

Our Curriculum

Our full academy curriculum is 11 weeks, we have six lecture weeks, three group project weeks and two individual assignment week. In the following pages the weeks are explained in more detail.

○ – Lecture Week

△ – Group Project Week

⬡ – Individual Assignment Week





BEGINNER BOOTCAMP

HTML, CSS & JavaScript

During this week you will learn the basics of programming, and how to think like a programmer. You will learn how to build an interactive front end application using HTML5, CSS3 & Javascript.

MONDAY

WELCOME, HTML & VERSION TRACKING

- › You will learn to navigate and do basic file system management from the shell.
- › You will learn to use git & GitHub for version tracking.
- › You will learn to use Chrome DevTools to inspect the HTML of a page.
- › You will learn to create an HTML file in your code editor and view the page in your browser.
- › You will learn to use (the latest HTML5) tags to structure content on your web page.
- › You will learn to explain how writing semantic HTML improves SEO & accessibility.

TUESDAY

CSS & RESPONSIVE WEB DEVELOPMENT

- › You will learn to use Chrome DevTools to inspect the CSS of a page using different styles.
- › You will learn to create a CSS page in your code editor and link it to your HTML file.
- › You will learn to use css selectors such as the `class` & `id` selector to style specific elements on a page.
- › You will learn to apply any CSS declaration or rule to style elements on your page, given some example code.
- › You will learn to use Chrome DevTools to style your page 'mobile first'.

WEDNESDAY

USER INPUT & THE DOM

- › You will learn to create a Javascript file in your code editor and connect it to your HTML file.
- › You will learn to use Chrome DevTools to inspect your javascript code.
- › You will learn to use DOM methods to `select`, `create` & `display` elements in your app.
- › You will learn to manipulate user input and page content by (re)assigning values to a property of a DOM element.
- › You will learn to write Javascript `functions` to keep your code `DRY`.
- › You will learn to build a form and display the user input in your app.
- › You will learn to use javascript control statements to adjust logic according to user input validation.

THURSDAY

DATA MANIPULATION

- › You will learn to recognize composite data structures.
- › You will learn to select specific data from complex data structures.
- › You will learn to use a loop or iterator to create elements on a page with content from an array.

FRIDAY

PROJECT DAY

- › You will learn to apply everything you learned this week without being told explicitly how to do it.
- › You will learn to build an interactive web app given a set of user stories.
- › You will learn to discuss the data flow in an app.
- › You will learn to collaborate with other developers to build a web app.

JavaScript

Introduction to object-oriented and functional programming paradigms in ES6 syntax.

MONDAY

OBJECT ORIENTED PROGRAMMING & JAVASCRIPT MODULES

- › You will learn to explain how OOP makes your code easier to understand.
- › You will learn to write code according some of the main principle of OOP.
- › You will learn to create modules to split your applications into multiple files, to keep your code organized.
- › You will learn to use `npm` to initialize a new NodeJS project and install dependencies.
- › You will learn to create a Javascript `class` to make your code more flexible.
- › You will learn to test your code using the `mocha` library.”

TUESDAY

FUNCTIONAL PROGRAMMING & ALGORITHMS

- › You will learn to identify pure functions by checking whether a function uses global variables or have side effects.
- › You will learn to explain the most important characteristics of functional programming and how they make your code more predictable.
- › You will learn to recognize a recursive function.
- › You will learn to apply the `map` method to transform all values of an collection and put them into a new collection of the same size.
- › You will learn to apply the `filter` method to filter a collection into a smaller collection with only some of the elements
- › You will learn to apply the `reduce` method to transform a collection into a single value.

WEDNESDAY

OOP/FP RECAP

- › You will learn to apply Object Oriented Programming in a larger application.
- › You will learn to apply functional programming in some game logic.

THURSDAY

DATA TRANSFORMATIONS

- › You will learn to describe the different steps of a simple algorithm in English.
- › You will learn to describe the difference purposes of objects and arrays.
- › You will learn to explain how declarative transformations are easier to read than imperatives one.
- › You will learn to use `reduce` to find the min/max of a list of items based on several properties.
- › You will learn to convert a list of items to a reference object for fast lookups.
- › You will learn to solve an algorithm that requires multiple steps/different data.

FRIDAY

DATA FETCHING

- › You will learn to explain what an HTTP request/response is.
- › You will learn to identify the `headers`, `body`, `status code`, `method` and `content-type` in an HTTP request/response.
- › You will learn to use requests/responses to fetch or modify data, given an http client library and its docs.
- › You will learn to explain how promises help you optimize the performance of your code.
- › You will learn to implement `.then`, `.catch`, `Promise Chaining` & `Promise.all` to handle requests/responses.
- › You will learn to create a promise given the values that it should return when it is resolved/rejected.

React & Redux

Build the front end of a single page application using React and Redux.

MONDAY

REACT INTRO

- › You will learn to explain that React is a library that helps you manage scalable complex UIs, and how it achieves this.
- › You will learn to explain that React manages performance optimization by using a Virtual DOM.
- › You will learn to create a new React project using `create-react-app` to create a SPA.
- › You will learn to compose components to render data, passing props to child components.
- › You will learn to manage state in a component, and have the component update (re-render) accordingly
- › You will learn to break up a user interface into components, and you can decide what data these components should receive (as props) and/or manage (as state).

TUESDAY

REACT ESSENTIALS

- › You will learn to use `PropTypes` to ensure that components are passing the right data needed for the job.
- › You will learn to explain what a callback prop is, and why it's useful.
- › You will learn to explain what lifting state means, and what kind of thinking goes into the decision to lift state up.
- › You will learn to describe the component tree structure of an application, figure out how data flows down through props, and how callback props are used to pass data back up.
- › You will learn to write callback props and pass them to child components.
- › You will learn to solve errors relating `this`` being `undefined`` in React components.
- › You will learn to style your React components with style imports.

WEDNESDAY

REAL WORLD REACT

- › You will learn to check out a web API, do some requests and fetch some data, using a tool such as HTTPie (or Postman or REST Client or ...).
- › You will learn to fetch data from an API and then render it in a React component
- › You will learn give some examples of life-cycle methods, explain when they're called, and when you'd use them.
- › You will learn to explain the difference is between container and presentational components, and why this separation of concerns is often introduced.
- › You will learn to make implement to `react-router`` library to create a React app with multiple pages.

THURSDAY

INTRO TO REDUX

- › You will learn to explain that Redux is a library to simplify state management.
- › You will learn to set up a Redux store to hold you state.
- › You will learn to create and dispatch actions to trigger your store to update its state.
- › You will learn to create reducers to define how the state of the store changes when an action is dispatched.
- › You will learn to explain how and why you should avoid mutations in your reducers.
- › You will learn to make the state of the Redux store available to React components.

FRIDAY

REAL WORLD REDUX

- › You will learn to build action creators to make your actions reusable.
- › You will learn to implement async actions & `redux-thunk`` to fetch data.
- › You will learn to implement the `redux`` library in a more complex app.



GROUP PROJECT

Project Week

Use agile methods & GitHub to build an application as a team.

MONDAY

AGILE & SCRUM

- › You will learn to list the main principles of agile development.
- › You will learn to explain that SCRUM is a process to implement these agile principles.
- › You will learn to list the standard 'events' of a *sprint*.
- › You will learn to describe the different roles in a scrum team.
- › You will learn to implement SCRUM/Agile values and methodologies while building a complex app in a team.

TUESDAY TO THURSDAY

GROUP PROJECT

- › You will work in groups to build a frontend application using React-Redux.

FRIDAY

PRESENT YOUR PROJECT

- › You will present & demo your application to the class!

Express & PostGres

Build a RESTful back end using Express and Postgres, connect it to a front end application and deploy these applications.

MONDAY

DATABASES

- › You will learn to explain what a database is and why you would want to use one.
- › You will learn to set up a postgres database.
- › You will learn to predict the outcome of various SQL queries using SELECT, ORDER BY, GROUP BY & JOIN.
- › You will learn to predict which SQL query has been used, given a particular result.
- › You will learn to explain what migrations, normalization and transactions are.
- › You will learn to distinguish normalized from not normalized tables.

TUESDAY

REST APIS

- › You will learn to explain what an API is.
- › You will learn to build a simple restful CRUD API using ExpressJS, Docker, Postgres & Sequelize.
- › You will learn to describe the main characteristics of a REST API.
- › You will learn to write SQL queries to perform all CRUD actions in your database.

WEDNESDAY

ADVANCED APIS

- › You will learn to create an API for realistic models, including relations.
- › You will learn to set up user registration & authentication for your back-end.
- › You will learn to apply pagination in a realistic API.

THURSDAY

REDUX + REST

- › You will learn to create full CRUD REST API.
- › You will learn to make calls from your frontend (React/Redux) to you backend (ExpressJS API).
- › You will learn to use simple React forms to update or create data in your backend.

FRIDAY

SECURITY & DEPLOYMENTS

- › You will learn to use parameterized queries to prevent SQL injection.
- › You will learn to explain why you should add rate-limit log-in requests or a CAPTCHA to protect yourself from a brute force attack.
- › You will learn to add salt to a hash to prevent sensitive exposure.
- › You will learn to set up authorization to prevent broken access control.
- › You will learn to prevent users from writing code that will be run on other peoples machines.
- › You will learn to deploy both your backend and frontend applications using Heroku.

TypeScript, GraphQL & React Native

Discuss the pros and cons of Typescript, GraphQL & React native.

MONDAY

TYPESCRIPT I

- › You will learn to list some arguments in favor of and against using a typed language.
- › You will learn to set up and compile a TypeScript project.
- › You will learn to create basic types, interfaces, unions and generics to specify the types of all sorts of data in your app.
- › You will learn to build a simple application using TypeScript.

TUESDAY

TYPESCRIPT II

- › You will learn to use decorators and explain why they are useful.
- › You will learn to use TypeScript to build a backend.

WEDNESDAY

PROJECT DAY: FULL STACK LABS

- › You will learn to discuss the data model, endpoints and views/pages of a potential app, given a set of user stories.
- › You will learn to set up a backend & frontend from scratch and connect the two.

THURSDAY

GRAPHQL

- › You will learn to explain how GraphQL optimizes performance by allowing to query data that would traditionally be exposed in multiple endpoints in one single request.
- › You will learn to write a GraphQL Query to request for specific data using the GraphQL playground.
- › You will learn to write a GraphQL Mutation to alter data in your back end using the GraphQL playground.
- › You will learn to create a GraphQL server containing a schema, root fields & resolvers.
- › You will learn to integrate GraphQL into a React app using the Apollo library.

FRIDAY

REACT NATIVE

- › You will learn to build a simple app using React Native and test it on your mobile device.
- › You will learn to use React Native's specific components.
- › You will learn to set up different screens in your app and navigate programmatically.



GROUP PROJECT

Game Project

Build a full stack application implementing a two way interactive communication session, advanced business logic & continuous deployment as an agile team.

MONDAY TO FRIDAY

GAME PROJECT

- › You will work in a team to build a full stack multiplayer game!
- › You will present and demo your game to the class on Friday.



GRADUATION ASSIGNMENT

Individual Assignment

During this week you will demonstrate what you've learned over the past weeks, and build a full stack application implementing an algorithm individually.

MONDAY TO FRIDAY

INDIVIDUAL ASSIGNMENT

- › You will work in alone to implement Express, React, Redux and an algorithm to build a full stack app, on your own.
- › You will hand in your assignment, and be evaluated, on Friday.



GROUP PROJECT

Real World Project

Two full weeks of projects, show everyone what you've learned!

MONDAY TO FRIDAY

LEGACY CODE

- › You will learn how to work with an existing code base, in a team you'll work on a project where the previous class left it.

MONDAY TO FRIDAY

PORTFOLIO PROJECT

- › You will work on an individual project; this will be used as your portfolio project. On Demo Night, you present the project, and the winner will receive a prize!



CAREER COACHING

Career Coaching

This week will consist of workshops and time to prepare you for the job market. After convincing Codaisseur that you can code after ten weeks of training, it is time to convince your future employer, too!

The workshops will take place in the mornings, in the afternoon, you will have time to implement the learnings and continue working on your project(s) which you will use for your portfolio.

MONDAY

JOB SEARCH KICKOFF

- › We'll kick off the search with a short presentation on what you can expect and what we expect of you the following weeks.

TUESDAY

HOW TO "GIT GUD"

- › Show the best of your code by highlighting it on your GitHub profile. Step by step, you'll learn how to set up a repository complete with an excellent README.

WEDNESDAY

WRITE A CV AND COVER LETTER AS A DEVELOPER

- › We explain to you all the best practices to write a clear CV and compelling cover letter that will get you invited for an interview.

THURSDAY

HOW TO ACE AN ASSIGNMENT

- › This is where employers want to test your technical capabilities. In this workshop, you will get the tips from an experienced developer how to approach it and impress the hiring manager.