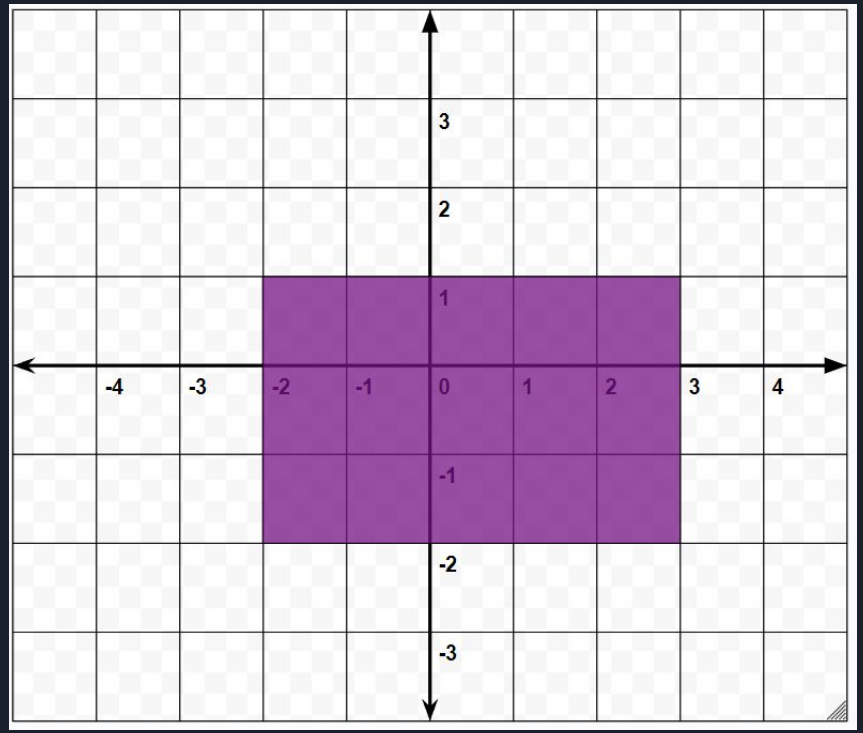
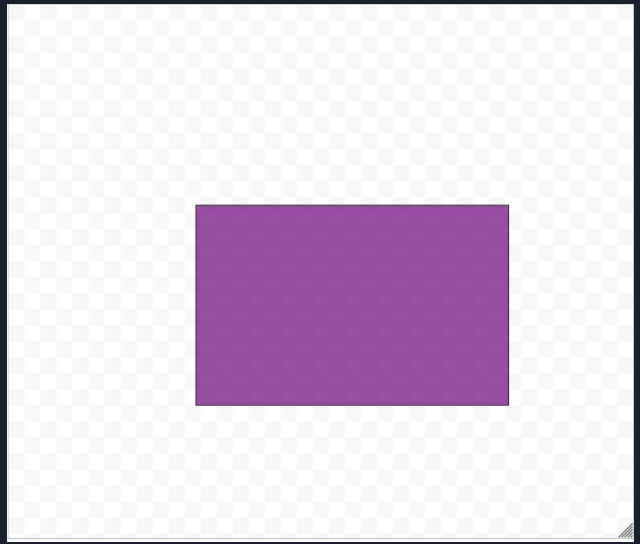
Aim: How Can We Use CSS to Resize and Position HTML Elements?

Do Now: What are the coordinates of each corner of the rectangle in the image?



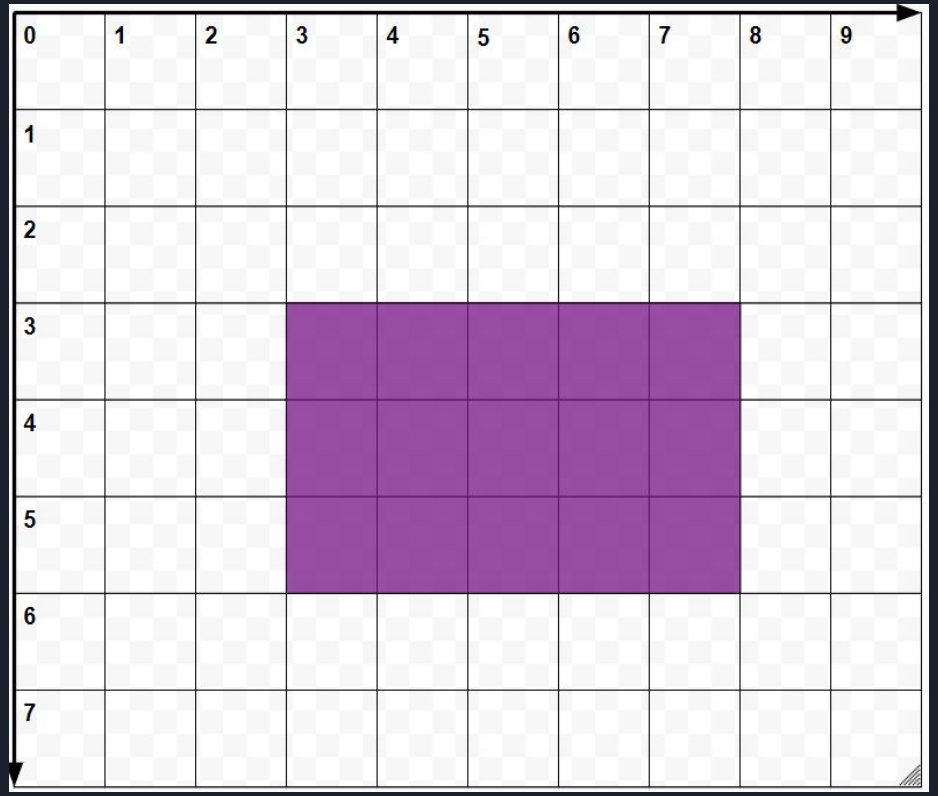
Unseen Coordinate System

- For the Do Now, we could use the coordinate grid to help us locate the corners of the rectangle
- Our websites work using a coordinate system like in Math as well
- However, the coordinate system used in our websites differs from what you may be more familiar with from Math in two important ways
- The first big difference is that unlike with a Cartesian coordinate grid, we can't actually see the lines or numbers making up the grid when we browse the web
- However, just because we can't see the underlying grid, doesn't mean it's not there



This is the same rectangle as before, but now with a coordinate system that matches what's used by web browsers.

What would be the coordinates of each corner of the rectangle now, using this new coordinate system?



The Top and Left Properties

- Much like how we can plot a point on a grid in Math given its coordinates, we can do something similar with our HTML elements using CSS
- The top and left CSS properties are analogous to the y and x coordinates in math respectively.
- Both CSS properties take numeric values, like with the font-size property from last lesson.
- There are many units that can be used for the top and left properties as well, but we'll stick to pixels (px) for now
- Although (0,0) is the top left corner in this coordinate system, it's still possible to use negative values for top and left.
- A negative value just means that part or all of that HTML element will be out of bounds

```
#photo {

    left:200px;

    top:400px;

}
```

The Bottom and Right Properties

- Elements in HTML can also be positioned using the bottom and right CSS properties
- Sometimes it may make more sense to say something should be near the bottom of the screen instead of saying it should be ~700px from the top
- The bottom and right properties work nearly identically to the top and left properties, but use a different frame of reference
- It may be easy to see something like left:50px and interpret that as meaning move something 50px to the left
- Instead, think of it as meaning move something 50px **from** the left-most side
- Or, think of it as moving it in the opposite direction, left:50px means moving 50px to the **right**

The Position Property

- The top, left, right, and/or bottom properties need to be combined with the position property for them to work properly
- Despite the name, the position property doesn't affect where elements are positioned, but rather how they are positioned, like choosing what set of rules to apply
- There are four values for the position property that we cover today,
- position:static;
- position:relative;
- position:absolute;
- position:fixed;
- We can demo and practice the difference between these values using this site: https://www.w3schools.com/css/css_positioning.asp

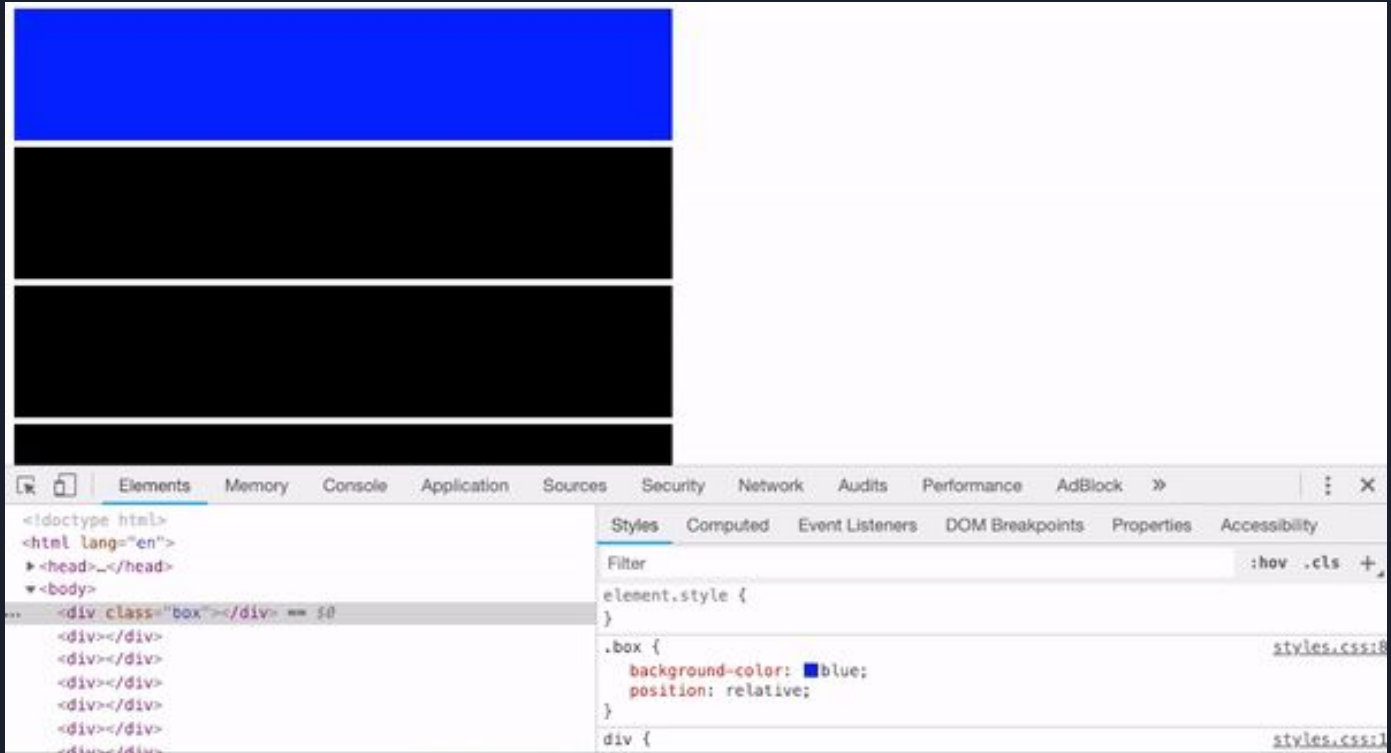
position: static

- Static is actually the default value for most HTML elements
- An element with position:static actually isn't affected by the top, left, bottom, or right properties
- Thus, an element with position:static tends to stay at their default position determined by the content and flow of the webpage

position: relative

- An HTML element with position: relative actually behaves like an element with position: static as long as no value for top, left, bottom, or right properties are given
- However, if an element has position: relative and one of the above properties are applied, the element will move the corresponding distance from it’s default position in a direction determined by the properties used
- Changing the position of an element with position: relative does not affect or move the other elements around it, even if it would overlap with something

position: relative

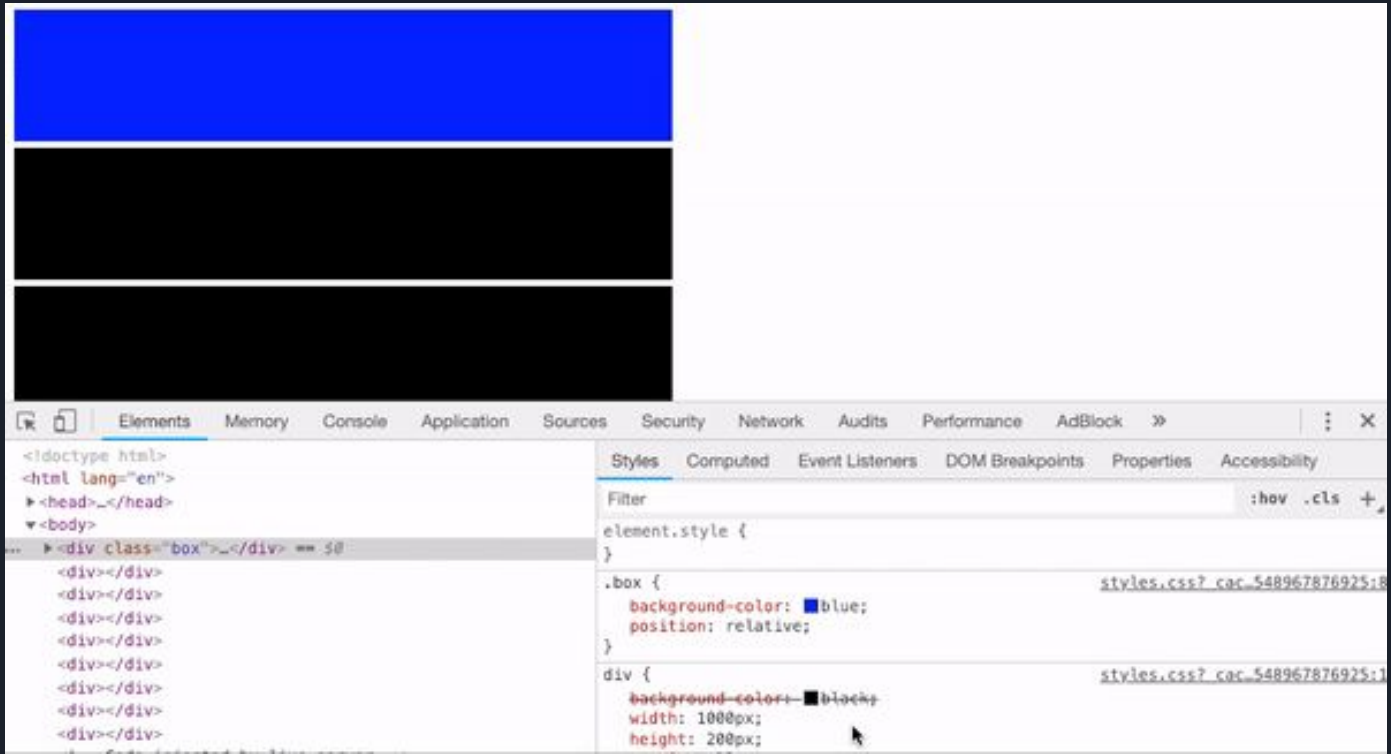


position: absolute

- An HTML element with position: absolute ignores it's default position dictated by the flow of the web page
- Instead, elements with position: absolute are positioned with respect to the window
- That means that while the below code would not cause any change to an element with position: relative, it would position an element with position:absolute to the top-left corner of the page

```
img {
    top:0px;
    left:0px;
}
```

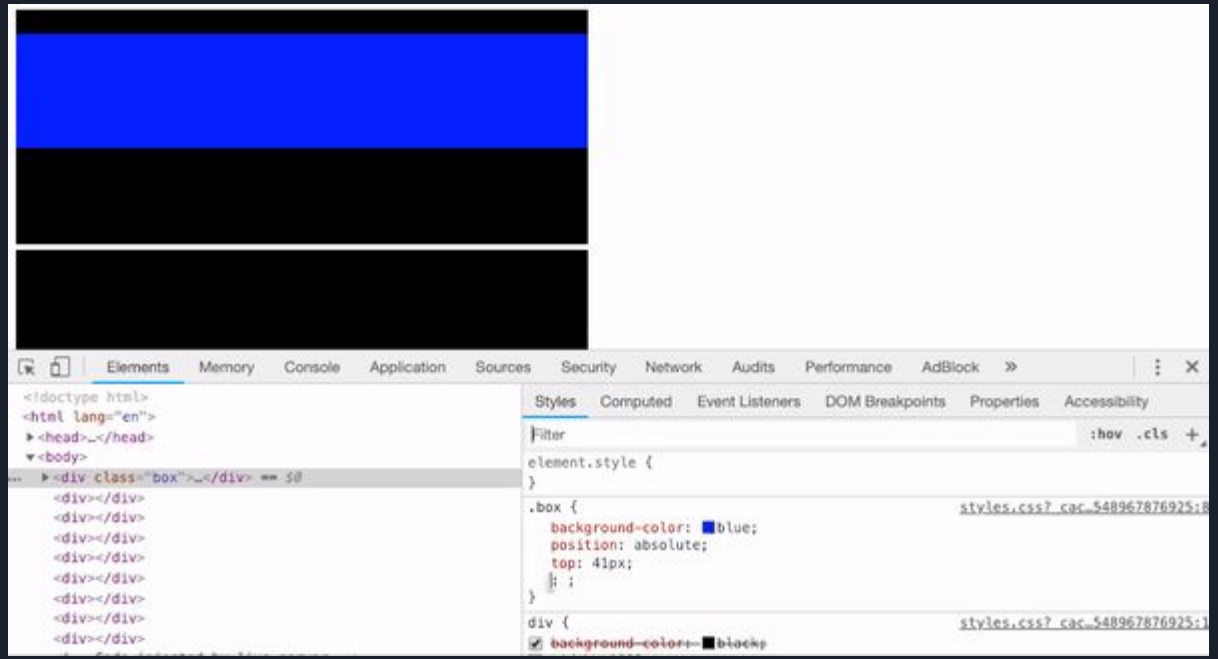
position: absolute



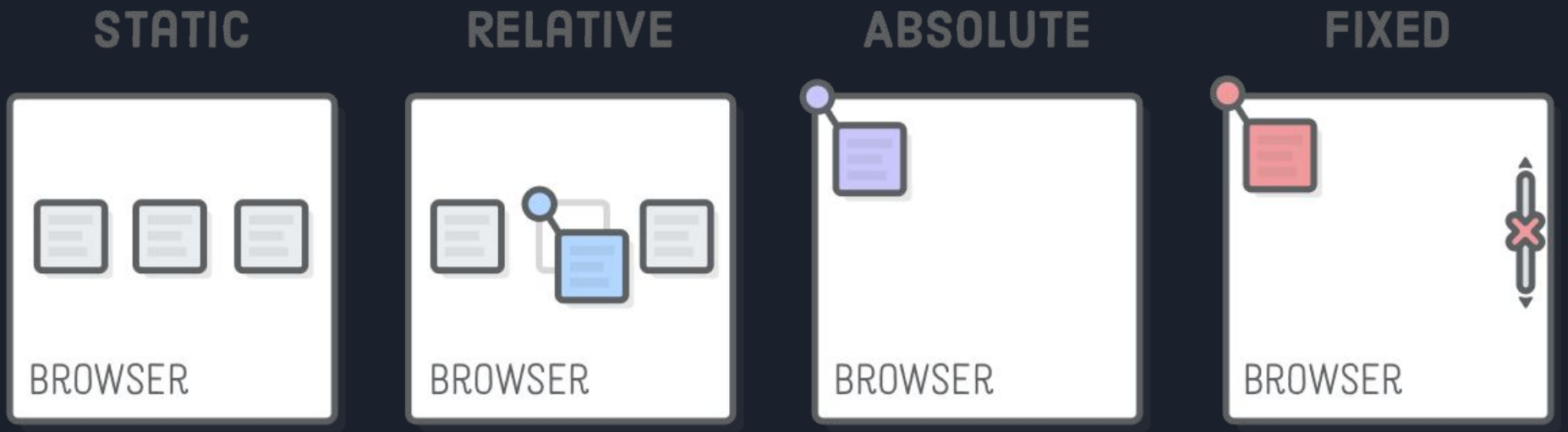
position: fixed

- An HTML element with position: fixed behaves very similarly to an element with position:absolute;

- However fixed elements are “fixed” in place on the user’s screen, staying in the same position even if the user scrolls up or down



Position Values Recap



Width and Height properties

- We can change the size of HTML elements by using the width and height properties
- These properties work especially well with images
- Like with the top, left, bottom, and right properties, the width and height properties use numeric values and we usually use pixels (px) as the unit
- If we use just one of the width or height properties for an image, it will automatically scale the other dimension to maintain the same aspect ratio
- For example, if an image is originally 200px x 400px and we set the width to 600px using CSS, the height will be automatically set to 1200px
- However, if we specify both the width and height for an image, it will not maintain the image's original aspect ratio and may look stretched or distorted

```
img {
    width: 600px;
}
```

Using a background image

- In a previous lesson, we learned how to change the background-color of an HTML element
- However, we are not just limited to using colors as backgrounds, we can use images as backgrounds as well
- We can set a background image for an HTML element by using the background-image property in CSS
- This is an example of code one might use to use a background image for their web page

```
body {

    background-image: url(“images/ocean.png”);

}
```

- Notice that the file path or URL for the desired background image needs to be put in quotes and put in between the parentheses of **url( )**