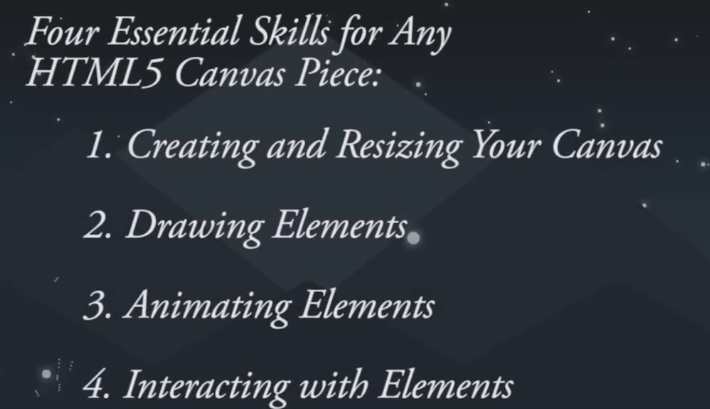
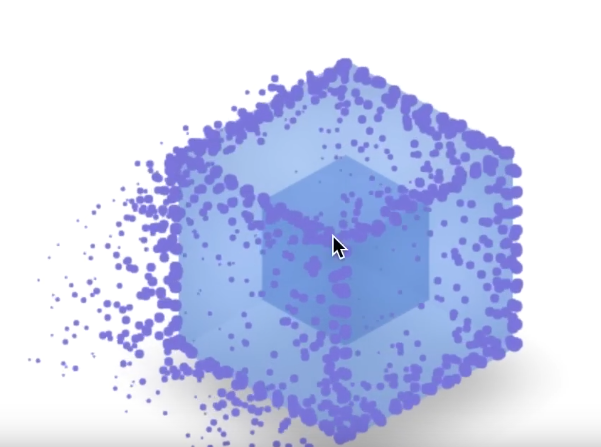
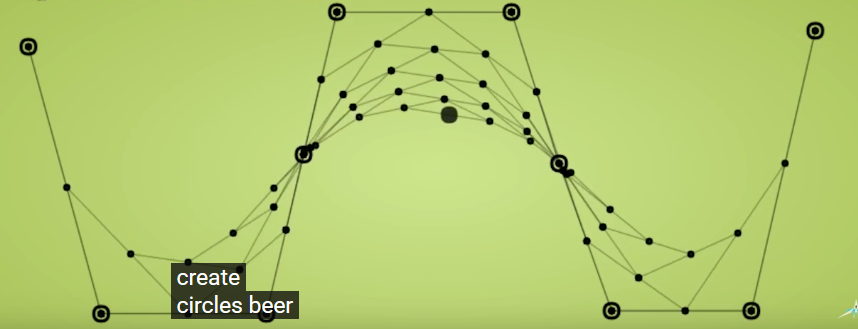
In this lesson we are going to be covering skill number two of how to become canvas pro that is Drawing on the canvas



In the last lesson we were covered skill number 1 of how to become a canvas pro of creating and resizing your canvas we also covered most rudimentary (or basic ) shape we could draw on the canvas the rectangle



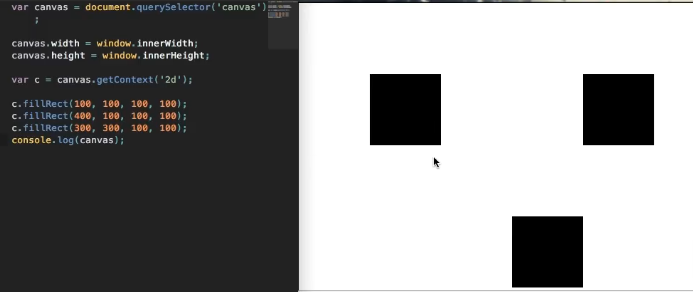
Now the our end goal is still to create amazing visuals but there’s only so much we can create which is rectangle and squares. To get the most out of these visuals. We need to understand the full extent as to what objects we can draw and how to draw them. To name some with the canvas we can draw rectangle, lines, arcs which we can also use to create circles, Bezier curve as shown below



Images and text as shown below



So for the sake of time we are not going to be covering how to draw all of these within the course but we will be covering those which are widely used which are the first three that is **rectangles**, **lines** and **arcs**. Once we have a solid foundation in regards to how to draw these shapes we will be moving we will be moving into some programming one by one and then the teacher will show you where teacher will show us how to effectively create hundred of these shapes at once using only couple of lines of code so without further due let’s get back to work and cover skill number 2 that is drawing on the canvas.

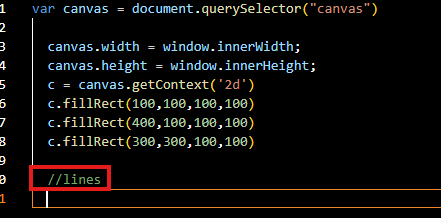


Now this is where we left off in the last lesson as you can see above there are few rectangles on the browser and we also resized our canvas to fit the entire width and height of the browser.

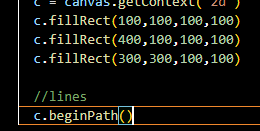
1:39

So as Teacher mentioned there are few different objects that we can draw on the canvas and we are going to be focusing on the lines and arcs which we can use to draw and circles mainly for time sake(who knows). Teacher would like to show us everything but unfortunately to keep this lesson short we can only focus on a few things at once but nevertheless the lines and the arcs and the circles are some of the most widely used pieces that we can draw on the canvas and if you would like to learn how to draw any of the other objects listed in the intro then teacher have links in the description of the lesson video in which you have the syntax on how you can go to drawing those.

So without further due lets go ahead and get started with drawing a line so lets go ahead and comment a section to differentiate between the line code and other canvas code as marked below



In order to get started with drawing a line we are going to need to our magic c variable an we are going to begin a path by beginPath method in c variable this is basically indicator for canvas saying okay we want to start a path (do not connect this path to anything proceeding it (maybe drawn over the previous shape (who knows))) So declared a canvas that we want to begin a path using beginPath method of c as shown below but we need to actually declare where on the canvas we want out path to start



So we are going to use moveto method of c which take two parameters that is x and y coordinate in numerical format so for x in first argument we take a 50 pixel and for y that is 300 in second argument now(I think by this we are moving the point from where we start making the line ) if you save the file and due to which the browser gets refreshed and you will not see anything because this point is visible until we call a stroke method as soon as we call the stroke method then you will see the line from point to point so now we have our starting point we can go ahead and create a line to a new point so to create a line to a new point

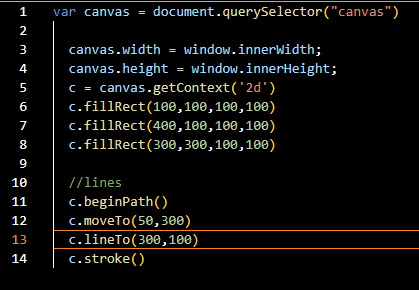
3:22

we will use a method lineTo in c object it also take two argument similar to moveTo

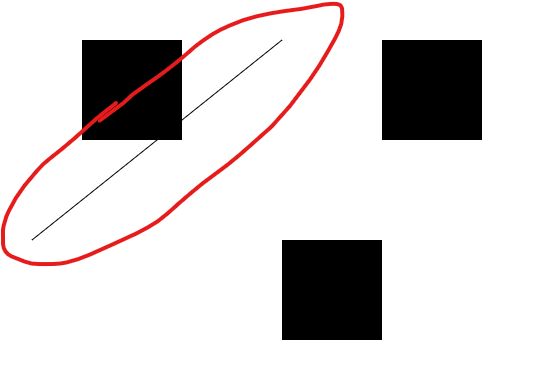
Here we specifies where we want our line to go to is we want it to go somewhere up in the screen because we have chosen the starting point near the bottom so we choose 300 for x and 100 for y



So we are basically drawing a line diagonal from bottom to up of I save the file and then the browser get refreshed because we need to call a stroke method for this line to show so call the stroke method in c variable now our code look like below



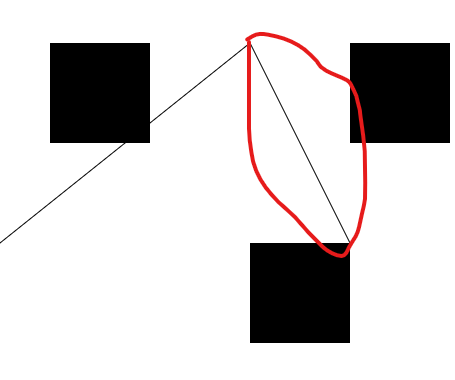
Now our line will be drawn as marked below



We now have a line expanding from one point to another and we can drawing on those lines by adding more lineTo method and our x is 400 and y is 300. As shown below



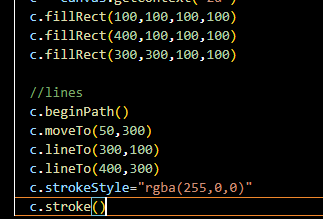
the line will start from the last point of previous line and now output will look like below

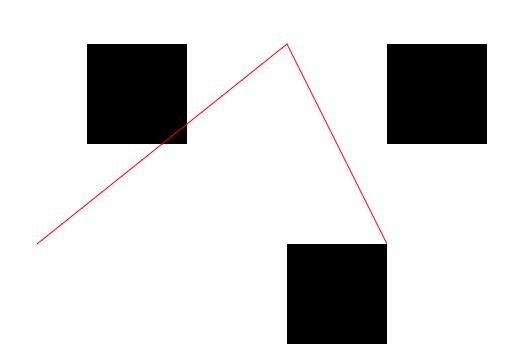


Our new line is marked above

We can keep doing these as many times as we would like but for lets go ahead and keep the thing as is. Now something else Teacher would like to cover is how to add colors to canvas first to our lines And colors to the inner Fills of our shapes So to add colors to our lines its actually simple before stroke method we need to add property to c (as object in js is expandable)

Now strokeStyle is equal to any css color it can be rgba value between double quotes by using rgba function like we use in css for colors and it can be text color value like blue, red etc or any hexadecimal value between double quotes. So I add a red color to the stroke as shown below

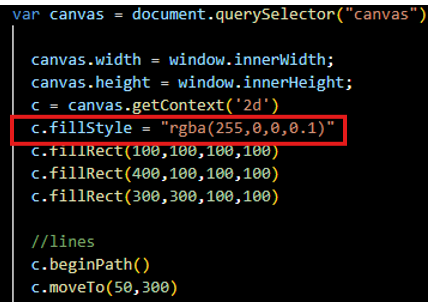




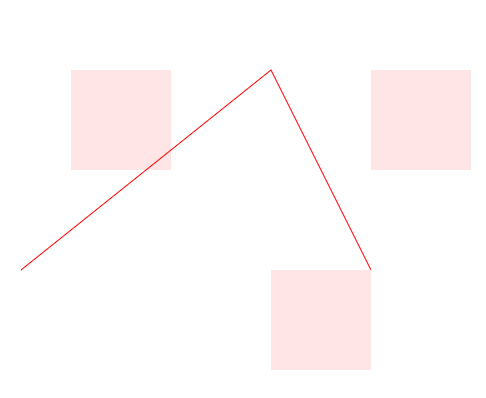
Now they are red in color So this is how we change our lines using a strokestyle property

Now our output will be like below but how do we change the colors for our filed rectangle well to change the color for our filled rectangle we need to add a fillStyle property rather than a stroke so here the value will be a color value which we discussed before.

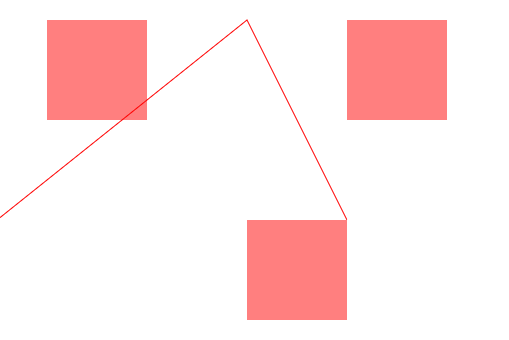
Now we add a fillStyle property as shown below



Now it will look like below

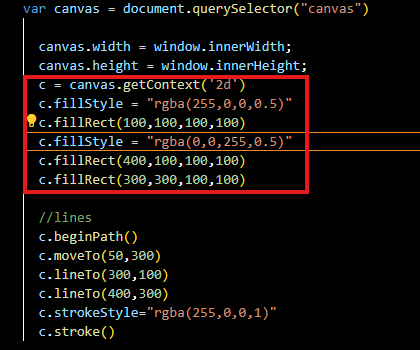


Now you can see that we have very light red rectangle instead of that harsh black rectangle that we saw earlier now if we want to change this color we increase the alpha of color in fillStyle so we can see it better as shown below

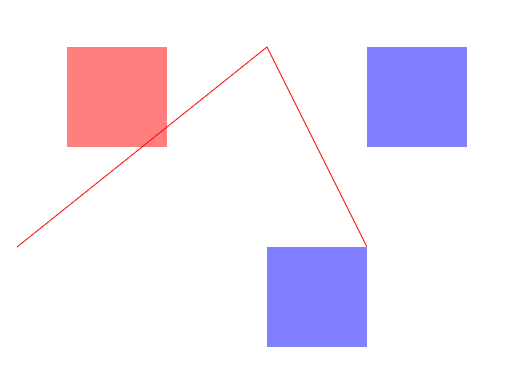


Now we have three semi transparent squares. But if we want to change one of these rectangle so only one of our rectangle is semitransparent. Well we can change the colors of all these rectangle by proceeding them with different fill styles so the fillRect is going to take whatever fillstyle before it and then itself with that color that is specified

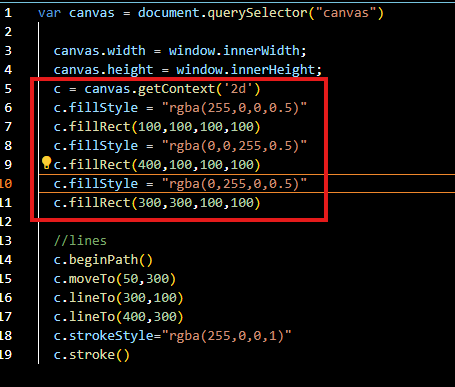
So we change other two rectangles to blue by doing this shown in the marked code below



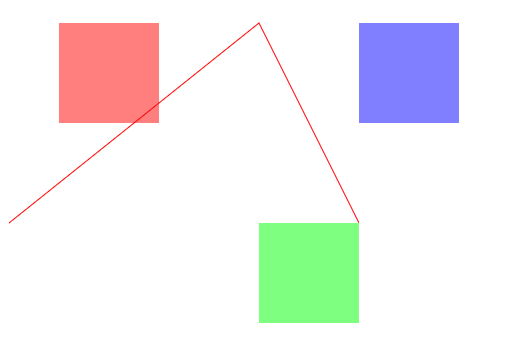
Now our output will look like below



So now we change the second blue rectangle to green as shown in the marked code below



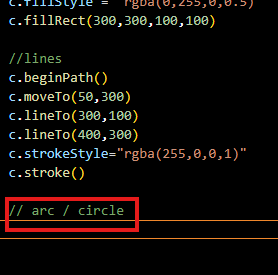
Now it will look like below



So now you can see we have different color line different colored rectangle and Teacher encourage us to play around with this a bit. Now go ahead and create some rectangles go ahead and create some lines with different colors

6:50

And kind of get a feel for things because that’s where you are going to learn the most is when you are actually playing around on your own trying to figure it out Okay How I do this , what’s the problem , How do I solve it on my own that’s when the knowledge is really going to be retained so Teacher really encourage us to play around with this for a little bit before we move on to the arc section so lets go ahead and learn how to create an arc with canvas we are going to be using arcs to create circles because once we create an arc that is 360 degree around that’s where we get our circles so to create an arc or a circle but at first we differentiate them by comments as marked below



We are going to write arc method of c and arc takes quite a few arguments so it takes an x value and it takes a y value telling the location of center from where the arch center itself to draw an arc and then it takes a radius we take radius that is 30(I think it w will tell is the difference between the center and the first pixel when the arc is starting to form) for now and it takes something called start angle and end angle now hey don’t take degrees they take radians

Essentially the start angle property says at what angle( between radius and the tangent and the straight line of 180 degree from the center ) would we like to start drawing our arc while the end angle property how long would we like the arc to go on (who knows) means what angle to stop ( between radius and the tangent and the straight line of 180 degree from the center ).

We have our start angle( between radius and the tangent and the straight line of 180 degree from the center ) . which declares at what angle do we want our arc to start we set it to 0 radian

And we want our arc to end at a radian of Math.PI \* 2 because that will give us the arc that will give us angle that extend all the way from the beginning to the very end of the circle and in next we tall that should we reach from starting angle to end angle clockwise and counterclockwise This is for our purpose is not a huge deal which one we choose so we are just going to say false for now ( means arc will be drawn clockwise ) and now we have this we actually have created an outline for arc so if you are refresh then you will not see anything

8:50

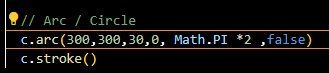
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Discovery:-**

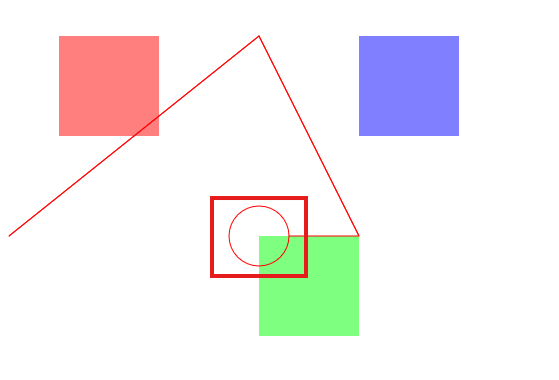
After begin path whatever you draw tht is line , circle they all will be connected to make a path

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Now we have just that outline but we need to fill that outline by using a stroke or a fill properly we are going to say c.stroke() as shown below



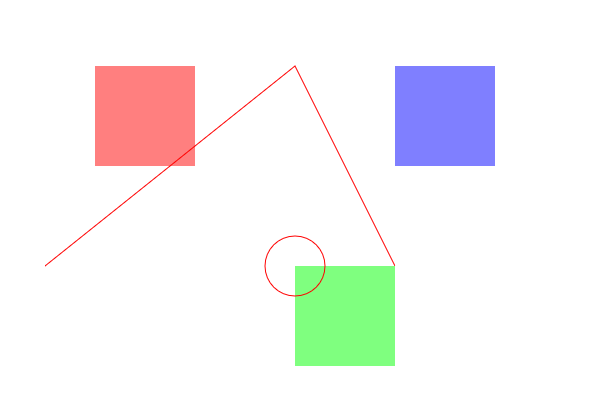
No in the output screen or circle will ok like marked below



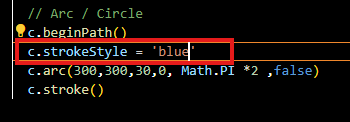
You can see the point where the circle was drawn and the last point of previous line as shown above

And we don’t want that where the beginPath method path comes in we want to make sure that we are proceeding any arc or any line with this beginPath method because it going to separate the two and prevent them from connecting to each other ( according to my observation that we use begin path then we have to start from beginning that is in case of line you have to use Moveto method again )

Now below you can see our circle is separated from our line as shown below

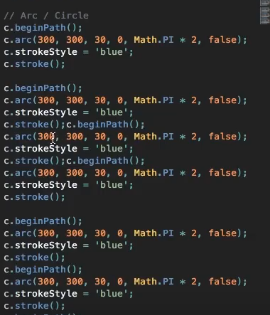


So now we know how to give our strokes and arc colors Lets go ahead and give our arc a different arc strokestyle and we give it a color blue as marked below in the code ( previously we set the c.strokestyle to red that why our line and circle was red in color bit now we are changing it in arc code that is to blue )



Now in our browser screen will look with disconnected blue circle

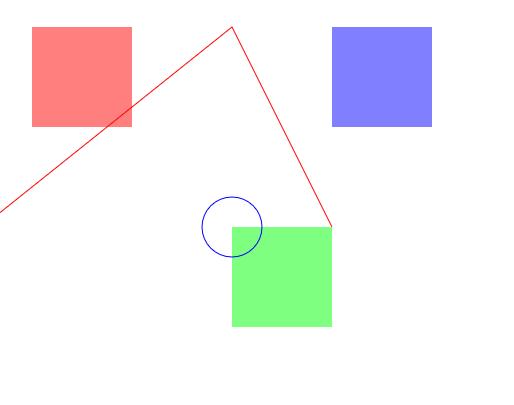
So these are the bare basics we know how to draw rectangles we know how to draw lines and connect the lines and we know how to draw a circle and arcs basically we also know how to change their stroke color and fill colors and this may not look great t this moment but as Teacher mentioned these are then main building blocks we will need to create any amazing canvas piece once we starts using colors and start animating these guys and creating multiple shapes that’s when the pieces start to come together and that’s when we will be having fun and creating eye appealing visuals so lets leave off on this let’s go ahead and ask ourselves a question what if we want to create 100s of circle How would we go about doing that well Teacher guess if we look at our code what we could do is copy and paste our circle code a 100 times as shown below



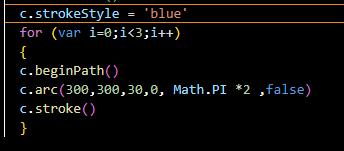
but you can see that’s going to get out of hand very quickly we would have to change the coordinates for each of these circles over and over again and that is just a maintenance nightmare you do not want to do that if you are creating multiple circles so let me go ahead and show you how to create multiple circles using a for loop Now Teacher will explain to whom who are not familiar with the concept of for loop I know it so I will skip it.

11:20

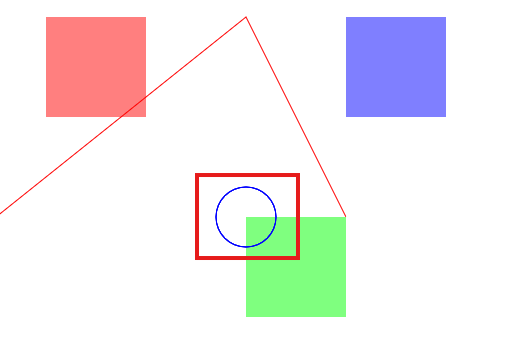
For now without loop our output will look like below



Now we add a for loop in our code as shown below

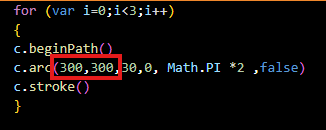


Now after adding it our output will look like below



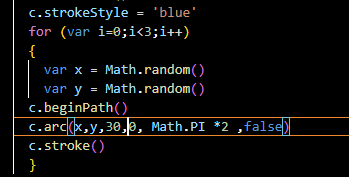
Now you can see in marked above that now the circle is looking slightly bolder as compared t when there was not loop

Because we are drawing three circles on top of each other Nowe this is not being effective we want that our circles are being drawn in wither random location on the screen or maybe coordinated location(x,y) on screen but definitely on top of each other because it does not make sense to draw three circles on each other just to change the linewidth when we can do It manually using a linewidth property but let’s go ahead and change the location of these three different circles. Now to change these location we need to make sure that we are changing these x and y value as marked below

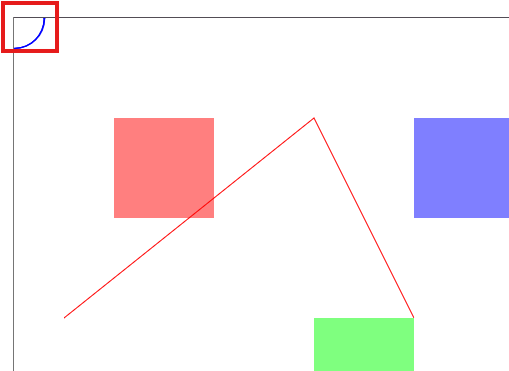


So they are different from each other right now they are the same so therefore being drawn on top of each other

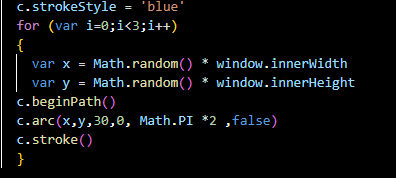
We need to make sure that these are different each time we run through this for loop so a really easy way to do it is adding some variable that is one for x coordinate and one for y coordinate we are going to set these variable to Math.random as we know Math.random returns a value anywhere between 0 and 1 to any random value it can be 0.4 or 0.6 or 0.65 it will store the value between x and y and that means each time we run though this loop we are going to return a random value.Now we pass these variable as the argument in the arc method for x and y position. Now our code will look like below



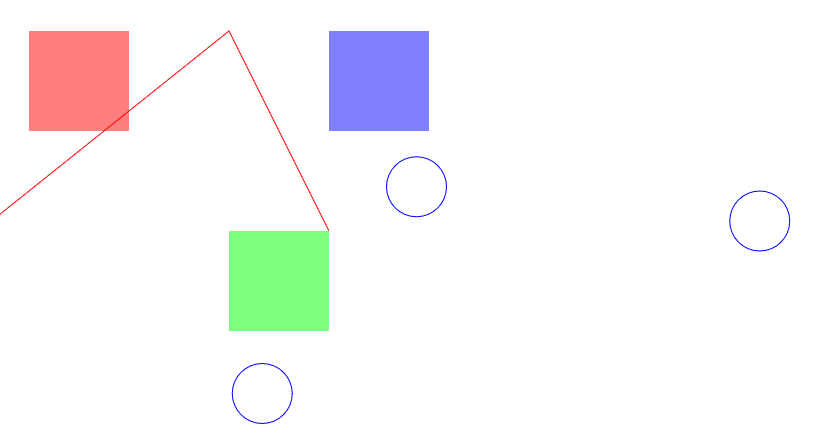
Now our output will look like below



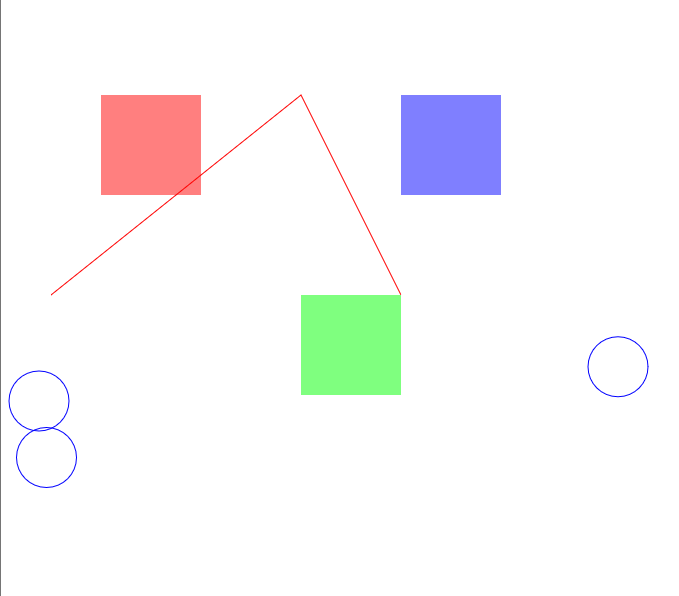
Now we can see our all circles are still on top of each other but now present on the top left corner of the canvas as marked above. Now we want to make sure that these circles are randomized throughout the full width and height of the browser window screen so in order to do that we are going to multiply Math.random to window.innerWidth fir our x value and then for y we are going to multiply Math.random by window.innerHeight and basically what we are doing So badically what we are doing is that we are returning a random value anywhere from 0 to the full innerwidth and then we are also saying for the y value that is give us any value anywhere from 0 to the full innerHeight of our screen Now our code look like below



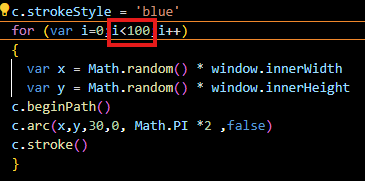
Now we save our file and now our output will look like below



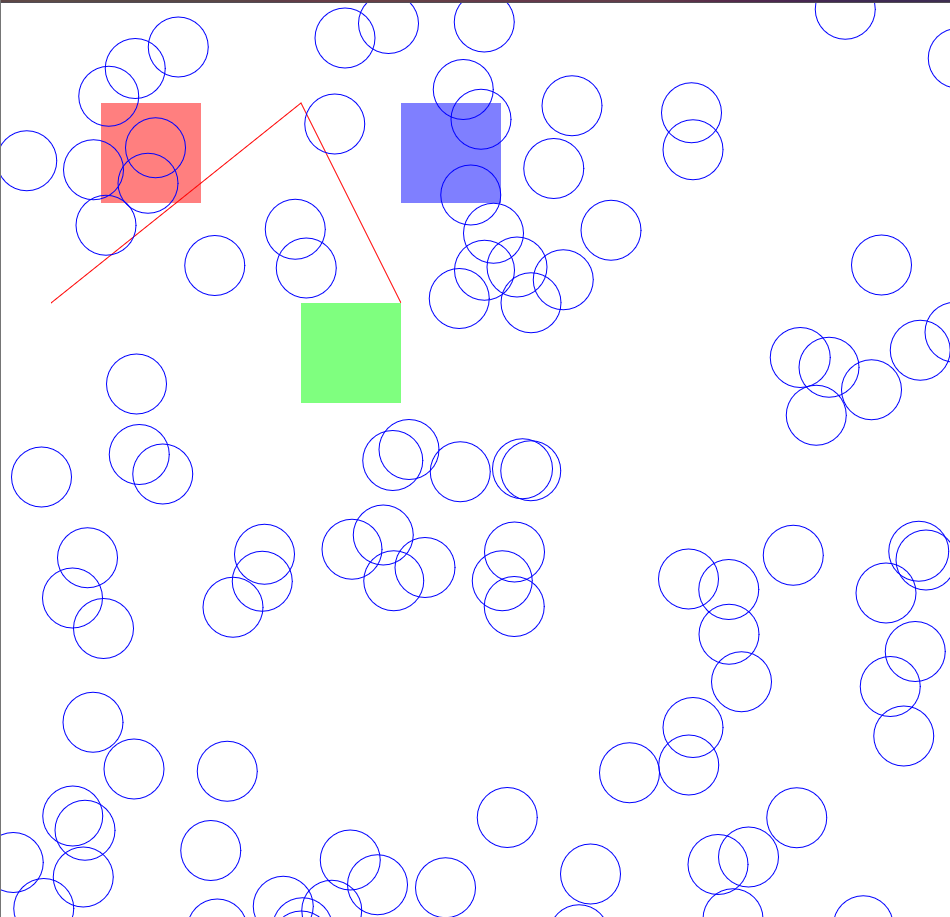
Now if we refresh the page then it will look like below



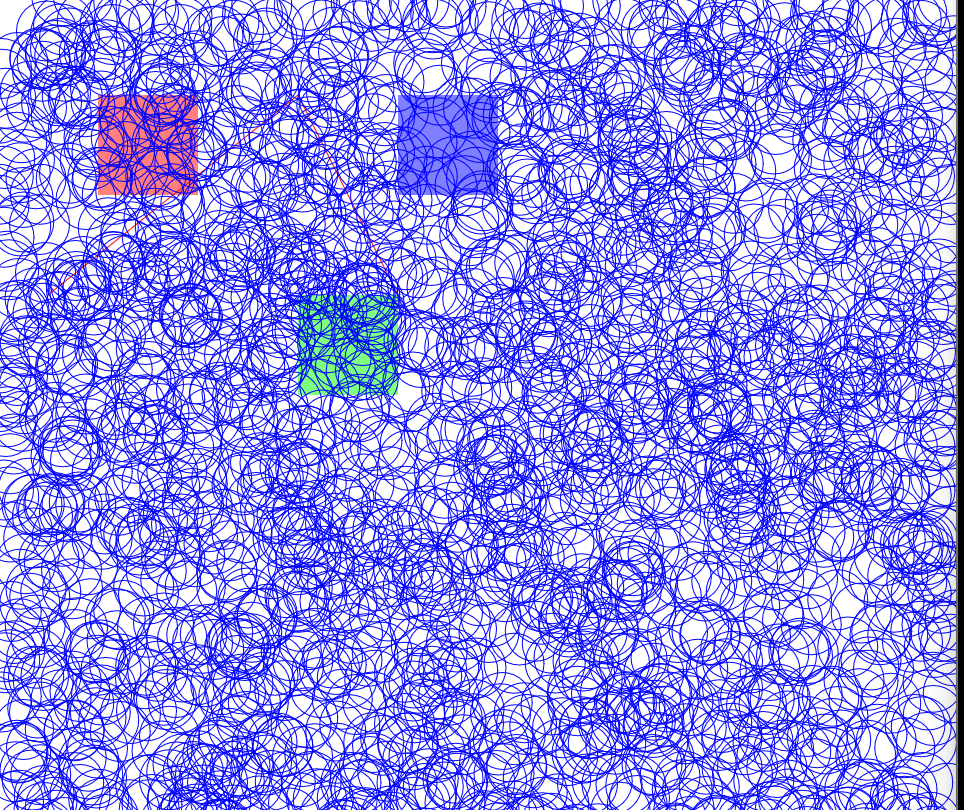
Now you can see our circles are randomized each time we refresh it and this is pretty cool this is starting to get a little more interesting then it was before because previously we just had static shop but now actually we are having some things move around randomly we are starting to procedurally generate thing which is actually pretty fun to play around with so you are not limited to just three circles you can change it in for loop and you can specific how much you want to run it if I use 100 in place of 3 in loop condition as marked below



Now we save our field and now our output will look like below



If we make 2400 then it will look like below



Now if we go higher then your computer may start to draw a lot of circles and it can’t actually handle it that computationally so teacher encourages to play around as well . Its really fun start changing x and y values and see what kind of stuff you can come up with start bringing line in here randomized

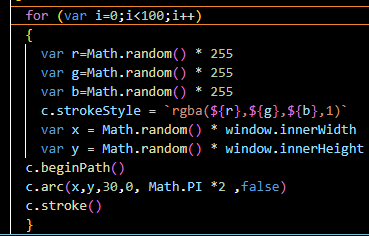
The value of the lines we made , randomize the values of your rectangle this Is exactly you need to do start creating good looking procedurally generated animation So as a challenge to you before we start heading to our animation section Teacher challenge us to randomize the color of these strokes od random circles we drawn any color from the color wheel go ahead and randomize this so that’s it ‘s calling any color

It will provide a good challenge so you familiar with the for loop and also familiar with the idea of randomizing values Get a feel for random number generators over here, the for loop

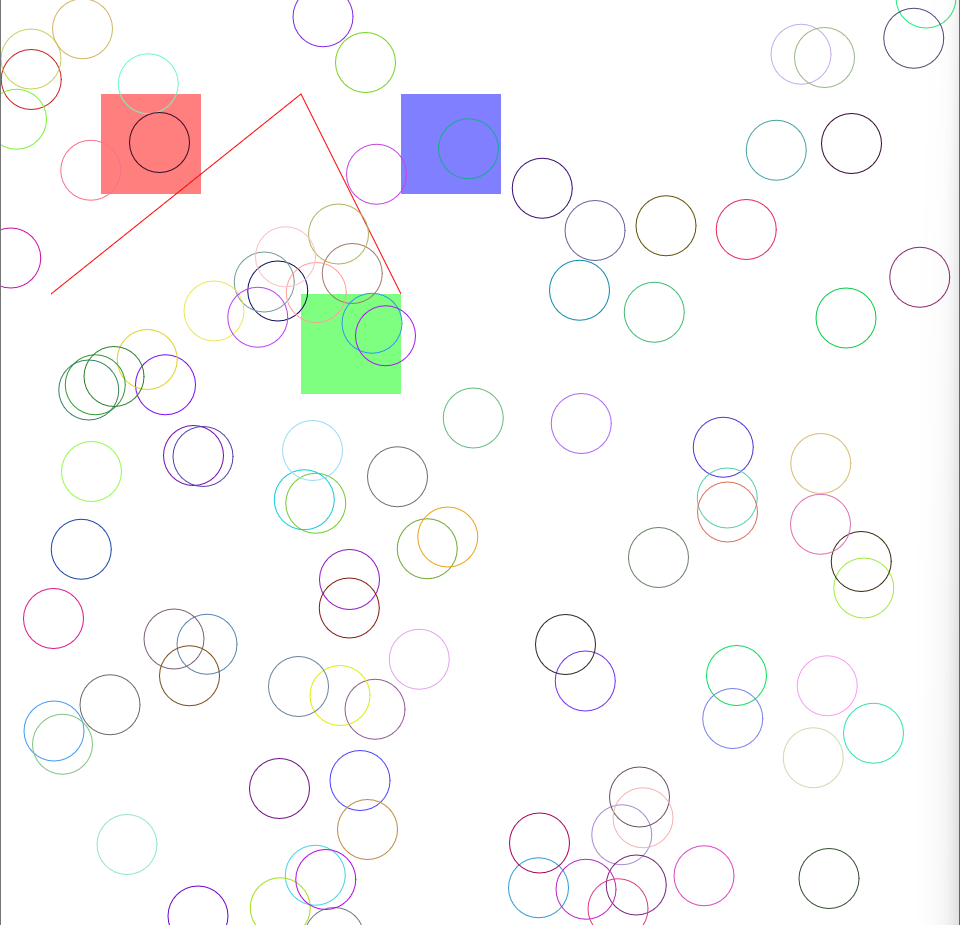
15:33

So we now know how to draw three most basic object onto the canvas. We know how to draw rectangle , how to draw lines , we also know how to do arcs/circles So the next one is where the magic really starts happening we created some magic here by randomizing and creating multiple circles at once but once we add animation on the mix that’s where things really start getting cooler we are going to start adding conditions for our circle to bounce off balls and then in the fourth episodes we are going to add event listener so that when we hover over the circles they actually grow in size So that’s it for this one folks get a feel for the random numbers generators over here , for loop because we are going to be using this over and over again in our future canvas pieces

Now I completed the challenge and it is my code as shown below



Now our output will look like below



See you in the next lesson.