Lesson Weekend

Intermediate JavaScript (/intermediate-javascript) / Object-Oriented JavaScript (/intermediate-javascript/object-oriented-javascript) / Homework and Class Structure

Text

So far our in-class time has followed a similar routine. We arrive or log on to class, find a partner, sign in, locate the day's lessons and exercises, then work through them in order. This usually involves reading lessons, completing a project or activity, reading a few more lessons, completing a few more projects or activities, etc. But now that we have the necessary skills to build larger, more complex projects in class, this structure will change.

New Daily Structure

Beginning with the first class of this section and continuing throughout the remainder of the program, exercises and lessons will be completed as follows:

- **Each class session will contain only a single coding exercise** (denoted by the pencil icon). It lists *all* of the class session's in-class projects.
- **All lessons** (denoted by the book icon) **are to be read outside of class**. We previously titled lessons meant to be completed outside of class with a *Homework* prefix. But from this point on,

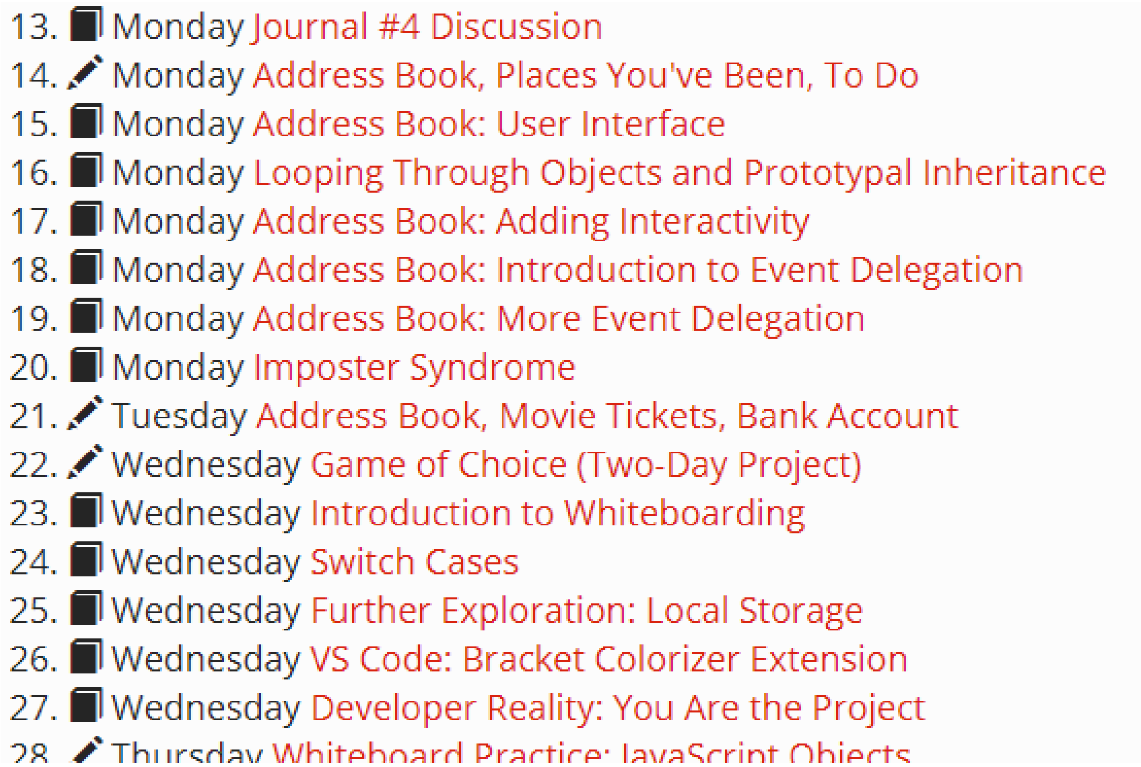
***all* lessons are homework by default**, regardless of title.

You are not required to code along with homework lessons, nor to fully understand the concepts in the lessons. Just read them, familiarize yourself with new concepts and vocabulary, and prepare to practice these concepts in class the following day.

As a tip, we suggest looking at upcoming homework in advance so you can plan time to work through it ahead of class.

Example

Let's work through an example based on the picture below:



13. 📓 Monday Journal #4 Discussion
14. ✏ Monday Address Book, Places You've Been, To Do
15. 📓 Monday Address Book: User Interface
16. 📓 Monday Looping Through Objects and Prototypal Inheritance
17. 📓 Monday Address Book: Adding Interactivity
18. 📓 Monday Address Book: Introduction to Event Delegation
19. 📓 Monday Address Book: More Event Delegation
20. 📓 Monday Imposter Syndrome
21. ✏ Tuesday Address Book, Movie Tickets, Bank Account
22. ✏ Wednesday Game of Choice (Two-Day Project)
23. 📓 Wednesday Introduction to Whiteboarding
24. 📓 Wednesday Switch Cases
25. 📓 Wednesday Further Exploration: Local Storage
26. 📓 Wednesday VS Code: Bracket Colorizer Extension
27. 📓 Wednesday Developer Reality: You Are the Project
28. ✏ Thursday Whiteboard Practice: JavaScript Objects

- The *Address Book, Places You've Been, and To Do* exercise contains all of Monday's in-class projects and activities.
- All lessons below this exercise, labeled *Monday*, are Monday homework assignments that must be read before class on Tuesday.

- The *Address Book, Movie Tickets, Bank Account* exercise contains all of Tuesday's in-class projects.
- There is no homework for Tuesday.
- The *Game of Choice (Two-Day Project)* exercise contains all of Wednesday and Thursday's in-class projects.
- Any other lessons labeled *Wednesday* are Wednesday homework assignments that must be completed before Thursday's class. So on, and so forth.

Homework and Pair Programming

Keep in mind that Epicodus follows the flipped classroom model, where reading is done outside of class so we can spend the time during class hours actually getting hands-on experience applying new concepts and practicing technical communication with our pair.

Coming to class repeatedly without having at least read through the material is unfair to your pair and sets you up for failure at Epicodus. When you repeatedly do this it inadvertently puts more pressure on your pair to know what to do and can lead them to burn out. It also creates a gap in the baseline knowledge (from the homework) shared between you and your pair, which can be stressful for everyone.

The expectation for all students is to have at least read through the homework before class. You are welcome to code through the material as homework if you have more time available to you outside of class. However, **please respect that not everyone in your class may have the same time available to dedicate to coding outside of class as you have.**

It’s not fair to repeatedly ask to only pair with others who are able to dedicate more hours to coding outside of the classroom or who are at the same comfort level with coding as you. When you repeatedly do this it has the effect of inadvertently shaming others and alienating you from your peers. This habit also does a disservice to you as well, as you are missing out on opportunities to truly test your understanding of the material by workshopping it with someone else.

Often the homework builds a project to put new concepts into action. Since we don’t expect students to code along with the homework, the first practice prompt for the following class session is usually dedicated to reviewing the homework and building the project. Even in these cases, you're expected to read lessons the night before and you should never be reading a lesson that was assigned as homework for the first time during class.

Keep in mind that understanding new concepts well takes repeated practice and review. This is why reading through the homework before you come to class is so important — it’s your first review (of many) that will prepare you to work with your pair during class.

As a guideline, if you and your pair are in a situation where one person has read through the homework and the other person has coded through the homework, know that this is very common and expect to start by pair programming through the homework to build the project.

Reminder: Be Comfortable With Not Fully Understanding Concepts Right Away

As you start your next course at Epicodus, we want to remind you that the path to fully understanding a new concept takes many forms. Sometimes, you may read about a new concept once, practice it once, and understand it immediately. For other concepts, you may need to read about it and practice it a lot before understanding it.

You likely will also experience that you understand a concept, but on a second review you understand it in a new or deeper way. Sometimes you’ll work with a concept for weeks before you experience the moment where it “clicks” in your mind.

Similarly, you may learn a new concept just enough to get by but only months later does it really click. Other times, understanding really comes down to seeing a concept applied in various scenarios, or hearing a metaphor to visualize it.

Whatever your experience is, we invite you to approach learning through the high and low moments with a growth mindset!

Summary

From this point forward, all in-class time is spent coding, and all lessons are homework by default. This allows us to maximize hands-on coding practice, and bolster our portfolios with bigger, more complex projects as we prepare for our eventual job search. Finally, coming to class having read the homework is required.

Previous (/intermediate-javascript/object-oriented-javascript/installing-and-using-node-version-manager)
Next (/intermediate-javascript/object-oriented-javascript/javascript-objects)

Lesson 3 of 33
Last updated March 23, 2023

disable dark mode

(http://www.epicodus.com)