CSS Advanced Selectors Lab

Developers tend to add more ids and classes to our HTML because we often don't know CSS selectors as well as we should. In fact, up to now, we've done exactly that. In this lab we'll fix some of those things.

Multiple selectors

1. All our headings (h1 - h6) have some features in common. Rather than having six different styles, let's combine them. Write one selector for all headings combined and apply this style to them:

```
{ font-weight: 500; line-height: 1.1;}
```

2. The h1 through h3 all have certain things in common that the h4-h6 do not. Add a single selector for only h1 through h3 and give them this style:

```
{margin-top: 20px; margin-bottom: 10px;}
```

3. Check it out. Did it work?

Descendant selectors

Descendant selectors help us by reducing the clutter on a page. Instead of applying a class to every element in a list, we can add the class to the list's parent and then select all the children of that parent. Here's an example.

4. On the index page you have featured items. Each <img> has a class of productImage. This is a waste. Remove all those classes from the HTML.
5. In site.css, change .productImage class to "#featured img" which means select all <img>s that happen to be inside something with an id of #featured.
6. Browse to that page. You should still see the images properly sized but now we have cleaner styles and cleaner HTML.
7. Let's do another. You have site links in all of your headers. You may have styled them so they're on the same line and you may have used classes to apply styles. If so, remove those classes.
8. Run and test. You should see them stacked again.
9. Line them back up by changing the selector in site.css to will point to those s as descendants of your nav. Adjust it as you see fit by pointing to the <ul> and/or the <nav> and/or the .jumbotron class.

Pseudo class selectors

Let's give the user a visual indicator when he/she is focused on an input box. We'll do that by applying a style when any input box has focus. And we'll need a pseudo-class to do that.

10. Edit site.css. Add a new style that will produce a blue glow. Your style will have properties kind of like this:

```
box-shadow: inset 0 0 8px rgba(102,175,233,.6)
```

But you'll need to write the selector. You want it to apply to everything with a class of form-control but only when they have focus. (Hint: you'll use the ":" character in your selector).

11. Browse to any pages with an input box and adjust the glow until you think the effect is ... well ... effective.

Note that all future form-controls will immediately get this feature. Nice.

Working with multiple advanced selectors

Now let's take on a pretty advanced example. It will be a challenge, but hang in there and experiment.

12. At the bottom of every page you added a class of *list-inline* to all the s. That's a waste. Instead remove all of the classes from each <li> and move that class to the parent <ul>. Now it shold say <ul class="list-inline">. As you know, this will mess up our layout because it is the s that need the style.
13. We can fix it! We simply need to change the CSS selector to point to all s that are a child of a <ul class="list-inline">. (Hint: you should use a child selector or a descendant selector).
14. Browse to any page and you should see them side-by-side again.
15. Now for some fun! Add this style:

```
li::before {
  content: "|";
  padding-right: 15px;
}
```

    You've just created a pseudo-element! It's an element that is added dynamically by the browser without adding it in the HTML structure.

16. Browse to multiple pages. You should see a "|" before every <li>. Which is cool! But we only wanted it in front of s inside a <ul> with a class of list-inline. Please change the selector so that only those s will get a bar in front of them. (Hint: You'll want to use a child or descendant selector again. You'll be putting a selector in front of the li::before).

One more step. If you got the above step right, you'll see the bar in front of every <li>. But we don't want it before every <li>, only between the s, so we need to exclude the first one. Another way of saying that is "We want the bar in front of every <li> that is after the first one."

17. Work on that for a few minutes. Try to use a sibling selector to make that happen. Don't get discouraged; this is a tough exercise. If you can't get it, your instructor can go over the solution with the whole class when the lab is finished.