**Arrays:**

By definition, an array is a collection of the same or different data types.

Such as you can create an array named friends and name of your friends in it like

friends = [“sam”, “john”, “ramy”]

One more example, if you want to store your marks of all the subjects, storing them in multiple variables would be a very tedious job. We have to define multiple variables. But instead we can create a single array and store all the marks in one place.

To add the elements in the array we set the array name equal to the **square bracket**, and then add the elements in the square bracket separated by a comma(,).

Once elements are added in the array, we can print the array in the console by using console.log() function.

To access the elements of the array, we need the index of that element.

**Index** is just like an address of the element inside the array. But the important thing is that the **index always starts from 0**. Our first element is always at 0 index.

To get the element we will write the name of our array, then square bracket and inside the brackets we will write the index; here it is 0.

**Edges** is an array

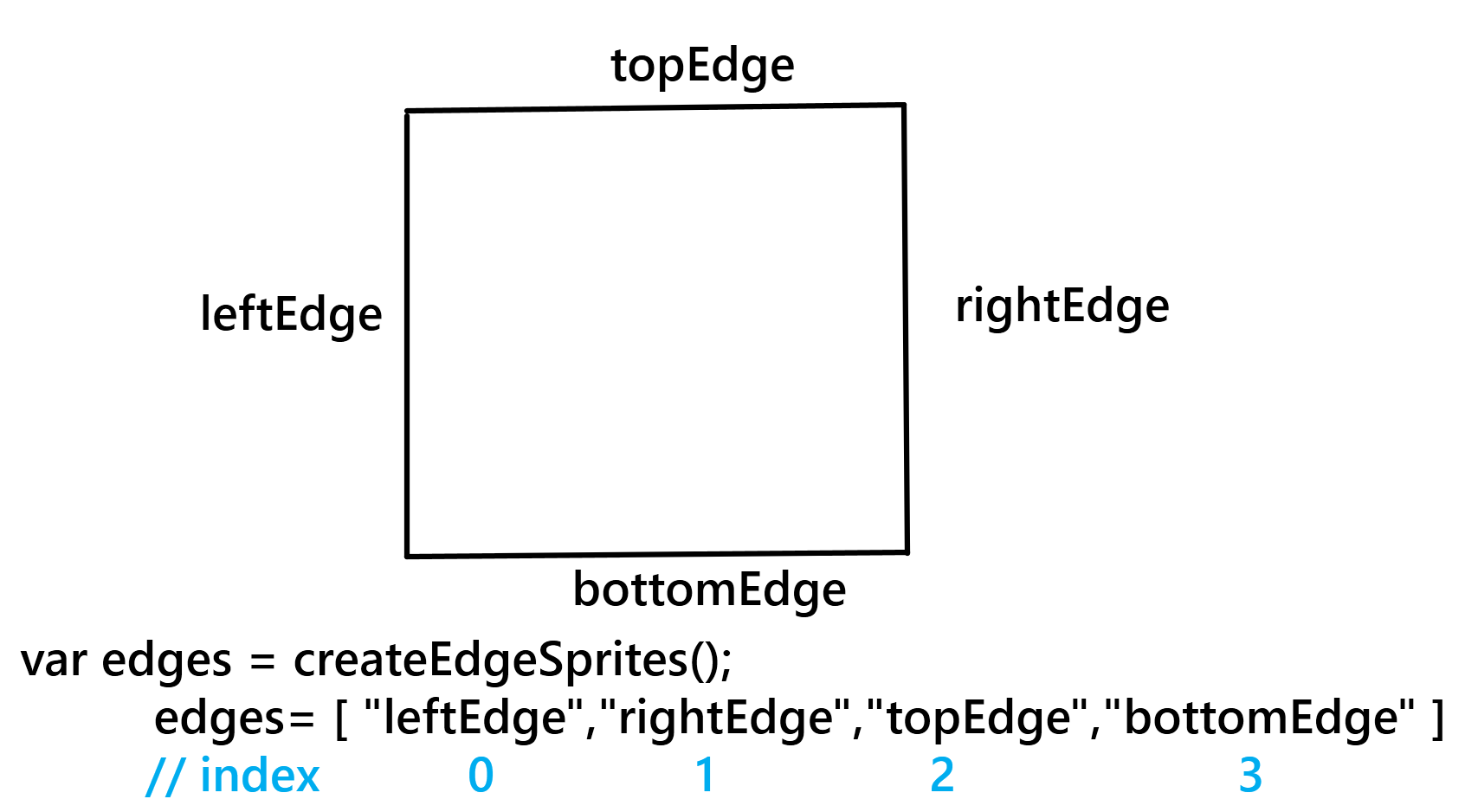
edges=[“leftEdge”, “rightEdge”, “topEdge”, “bottomEdge”]

so 0th position- leftEdge -- edges[0]

1st position- rightEdge -- edges[1]

2nd position- topEdge -- edges[2]

3rd position- bottomEdge -- edges[3]



**Position**

var s1=createSprite(100,100,20,20);

var s2=createSprite(200,100,20,20);

var s3=createSprite(300,100,20,20);

var sprites= [ s1, s2, s3 ]

To access the x position we need to write sprites[0].position.x.

**p5.play.js**.

This is a game library which code.org was internally using to allow you to create games!

Creating animation is similar to adding images. p5 has something called **loadAnimation** and **addAnimation** to add animations to the sprite.

We use **loadAnimation** to load all the images for the animation and then use **addAnimation** to add the animation to the sprite.

Load our animation in the **preload**() function and in the preload function we will define our animation using loadAnimation() function.

Note: Using loadAnimation() to load the animations in the preload function, you can load multiple images in a single variable to make it animated. addAnimation takes two arguments - a label which could be any string and the name of the animation which was loaded.

**What does gravity do?**

Due to gravity the velocity keeps reducing. It becomes 0 at the top but due to gravity it starts gaining velocity again when it starts falling towards the ground.

Gravity will slow down the object and reduce its velocity to 0. It will then pull the object down.

**Comments**

When we write the code, we understand what is happening in each line. If you share the program with your friends, they may not understand the meaning of each and every line. And also if you open your own code after some time you also may not understand each and everything at once. To overcome this problem we can use **comments** in the code.

Comments are written in plain English to explain the purpose or working of code. Comments are not executed when we run the program. In JavaScript to add the comment we first write 2 forward slashes (//); after this you can write the comment.

**draw()**

Do you know just like movies, games are made up of many frames? Anything we write inside the function draw() runs for every frame. Anything we write outside the function draw() runs only once.

**Trex game**

There are two ways to make the player feel that the dinosaur is moving. One would be to give some forward velocity to the Trex and the other would be to give some backward velocity to the ground.

Can you think of a way where the Trex never falls and the ground keeps scrolling infinitely?

One simple way would be to reset the ground back to the center of ground.x < 0. Once the ground has crossed the screen to the left, we are bringing it back to its original position. This way the ground will always be there.