Asynchronous Code in JavaScript

Let's have some fun working with callbacks and promises! You'll complete these exercises twice, using two different methods. First, you'll use promises! Then, in the exercises at the end of the next subunit, you'll use ***async*** / ***await***!

If this is your first time seeing these challenges, start by solving them with promises.

Once you've solved this using promises continue to the next subunit and after learning about async functions solve these using async and await.

Part 1: Number Facts

1. Make a request to the Numbers API (http://numbersapi.com/) to get a fact about your favorite number. (Make sure you get back JSON by including the ***json*** query key, specific to this API. Details.
2. Figure out how to get data on multiple numbers in a single request. Make that request and when you get the data back, put all of the number facts on the page.
3. Use the API to get 4 facts on your favorite number. Once you have them all, put them on the page. It's okay if some of the facts are repeats.

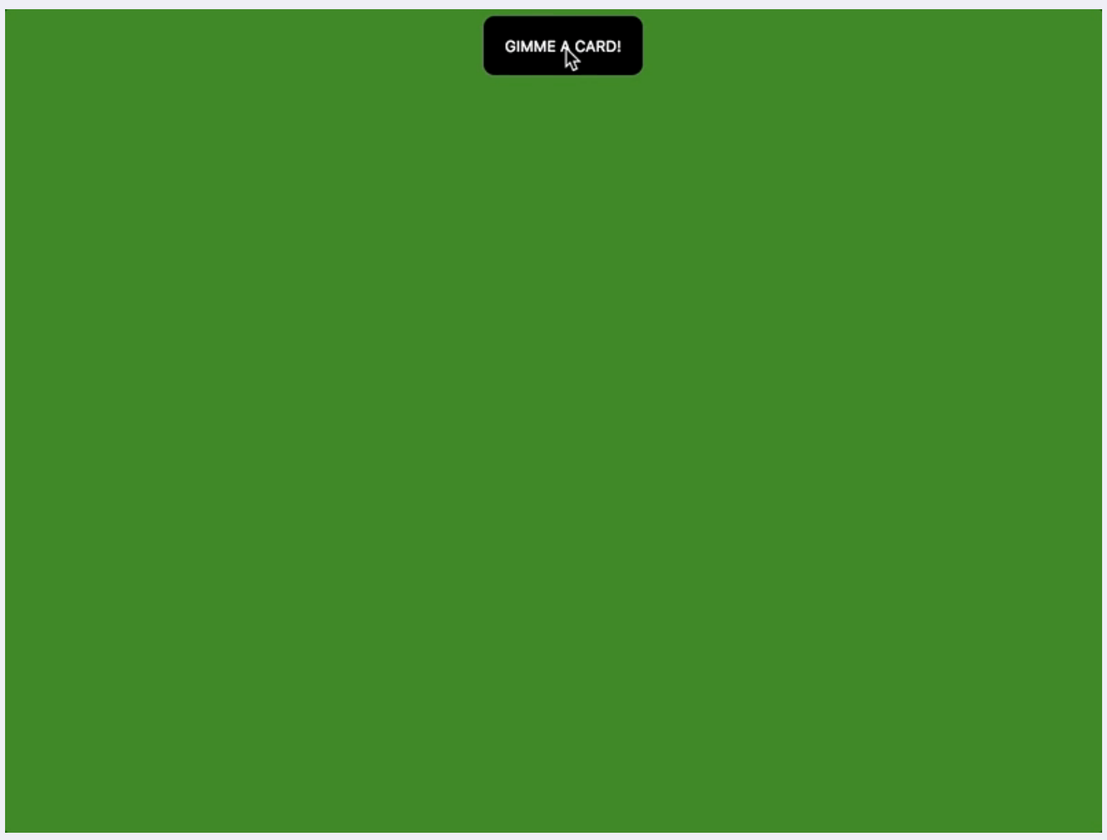
   (Note: You'll need to make multiple requests for this.)

Part 2: Deck of Cards

1. Make a request to the Deck of Cards API to request a single card from a newly shuffled deck. Once you have the card, ***console.log*** the value and the suit (e.g. "5 of spades", "queen of diamonds").
2. Make a request to the deck of cards API to request a single card from a newly shuffled deck. Once you have the card, make a request to the same API to get one more card from the **same** deck.

   Once you have both cards, ***console.log*** the values and suits of both cards.
3. Build an HTML page that lets you draw cards from a deck. When the page loads, go to the Deck of Cards API to create a new deck, and show a button on the page that will let you draw a card. Every time you click the button, display a new card, until there are no cards left in the deck.

Here's how this might look (with styling added):

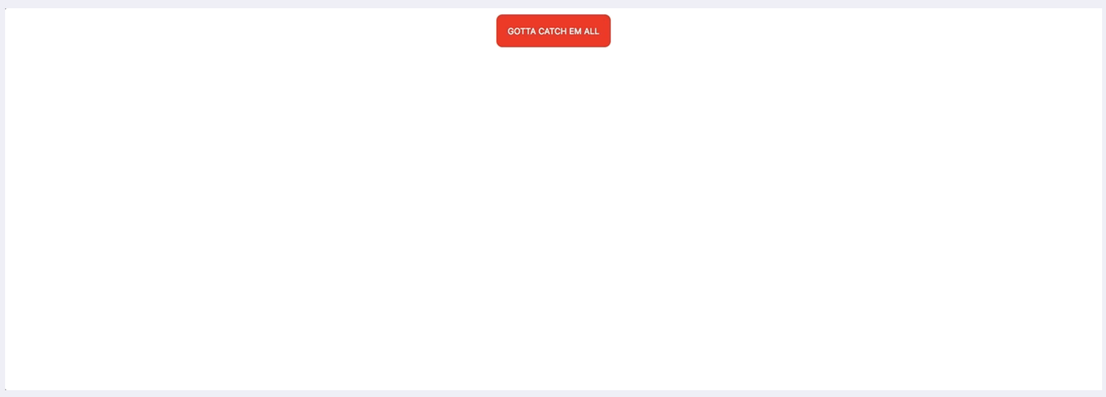


Further Study

1. Figure out how to make a single request to the Pokemon API to get names and URLs for every pokemon in the database.
2. Once you have names and URLs of all the pokemon, pick three at random and make requests to their URLs. Once those requests are complete, ***console.log*** the data for each pokemon.
3. Start with your code from 2, but instead of logging the data on each random pokemon, store the name of the pokemon in a variable and then make another request, this time to that pokemon's ***species*** URL (you should see a key of ***species*** in the data). Once *that* request comes back, look in the ***flavor_text_entries*** key of the response data for a description of the species written in English. If you find one, ***console.log*** the name of the pokemon along with the description you found.

   Example: "ducklett: They are better at swimming than flying, and they happily eat their favorite food, peat moss, as they dive underwater."
4. **BONUS** Instead of relying on ***console.log***, let's create a UI for these random pokemon. Build an HTML page that lets you click on a button to generate data from three randomly chosen pokemon. Include the name of the pokemon, an image of the pokemon, and the description of its species which you found in 3.

Here's how this could look:



Good luck!

Solution

View our Solution