Learning Objectives

Advanced Testing

- What is a test runner?
- How do I install Mocha and execute it?
- What is an `assert` ?
- How can I use `assert` with Mocha?
- What is Chai?

HTTP\/Request

- What is HTTP?
- What is HTTP used for?
- What are HTTP status codes? What are some examples?
- What are the common verbs?
- What are headers?
- How can I generate HTTP requests? *(Postman)*
- What is a query param?
- How can I pass a JSON payload?
- What is the request module?
- How can I use the request module?

Servers\/Express

- What is a request context?
- What is a response?
- What is Express?
- How can I build a `Hello World` application using Express?
- What is a route?
- How can I support different HTTP verbs?
- How to read query parameters in Express?
- How to read route parameters in Express?
- What is a middleware?
- How can I use request context to my benefit?
- How to read a JSON body in Express? *(body-parser)*

Required Reading & Prep-Work

Complete the following exercises to begin learning about anonymous functions and callbacks. These exercises should be completed *prior* to meeting with your mentor or class for this module:

Testing - *Dive into the world of automated testing with the following materials:*

- The Node Way - Testing Essentials

HTTP - *The protocol for how data is requested and sent over the internet. Begin learning with the exercises below.*

- Anatomy of an HTTP Transaction
- Docs: Postman

Express - *Get to know Express using the following materials:*

- Hello World App with Express
- Docs: Express (?)

Required Exercises

WIP - TODO verify

Testing in Node with Mocha - *Complete the following tutorials to practice testing with Mocha*

- Getting Started with Node.js and Mocha
- Automated Mocha Tests for Node.js

Codetrotters Github *TODO*

- GITHUB LINK HERE

1. Create a module that uses the \`request\` module in order to interface with the BuzzFeed API.

2. Create an express app that uses querystring params to return Human objects as JSON.

Additional Exercises

For the speedy ones out there:

TODO - Github?

1. Convert your tests for the random Human generator into Mocha tests.

2. Write tests for your BuzzFeed client.

Resources

Docs, Docs, Docs...

- NPM Docs - Requests
- Express Docs