Exercise Tuesday

Intermediate JavaScript (/intermediate-javascript) / Object-Oriented JavaScript (/intermediate-javascript/object-oriented-javascript) / Address Book, Movie Tickets, Bank Account

Text

Goal: The goal for this class session is to continue to practice creating objects using constructors and prototypes, and to connect those objects to a user interface.

Warm Up

- When would you use a constructor, and when would you use a prototype? What functionality do they offer?
- What is the design principle 'separation of concerns'? What benefits does it offer?
- What is event delegation and how do we use it?

Code

First complete the Address Book project, and then select either the Movie Tickets or Bank Account project to complete.

Address Book

Follow along with the Address Book lessons to add the new concepts discussed in the homework. Complete these items for it too:

- Add functionality to record and display a contact's email address.
- Add functionality to record and display a contact's physical address.
- Then, add functionality that allows a user to record *multiple* addresses (email or physical) for a single `Contact`, and what type each address is (ie: "work", "personal", etc.) *(Hint: `Address` will need to be an object with multiple properties saved within the `Contact` object.)*

If you wish, work on this too:

- Add support for *multiple* `AddressBook`s, each with their own unique set of `Contact`s. (For instance, you could have *School*, *Personal*, and *Work* address books to organize different types of friends, co-workers, and acquaintances.)

Movie Tickets

Create a webpage where a user can select the name of a movie, the time of day that they would like to see the movie and their age. The webpage should let them know how much their movie ticket will cost, based on those three factors. Consider that non-"first-release" movies, matinee, and senior tickets tend to be cheaper than the regular priced ticket.

Your constructor and prototype could be called `Ticket` and you can come up with the formula for determining how the price is calculated depending on the input from the user.

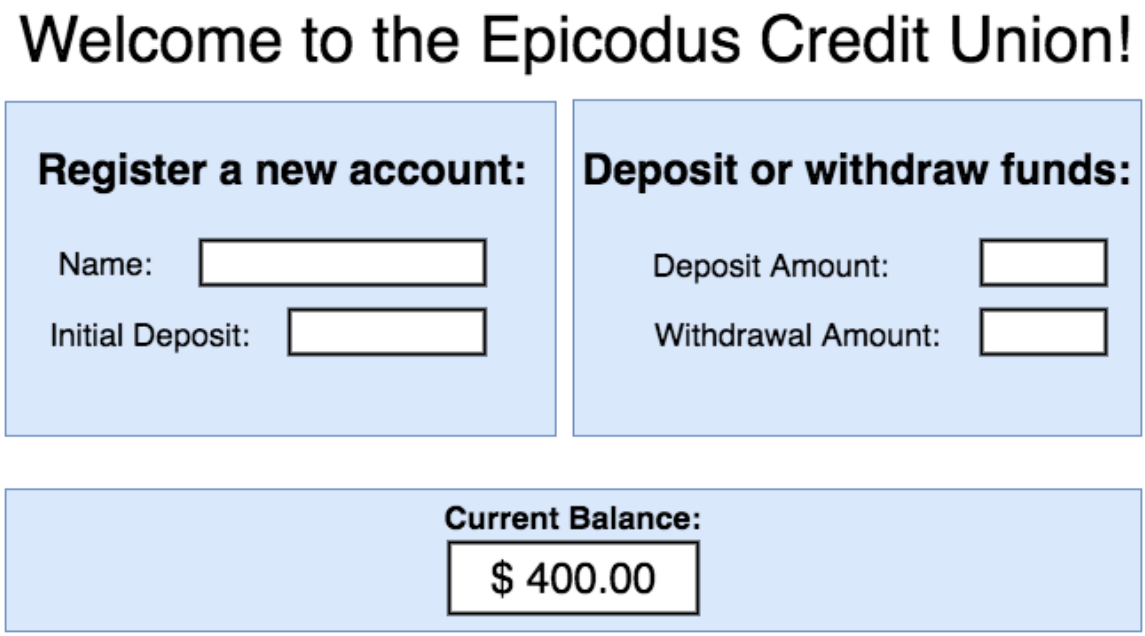
Start by completing your business logic, and then move onto your user interface logic. Use test-driven development to complete your business logic and add your pseudocode tests to your project's README.md. After every passing test, make sure to commit your code.

Bank Account

Use constructor functions and prototypes to create a webpage where a user can create a single bank account with an initial deposit amount. Then allow the user to make withdrawals and deposits, and see the balance of the account.

Start by completing your business logic, and then move onto your user interface logic. Use test-driven development to complete your business logic and add your pseudocode tests to your project's README.md. After every passing test, make sure to commit your code.

Remember to create a `BankAccount` constructor and a prototype that includes methods for deposit and withdrawal and any other properties needed. Here is an example wireframe to help you visualize the app:



Your project is not required to look like this; this is only an example.

Peer Code Review

- Does the web app work as expected?
- Is business logic and user interface logic kept separate?
- Were constructors and prototypes used to create new objects?

Previous (/intermediate-javascript/object-oriented-javascript/imposter-syndrome)
Next (/intermediate-javascript/object-oriented-javascript/introduction-to-the-node-object)

Lesson 23 of 33
Last updated March 23, 2023

disable dark mode

(http://www.epicodus.com)