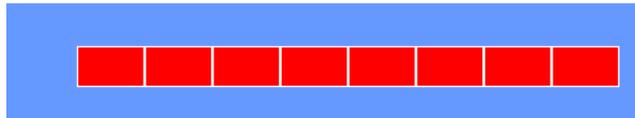


TCS 007D Box Model In-Class Exercise

For this exercise, we are going to use our knowledge of the box model to build a brick wall. You would probably just use a picture of a brick wall in you website, but this exercise will help to reinforce the concept of the box model.

1. Open `brick_wall.html` and take a look at the HTML and CSS. There are a number of `<div>` elements in the body. A `<div>` is a generic container element. They have no semantic meaning, and are widely used for CSS styling. There is one that I've used to designate a row of bricks (`.brick_row`) and a number of nested divs that will represent individual bricks (`.brick`). I've also written some CSS rules for you already. You won't need to change any of these. There is one CSS rule we haven't discussed yet: `float`. We will get to it later, but for now just understand that it allows the bricks to line up horizontally instead of vertically.
2. There are eight bricks inside of a brick row. Look at the CSS rules for `.brick_row`, and determine what the width and height of your bricks must be in order to fill up the entire row. Set these by giving `.brick` width and height values. When you are done, you shouldn't see any green on your screen.
3. Suppose that we wanted actually see the individual bricks, like in a real brick wall. We could do this by adding a border to `.brick`. Try adding a 1px solid white border to the bricks. What happens?
4. The reason that a brick is being dropped down is that all of our bricks cannot fit inside the brick row anymore. See if you can adjust the values of your bricks to get it to fit again. When you are finished, it should look like this:



The following parts of the exercise are optional, and should only be attempted if you've finished everything above.

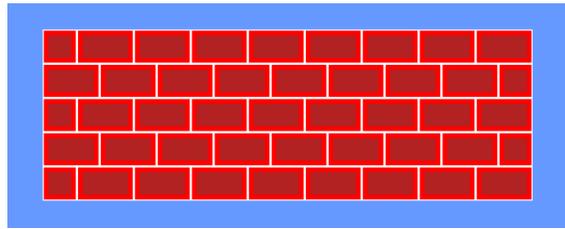
5. Suppose we wanted to make our bricks have a slightly different interior color. We can do this creating a new class of element and inserting it into our brick divs to create the following:

```
<div class="brick">  
  <div class="inner_brick"></div>  
</div>
```
6. Next, we need to give the `inner_brick` some CSS rules. Try setting its width and height values to 100%, and its `background-color` to `'firebrick'`. How does this display?
7. All of our bricks are completely firebrick colored now, because we've told our inner bricks to take up 100% of their parent elements width and height. We can fix this by adding some padding to our bricks. Try this now. Insert padding equally on all sides in order to center the inner brick.

8. The bricks dropped down again! This is because, like the border, the addition of padding makes our elements larger. Therefore, readjust the width and height values of `.brick` to make the bricks all line up correctly again. When you are finished, it should look like this:



9. Next, copy the brick row a couple of different times to create a brick wall. The bricks are all lined up directly on top of each other. In real brick walls, the bricks are staggered somewhat. Let's do this now. You're going to need to create a new class, called `half_brick`, and you'll need to insert it at the beginning of your first row, at the end of your second row, at the beginning of your third row, and so on. Give this class a width that is half of your other bricks. Other than the width, apply the same rules to it that you applied to `.brick`. You may also need to adjust the width of `.brick_row` to make room for this additional `half_brick`. When you are done, it should look something like this:



10. We can add a roof onto our brick wall and make a house! Directly above the first row, create a div with a class of "roof". Give it the following rules. Can you figure out why we get a triangle? What happens if we adjust the widths of the different border sides?

```
.roof{  
  width:1px;  
  margin: 0 auto;  
  border-left: 240px solid transparent;  
  border-right: 240px solid transparent;  
  border-bottom: 240px solid brown;  
}
```

11. When you are finished, you should have something that looks like this:

