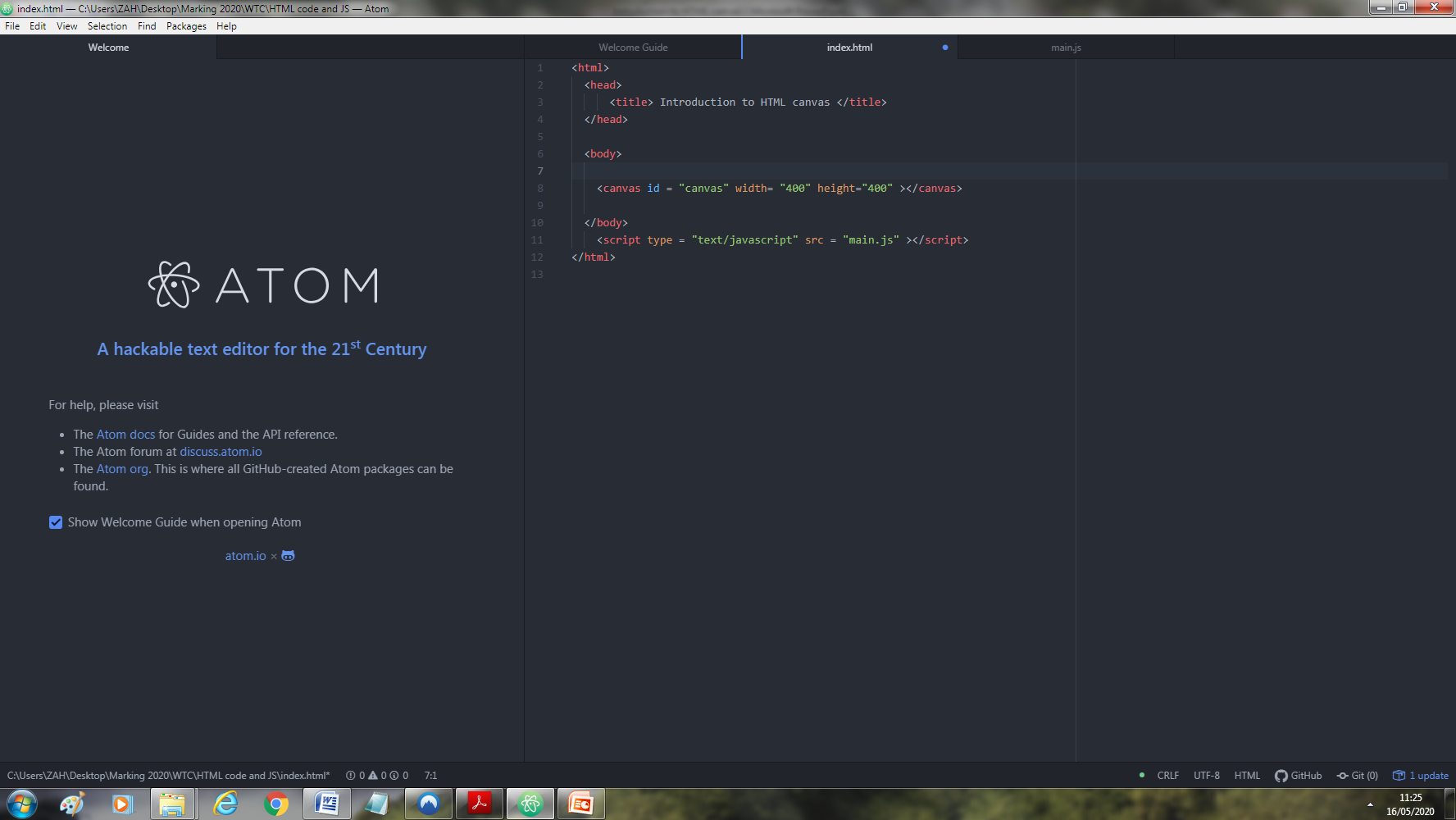
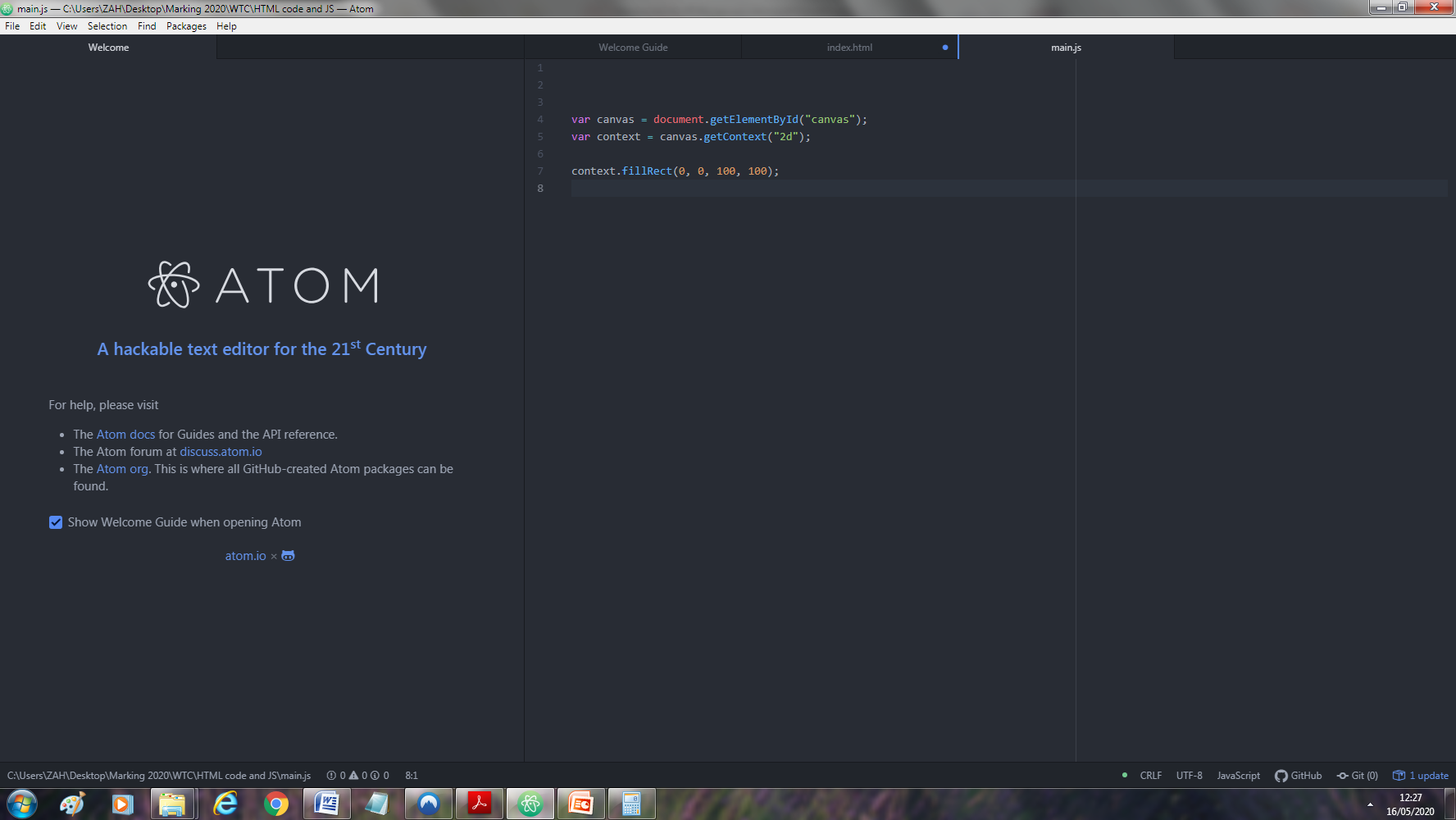
**Introduction to HTML Canvas**

HTML

****

Copy the code above into your index.html document.

Javascript



Copy the code above into your main.js document. Save it then open your index.html file.

Drawing square\rectangle code

context.fillStyle="red";

context.fillRect(10,10, 10, 10);

context.strokeStyle = "black";

context.strokeRect(10, 10, 10, 10);

**Activity 1**

Now try creating several squares or rectangles of different sizes, different colours and on different points on the canvas.

Provide code here:

*JavaScript understands more than 100 color names, including Green, Blue, Orange, Red, Yellow, Purple, White, Black, Pink, Turquoise, Violet, SkyBlue, PaleGreen, Lime, Fuchsia, DeepPink, Cyan, and Chocolate. Try any of these colours.*

End of activity 1

Drawing line\s on canvas

Now you will learn how to draw a line on the canvas. Remember to continue entering the code in the main.js file.

context.moveTo(0, 0);  
context.lineTo(200, 100);  
context.stroke();

Create more than one line on the canvas

context.beginPath();

context.strokeStyle = 'blue';

context.moveTo(20, 20);

context.lineTo(200, 20);

context.stroke();

context.closePath():

context.beginPath();

context.strokeStyle = 'green';

context.moveTo(20, 20);

context.lineTo(120, 120);

context.stroke();

context.closePath():

Creating a triangle using lines

context.strokeStyle = "SkyBlue";

context.beginPath();

context.moveTo(100, 100);

context.lineTo(10, 200);

context.lineTo(200, 200);

context.lineTo(100, 100);

context.stroke();

**Activity 2**

Now try creating the following image below:

****

Provide code here:

End of activity 2

Drawing circle\s on canvas

context.beginPath();

//parameters x, y, radius, start and end

context.arc(400, 400, 50, 0, 2 \* Math.PI);

context.fill()

Drawing text on canvas

context.font = "30px Arial";  
context.fillText("Hello World", 10, 50);

change text colour and position

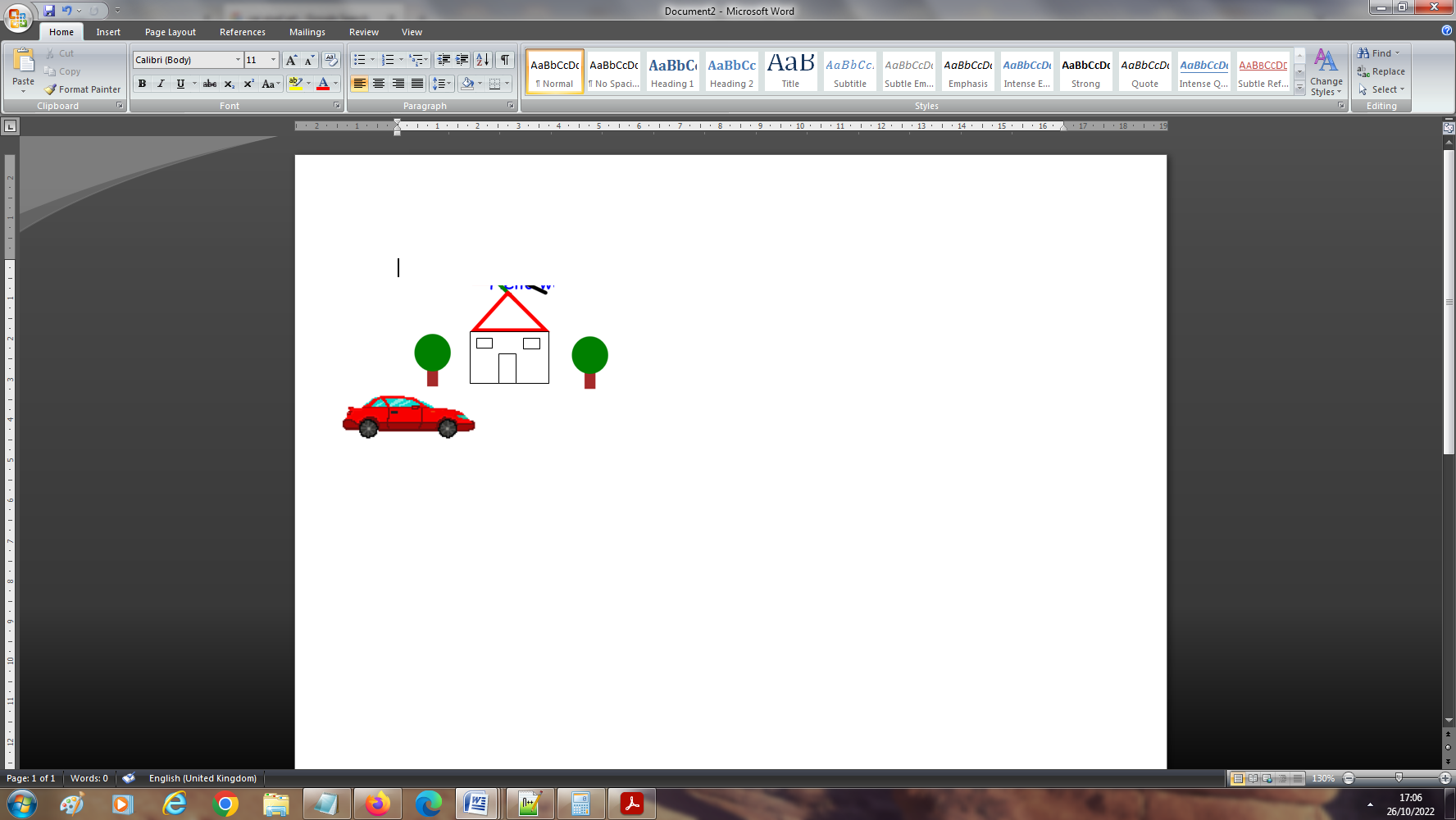
context.font = "30px Comic Sans MS";  
context.fillStyle = "red";  
context.textAlign = "center";  
context.fillText("Hello World", 30, 30);

change text to stroke

context.font = "30px Arial";  
context.strokeText("Hello World", 10, 50);

**Activity 3**

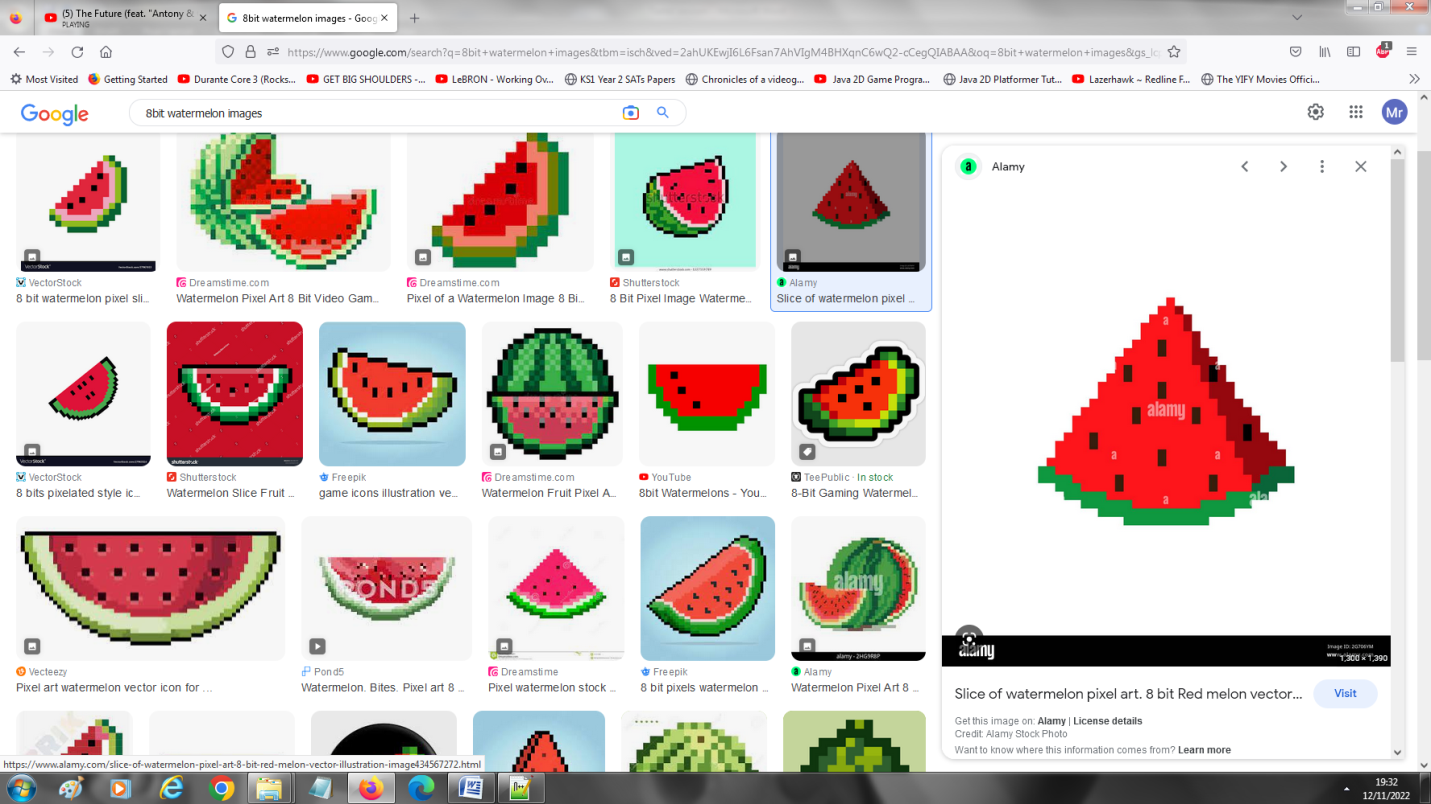
Add to your current drawing so that it has two trees beside the house along with text below it. The text should have “This is a house” in any font style and colour of your choice.



*This is a house*

Creating images using rectangles

So now we will combine what we already know about drawing on a canvas to create images via pixel art. We will create the image below:



If we want to create a line of squares then we would need to find a way to repeat the process of creating a square\rectangle. We will use a loop to produce this using the code below:

for (var x=90; x < 110; x+=10) {

Repeat method

var y=120;

context.fillStyle="red";

context.fillRect(x+10, y, 10, 10);

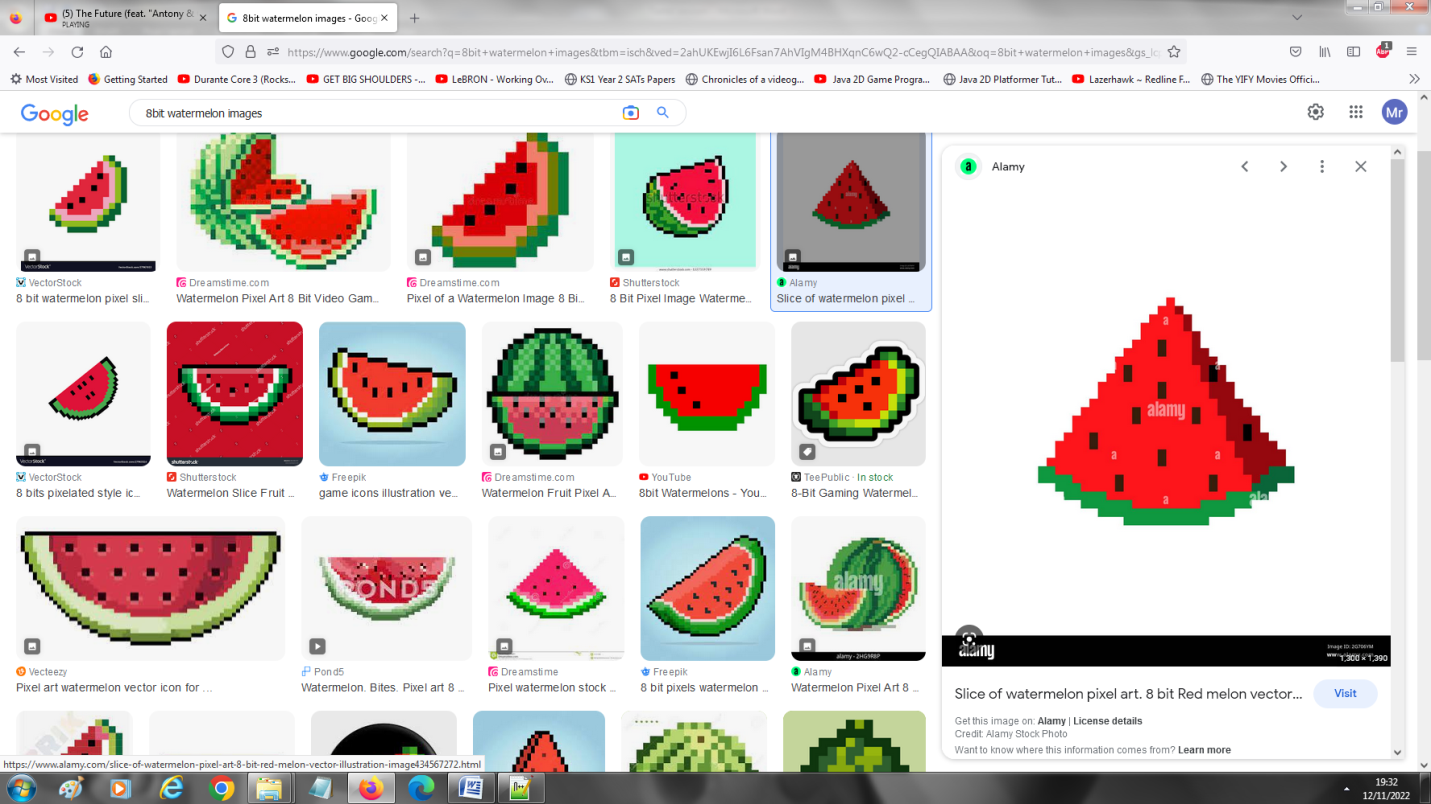
}

The image can be attained by going to the following link

<https://github.com/zahmedSSFC/CatBalloon.git>

**Activity 4**

Create the image of the watermelon using the repeat method shown above.



CatBalloon activity

In this activity we will learn how to add images, create animation along with interactivity between the user and image.

All files for this can be downloaded from link below:

<https://github.com/zahmedSSFC/CatBalloon.git>

how to add images

var img = new Image();

img.onload = function(){

// execute drawImage statements here

context.drawImage(img,20,100);

};

img.src = "images/dino.png";

Make cat move up

var balloon = -1.5;

var catY = 150;

function draw(){

context.clearRect(0,0, canvas.width,canvas.height);

context.drawImage(img, 20 ,catY);

catY += baloon;

};

let game = setInterval(draw,10);

Add interactivity

Now go back to your code where you have set catY to 150 and add the following code below it. It should look like this.

var catY = 150;

Here is the new code added below:

document.addEventListener("click",moveDown);

function moveDown(){

catY += 75;

}