
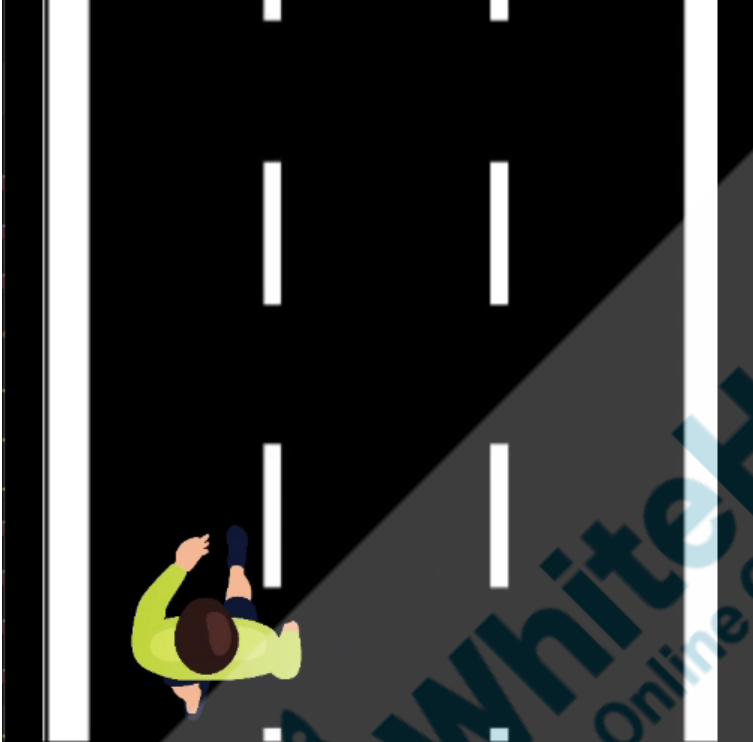



| Topic | RANDOM NUMBERS | |
|---|--|---|
| Class Description | Students learn how to spawn game objects at random positions in the game. | |
| Class | C12 | |
| Class time | 45 mins | |
| Goal | <ul style="list-style-type: none"> • Generate random numbers and use them inside a game. • Use the concept of frameCount to give a delay in the game. • Spawn a sequence of game objects at different positions. | |
| Resources Required | <ul style="list-style-type: none"> • Teacher Resources: <ul style="list-style-type: none"> ○ VS Code Editor ○ Laptop with internet connectivity ○ Earphones with mic ○ Notebook and pen • Student Resources: <ul style="list-style-type: none"> ○ VS Code Editor ○ Laptop with internet connectivity ○ Earphones with mic ○ Notebook and pen | |
| Class structure | Warm-Up Student - led Activity 1 Teacher - led Activity Student - led Activity 2 Wrap-Up | 10 mins 10 mins 10 mins 10 mins 5 mins |
| WARM UP SESSION - 10mins | | |
| <div>  </div> <p>Teacher starts slideshow from slides 1 to 7</p> <p>Refer to speaker notes and follow the instructions on each slide.</p> | | |

| Activity details | Solution/Guidelines |
|--|--|
| <p><i>Hi, so good to see you again! How have you been? Are you excited to learn something new?</i></p> <p>Run the presentation from slide 1 to slide 4.</p> <p>Following are the warm up session deliverables:</p> <ul style="list-style-type: none"> Connecting students to the previous class. <p>Excited to start today's class?</p> | <p>ESR: Thanks, yes I am excited about it.</p> <p>Click on the slide show tab and present the slides.</p> |
| QnA Session | |
| Question | Answer |
| <p>Which line of code will give a moving animation to the boy?</p>  <p>A. <code>boyImg = loadAnimation("Runner-1.png");</code></p> <p>B. <code>boyImg = loadAnimation("Runner-2.png");</code></p> | <p>C</p> |

| <p>C. <code>boyImg = loadAnimation("Runner-1.png","Runner-2.png");</code></p> <p>D. <code>boyImg = loadAnimation("Runner-1.png","Runner-1.png");</code></p> | |
|---|---|
| <p>Select the line of code to move the boy left and right using a mouse.</p>  <p>A. <code>boy.y = World.mouseY;</code> B. <code>boy.y = World.mouseX;</code> C. <code>boy.x = World.mouseY;</code> D. <code>boy.x = World.mouseX;</code></p> | <p>D</p> |
| <p>Continue the warm up session</p> | |
| Activity details | Solution/Guidelines |
| <p>Run the presentation from slide 5 to slide 7 to set the problem statement.</p> <p>Following are the warm-up session deliverables:</p> <ul style="list-style-type: none"> Introduce students to the problem statement for student activity. | <p>Narrate the story by using hand gestures and voice modulation methods to bring in more interest in students.</p> |

| | |
|--|---|
| | |
| <div>  </div> <p>Teacher ends slideshow</p> | |
| STUDENT-LED ACTIVITY 1 - 10 mins | |
| <ul style="list-style-type: none"> Ask the student to press the ESC key to come back to the panel. Guide the student to start Screen Share. The teacher gets into Fullscreen. | |
| <p>ACTIVITY</p> <ul style="list-style-type: none"> Display the elements greater than a number from the array. | |
| <p>Step 1: Student-Led Activity</p> <p>In the previous classes, we have seen how to create arrays, access elements manually as well as using a for loop.</p> <p>In this class we are going to perform an activity using the arrays but it will be different from the previous one. We are going to use the same concepts but this is like a challenge for you. But don't worry I am here to guide you.</p> <p>We have an array of marks, 10 students took the math test and you need to find out how many of them passed the test.</p> <p>We have the marks stored in the array, passing marks are 45. We only need to display the marks of the students having marks more or equal to 45.</p> <p><i>Encourage the student to solve the problem by writing code.</i></p> <p>The problem is we need to display very specific elements from the array having value equal or more than 45.</p> | <p>The student downloads the code from the Student Activity 1 and opens it in the VS code editor.</p> |

The first step is to use a for loop to go through each element of the array, we need to compare each and every element with 45. If that particular number is more than or equal to 45, then we will display that using **console.log()**.

Write the for loop in the setup() function. In the for loop, our variable i will go from 0 to the last element of this array with increments of 1.

In the curly brackets of the for loop, we are going to use the if condition.

The condition says:

if(marks[i] >=45

then console.log(marks[i])

marks[i] stores the element of the array at the current iteration. As the value of i increases we check each and every element of this array.

And if our condition gets satisfied, an element will be displayed on the console.

```
var marks = [35,38,42,45,43,34,46,41,48,32];

function setup() {
  createCanvas(400,400);

  for(var i = 0; i<marks.length; i++)
  {
    if(marks[i]>= 45)
    {
      console.log(marks[i])
    }
  }
}
```

OUTPUT:

```
45
46
48
```

Teacher Guides Student to Stop Screen Share

Now let's move on to the game activity of the class.

TEACHER-LED ACTIVITY - 10mins

Teacher Initiates Screen Share

CHALLENGE

- **Generate clouds in the background at different positions in the game.**



Teacher can show slideshow from slides 8 to 18.
 Refer to speaker notes and follow the instructions on each slide.

Step 3:

Teacher-led Activity

(10 mins)

The teacher opens the [Teacher Activity 2](#)

In the previous classes we have added T-rex and made it jump.

Here will add the clouds in the sky, but we want to have the clouds at different positions and that position should be changing randomly.

This we can do using 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.

The teacher opens the documentation from [Teacher Activity 4](#) and goes through the random functions.

The student opens the documentation from [Student Activity 3](#) and follows with the teacher.

The teacher writes code to generate a random number between 1 to 100 and stores it in a variable called 'rand'.

*The teacher comments any other **console.log** message and prints this variable 'rand' on the console.*

The student observes and learns.

```

34
35 //creating invisible ground
36 invisibleGround = createSprite(200,190,400,10);
37 invisibleGround.visible = false;
38
39 var ran= Math.round(random(10,60));
40 console.log(ran);
41
42 }
43
44 function draw() {
45 //set background color
46 background(220);
47
48 console.log(trex.y);
49
50 // jump when the space key is pressed
51 if(keyDown("space")&& trex.y >= 100) {
52   trex.velocityY = -10;
53 }
54
55 trex.velocityY = trex.velocityY + 0.8

```

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

The teacher runs the code a number of times to show the different numbers being shown as an output on the console each time the program is run.

The student observes.

So far, our program has always run in a similar way each time we run the code. But now our program runs differently each time we run the code.

When the same program runs differently each time the player plays, it can allow for different game scenarios.


Can you imagine ways in which we can use this feature in different games that we might design in the future?

| | |
|--|-------------------------------------|
| | ESR: Varied. |
| <p>Let's start with a simple scenario.</p> <p>Let's start by spawning clouds in the game at different random heights.</p> | |
| <p>We will need to teach the computer how to spawn the clouds - that is create clouds programmatically which appear at different points in the game.</p> <p>Remember we can teach the computer how to do anything with the use of functions. Let us create a function called spawnClouds() and use it in our program. You are going to teach the computer to spawn the clouds.</p> <p><i>The teacher writes the code for an empty spawnCloud function and uses it inside the program.</i></p> | <p><i>The student observes.</i></p> |

JS sketch.js > spawnClouds

```
43
44 function draw() {
45   //set background color
46   background(240);
47
48   console.log(trex.y);
49
50   // jump when the space key is pressed
51   if(keyDown("space")&& trex.y >= 100) {
52     trex.velocityY = -10;
53   }
54
55   trex.velocityY = trex.velocityY + 0.8
56
57   if (ground.x < 0){
58     ground.x = ground.width/2;
59   }
60
61   //stop trex from falling down
62   trex.collide(invisibleGround);
63   spawnClouds();
64   drawSprites();
65 }
66
67
68
69 function spawnClouds() {
70   //Write code here to spawn the clouds
71
72 }
73
74
```

Teacher Stops Screen Share

| | | |
|---|---|--|
| | | <i>The student opens Student Activity 1 and creates a duplicate.</i> |
| | | |
| STUDENT-LED ACTIVITY 2 - 10 mins | | |
| | Now it's your turn. Please share your screen with me. | |
| <ul style="list-style-type: none"> • Ask the student to press the ESC key to come back to the panel. • Guide the student to start Screen Share. • The teacher gets into Fullscreen. | | |
| <p style="text-align: center;"><u>ACTIVITY</u></p> <ul style="list-style-type: none"> • Use a random math function to generate random positions for the clouds in the game. • Spawn clouds in a sequence. • Arrange the game objects at appropriate depths. | | |
| <div style="text-align: center;">  </div> <p>Teacher starts slideshow :Slide 19-20 (Only 2 slide for this Activity)</p> | | |
| <p>Run the presentation slide 19 & 20 to set the student activity context.</p> <p>Now it's your turn. We will be adding the code in the function we just created, function <code>spawnClouds()</code>.</p> | | |
| <p>Why don't you share your screen and try to spawn the clouds?</p> <p>Let's open the student activity and start writing the code to spawn the clouds.</p> | | |



Teacher ends slideshow

Step 4: Student-Led Activity

Let us write code to create just one small cloud sprite.
Generate it outside the screen and give it some x velocity
so that it appears moving.

*(Use the following questions to revise some concepts with
the student before or during coding of this point.)*

- How do you create a sprite?
- What position would you give to the cloud if you want the cloud outside the screen to the right?
- What would be the x velocity of the cloud if you want the cloud to move from right to left?

*Student download the
[Student Activity 2](#) code.*

*And writes code to
generate a single cloud
sprite and gives it some
velocity.*

*The student runs the code
and observes the output.*

```

48   trex.velocityY = trex.velocityY + 0.8
49
50   if (ground.x < 0){
51     ground.x = ground.width/2;
52   }
53
54   trex.collide(invisibleGround);
55
56   //spawn the clouds
57   spawnClouds();
58
59   drawSprites();
60 }
61
62 function spawnClouds() {
63   cloud = createSprite(600,100,40,10);
64   cloud.velocityX = -3
65
66   |
67 }
68
69

```

What do you see happening?

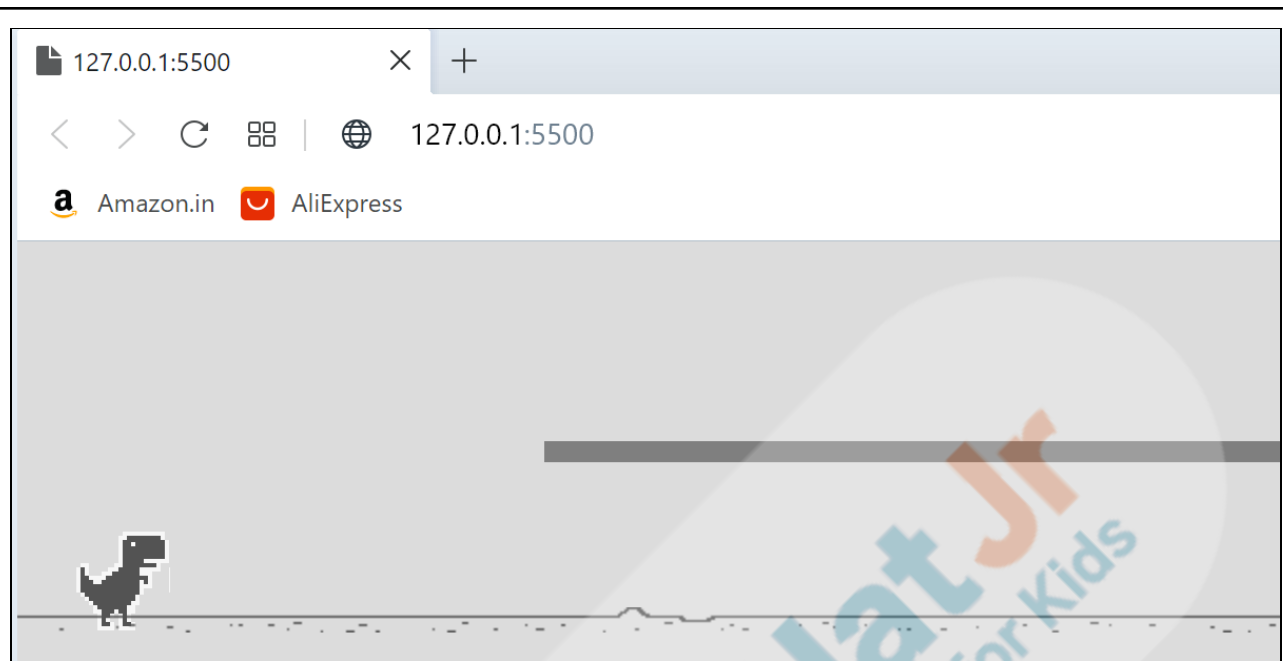
ESR:

A long line of many cloud sprites are appearing without any space.

ESR:

Varied

Can you guess why this is happening?



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. For each frame, **spawnClouds** are also called because it is inside the draw function.

This is why even though we created just one cloud sprite, we get a sequence of cloud sprites all joined together.

Every time **spawnCloud** is called, a new cloud is made. A new cloud is being made for every frame in our game.

Student listens and asks questions to understand.

Games run from one frame to another frame. You can imagine a frame as an image.

Each image has the state of the game. For example, in our Trex game, the ground is moving and clouds are moving; they look like they are moving because our computer is

The student writes the code, runs it and observes the output in the console.

| | |
|---|---|
| <p>creating one image after another and the position of the game character is updated in the image.</p> <p>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.</p> <p>You can actually print the frameCount of the game in the console by displaying in console.log(frameCount).</p> <p><i>The teacher guides the student to write the code for displaying the frameCount of the game.</i></p> <p>What do you see?</p> <p>These are the different frames in our game.</p> | <p>ESR: A sequence of numbers printed on the screen.</p> |
| <p>Add the following line to any line in the code inside function draw() :</p> <p>console.log(frameCount);</p> | |
| <p>Do we want a cloud in every frame?</p> | <p>ESR: No.</p> |
| <p>Let's say we want a cloud only after every 60 frames. How do we instruct the computer to create a cloud after every 60 frames?</p> | <p>ESR: Varied.</p> |
| <p>Let's see.</p> <p>We want a cloud when frameCount is 60, then when the frameCount is 120, then when the frameCount is 180, then again when the frameCount is 240and so on.</p> <p>All these numbers are multiples of?</p> | <p>ESR: 60.</p> |

| | |
|---|---|
| <p>We can tell the computer to create a cloud only after 60 frames. How do we do this?</p> <p>Right!</p> <p><i>The teacher makes the student do this for a different frameCount.</i></p> <p>What will the computer do if the frameCount is 10? For now, you can assume we are doing the division in whole numbers only.</p> <p>What will the computer do if the frameCount is 30?</p> <p>What will the computer do if the frameCount is 60?</p> | <p>ESR: We can tell the computer to divide each frameCount by 60 and if the remainder is 0, create a cloud for that frame.</p> <p>ESR: The computer will divide 10 with 60. Remainder will be 10. So no clouds will be made.</p> <p>ESR: The computer will divide 30 with 60. Remainder will be 30. So no clouds will be made.</p> <p>ESR: The computer will divide 60 with 60. Remainder will be 0. So a cloud will be made.</p> |
| <p>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.</p> <p>Remember how we compared in if loops using the comparison ("===") sign?</p> <p>Also remember to indent your code.</p> | <p><i>The student writes the code to generate a cloud for every 60 frames with the help of the teacher.</i></p> |

JS sketch.js X

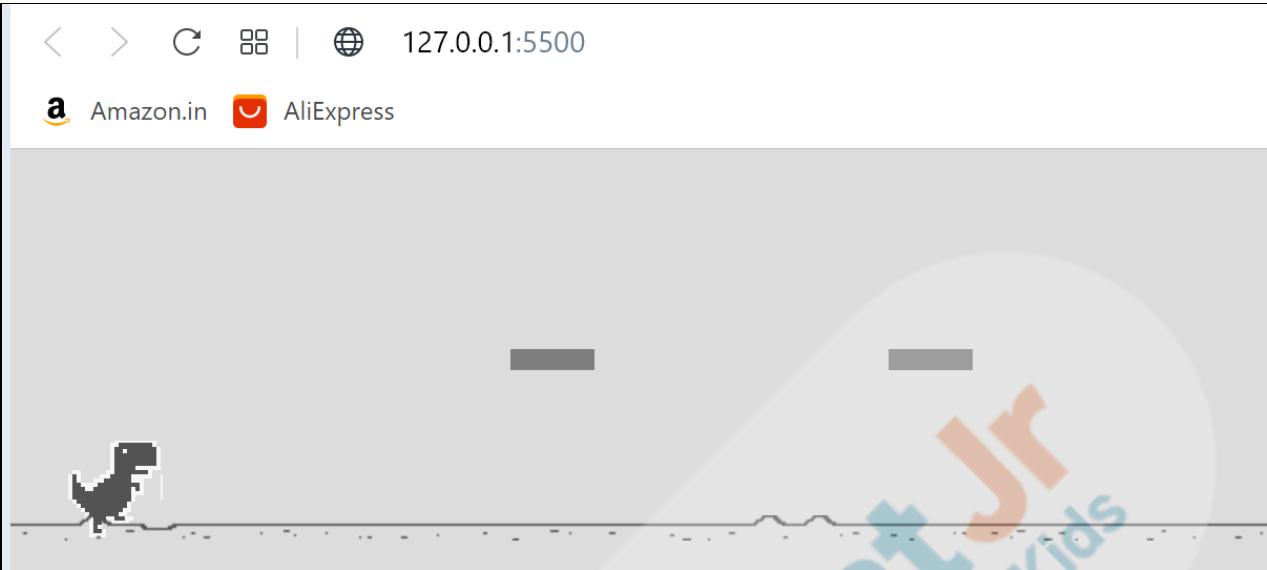
JS sketch.js > ...

```
57     if (ground.x < 0){
58         ground.x = ground.width/2;
59     }
60
61     //stop trex from falling down
62     trex.collide(invisibleGround);
63     spawnClouds();
64
65     drawSprites();
66
67 }
68
69 function spawnClouds() {
70     if (frameCount % 60 === 0) {
71         cloud = createSprite(600,100,40,10);
72         cloud.velocityX = -3;
73     }
74
75 }
76
77
```

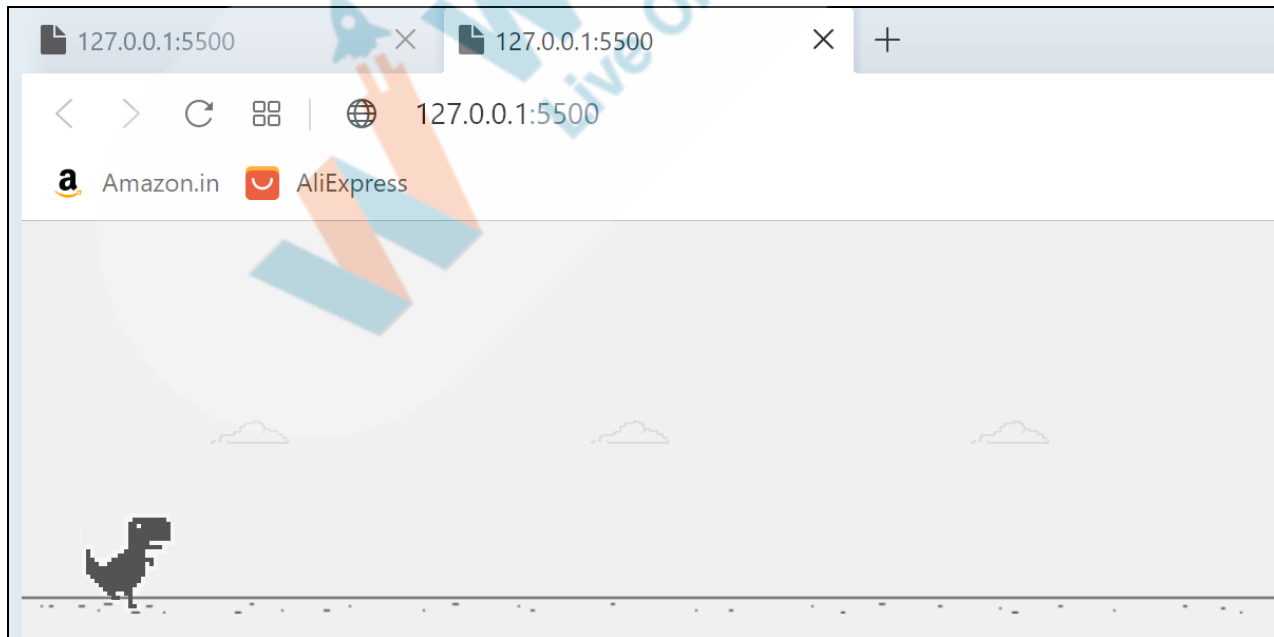
What do you see now?

ESR:

There is some space between the clouds because the clouds are generated after every 60 frames.

| | |
|---|---|
|  | |
| <p>Let's add the animation for the cloud.</p> <p>The image is already uploaded in the editor.</p> | <p><i>The student writes code to add animation for the cloud.</i></p> |

```
53
54   trex.collide(invisibleGround);
55
56   //spawn the clouds
57   spawnClouds();
58
59   drawSprites();
60 }
61
62 function spawnClouds() {
63   //write code here to spawn the clouds
64   if (frameCount % 60 === 0) {
65     cloud = createSprite(600,100,40,10);
66     cloud.addImage(cloudImage)
67     cloud.scale = 0.4;
68     cloud.velocityX = -3;
69
70
71   }
72 }
73
```



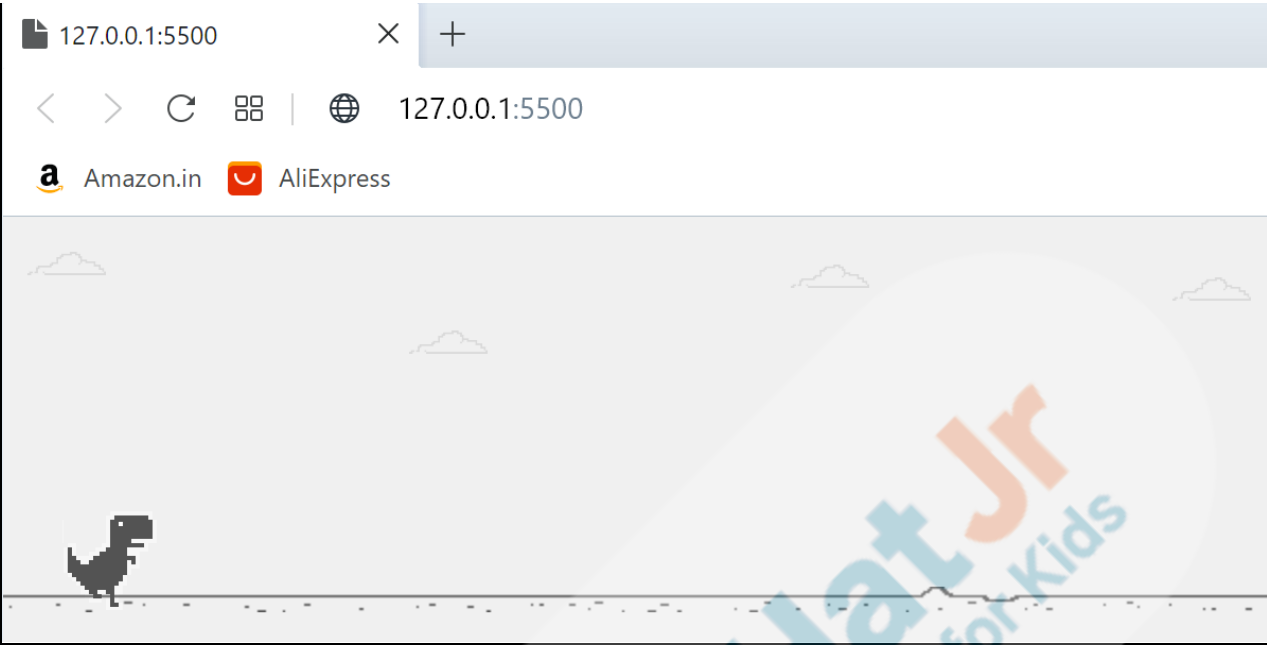
Our clouds all have the same height now. Let's make it more random. Let's change the y position of the cloud and give it a random number between 280 and 320.

The student writes the code with the guidance of the teacher.

The student runs the code and sees the output.

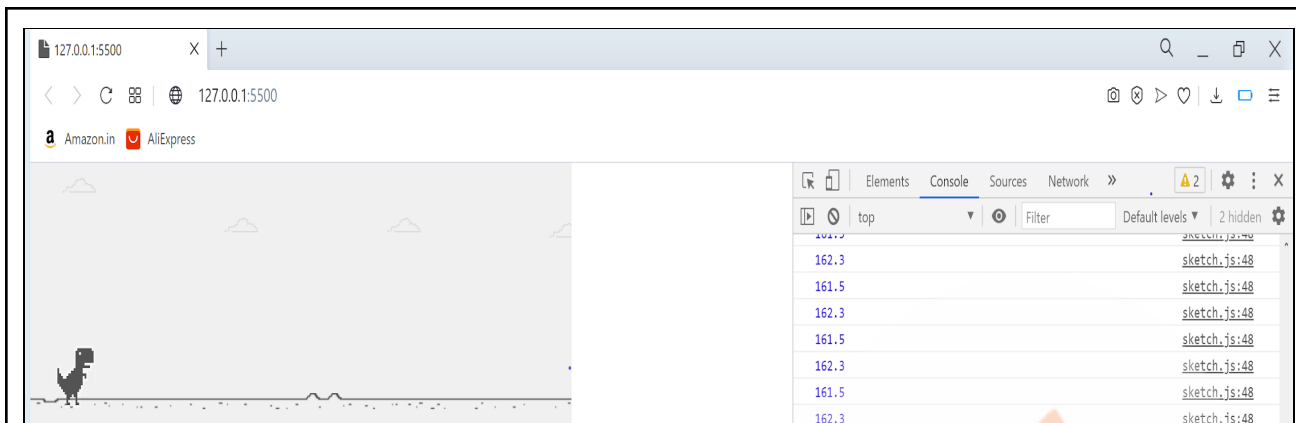


```
JS sketch.js X
JS sketch.js > spawnClouds
57   if (ground.x < 0){
58       ground.x = ground.width/2;
59   }
60
61   //stop trex from falling down
62   trex.collide(invisibleGround);
63   spawnClouds();
64
65   drawSprites();
66
67   }
68
69   function spawnClouds() {
70       if (frameCount % 60 === 0) {
71           cloud = createSprite(600,100,40,10);
72           cloud.addImage(cloudImage)
73           cloud.y = Math.round(random(10,60))
74           cloud.scale = 0.4;
75           cloud.velocityX = -3;
76       }
77   }
78
79
```

| | |
|--|----------------------------|
|  | |
| <p>One last thing that we need to fix!</p> <p>Can you observe it?</p> | <p>ESR: Varied.</p> |
| <p>When the dinosaur jumps, the cloud appears to be in front of the dinosaur.</p> <p>This is because in the game, the clouds are created after the Trex was created.</p> <p>Imagine each object being created on a layer above another object. So the Trex is created first and has the lowest depth. Clouds are created later and have higher depth.</p> <p>You can print the depth of any sprite using sprite.depth.</p> <p>Why don't you print the depth of the clouds and the Trex and see what they are?</p> | |

The student writes to print the depth of the Trex and cloud sprites.

```
JS sketch.js ×  
JS sketch.js > ...  
57     if (ground.x < 0){  
58         ground.x = ground.width/2;  
59     }  
60  
61     //stop trex from falling down  
62     trex.collide(invisibleGround);  
63     spawnClouds();  
64  
65     drawSprites();  
66  
67 }  
68  
69 function spawnClouds() {  
70     if (frameCount % 60 === 0) {  
71         cloud = createSprite(600,100,40,10);  
72         cloud.addImage(cloudImage)  
73         cloud.y = Math.round(random(10,60))  
74         cloud.scale = 0.4;  
75         cloud.velocityX = -3;  
76  
77         console.log(trex.depth);  
78         console.log(cloud.depth);  
79     }  
80  
81 }
```



Let us change the depth of the clouds to be the same as the T-rex and then increase the depth of the T-rex by 1.

This will ensure that the T-rex has a higher depth than the clouds.

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.

The student writes to adjust the depth of the T-rex and the clouds.


The student runs the code and checks the depth of the clouds and the T-rex.


JS sketch.js ×

JS sketch.js > ...

```
57     if (ground.x < 0){
58         ground.x = ground.width/2;
59     }
60
61     //stop trex from falling down
62     trex.collide(invisibleGround);
63     spawnClouds();
64
65     drawSprites();
66
67 }
68
69 function spawnClouds() {
70     if (frameCount % 60 === 0) {
71         cloud = createSprite(600,100,40,10);
72         cloud.addImage(cloudImage)
73         cloud.y = Math.round(random(10,60))
74         cloud.scale = 0.4;
75         cloud.velocityX = -3;
76
77         //adjust the depth
78         cloud.depth = trex.depth
79         trex.depth = trex.depth + 1;
80     }
81
82 }
```

Teacher Guides Student to Stop Screen Share**WRAP UP SESSION - 5 Mins**

| <div>  Teacher starts slideshow Slide 21-32 </div> | |
|---|--|
| Activity details | Solution/Guidelines |
| <p>Run the presentation from slide 21 to slide 32.</p> <p>Following are the warm up session deliverables:</p> <ul style="list-style-type: none"> • Explain the facts and trivias. • Next class challenge. • Project for the day. • Additional Activity. | <p>Guide the student to develop the project and share with us.</p> |
| Quiz time - Click on in-class quiz | |
| Question | Answer |
| <p>What is traversing through an array?</p> <p>A. Creating an array and displaying it</p> <p>B. Accessing elements of the array through index.</p> <p>C. Adding an element to the array</p> <p>D. Every time when we need to delete each element of the array by using the for loop</p> | B |
| <p>What does the following piece of code do?</p> <pre>function spawnClouds() { if (frameCount % 60 === 0) { cloud = createSprite(600,100,40,10); cloud.velocityX = -3; } }</pre> <p>A. 60 Sprites are being created after every frame</p> | B |

| | |
|---|---|
| <p>B. Cloud sprites are being created after every 60 frames</p> <p>C. Cloud sprites are being created after every frame</p> <p>D. None of the above</p> | |
| <p>Which of the following is true about the depth of a sprite?</p> <p>A. Every sprite is given a depth</p> <p>B. Depth of a sprite depends on when the sprite is created in the game.</p> <p>C. Sprite created first will have lowest depth and sprite created at the last will have the highest depth</p> <p>D. All of the above</p> | D |
| End the quiz panel | |
| <p><u>FEEDBACK</u></p> <ul style="list-style-type: none"> • Thoughts about a random number in a program. • Appreciate the student for their efforts in the class. • Review the content of the class. | |
| <p>We have come a long way since we started making this Trex runner game.</p> <p>How are you feeling?</p> | ESR: Varied. |
| <p>Does it feel that we are almost there?</p> | ESR: Yes. |
| <p>You get Hats Off for your excellent work!</p> | <p><i>Make sure you have given at least 2 Hats Off during the class for:</i></p> <div>  </div> |

| | |
|---|--|
| <p>There is one problem with our current code.</p> <p>We are creating a variable to store the cloud sprite after every 60 frames but when the clouds move out of the screen, the cloud sprite continues to be stored in these variables. And still we are creating more variables every 60 frames. What do you think will happen if this keeps happening for a long time?</p> | <div data-bbox="1031 241 1295 331">Great Question +10</div> <div data-bbox="1031 352 1295 443">Strong Concentration +10</div> <p>ESR: The game might slow down and then crash.</p> |
| <p>Yes! This is called a memory leak in the program.</p> <p>Have you seen applications which slow down your phone or computer?</p> <p>This happens mostly because the programs have memory leaks. We will fix this memory leak in our program in the next class.</p> | <p>ESR: Varied.</p> |
| <p>* This Project will take only 30 mins to complete. Motivate students to try and finish it immediately after the class.</p> <p>Project Overview FEED THE RABBIT</p> <p>Goal of the Project: Today, we have learned to use random numbers to generate clouds at random positions. We also learned about frame count to introduce a delay in the game.</p> <p>In this project, you have to spawn the first sprite i.e apples at random positions and add code for moving the rabbit sprite using a mouse.</p> | <p>Note: You can assign the project to the student in class itself by clicking on the Assign Project button which is available under the projects tab.</p> <p><i>Students engage with the teacher over the project.</i></p> |

Story:

Shailey loves rabbits and she wants to create a game in which a rabbit can eat different leaves and carrots. We have already helped Shailey in creating the design of the game with a moving background and a rabbit. Now she wants to make the game a bit challenging. She wants to spawn apples at random positions.

Are you up for the challenge?

I am very excited to see your project solution and I know you will do really well.

Bye Bye!

Teacher Clicks

✕ End Class

Additional Activities

Encourage the student to write reflection notes in their reflection journal using markdown.

Use these as guiding questions:

- What happened today?
 - Describe what happened.
 - The code I wrote.
- How did I feel after the class?
- What have I learned about programming and developing games?
- What aspects of the class helped me? What did I find difficult?

Student uses the markdown editor to write her/his reflection in a reflection journal.

Additional Activity 1

Teacher-led Activity- 10 Mins

In the last class, we worked with the arrays.

Here, we are going to do an interesting activity, which is very crucial for a lot of websites to work properly.

Every time you log in to your email, or your favorite social media website, behind the scenes this activity is happening in some form.

That activity is 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, like when you log on to the social media website, during login it checks whether your username and password is present in their database or not.

But how do we do it? First of all:

- 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.

The teacher downloads the code from [Teacher Activity 1](#) and opens in the VS code editor.

```

1  var arr = [1,2,3,4,5,6,7,8,9,10];
2
3  function find_num(arr,x)
4  {
5
6
7  }
```

The teacher writes the code.

| | |
|---|--|
| <p>Here we first created an array of 10 elements from 1 to 10, then we defined a function find_num().</p> <p>In the function brackets we have 2 things, arr and x, these are the arguments for this function, which means when we call this function, we need to give these two things, one of them is the array in which we are looking for a number and the second argument is the number itself.</p> | |
| <p>But how do you find a number in the array?</p> <p>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.</p> <p>So how do you find which room your friend is hiding?</p> <p>Great!</p> <p>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.</p> <p>If the number matches with that element, we will return the index of the number else we will return -1.</p> | <p>ESR: By checking each room one by one.</p> |
| <p>In the code we are using return statements two times. When we find the number, we are returning the index of the number and if we don't find we are returning -1.</p> <p>Return statement means what will be the output of this function when it will run. So when this function runs we can</p> | |

have 2 outputs—either the index of the element or -1. So now based on this knowledge we are writing a condition that gives us the result.

```
function setup()
{
  createCanvas(400, 400);
  var result = find_num(arr,20);
  if(result==-1)
  {
    console.log("number not found!");
  }
  else{
    console.log("number found at index :",result);
  }
}
```

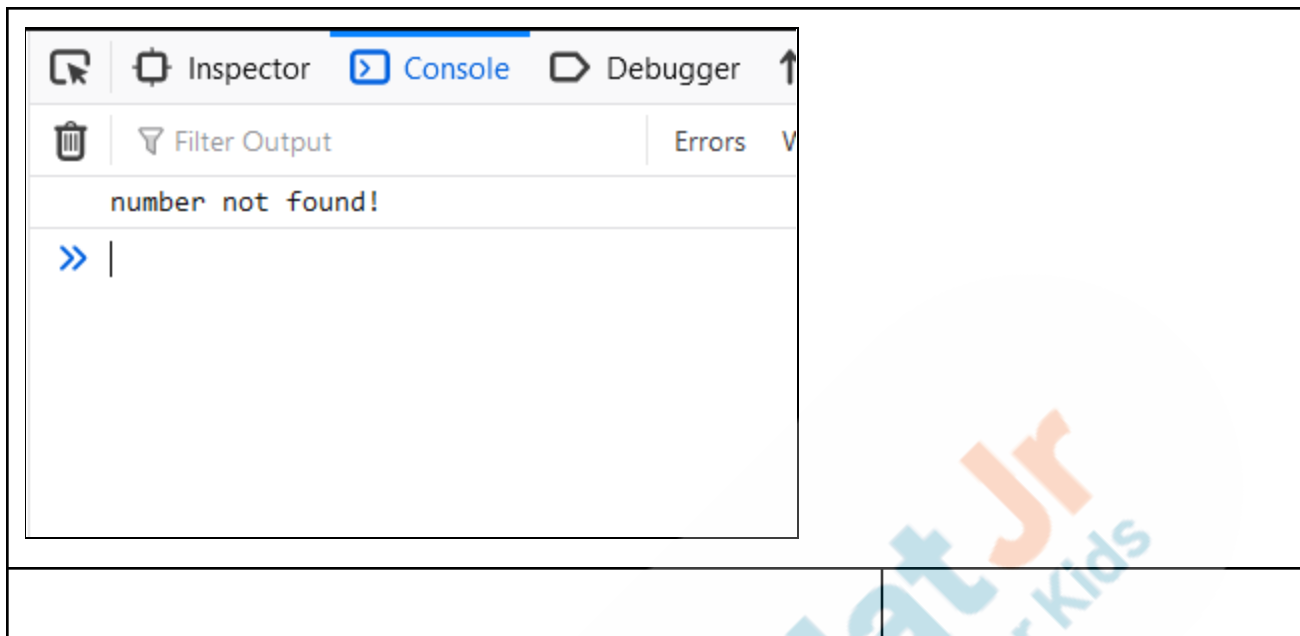
In the setup function first define a variable named **result** then set it equal to the function **find_num()** (when this function will be executed, the result will be stored in the **result variable**) and we also need to provide the array name and a number inside the function call. In this case the array name is arr and the number is 20 which is clearly not present inside the array.

The condition is if function returns -1. **This return value will be stored in the result variable.**

We can say that the number is not present in the array, else we will say the number is present and we will display the index of the element which is stored in the result variable.

Click on the live server button and bottom left corner to run the code. The browser window will open, then press F12 to open the console window.

```
1  var arr =[1,2,3,4,5,6,7,8,9,10];
2
3  function find_num(arr,x)
4  {
5      for(var i = 0; i<arr.length; i++)
6      {
7          if (arr[i]==x)
8          {
9              return i;
10             }
11         }
12     return -1;
13 }
14
15 function setup()
16 {
17     createCanvas(400, 400);
18     var result = find_num(arr,20);
19     if(result== -1)
20     {
21         console.log("number not found!");
22     }
23     else{
24         console.log("number found at index :",result);
25     }
26 }
```

| Activity | Activity Name | Links |
|-----------------------------|--------------------------------------|---|
| Teacher Activity 1 | Trex stage 2 | https://github.com/pro-whitehatjr/C-12_Trex_stage_2 |
| Teacher Activity 1 solution | Reference indented Code | https://github.com/pro-whitehatjr/C12_TA-3_reference_code |
| Teacher Activity 4 | Math functions | https://p5js.org/reference/#/p5/random |
| Teacher reference | Reference code of student Activity 1 | https://github.com/pro-whitehatjr/pro-c12-sa-reference-code |
| Student Activity 1 | Template Code | https://github.com/pro-whitehatjr/pro-c12-sa1-template |

| | | |
|-----------------------------------|---------------------|---|
| | | |
| Student Activity 2 | Trex stage 2.5 | https://github.com/pro-whitehatjr/C-12_Trex_2.5 |
| Student Activity 3 | Math Functions | https://p5js.org/reference/#/p5/random |
| Additional Activity 1 | Array Linear Search | https://github.com/pro-whitehatjr/C-12_additional_activity_1 |
| Additional Activity 1 solution | Reference link | https://github.com/pro-whitehatjr/c12_AA1_solution |
| Teacher Reference visual aid link | Visual aid link | https://curriculum.whitehatjr.com/Visual+Project+Asset/PRO_VD/BJFC-PRO-V3-C12-withcues.html |
| Teacher Reference In-class quiz | In-class quiz | https://s3-whjr-curriculum-uploads.whjr.online/a7638c74-9a14-4896-8b8e-7d880809cf90.pdf |
| Project Solution | Feed The Rabbit | https://github.com/pro-whitehatjr/Project_C12_Feed_The_Rabbit |