Rev. 08Aug2019

Exercise 38

HELPING OUT TONY

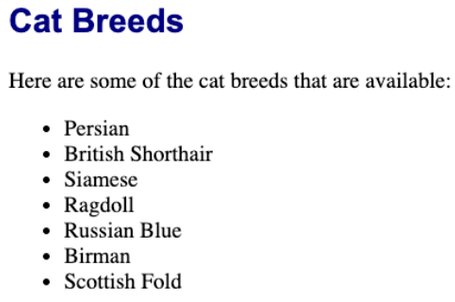
Remember Tony from the last ~~episode~~ exercise? He's in a jam and he needs your help. He finished his web page last night, then dashed off to hop a plane to Early Tibet. And then you got a phone call from him.

Tony: *Hey, I'm in a jam! You know that web page about cat breeds I did yesterday? Well, I forgot to give a class to each of the breeds, so now it looks crummy. And if the boss sees it, I'm going to get in trouble. Can you help me out?*

You: *I'd like to help, but the boss just implemented a policy that JavaScript developers aren't allowed to touch finished HTML code.*

Tony: *Yeah. I was hoping you might have something in your bag of tricks?*

And, of course, you do. Or, you're about to. First, you look at the web page in a browser.



Cat Breeds

Here are some of the cat breeds that are available:

- Persian
- British Shorthair
- Siamese
- Ragdoll
- Russian Blue
- Birman
- Scottish Fold

Yes, it definitely could use some styling. Tony says he already create a class called `breed` that will style them properly, but you look at his HTML code:

```
<div>
  <h2>Cat Breeds</h2>
  <p>Here are some of the cat breeds that are available:</p>
  <ul>
   <li>Persian</li>
   <li>British Shorthair</li>
   <li>Siamese</li>
   <li>Ragdoll</li>
   <li>Russian Blue</li>
   <li>Birman</li>
   <li>Scottish Fold</li>
  </ul>
</div>
```

And you see what Tony was talking about: there's no `class` property on his `<li>` elements.

At first, you think you have a solution — one you've used before:

```
let breeds = document.querySelector('li')
let classProperty = document.createAttribute('class')
classProperty.value = 'breed'
breeds.setAttributeNode(classProperty)
```

You run the code and look at the web page in a browser:

Cat Breeds

Here are some of the cat breeds that are available:

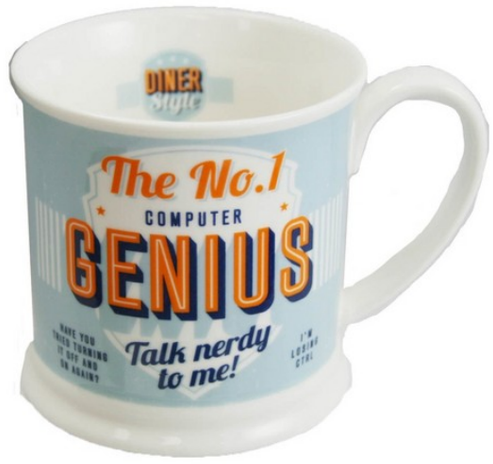
- Persian
- British Shorthair
- Siamese
- Ragdoll
- Russian Blue
- Birman
- Scottish Fold

But it only changed the *first* `<li>`. What's up?

There's nothing wrong with your code. `document.querySelector` worked — but it turns out that it only returns the *first* element it finds. A quick Google search and you see that there is also a `document.querySelectorAll` that returns *all* of the matched elements. That's what you want.

But there's a problem.

The function, `document.querySelectorAll`, returns something called a *nodelist*. It's a group of all the HTML elements. Did you, by chance, think "Arrays are good for dealing with groups of things"? If you did, you are officially awarded this mug:



While you gloat, the rest of us will figure this out. Yes, we do want to use an array. Then we could use the array's `forEach` method, passing it a function that would add a `class` to each of the elements in the array. There's only one problem: a nodelist isn't an array.

But that turns out to be easy to solve. The code snippet...

```
let breeds = document.querySelectorAll('li')
let breedsArray = Array.prototype.slice.call(breeds)
```

...will convert the nodelist into an array! And from there, you can write a function that will be passed into `breedsArray.forEach` function. Your function will accept a single argument. For every element inside `breedsArray`, your function will be called once and will be passed the current element. Then, inside your function body, you can create a `class` attribute, assign it a value, and attach the attribute to the element.

- [ ] Write the code! It's really not that difficult. Follow the comment instructions in `index.js` and, if you get stuck, take a peek at my `answer.js`. But, really, you've done this sort of thing multiple times. Just go slow.