"The Web Design Workshop" Decal – Spring 2014
Programming Handout 4

Goals:

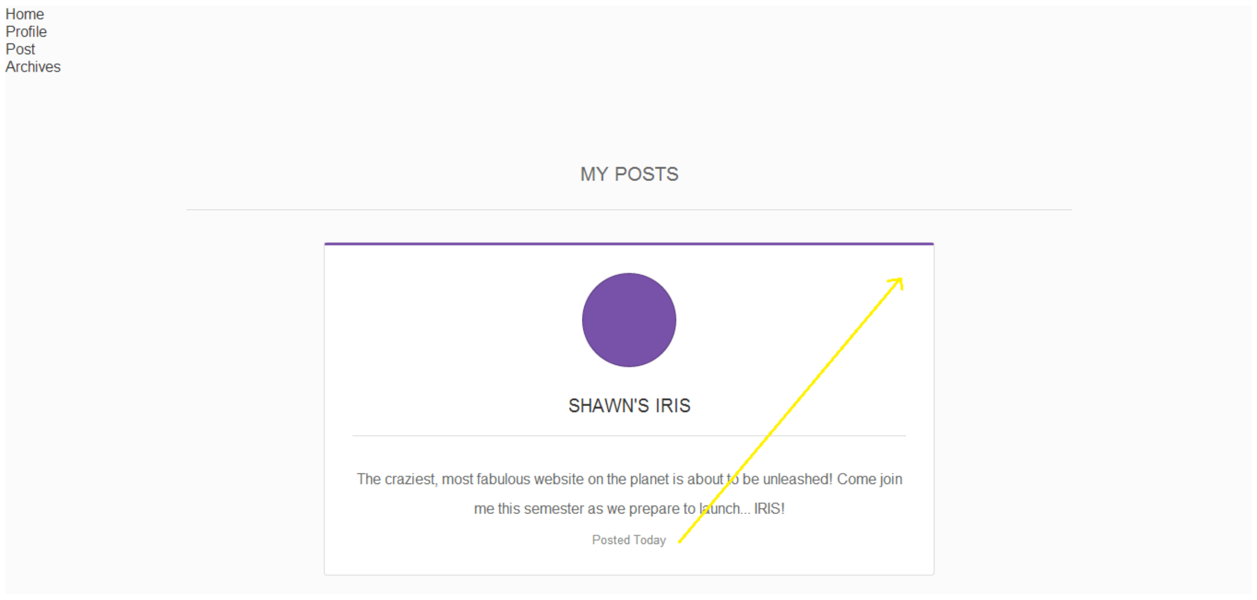
1. Using Position Fixed & Floats
2. Using Position Absolute and Relative

Before we begin, make sure you have downloaded the *handout4.zip* file. Unzip this file, and you should see 2 files inside the *handout4* folder: *positioning.html* and *styles.css*

1 – Using Position Fixed & Floats

1. Open both *positioning.html* and *styles.css* in Sublime. Take a look at the *positioning.html* elements to get an understanding of the HTML structure. Then, pull up *styles.css*. We will start adding styles to *styles.css*.

   If curious, you can open up *positioning.html* in your browser and you will see this:


Home
Profile
Post
Archives
MY POSTS
SHAWN'S IRIS
The craziest, most fabulous website on the planet is about to be unleashed! Come join me this semester as we prepare to launch... IRIS!
Posted Today

   Here's our goal: To make the top left text (Home, Profile, etc.) into a **fixed** menu. Later, we will use **position: absolute** to get our "Posted Today" date on the top-right corner of our post.

2. First, let's understand the *#menu-container*.

   #menu-container will be the menu element that stretches across the screen, and will be fixed as we scroll down the page.

   #menu will be the div inside *#menu-container*, that is set to 960px and is centered in the middle of the screen (we learned in design section to contain our elements in some

width, say 960px, or else our text/elements span the entire browser window.

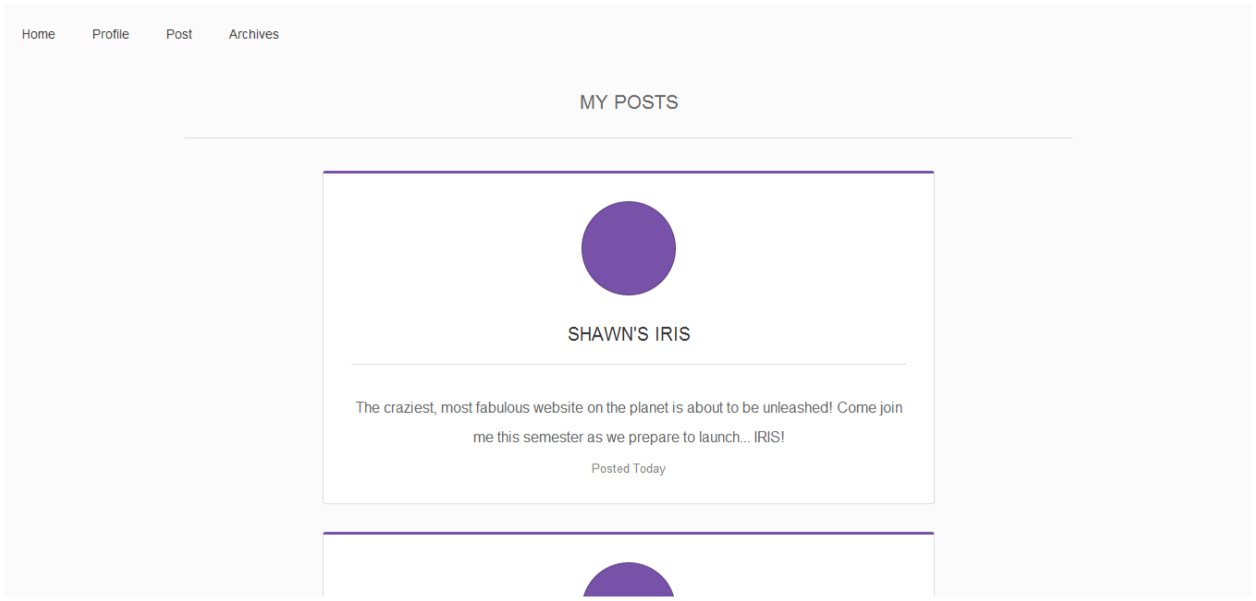
.menu-item will be the individual menu items that we want horizontally, away from the left side of the browser window.

Now, let's first make our **.menu-item** elements next to each other, using **float:**

float: left;
padding: 25px 20px;
font-size: 25px;
cursor: pointer;

We added padding too, to space the elements from each other. We also added *cursor: pointer,* so that hovering over a *.menu-item* causes the mouse to transform into a pointer.

Open or refresh your *positioning.html* page in the browser. You get:



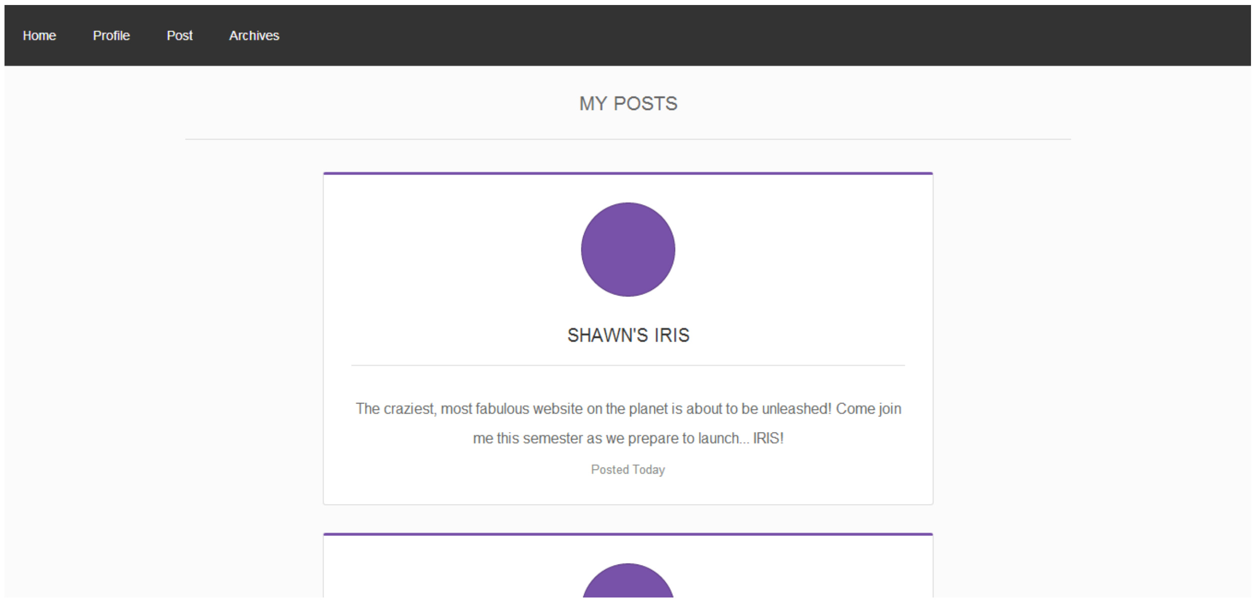
The craziest, most fabulous website on the planet is about to be unleashed! Come join me this semester as we prepare to launch... IRIS!

Posted Today

Woah! Our menu is already looking good!

3. Next, let's add a **background** to our menu that stretches the browser window. Also let's make it **position: fixed** and throw in a few styles. Find *#menu-container*, and add the following styles:

   background-color: #333;
   color: white;
   position: fixed;
   width: 100%;



MY POSTS

SHAWN'S IRIS

The craziest, most fabulous website on the planet is about to be unleashed! Come join me this semester as we prepare to launch... IRIS!

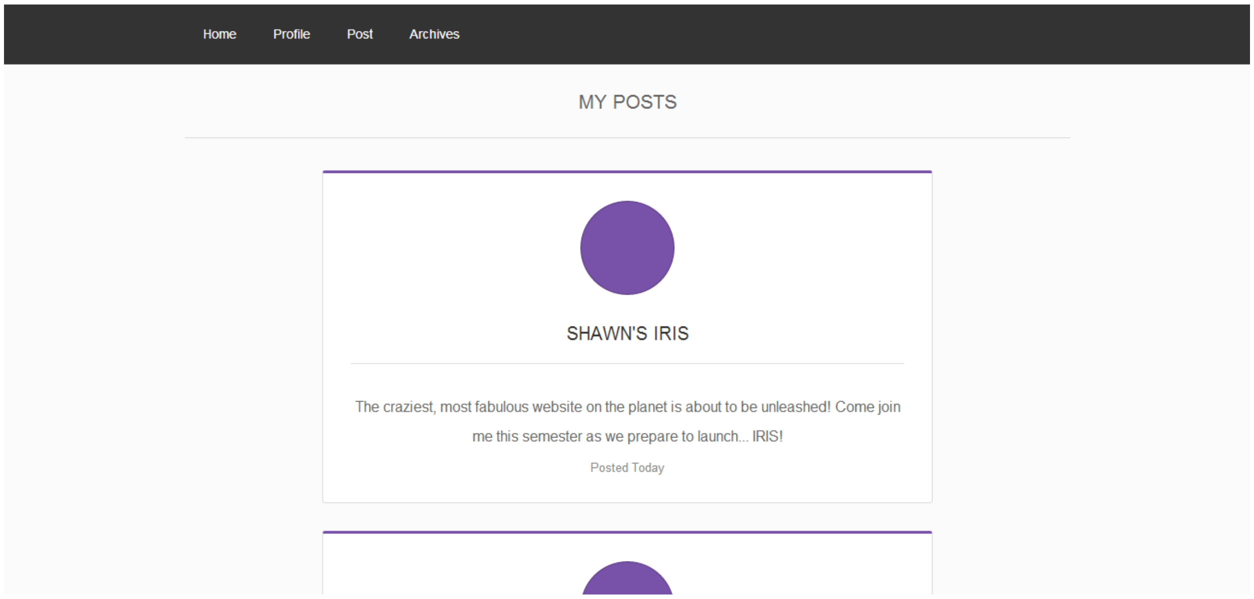
Posted Today

Wowza! A lot has happened. Try **scrolling** down the page, and you will notice that the menu is **fixed**!

4. What do we have so far? We have a fixed menu, stretching the entire page, and menu elements that float left to each other (and spaced away by padding).

   However, we don't want the *#menu-item* elements to hug the left side of the browser window! We want it to hug the left side of a **centered 960px div**! Let's add the last styles to *#menu:*

   margin-left: auto;
   margin-right: auto;
   width: 960px;



The craziest, most fabulous website on the planet is about to be unleashed! Come join me this semester as we prepare to launch... IRIS!

Posted Today

Nice! Notice how much better a menu is in a **fixed width container**.

For fun, you can also add a new style:

```
.menu-item:hover {
        background-color: #444;
}
```

We will talk about *:hover* and other *CSS pseudo-classes* next week. What the above style will do is if you **hover** over a *.menu-item*, the background-color will change! Cool right??

Extra:

- What happens if you do a *float: right* instead of a *float: left?* How could you fix the problem?
- Know that you can also make menu items using *display: inline-block*. This actually works better if you have a **centered menu**, as opposed to a floated left menu in this example.

2 – Using Position Absolute and Relative

1. Next, we want to **move** our “Posted Today” date (and the other “Posted …” dates for the other posts) on the top-right corner of the post. What things should we do?

   First, let’s find the first *.post* element. Notice that it contains the *.date* element with “Posted Today” as the content.

If we want to shift *.date* to the top-right of the box, we could first make *.post* a **position: relative**, and make *.date* a **position: absolute.** If we don't do this, *.date* will not move relative to *.post* if you use top, left, right or bottom!

```
.post {
    position: relative;
}
.date {
    position: absolute;
}
```

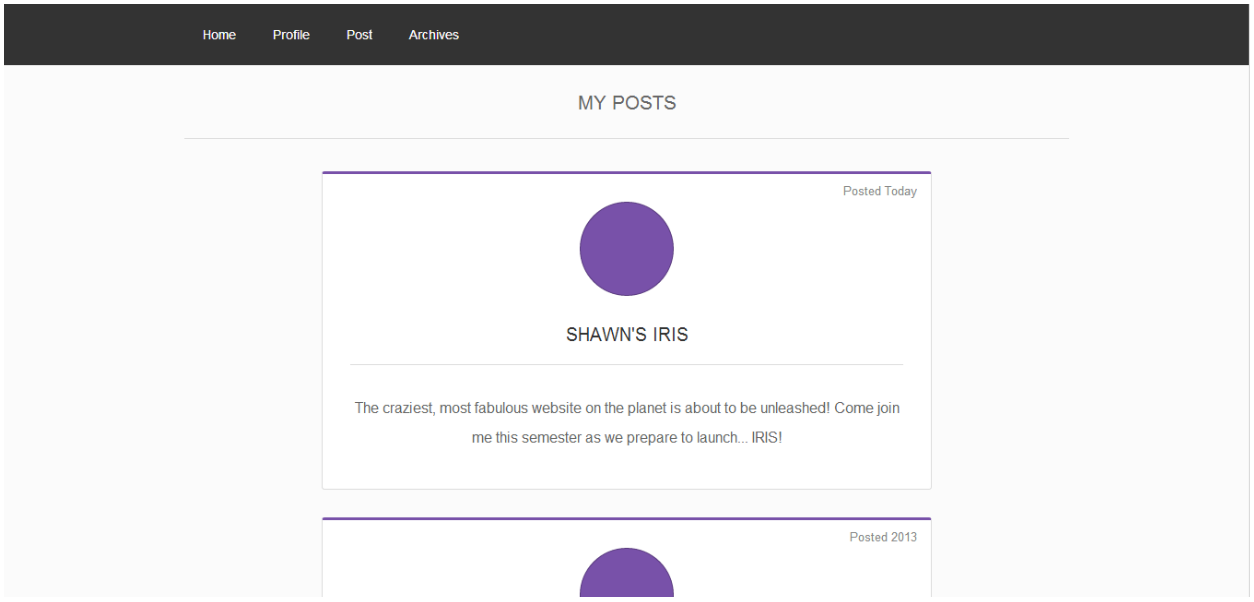
But we're not done!

You made *.date* **position: absolute**, but where (relative to *.post*) do you want *.date* to be?

Let's also add **top** and **right** values to *.date!*

```
.post {
    position: relative;
}
.date {
    position: absolute;
    top: 10px;
    right: 15px;
}
```

Our product?



MY POSTS

Posted Today

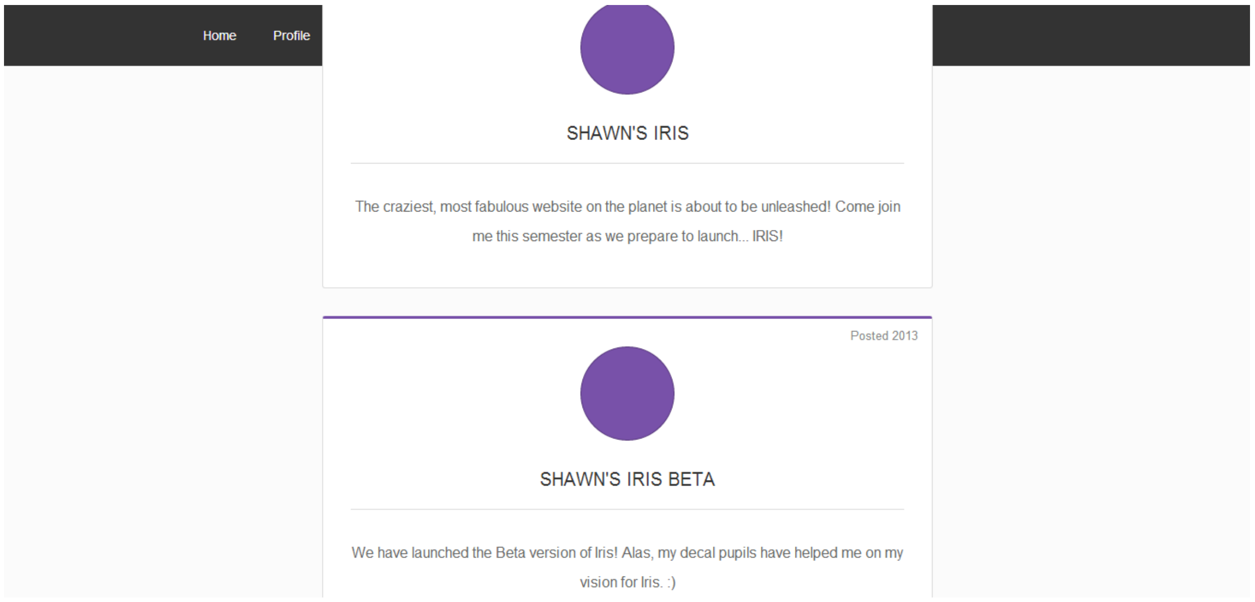
SHAWN'S IRIS

The craziest, most fabulous website on the planet is about to be unleashed! Come join me this semester as we prepare to launch... IRIS!

Posted 2013

Awesome. We have a date on the top-right corner of our post!

2. Hmm. We seem done! … Or are we?


Home
Profile
SHAWN'S IRIS
The craziest, most fabulous website on the planet is about to be unleashed! Come join me this semester as we prepare to launch... IRIS!
Posted 2013
SHAWN'S IRIS BETA
We have launched the Beta version of Iris! Alas, my decal pupils have helped me on my vision for Iris. :)

Scroll down the page and you get the above image.

What did we do… wrong?

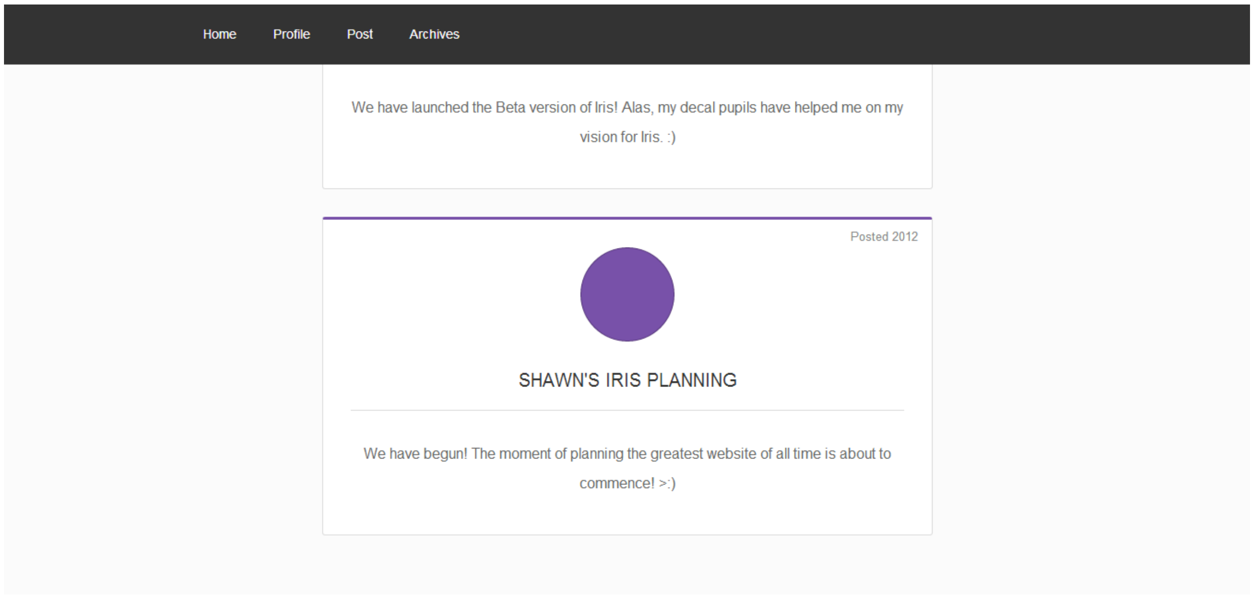
Well, nothing really. This is a common problem when you have lots of **position: absolute, position: relative** and **position: fixed** elements on the page.

Elements compete over layers. Say you have two **position: absolute** elements. You create the first one, and then you create a second one in the same spot as the first. The second element will show instead of the first. This is because elements are created in an order, and the newest **absolute/relative/fixed** element will be given a higher layer, or in terms of CSS: a higher **z-index**.

This gets crazier when you may have dozens of these elements on your page. To assure that **our** menu is over *all* of the **absolute/relative/other fixed** elements on the page, set *#menu-container*'s **z-index** to a high value, such as 999999;

z-index: 999999;

Now try scrolling again!



We have launched the Beta version of Iris! Alas, my decal pupils have helped me on my vision for Iris. :)

Posted 2012

SHAWN'S IRIS PLANNING

We have begun! The moment of planning the greatest website of all time is about to commence! >:)

Aha! We fixed it! ☺

Keep in mind, *z-index* only applies to **position: relative, position: absolute** and **position: fixed** elements.

Congratulations! Let's review what we learned:

- We learned to float elements with *float: left*
- We learned how to make a horizontal menu, in a fixed width div
- We learned how to make use of *position: absolute* and *position: relative*
- We learned to use *z-index*