INFO/CS 1300: Lab 4 (9/15)

By Katerina Prastakou and William Chung

Due at the end of your lab section. Show the TA your work to get your participation credit.

Overview

In this lab, we'll cover the box model, floating, and two important elements: the *div* and the *span*.

What You Need

Bring your **laptop** to your lab section. You may also want to **print** out this lab.

Part 1: The Box Model

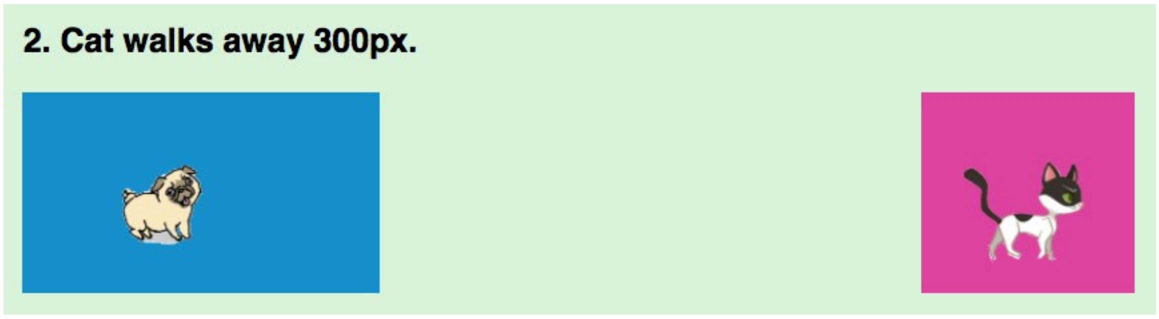
Create a folder called **lab4-<NetID>** (example: lab4-kjh235).

Download the lab4.zip file from CMS and unzip the contents to your folder. We have created an HTML file, **index.html**, designed to help you understand today's topics.

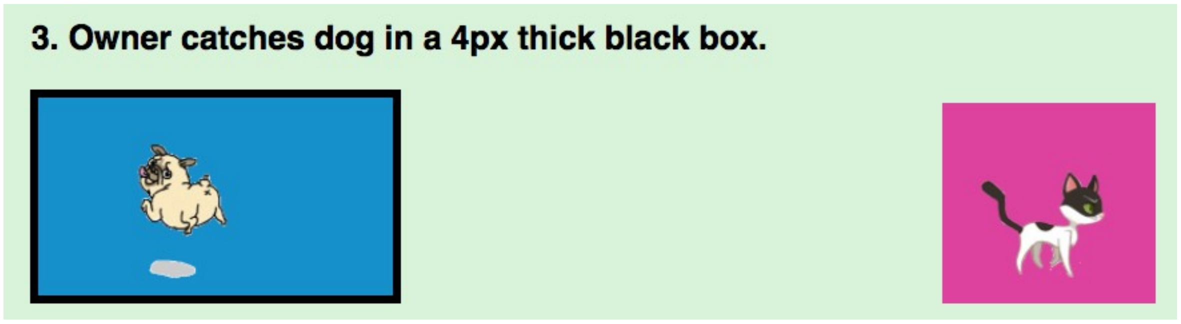
The first part of this lab involves telling a short story about a fugitive cat, excited dog, and hapless owner. **Do not change the HTML *at all* for this part**. Only change the styles in the appropriate part of **all.css**.

This story has five chapters, and right now all of them look the same. Let's fix this!

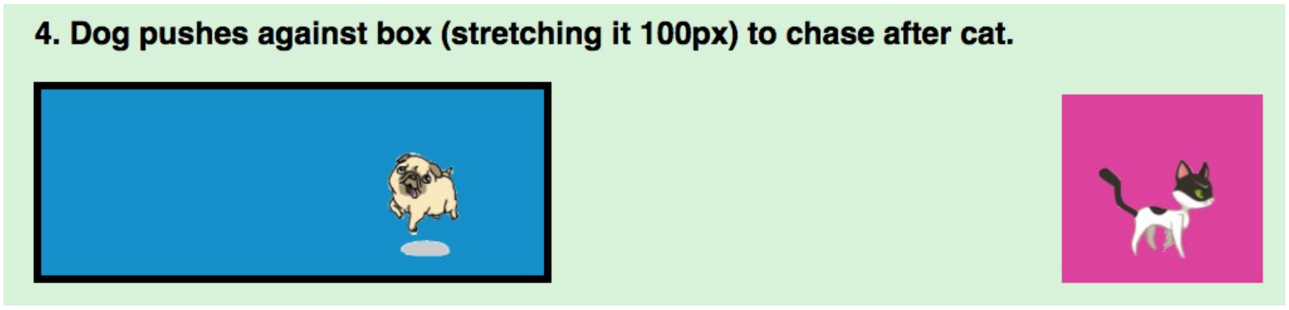
1. Chapter 1 is okay as-is, so you can just leave it.
2. For chapter 2, you'll need to use your knowledge of the box model (padding, border, and margin) to show that that cat has walked 300 pixels to the right:



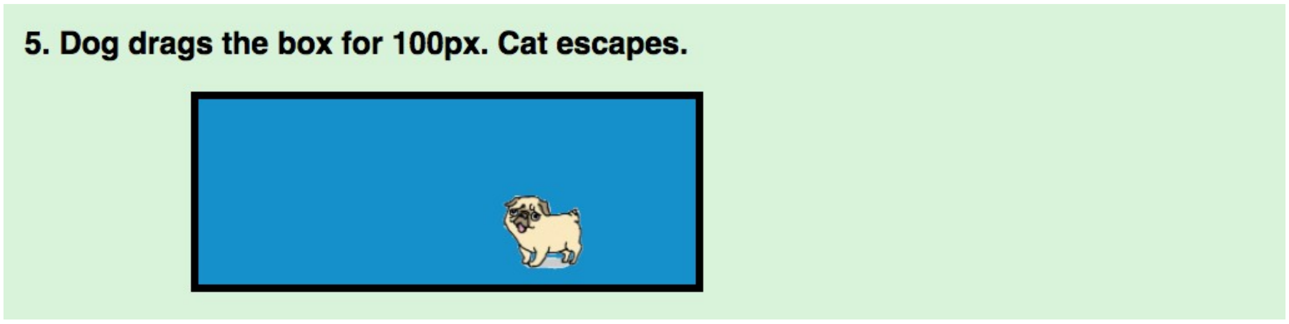
3. For chapter 3, you'll want to show how the owner has caught the dog in a box (again using the box model). Remember that the cat is still 300 pixels away!


3. Owner catches dog in a 4px thick black box.

4. For chapter 4, the dog chases after the cat, stretching the box so that it is 100 pixels larger. Again, you will need to show this with the box model, and maintain things from the previous chapter.


4. Dog pushes against box (stretching it 100px) to chase after cat.

5. For the final chapter (chapter 5), the dog drags the box 100 pixels to the right, which you will once again need to demonstrate using the box model. The cat also finally manages to escape. Remember you can't modify HTML for this part, so you'll have to use CSS (Hint: use your documentation lookup skills to learn about the "visibility" property).


5. Dog drags the box for 100px. Cat escapes.

Part 2: Floats

You've seen floats in lecture, but let's get some hands on experience with them (after a quick aside).

Lorem Ipsum

You will notice that the text for this section all starts with "Lorem ipsum" and doesn't appear to be in English. "Lorem Ipsum" is from a Latin text from the works of the Roman orator Cicero, and website designers use it to generate lots of text so that they can see what a layout looks like with text without filling in the actual content.

Why Latin? If you used random English text, you might be distracted by what it actually *says*. With Latin, you can just see how text looks and feels on the page without any impact from the actual content of the text. Latin is also close enough to English that you still get a good sense of what English text will look like. If you used Egyptian Hieroglyphs or the Russian Cyrillic Alphabet, you wouldn't be able to see what English letters look like with your given text styling.

And if Latin seems boring, you can find lots of cool Lorem Ipsum generators online, including variants like Bacon Ipsum, DJ Khaled Ipsum, and many more.

Note: For assignments in this class, you need **real content**. Lorem Ipsum is a tool for testing design, not a substitute for final content.

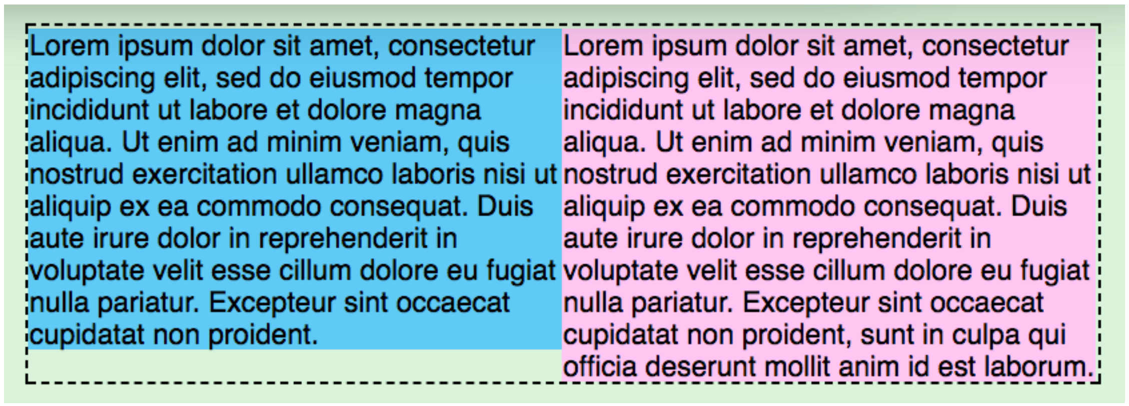
Two-Column Layouts

Now, on to floats. First, we'll look at creating a two column layout. Again, you should **not change anything in the HTML file**. Just modify **all.css**.

We have two simple paragraphs, one blue and one pink, that are in a vertical stack. Instead, we want them to be horizontal, in two columns. Let's float the blue paragraph left by setting the float property of the paragraph with ID `#one` to the left. *Note: you can float any block element, not just paragraphs!* We just want to use something familiar for this example.

If you refresh in Chrome, you'll see… nothing. **Floats only work on containers that have a specified width!** Otherwise, the behavior is undefined, and you probably won't get any floating. Let's add a width of 300px (half the container size) to `#one` and see it float to the left.

You'll see that it still looks a little odd. The text from the pink paragraph wraps around and under the blue paragraph. We can fix this by giving the right column (`#two`) a left-margin of 300px (the same as the width of the right column). You could float the right column to the right. The final result should look like this:

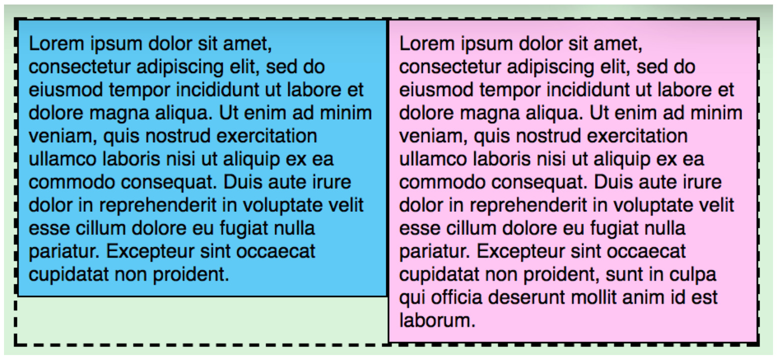


Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Two-Column Layouts (with padding and borders)

Now, let's look at the next set of paragraphs. These two have a border of 1 pixel and padding of 9 pixels. Try copying the styles from the previous two-column layout into the styling for these two new columns (`#three` and `#four`). You might be surprised by the result! When you do floating, remember that the border and the padding are added *in addition* to the width. So if you want your left column to be 300 pixels from border to border, the width property will have to be *less* than 300 pixels. To calculate the horizontal space a container takes up, use (*left margin + left border width + left padding + width + right padding + right border width + right margin*). This container will have no margin, a border of 1 pixel on both sides, and a padding of 9 pixels on both sides, so you need to find a width such that (0 + 1 + 9 + *width* + 9 + 1 + 0) = 300. The final result should look like this:

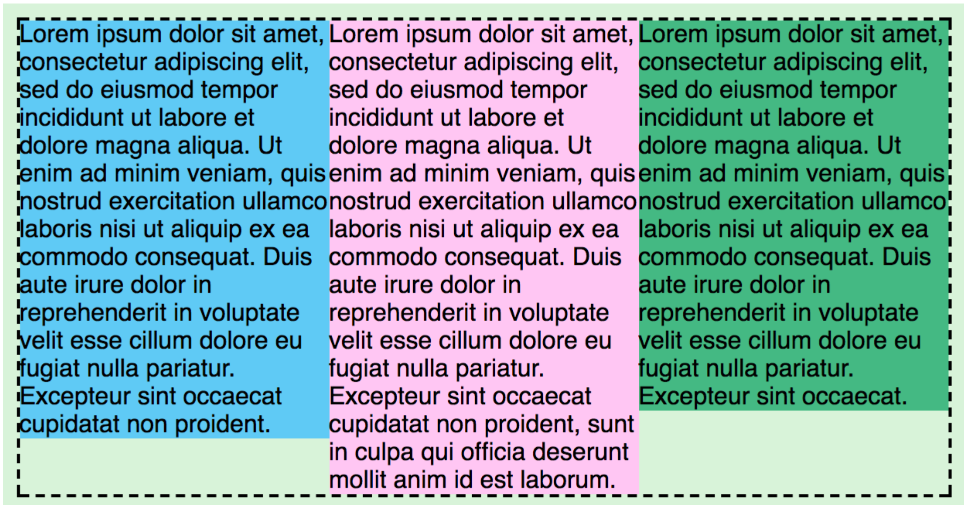


Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Three-Column Layout

Now let's create a three column layout. Float the blue paragraph to the left, the green paragraph to the right, and leave the pink paragraph in the middle. The final result should look like this:



Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident.

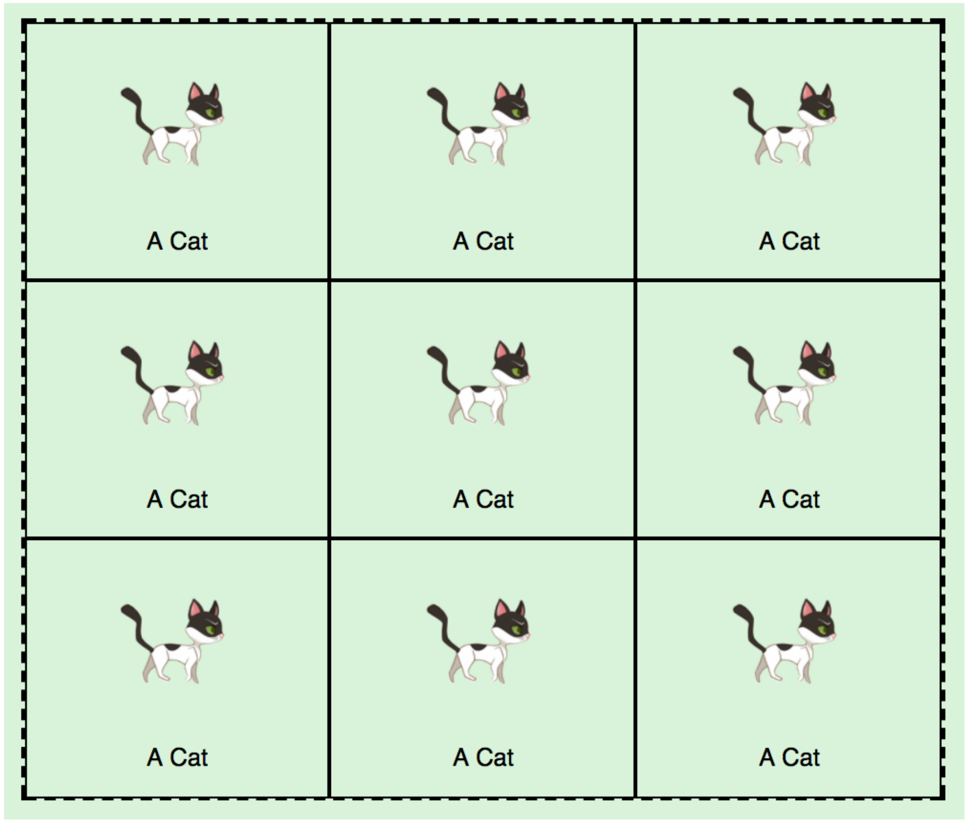
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat.

Image Gallery

This next sections include a series of cats in containers with the class `.catContainer`. Instead of 9 blocks stacked on top of each other, we want to create a 3 by 3 grid. We want three containers to fit exactly in the 600 pixel container, so give the "`.catContainer`"s a width of 198px (they have a 1 pixel margin on both sides, so the container will take up 200 pixels horizontally). When you make this change, you'll notice that the cat containers are smaller, but they still each wrap onto their own line.

To fit them all next to each other, we will need to use floats. Modify all.css so that the "`.catContainer`"s float left. Now, refresh you page. Are you surprised by what happens? The final result should look like this:



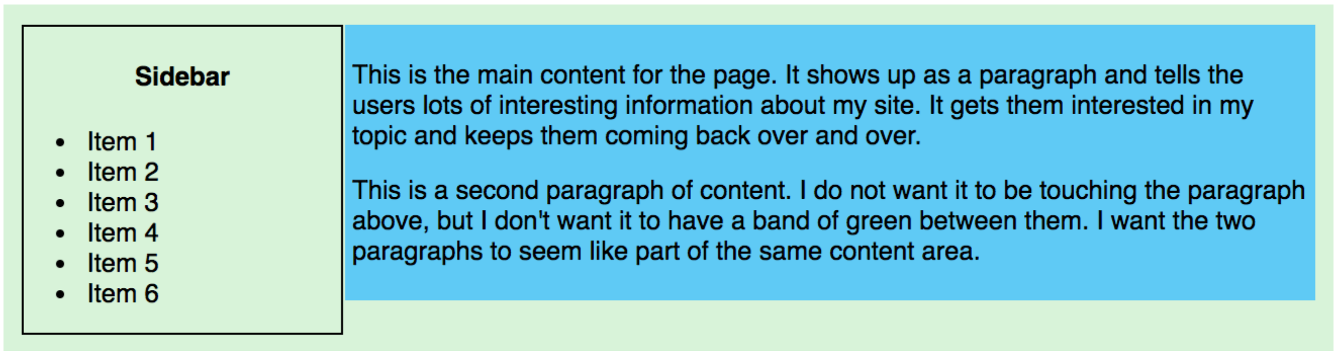
Part 3: Divs and Spans

You may have heard of `div`s and `span`s before (or on Piazza), so what are they?

Divs

Divs are generic block elements. They let you create generic containers on your page so that you can style things like sidebars, images with captions, alert blocks, and many others. With your div containers you group related content for styling purposes. You will find that they are quite useful for creating complicated websites.

Let's look at a case where divs can be very useful. Say I wanted to create a page with a sidebar that has a header and a list, with a main content area containing two paragraphs with a blue background, like this:



Sidebar

- Item 1
- Item 2
- Item 3
- Item 4
- Item 5
- Item 6

This is the main content for the page. It shows up as a paragraph and tells the users lots of interesting information about my site. It gets them interested in my topic and keeps them coming back over and over.

This is a second paragraph of content. I do not want it to be touching the paragraph above, but I don't want it to have a band of green between them. I want the two paragraphs to seem like part of the same content area.

If you scroll down to the "Divs" section of index.html, you can see a first attempt at creating the layout above. It clearly does not look like what we have above, because I applied the styles *directly to the elements* in the sidebar and main content rather than to a container surrounding these elements.

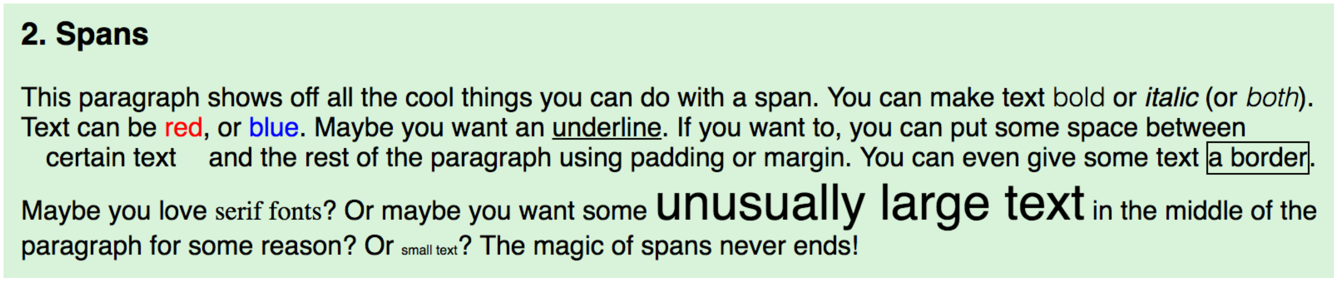
Go into the HTML in index.html and add two divs **inside the areas denoted by comments**. One div should have the class `sidebar2` and the other should have the class `mainContent2`.

Then, modify the `sidebar1, sidebar2, mainContent1`, and `mainContent2` classes in all.css so that the HTML looks like the picture above (hint: *almost* all of the CSS is already there; you just have to move it).

One last point: you should only use divs when you *need* them. HTML5 provides many elements like, header, section, article, main, aside, footer, for describing your content. If you are typing text and a paragraph element makes sense, use <p>. If you are wrapping a navigation bar, use <nav>. Divs should be part of your toolset, not the only tool you have.

Spans

Spans are a generic inline element. They let you style specific pieces of inline text, so you can make them bold, or italic, or green, or a different font, etc. You just need to use spans in conjunction with CSS classes. Look at the HTML for the spans paragraph, and you will see spans spread throughout the paragraph. Each of these spans has a class on it, so you should go into **all.css** and style these classes appropriately (so `.redText` should have red text). Once you are done, the paragraph should look much more interesting than it did before:



2. Spans

This paragraph shows off all the cool things you can do with a span. You can make text bold or *italic* (or *both*). Text can be red, or blue. Maybe you want an underline. If you want to, you can put some space between certain text and the rest of the paragraph using padding or margin. You can even give some text a border.

Maybe you love serif fonts? Or maybe you want some unusually large text in the middle of the paragraph for some reason? Or small text? The magic of spans never ends!

User Testing

If you finish early, use this as an opportunity to user test your Project 1 webpage. The more feedback you get, the better your site will be (and the more notes you will have for your rationale).

You may also try sketching out a layout that uses floats and try integrating floats into your Project 1.

Credit

Once you have finished this task, show your page to your TA to make sure you have done everything correctly and get checked off for participation in the lab.