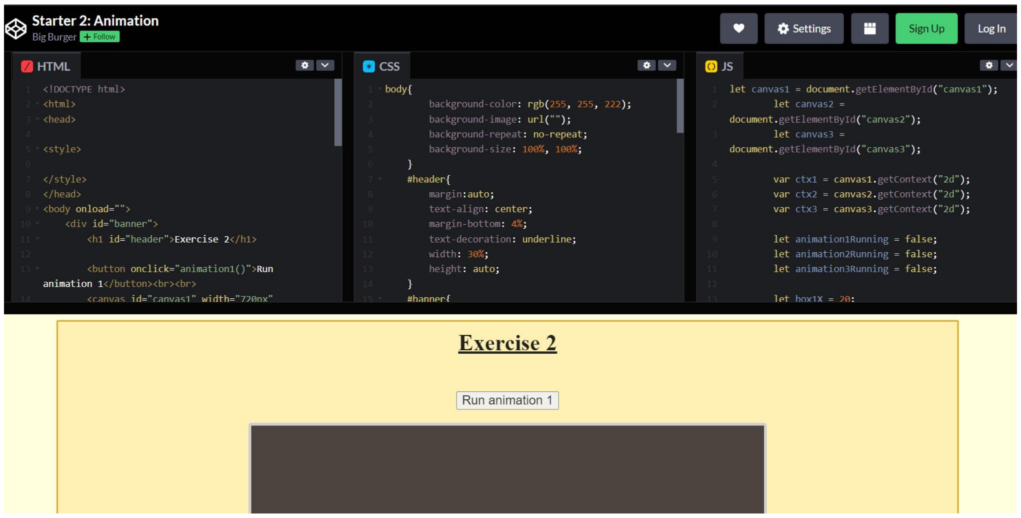
Foes - Workshop 2

Below are the detailed instructions intended to help guide you through the set-up and creation of Workshop 2. This workshop will help users focus solely on JavaScript and how it can be used to create animations on 2D objects. These concepts are meant to mimic some of the functions used to build the Foe's final game.

Guided Steps:

1. To access Workshop 2 please click the following link to be taken to the second page of the Workshop on Code Pen: https://codepen.io/Alreeshid/pen/wvYaKjw.

NOTE: It is recommended to use the direct link to the CodePen site (Method 1) versus clicking the link at the bottom of Workshop 1 as the screen will appear different.

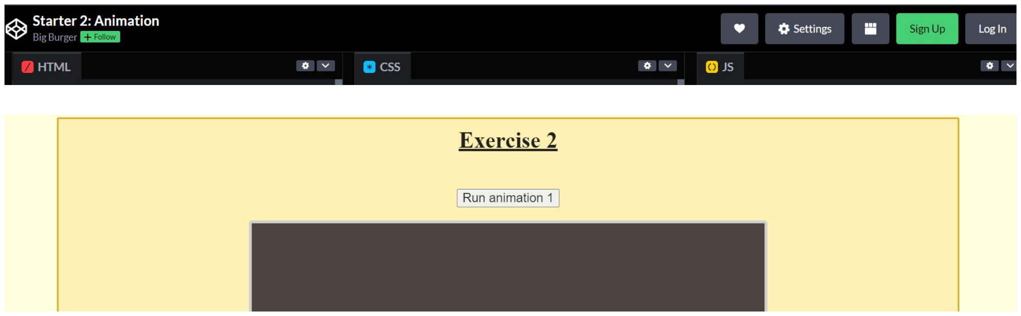


```html
<!DOCTYPE html>
<html>
<head>

<style>

</style>
</head>
<body onload="">
    <div id="banner">
        <h1 id="header">Exercise 2</h1>

        <button onclick="animation1()">Run
animation 1</button><br><br>
        <canvas id="canvas1" width="720px"
```

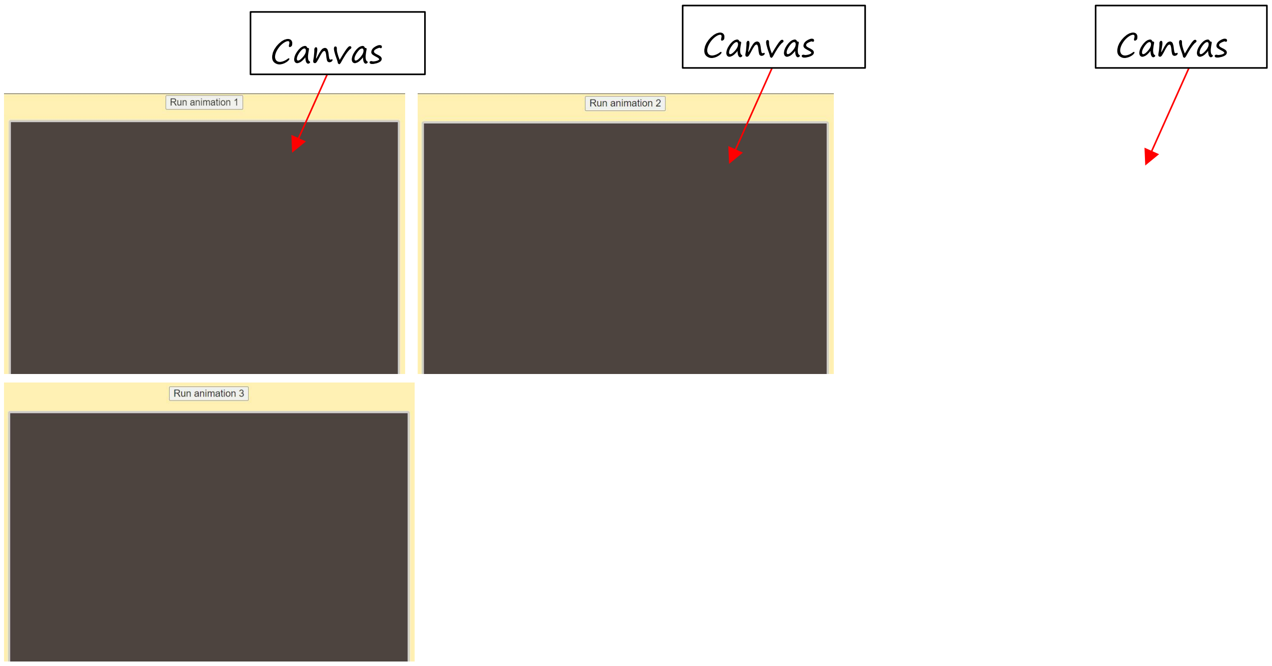
```css
body{
        background-color: rgb(255, 255, 222);
        background-image: url("");
        background-repeat: no-repeat;
        background-size: 100%, 100%;
    }
    #header{
        margin:auto;
        text-align: center;
        margin-bottom: 4%;
        text-decoration: underline;
        width: 30%;
        height: auto;
    }
    #banner{
```

```javascript
let canvas1 = document.getElementById("canvas1");
        let canvas2 =
document.getElementById("canvas2");
        let canvas3 =
document.getElementById("canvas3");

        var ctx1 = canvas1.getContext("2d");
        var ctx2 = canvas2.getContext("2d");
        var ctx3 = canvas3.getContext("2d");

        let animation1Running = false;
        let animation2Running = false;
        let animation3Running = false;

        let box1X = 20;
```

2. The link should take you to the new webpage on CodePen titled **Starter 2: Animation**. There should be three coding screens (HTML, CSS, JS) and at the bottom a webpage with the title "Exercise 2".

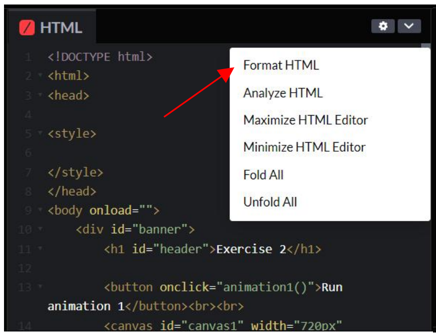


3. Here we will be working more with JavaScript to make objects animate in the three canvases on our webpage on the bottom (scroll to view all black/gray boxes). Canvases are block areas where we will run animations.

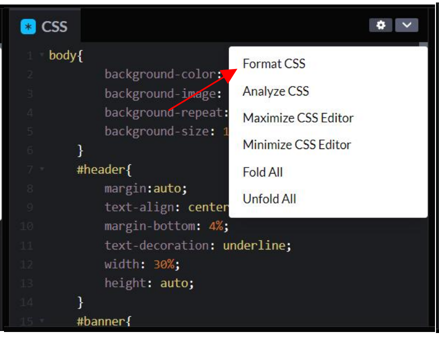


4. During this workshop please refrain from hitting the refresh button as it will cause all your changes to disappear as CodePen requires an account to save any changes made during a session.

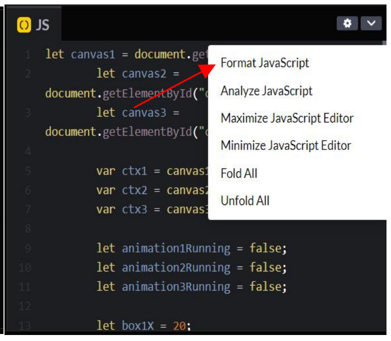
5. Before we begin let's format our pages. On the HTML screen there will be a down arrow in the upper right corner next to a gear. Click the down arrow and press "Format HTML", this will ensure that your code is easier to read for our workshop. Repeat this step for the CSS "Format CSS" and JS "Format JavaScript" mini screens.



```html
<!DOCTYPE html>
<html>
<head>

<style>

</style>
</head>
<body onload="">
    <div id="banner">
        <h1 id="header">Exercise 2</h1>

        <button onclick="animation1()">Run
animation 1</button><br><br>
        <canvas id="canvas1" width="720px"
```



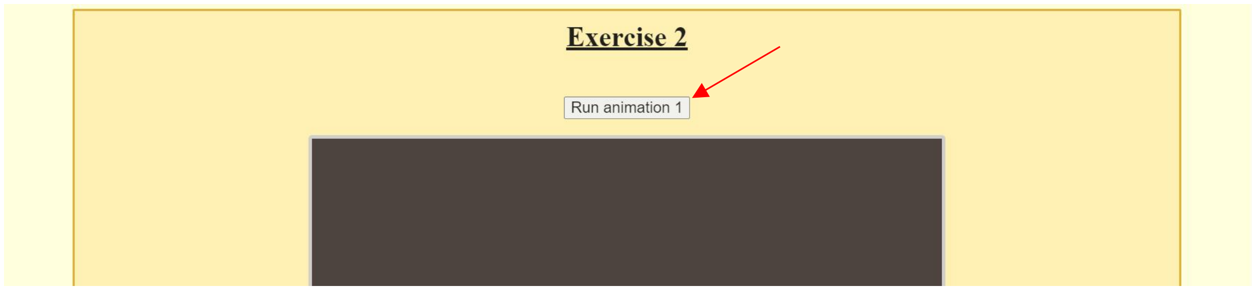
```css
body{
        background-color:
        background-image:
        background-repeat:
        background-size: 1
    }
    #header{
        margin:auto;
        text-align: center
        margin-bottom: 4%;
        text-decoration: underline;
        width: 30%;
        height: auto;
    }
    #banner{
```

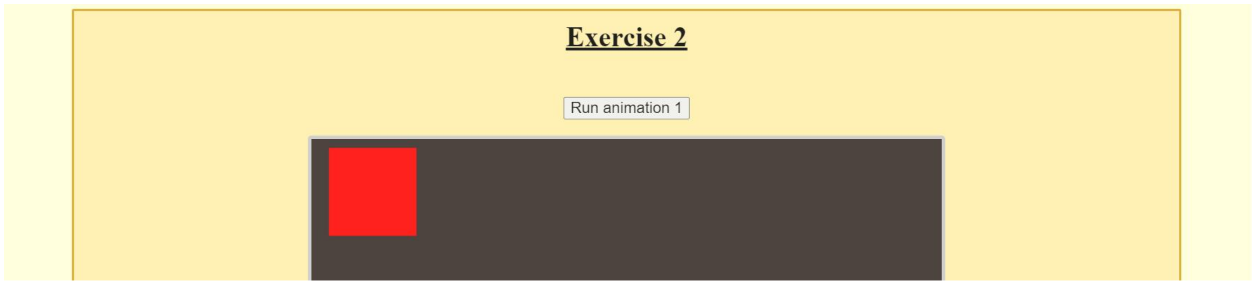


```javascript
let canvas1 = document.get
        let canvas2 =
document.getElementById("c
        let canvas3 =
document.getElementById("c

        var ctx1 = canvas1
        var ctx2 = canvas2
        var ctx3 = canvas3

        let animation1Running = false;
        let animation2Running = false;
        let animation3Running = false;

        let box1X = 20;
```

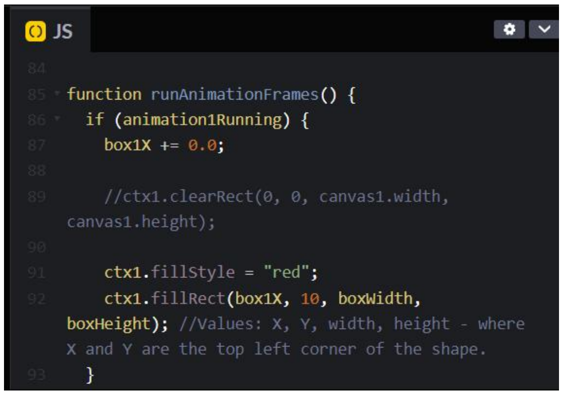
6. For the first part of our workshop we will focus on animation 1.

7. Right now when we click the "Run animation 1" button a red box appears but the red box has no movement.



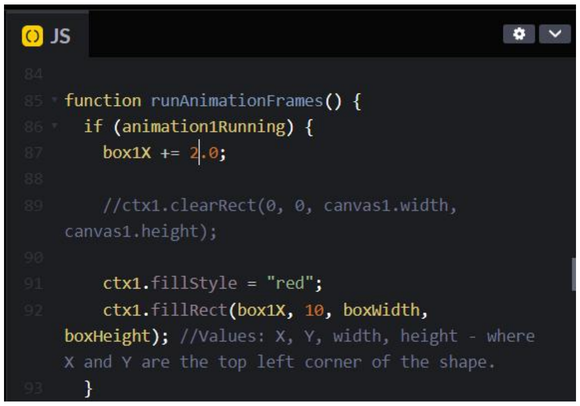


8. To fix this we want to go to the JS mini screen and scroll until we see the function **runAnimationFrames.** This function will be where we change most of our code to add functionality.


JS

```
function runAnimationFrames() {
  if (animation1Running) {
    box1X += 0.0;

    //ctx1.clearRect(0, 0, canvas1.width,
canvas1.height);

    ctx1.fillStyle = "red";
    ctx1.fillRect(box1X, 10, boxWidth,
boxHeight); //Values: X, Y, width, height - where
X and Y are the top left corner of the shape.
  }
```

9. Under the first if statement we will change the augmented value of box1X to **box1X += 2.0;**.


JS

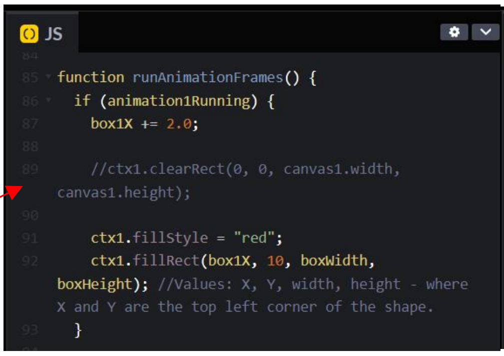
```
function runAnimationFrames() {
  if (animation1Running) {
    box1X += 2.0;

    //ctx1.clearRect(0, 0, canvas1.width,
canvas1.height);

    ctx1.fillStyle = "red";
    ctx1.fillRect(box1X, 10, boxWidth,
boxHeight); //Values: X, Y, width, height - where
X and Y are the top left corner of the shape.
  }
```

10. By changing the value to 2, we are letting JS know that every time this method is called for the first animation, we are going to change the x position of the boxes' coordinate system by 2, thus allowing the box to "move" across the screen. So right now, the box is at the coordinates (20, 10) and when we call the runAnimationFrames the X coordinate will change by 2 and the new coordinates will become (22, 10). And if we call the runAnimationFrames function again our coordinates would change from

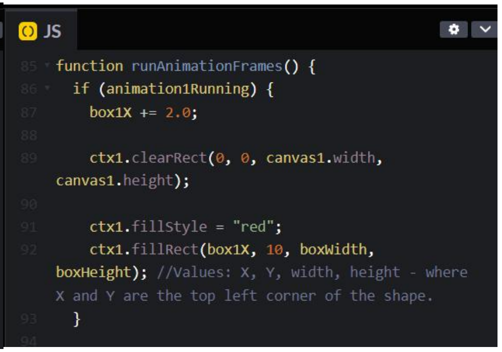
(22, 10) to (24, 10). This will repeat continuously for as long as the program continues to run.

11. Let CodePen save and approve our changes and click the "Run animation 1" button again. What happened to the box?

12. Instead moving across the screen the box just grows wider and wider. To fix that go back to your JS mini screen and remove the two slashes on line 89 to uncomment our code.



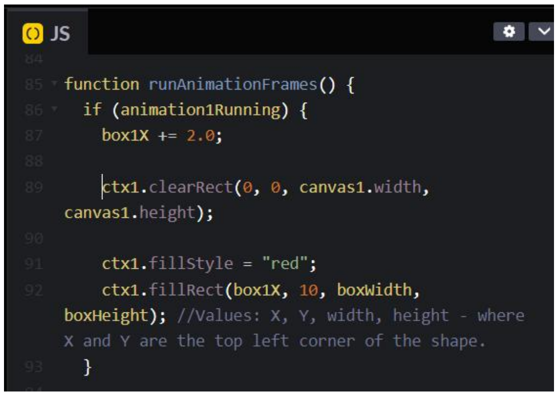
```
85 function runAnimationFrames() {
86   if (animation1Running) {
87     box1X += 2.0;
88
89     //ctx1.clearRect(0, 0, canvas1.width,
   canvas1.height);
90
91     ctx1.fillStyle = "red";
92     ctx1.fillRect(box1X, 10, boxWidth,
   boxHeight); //Values: X, Y, width, height - where
   X and Y are the top left corner of the shape.
93   }
```



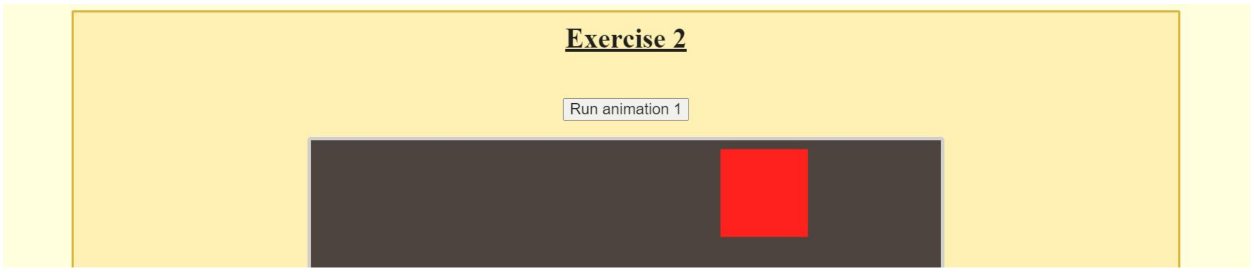
```
85 function runAnimationFrames() {
86   if (animation1Running) {
87     box1X += 2.0;
88
89     ctx1.clearRect(0, 0, canvas1.width,
   canvas1.height);
90
91     ctx1.fillStyle = "red";
92     ctx1.fillRect(box1X, 10, boxWidth,
   boxHeight); //Values: X, Y, width, height - where
   X and Y are the top left corner of the shape.
93   }
94
```

13. Let CodePen save and approve our changes and click the "Run animation 1" button again. Now what happened to the box?

14. It moved! This is because of the clearRect JavaScript method. ClearRect constantly erases the previous location of the red box on our canvas, so that when we redraw the box (line 92) it's as if the box is moving across the screen. So now every time we see the box it's in a new position.

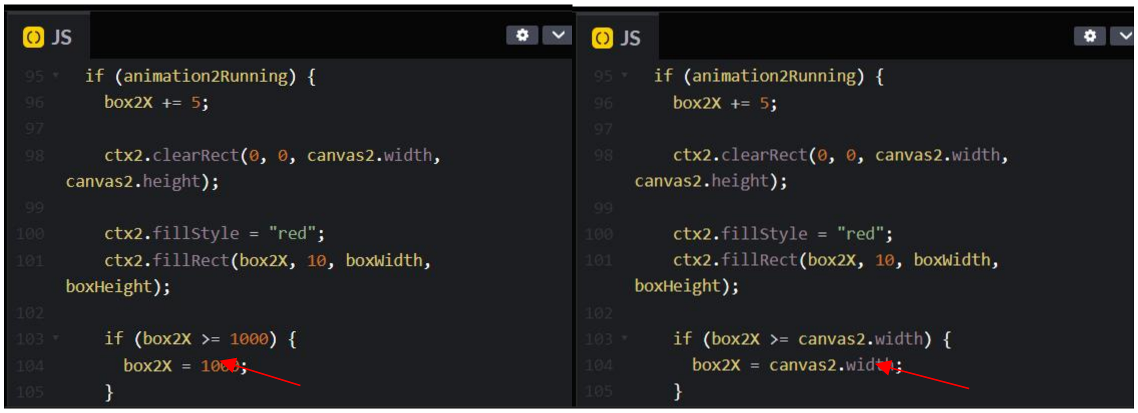


```
function runAnimationFrames() {
  if (animation1Running) {
    box1X += 2.0;

    ctx1.clearRect(0, 0, canvas1.width,
canvas1.height);

    ctx1.fillStyle = "red";
    ctx1.fillRect(box1X, 10, boxWidth,
boxHeight); //Values: X, Y, width, height - where
X and Y are the top left corner of the shape.
  }
```



15. Let's move on to the second part of our workshop – animation 2.

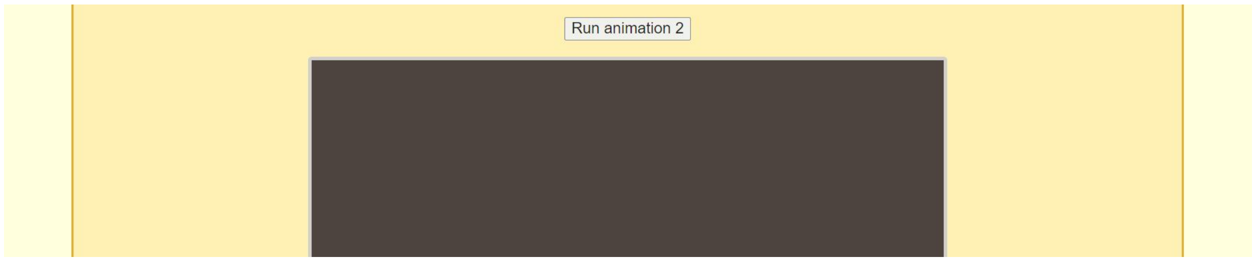
16. Under the JS mini screen scroll down to lines 103/104 and replace both 1000's with canvas2.width.

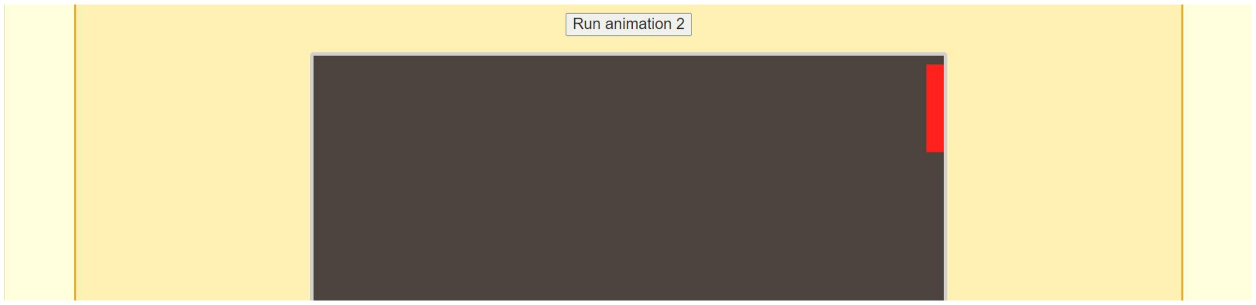


```
95    if (animation2Running) {
96      box2X += 5;
97
98      ctx2.clearRect(0, 0, canvas2.width,
    canvas2.height);
99
100     ctx2.fillStyle = "red";
101     ctx2.fillRect(box2X, 10, boxWidth,
    boxHeight);
102
103     if (box2X >= 1000) {
104       box2X = 1000;
105     }
```

```
95    if (animation2Running) {
96      box2X += 5;
97
98      ctx2.clearRect(0, 0, canvas2.width,
    canvas2.height);
99
100     ctx2.fillStyle = "red";
101     ctx2.fillRect(box2X, 10, boxWidth,
    boxHeight);
102
103     if (box2X >= canvas2.width) {
104       box2X = canvas2.width;
105     }
```

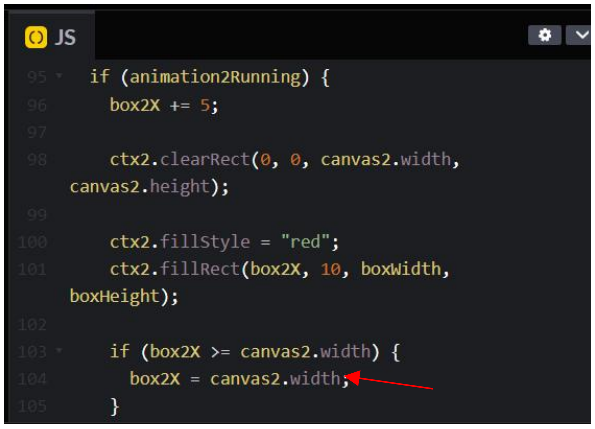
17. Click the "Run animation 2" button on your webpage and watch as a red box quickly moves across the page just like our first program. But now we want to make sure our box does not go off our canvas.





18. Under the function **runAnimationFrames** we are going to change some code.

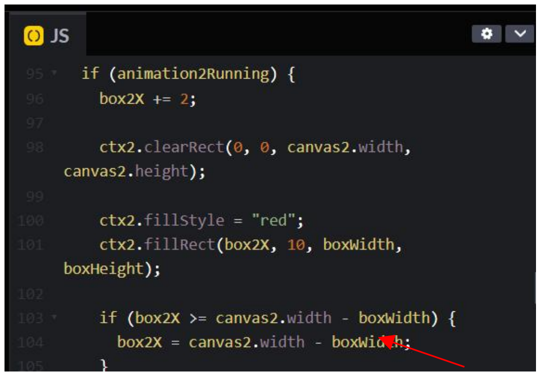
19. Go back to line 103/104 where we changed 1000 to canvas2.width.



```
95   if (animation2Running) {
96     box2X += 5;
97
98     ctx2.clearRect(0, 0, canvas2.width,
   canvas2.height);
99
100    ctx2.fillStyle = "red";
101    ctx2.fillRect(box2X, 10, boxWidth,
   boxHeight);
102
103    if (box2X >= canvas2.width) {
104      box2X = canvas2.width;
105    }
```

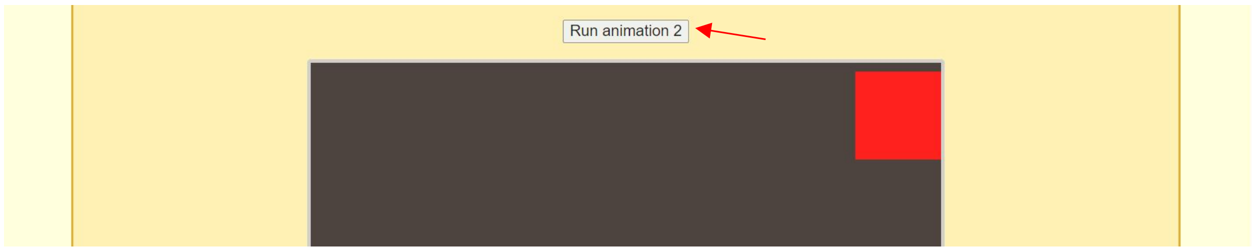
20. Right now our code wants the box to stop when the box2X coordinates becomes greater than or equal to canvas2.width, but the problem we run into is that the box is still appearing off the screen.

21. In the parentheses of our if statement after canvas2.width add – **boxWidth**. This will help fix our problem of the box running off the screen.



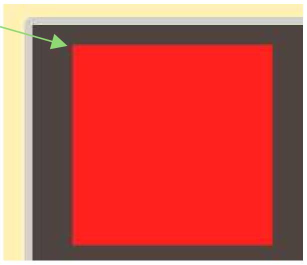
```
if (animation2Running) {
  box2X += 2;

  ctx2.clearRect(0, 0, canvas2.width,
canvas2.height);

  ctx2.fillStyle = "red";
  ctx2.fillRect(box2X, 10, boxWidth,
boxHeight);

  if (box2X >= canvas2.width - boxWidth) {
    box2X = canvas2.width - boxWidth;
  }
```

22. Let CodePen save and approve our changes and click the "Run animation 2" button again. Now what happened to the box? It stopped right before it ran off the screen.



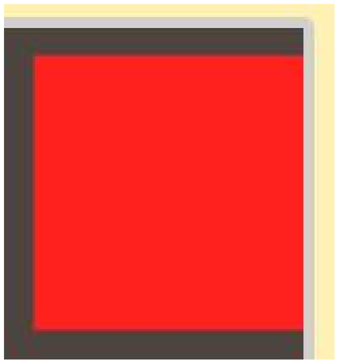
23. This is because when we created our box on the canvas, we didn't create it at the coordinates (0, 0) but instead at (20, 10), and the height and width of the box was created to be 100 pixels. When doing animations, we must consider not only the initial positions of our shapes but also the

Starting X
Coordinate of our



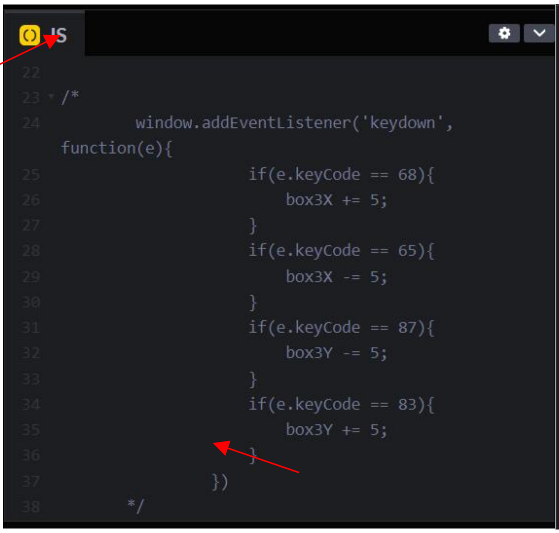
IMPORTANT: When we run an animation the x coordinate in the upper left corner is the one that matters the most to JavaScript. So, by the time our box2X equaled to canvas2.width the rest of our box was already off the page.

X Coordinate that will be tracked by

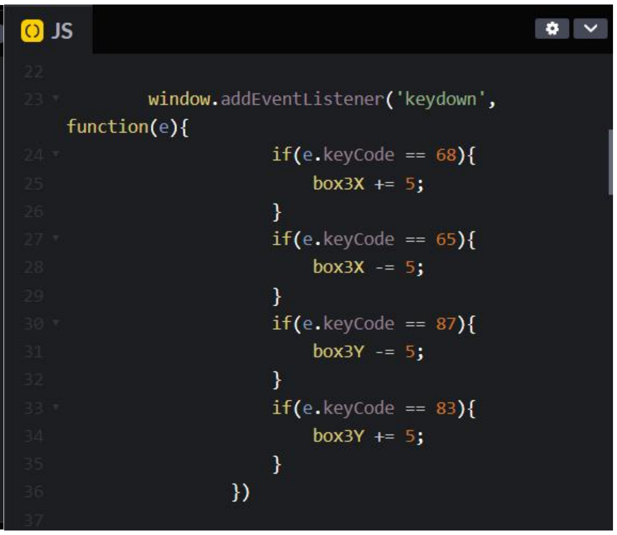


24. Lastly let's move on to the last part of Workshop 2, which allows the user to change the box's location using our keyboard.

25. In the JS mini screen scroll to line 23 and uncomment out that chunk of code by removing the slash and star characters; do the same on line 38 so that our code will run with no errors.

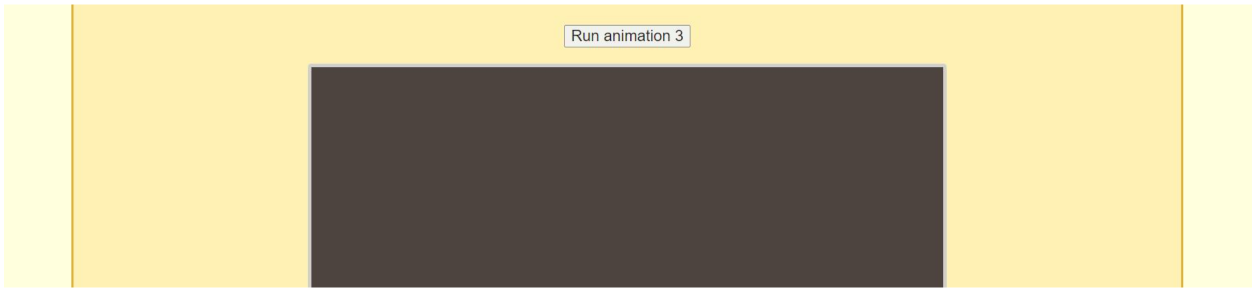


```
22
23  /*
24          window.addEventListener('keydown',
    function(e){
25                          if(e.keyCode == 68){
26                              box3X += 5;
27                          }
28                          if(e.keyCode == 65){
29                              box3X -= 5;
30                          }
31                          if(e.keyCode == 87){
32                              box3Y -= 5;
33                          }
34                          if(e.keyCode == 83){
35                              box3Y += 5;
36                          }
37                      })
38          */
```



```
22
23          window.addEventListener('keydown',
    function(e){
24                  if(e.keyCode == 68){
25                      box3X += 5;
26                  }
27                  if(e.keyCode == 65){
28                      box3X -= 5;
29                  }
30                  if(e.keyCode == 87){
31                      box3Y -= 5;
32                  }
33                  if(e.keyCode == 83){
34                      box3Y += 5;
35                  }
36              })
37
```

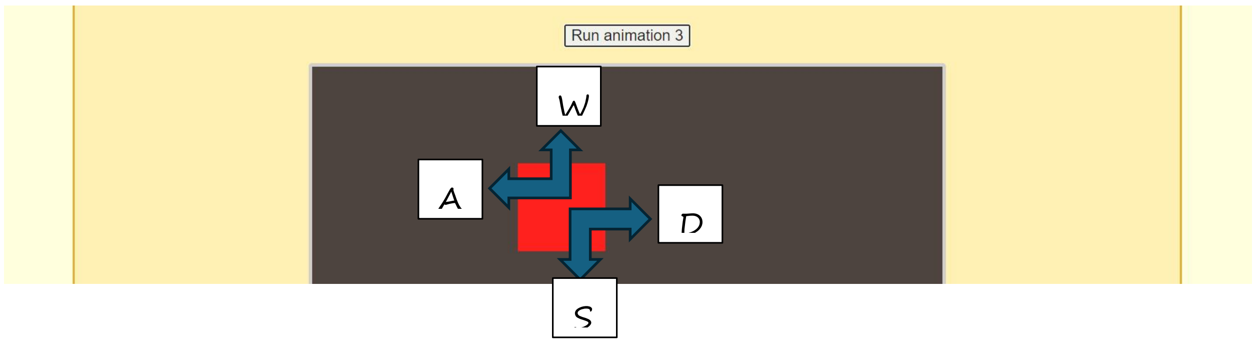
26. Once your code is fully commented out. Let CodePen save your changes then scroll your webpage to show the button "Run Animation 3" and the canvas.

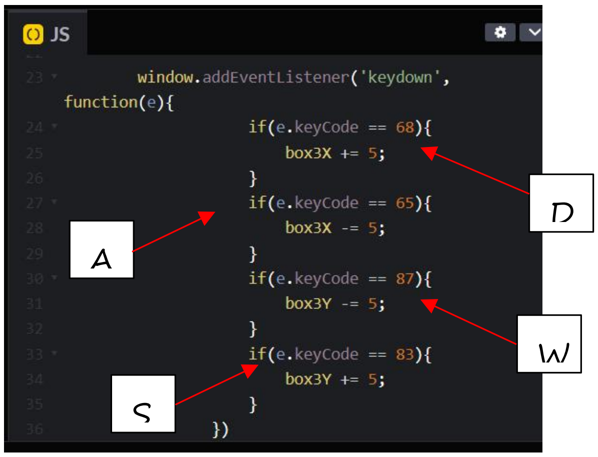


27. Click the "Run animation 3" button on your webpage.

28. To see changes reflected on the canvas you must interact with the webpage by pressing the keys: W, A, S, and D on your keyboard. What do each of the keys do to the box? They move it!

29. The code we uncommented in our code uses the addEventListener method which allows us to specify the action we want JavaScript to "listen" for, in this case '**keydown**', and what we want the code to do once they are pressed. So, the keys in this code when pressed will either add or subtract 5 from the box3X and box3Y coordinates allowing the box to move with user interaction.





```
window.addEventListener('keydown',
function(e){
            if(e.keyCode == 68){
                box3X += 5;
            }
            if(e.keyCode == 65){
                box3X -= 5;
            }
            if(e.keyCode == 87){
                box3Y -= 5;
            }
            if(e.keyCode == 83){
                box3Y += 5;
            }
        })
```

30. Now you try changing the amount we subtract/add and note the changes you see.

31. Next try changing the keycode values so that the user can press different keys to make the box go up, down, left, and right by using this website: https://www.exeideas.com/2014/05/javascript-char-codes-key-codes.html

32. Happy coding!