**Searching for any element in an array**.

Let’s say, we have an array of numbers. Now, we need to check whether a certain number is present in that array or not.

● We need to create an array of numbers.

● Then we will write a function that checks for the number we are looking for.

● This function will give us the result and based on the result we check whether a number is present in the array or not.

Let's consider a scenario where you are in a hotel with your friend and playing hide and seek and there are 10 rooms in the corridor and your friend can hide in any of the rooms. So how do you find which room your friend is hiding?

First you go to the first room, then second and so on until you find your friend. So here we are going to do the similar process. We will create a for loop that will go to each element of the array and inside the for loop we will compare every element with the number we are looking for.

If the number matches with that element, we will return the index of the number else we will return -1.

So now based on this knowledge we are writing a condition that gives us the result.

**Random number**.

First let us learn to generate a random number, save it in a variable and then print it on a console. Math operations/functions are inbuilt in JavaScript. Math provides us with a random function which we can use to get the random number.

Everytime we run the code, a different random number will be printed on the screen.

When we create the random number using the random() function. It gives us numbers without decimals which are called integers. But we only need whole numbers such as 1,2,3,4 etc. To make sure we get the whole numbers we need to round off the numbers generated by the random() function. Rounding off means if the number is having a decimal value such as 9.7, Math.round() function will make it 10. The round() function rounds off the number to the closest whole number

var rand= Math.round(random(1,6));

**frameCount**

Our game is made up of frames. These frames run together to give the appearance of movement. The draw() function is called by our game for each frame.

You can imagine a frame as an image. Each image has the state of the game.

The number of images our computer can show per second is called a framerate or frameCount; the more frameCount we have, the smoother the game looks. You can actually print the frameCount of the game in the console by displaying in console.log(frameCount).

We can tell the computer to create a cloud only after 60 frames. How do we do this? Right! The teacher makes the student do this for a different frameCount. What will the computer do if the frameCount is 10? For now, you can assume we are doing the division in whole numbers only. What will the computer do if the frameCount is 30? What will the computer do if the frameCount is 60? ESR: We can tell the computer to divide each frameCount by 60 and if the remainder is 0, create a cloud for that frame. ESR: The computer will divide 10 with 60. Remainder will be 10. So no clouds will be made. ESR: The computer will divide 30 with 60. Remainder will be 30. So no clouds will be made. ESR: The computer will divide 60 with 60. Remainder will be 0. So a cloud will be made.

There is an operator in Math called **modulus (%)** which gives you a remainder. Let us write code to generate clouds only if frameCount divided by 60 gives a remainder of 0. using the comparison (“===”) sign?