Lesson 3 Challenge

CSS Style Inheritance:

A Basic Example: One Level of Inheritance

To see how inheritance works, start by adding a single tag style and see how it affects the tags nested inside. The next two parts of this tutorial will build upon your work here, so save the file when you're done.

1. **Open the file <u>inheritance.html (in lesson files)</u> in sublime text.**Now add an internal style sheet to this file.
2. Now, you'll create a style that applies to all <p> tags.

```
p{
        color: #666666;
}
```

3. **Open the page in a web browser to preview your work.**

The color of the page's four paragraphs has changed.

Using Inheritance to Restyle an Entire Page

Inheritance works with class styles as well—any tag with any kind of style applied to it passes CSS properties to its descendants. With that in mind, you can use inheritance to make quick, sweeping changes to an entire page.

You'll add a new style below the <p> tag style you created.

The whole thing should look like this:

```
.pageStyle{

    font-family: "Helvetica Neue", Arial, Helvetica, sans-serif;
    font-size: 18px;
    color: #7C3100;
    width: 900px;
    margin: 0 auto;
}
```

This completed class style sets a font, font size, and color. It also sets a width
and centers the style on the page

4. **Find the opening <body> tag (just a couple lines below the style you just created), and then type class="pageStyle".**

Thanks to inheritance, all tags inside of the body tag
(which are also all the tags visible inside a browser window) inherit this style's properties and therefore use the same font.

5. **Save and preview the web page in a browser.**

Your class style has created a seamless, consistent appearance throughout all text in the body of the page. Both headings and paragraphs inside the <body> tag have taken on the new font styling.

The page as a whole looks better, but now look more closely: The color change affected only the headings and the bulleted list on the page, and even though the style specified an exact font-size, the headline text is a different size than the paragraphs.

Inheritance doesn't always apply, and that isn't necessarily a bad thing. For some properties, inheritance would have a negative effect on a page's appearance. Margins, padding, and borders (among other properties) don't get inherited by descendant tags—and you wouldn't want them to, as you'll see in this example.

Locate the p style

You'll indent the paragraphs on the page by adding a left margin.

Add three properties to the style so that it looks like this:

```
p{
    color: #666666;
    margin-left: 50px;
    padding-left: 20px;
    border-left: solid 25px  #FF6A00;
}
```

The margin-left property indents the paragraph 50 pixels from the left; the padding property indents the paragraph text 20 pixels from the border.

4. **Save the file and preview it in a web browser.**

Notice that all of the <p> tags are indented 50px from the left edge of the browser window and that they each have a thick brown border on the left. However, the tags inside the <p> tag (for example, the <em> tag) don't have any additional indentation or border . This behavior makes sense:
It would look weird if there were an additional 50px of space to the left of each <em> and each <strong> tag inside of a paragraph

To see what would happen if those properties were inherited, edit the p selector so that it looks like this: p, p * which makes it into a group selector.

The first part is just the p selector you already created. The second part—**p ***—means “select all tags inside of a p tag and apply this style to them.”

```
p, p * {
	color: #666666;
	margin-left: 50px;
	padding-left: 20px;
	border-left: solid 25px  #FF6A00;
}
```