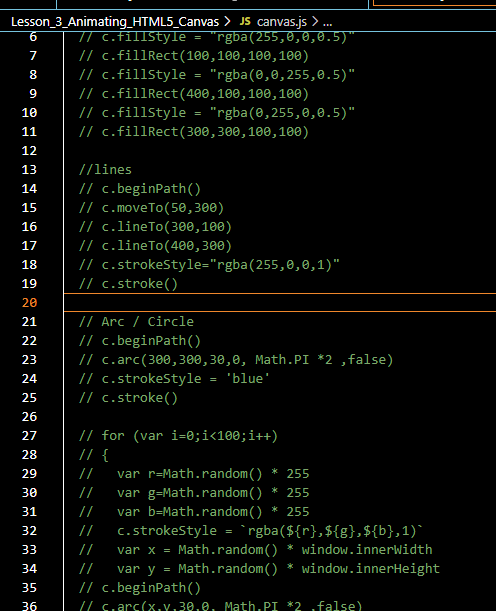
In the last lesson we covered how to draw circles on our canvas in random location but that only get us so far we only have static circles on the screen we want to make sure these things are animating so that we give the user an interesting experience something that might not be used to when they visit our site so how do we get our circles moving on the screen well instead of dealing all the circles it’s going to be easier for us to deal with just one circle and get that thing moving before we go on and moving all the circles at the same time sop lets go ahead and comment all the code we have right now just dop that we can focus on one individual circle We comment out rectangles and then we will comment out our for loop as well as shown below



We are commenting instead of deleting it because its going to be useful for us to reference later on. Because we are going to be using the same concept Now we save our file and then it get refreshed Now it will look like below

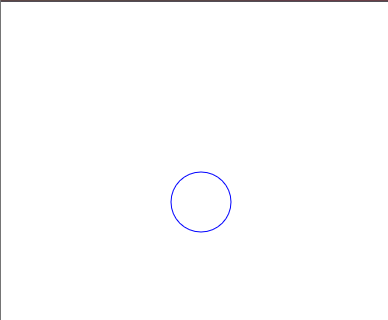


You are going to see we have a blank canvas but now we want to actually create a circle that we can get moving and basically bouncing off the sides so our screen so as you know based on the last lesson we can create a circle using the arc method that we have right here so lets go ahead and copy the code for drawing circle in the loop as shown below



Now paste it in the bottom of the code and uncomment them

And since our arc method is using x and y variable that we set within this for loop. Let’s go ahead and give it a hard-coded value that is 200 and 200 and if we refresh the page then we are going to see that we have a circle on our screen as shown below



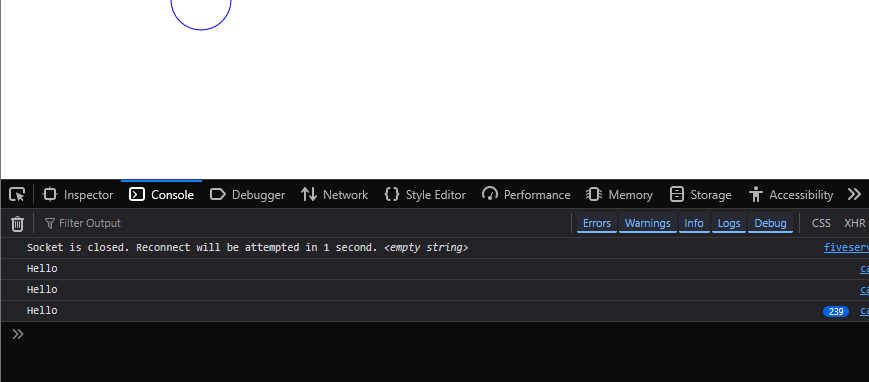
But now we need to be able to move out the circle around we need to go from left to right and up and down. Something to get moving and you may be wondering how do we do that well there’s actually particular method hat we need to go about to get the circle moving and this method is first creating a function and you can name this function anything you want we are going to call it animate because this function is going to be meant for animating and within this function we are going to be using another function called a requestAnimationFrame() this function take another function as an argument basically what’s going to happen is we are going to use this animate function by passing that to requestAnimationFrame as an argument (I think basically recursion is happening there and we pass animate function as a call back for requestAnimationFrame and when event happens associated with requestAnimationFrame then animate function is called and so on.)

2:05

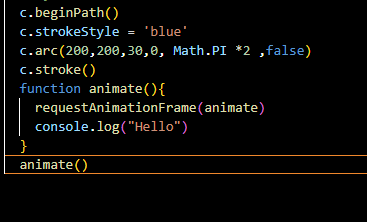
Once we call this animate function this requestAnimationoFrame function is basically going to be creating a loop for us it’s going to say okay we want to be able to create loop basically and what function to call to loop though its going to say call this requestAnimationFrame and then call animate again then it’s goinv to call this requestanimationFrame(who knows) we are basically creating a loop it sycling through over and over again until we tell it to stop but his is only going to work if were actually call animate at the bottom of the screen if I save the file then I will not see anything because we don’t have actually any data or any input being put in the browser for it to be read(who knows)

So to put some data in there we are going to put a console statement within our animate function we are going to console “Hello”

And if we save the file and page get refreshed then you will see it is being called over and over again as shown below

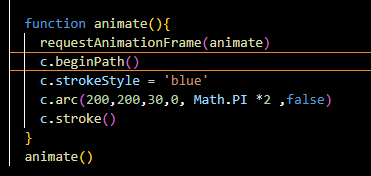


Our code will look like below

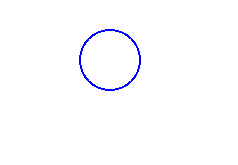


Iin console you can see Hello print is incrementing up and up. And it will keep going until it stop and there are certain ways to stop it. But for the purpose of this video lets just focus on the basics of getting our circle moving. So to get this thing moving its going to be good for you to understand how animation work in general

Animation works by refreshing the page we are basically going to be refreshing the page and each time we refresh the page we are going to be moving the circles x and y value incrementally very little so we are basically we are refreshing the page and then we are going to be incrementing our x value by one and as we increment it by one the circle is going to be move over to the right one by one. And we are just going to keep doing over and over again and its going to give us the illusion of animation   
It's going to be look like the circle is moving from its current location all the way to the right hand of the screen unless we tell it otherwise to do so or stop(not clear(who knows)) if we want to be incrementing our x values for our circles it really makes sense for use to grab our circle code and place it within your animate function so if we put this within our animate function and save the file and page is reloaded now the code will look like below

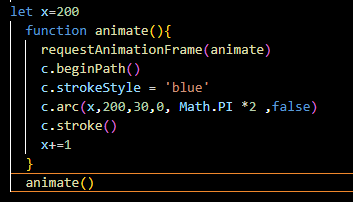


And our output will look like below



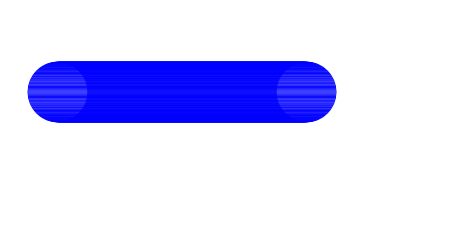
Now you can see that we are going to see that we have now have a bold circle that is because the circle is being drawn on top itself over and over again we only have x and y as 200 and 200 and since we are just drawing the circle on top itself but we want to make sure that we are changing these x and y values correspondingly so that they are not stuck at 200 we want to make sure that when the page refreshes itself then the c becomes 201 after refreshing and the 202, 203 and so forth and once we do that we are going to get the illusion of animation so whenever we want to change a number we need to create a variable and we are going to declare this outside our animate function. This is our global variable we assign it 200 so instead of using 200 within our x value for arc which we are going to be using our new x variable and then we are going to add one onto the current value of x each time the page is refreshed each time this animate function is called adding one onto x

Now our code will look like below

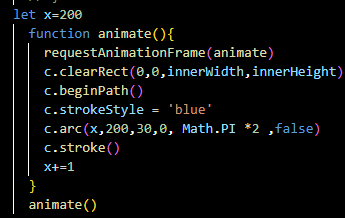


4:48

Now we save our file and page gets refreshed then you can see that output below



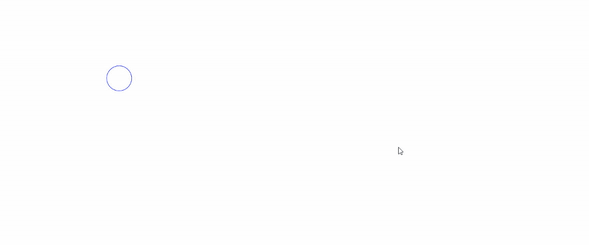
Now you can see our circles are moving from the left to right hand side of the screen but we are actually creating lots of circle in the process. And circular shapes are drawn on top of each other thus creating this line from left to right side of the browser screen Now to fix that basically what’s happening is that we are not actually clearing the canvas each time the page is refreshed each time our canvas is refreshing so we need to make sure that each time before we call arc we are clearing the canvas using clearRect() method and this is the same thing as fillRect method that we used up earlier It’s going op make an x value or y value and width and height so we are given to make sure we are clearing the entire canvas from 0, 0 to the canvas width and canvas height Now our code will look like below



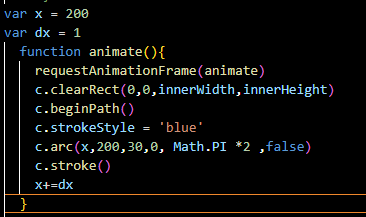
**Note:-**

The innerWidth actually means window.innerWidth as window is global

So this is going to clear canvas each time which give us an illusion such that when a next circle is drawn then previous circle will be removed Our output will look like below



Now you can see the circle Is not being drawn on top of each other over and over again we have this nice circle is being created for because we are clearing canvas each time this animate function is being called so its only getting us so far right now our circle is going off the screen and we don’t really want that there is no use of it going out of the screen because we can’t see it let’s go ahead and make it so that once hit side of the screen it bounces backward once it hit the left side of the screen it bounces backward as well it’s going to be bouncing left and right side of our screen in order to do it we need to make sure that we have conditional for our x velocity right now our x velocity is 1 and if you are unfamiliar with velocity then basically it is the speed or rate at which something moves in a particular direction so right now we are moving x coordinate at a speed of 1 pixel each time animate function is called and it is moving in a positive direction so therefore our velocity is 1 pixel per frame refresh to the right now since we want to make sure our velocity can be negative or positive we are going to want to create a variable for that as well so. Our variable is going to be called the dx and that just a standard acronym for our x velocity and this is going to be equal to 1 so instead of just adding 1 for x values we are going to adding dx instead as shown in our code



And play around with dx.

7:02

You can add 4 then it will now move very fast

We will keep it 1 but may change later once we can detect that right side of the browser(who knows)

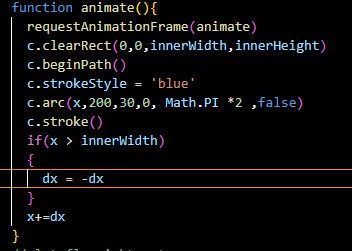
So we know we have a velocity of 1 but we need a condition such that x equal to the innerwidth of the screen and go ahead and change our velocity to negative value so actually we are subtracting negative number from x we are actually subtracting the number which make the circle go to left hand side from right hand side to the left hand side

So we are going to say x > innerwidth of our browser screen then make the dx negative and assign it to itself so that it becomes negative when x value is greater than innerwidth

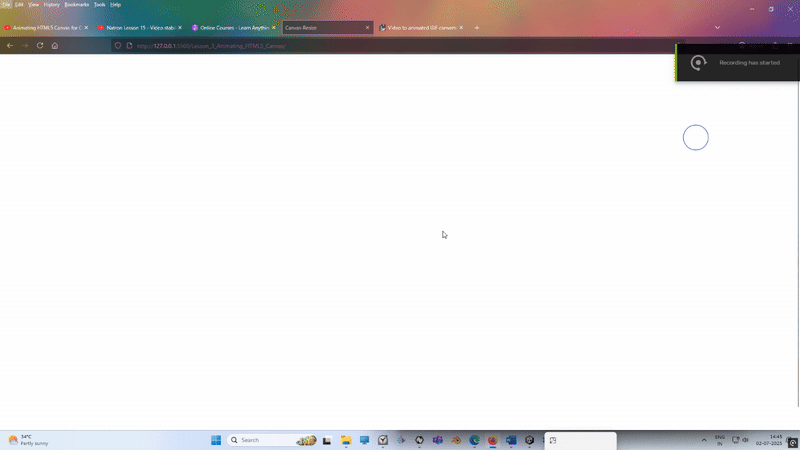
Lets see what will happen now

Now you will see that the ball is bouncing off the right hand side

Now our code will look like below



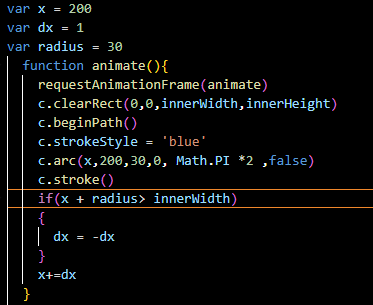
Now out output will look like below



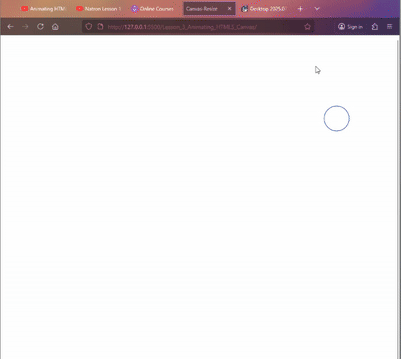
So you can see that the circle is bouncing off right side of the browser window but if you see closely then you will see the edge is not bouncing outside of the screen Its actually the center of the circle we want to take into account we want to take into account our circle had a radius and that edge should be bouncing off screen and not the center of the circle

We need to say that is x plus circle radius and circle radius id equal to 30 but lets go ahead and create a variable for now so we again declare a global variable that is radius of 30

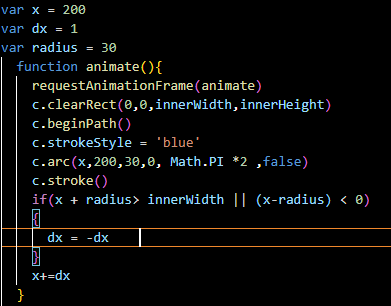
So we say id x value plus the circle radius is greater than innerWidth of our screen then go ahead and bounce it off now our code will look like below



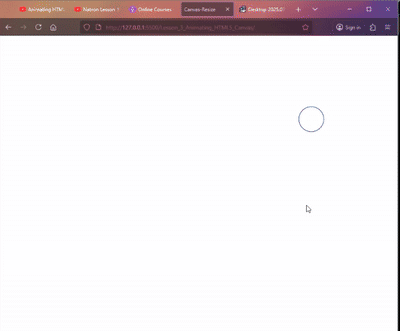
Now if we save this file and then in the browser after refresh you will see the output below



So now once the edge of the circle hits the screen we are now reversing the velocity so that the circle is going from right to left instead of left to right but one it hit zero you can see that it keeps on going past the width of the browser so we also need to take into account that conditional as well we can add an or operator in here and we can say if x – radius < 0 then reverse the velocity of our circle let see if that works out code looks like below



Our output will look like below



Now we have a circle that’s bouncing from the right to the left hand side of the screen.

9:20