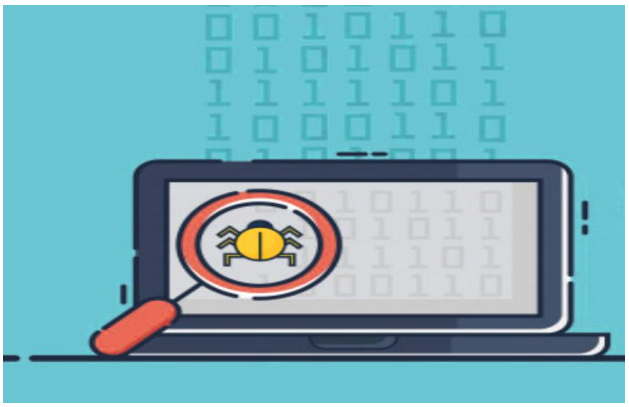
INFINITE GAME WORLD



What is our GOAL for this MODULE?

We created a Trex Game similar to what we see in the Google Chrome browser when it is not connected to the internet.

What did we ACHIEVE in the class TODAY?

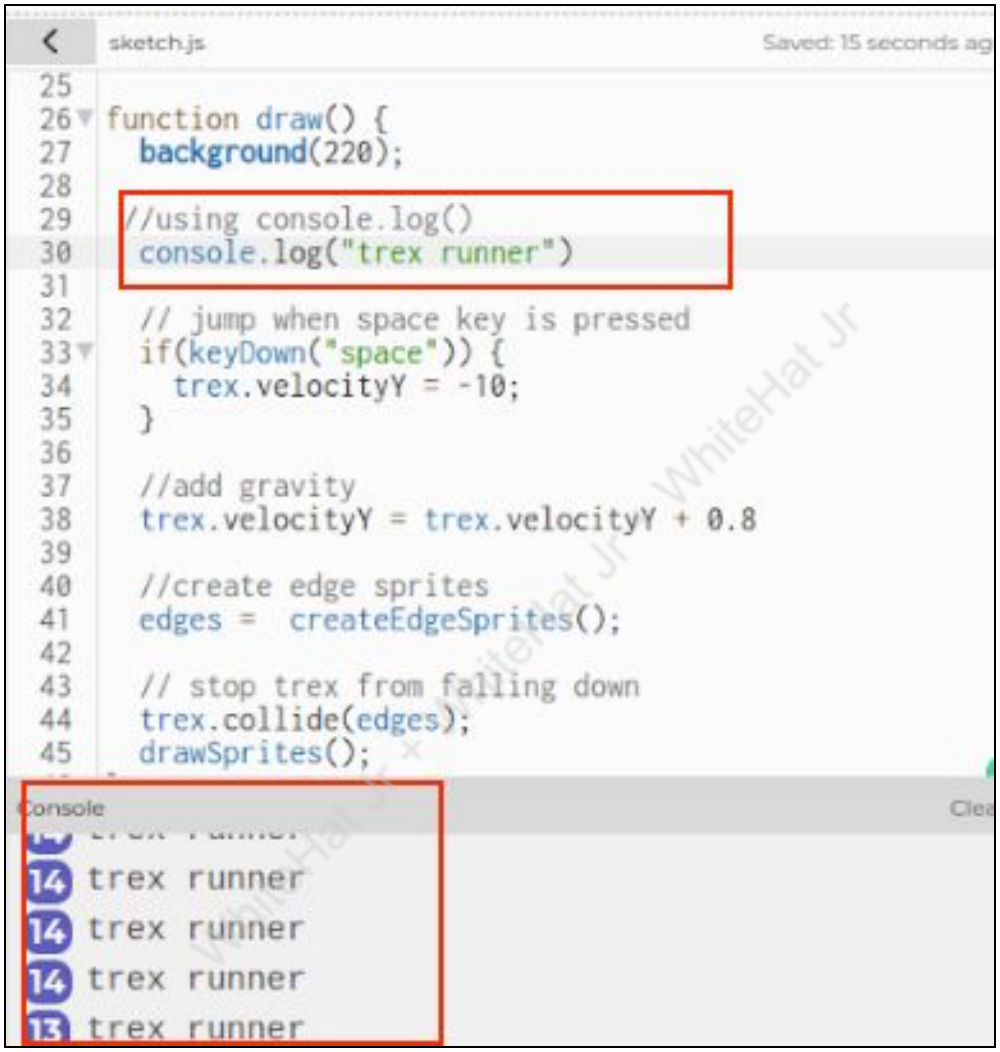
- Learned to log messages/outputs from the program into the console for testing purposes.
- Learned to create an infinitely scrolling ground for the dinosaur to run on.
- Learned to identify an additional condition needed in the program to stop the Trex from jumping again while it is in the air.
- Created an invisible ground sprite to make the Trex run below the ground.

Which CONCEPTS/ CODING BLOCKS did we cover today?

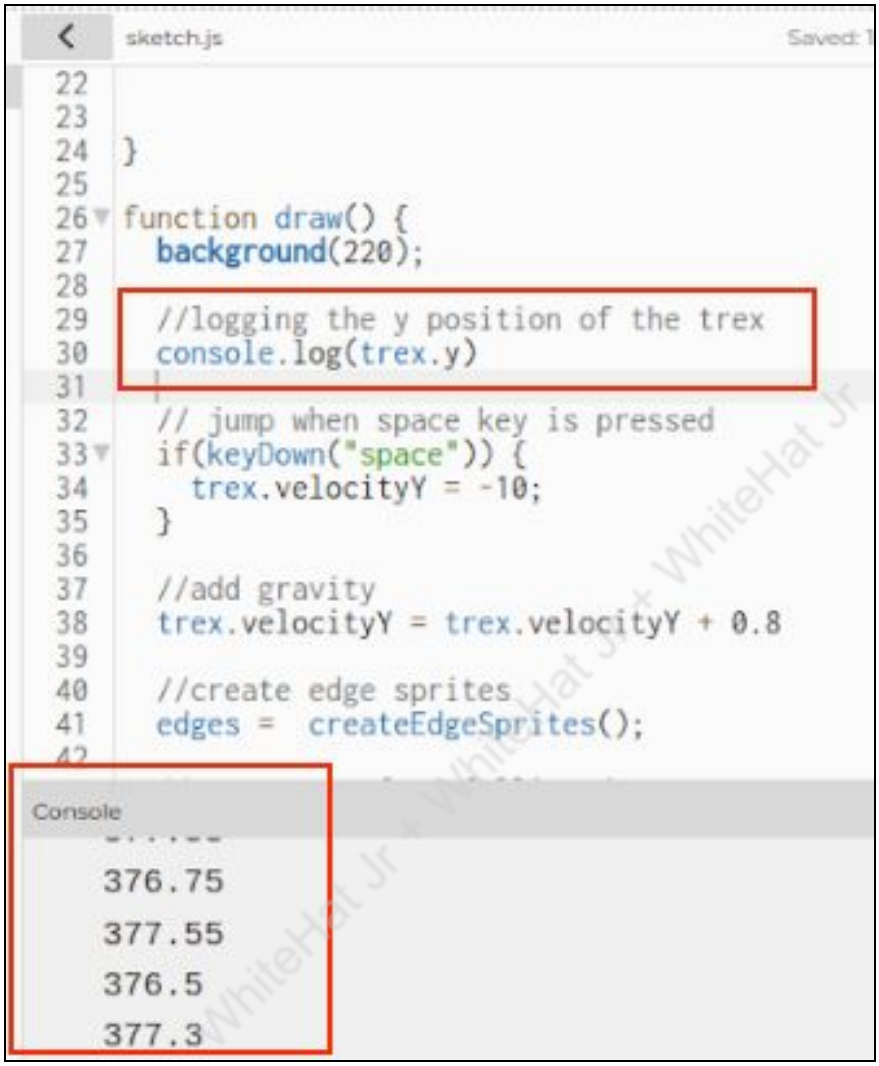
- Logging messages on the console using console.log() to test programs.
- Using logic to create a ground that gives the perception of scrolling infinitely.

How did we DO the activities?

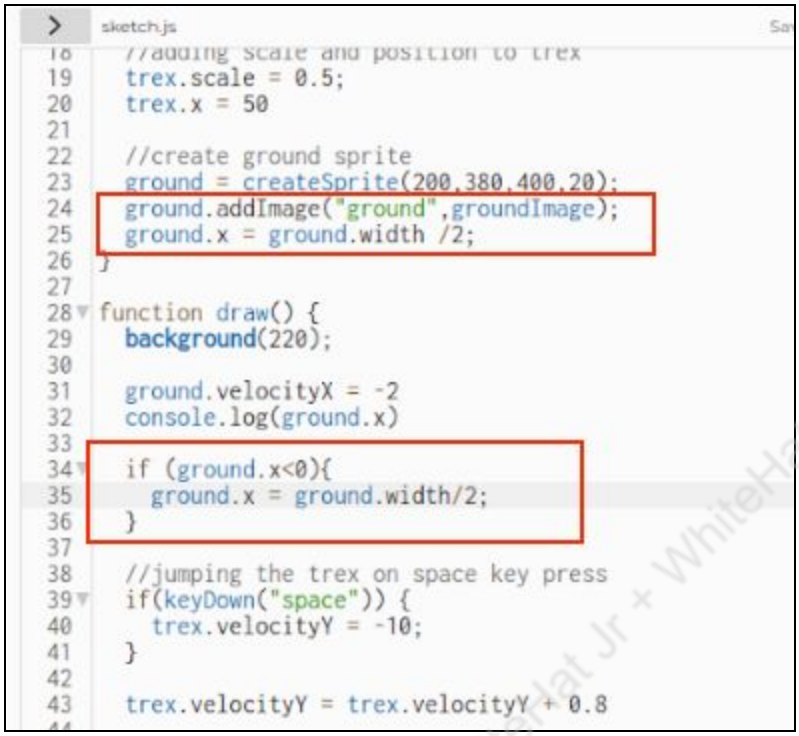
1. Learn to use **console.log()**. The P5 editor has a console window where we can log any message while the program is running. We do this using **console.log()** instruction.



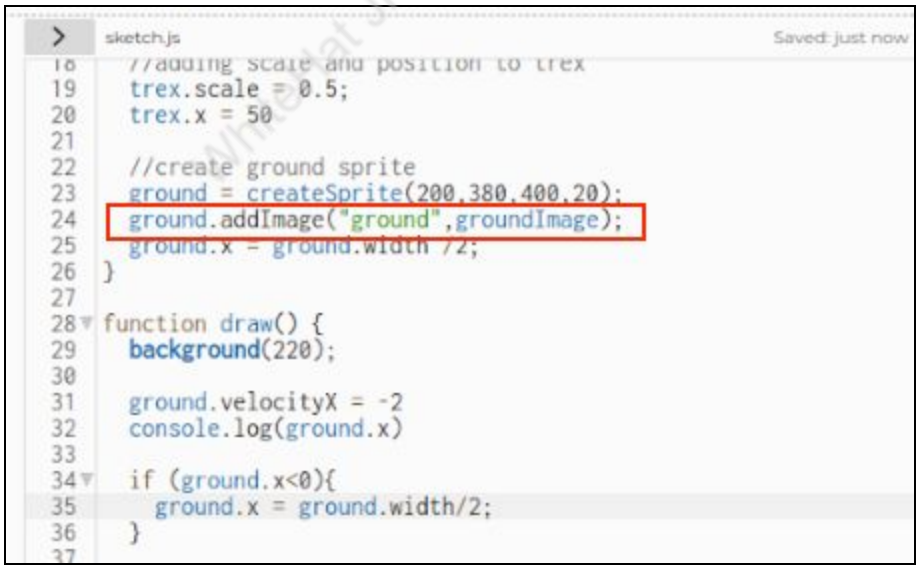
2. Write the **console.log()** instruction inside the **draw()** function. Try logging the y-position of the Trex sprite when it jumps.



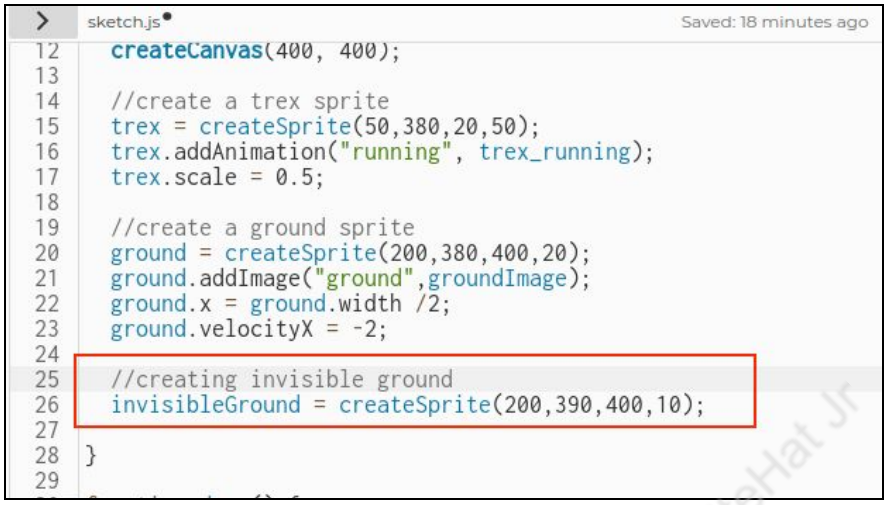
3. Move the dinosaur. Give a backward velocity to the ground; add the code to reset the ground.



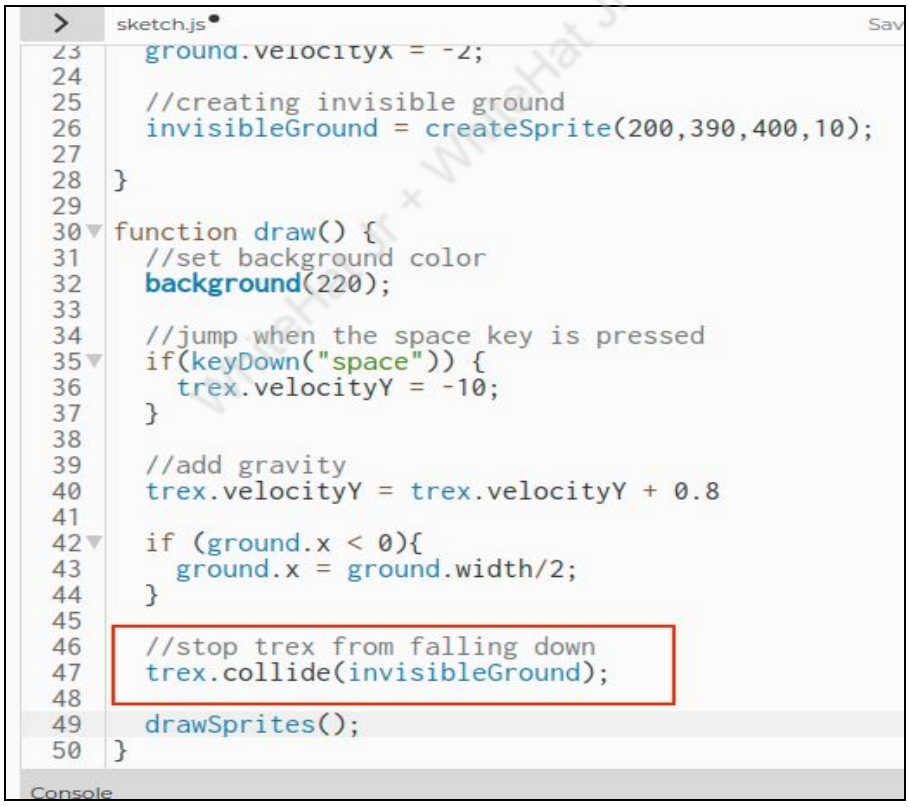
4. Use an actual ground image.



5. Create an invisible ground sprite just below the actual ground sprite since the dinosaur is running above the ground during bug fixes.



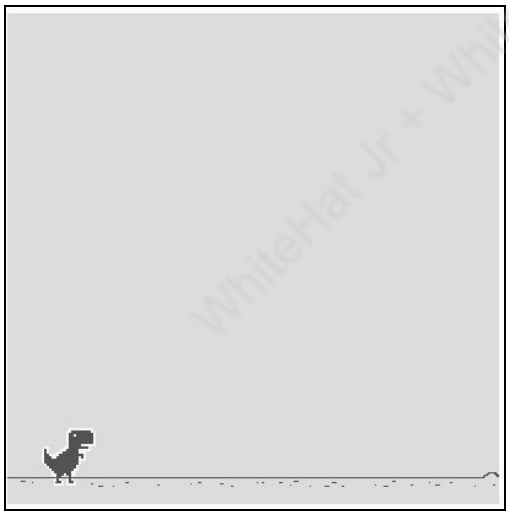
6. Collide the Trex with the invisible ground.



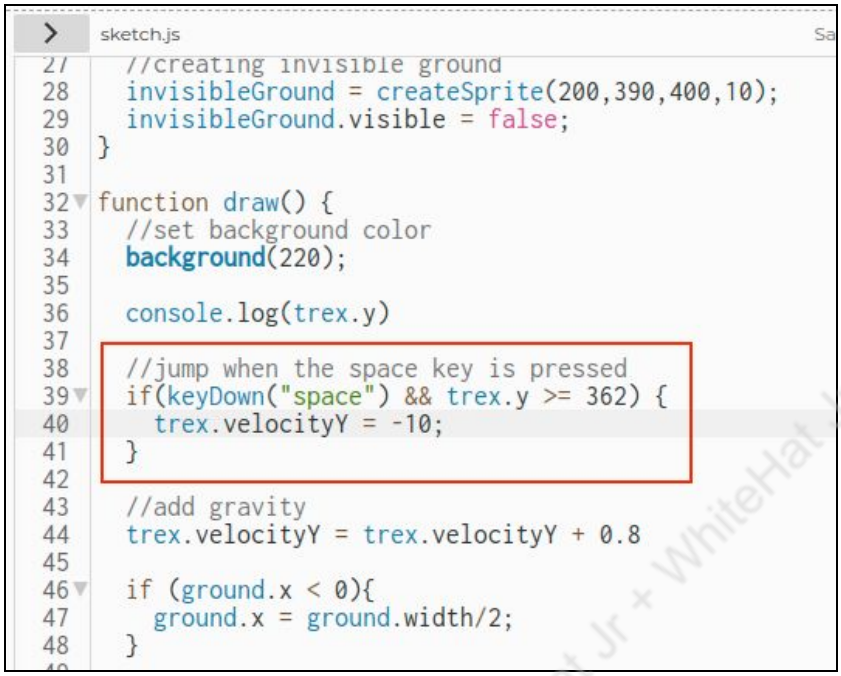
7. Add the following line of code anywhere outside the function draw() and after creating the invisible Ground Sprite: invisibleGround.visible = false;

```
29    //create a ground sprite
30    ground = createSprite(200,180,400,20);
31    ground.addImage("ground",groundImage);
32    ground.x = ground.width /2;
33    ground.velocityX = -4;
34
35    //creating invisible ground
36    invisibleGround = createSprite(200,190,400,10);
37    invisibleGround.visible = false;
38
39  }
40
41  function draw() {
42    //set background color
43    background(220);
44
```

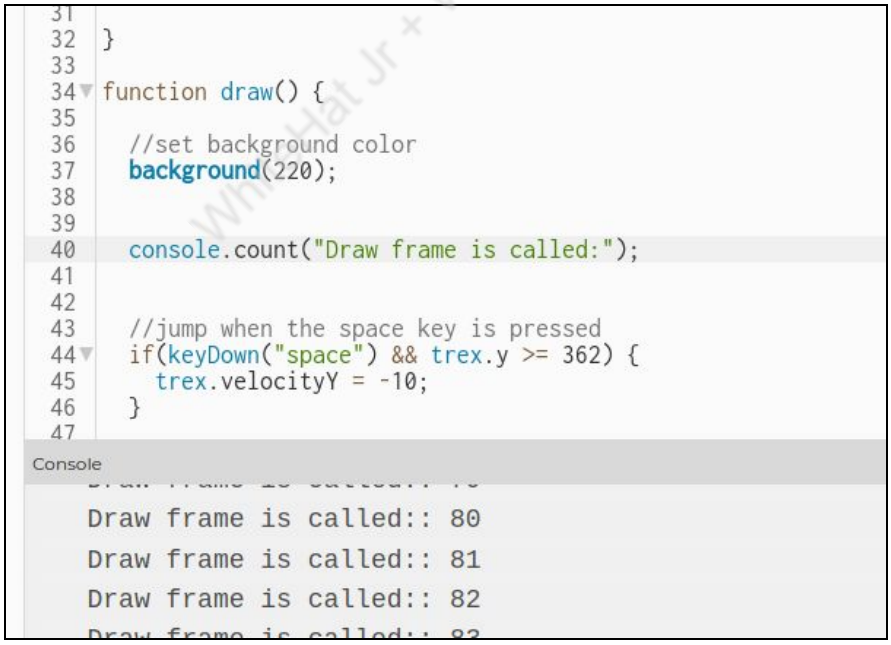
Output:



8. Add an additional condition inside the if block where we make the Trex jump only when it is on the ground while fixing the second bug.



9. Use **console.count()** to count how many times a particular program is called.



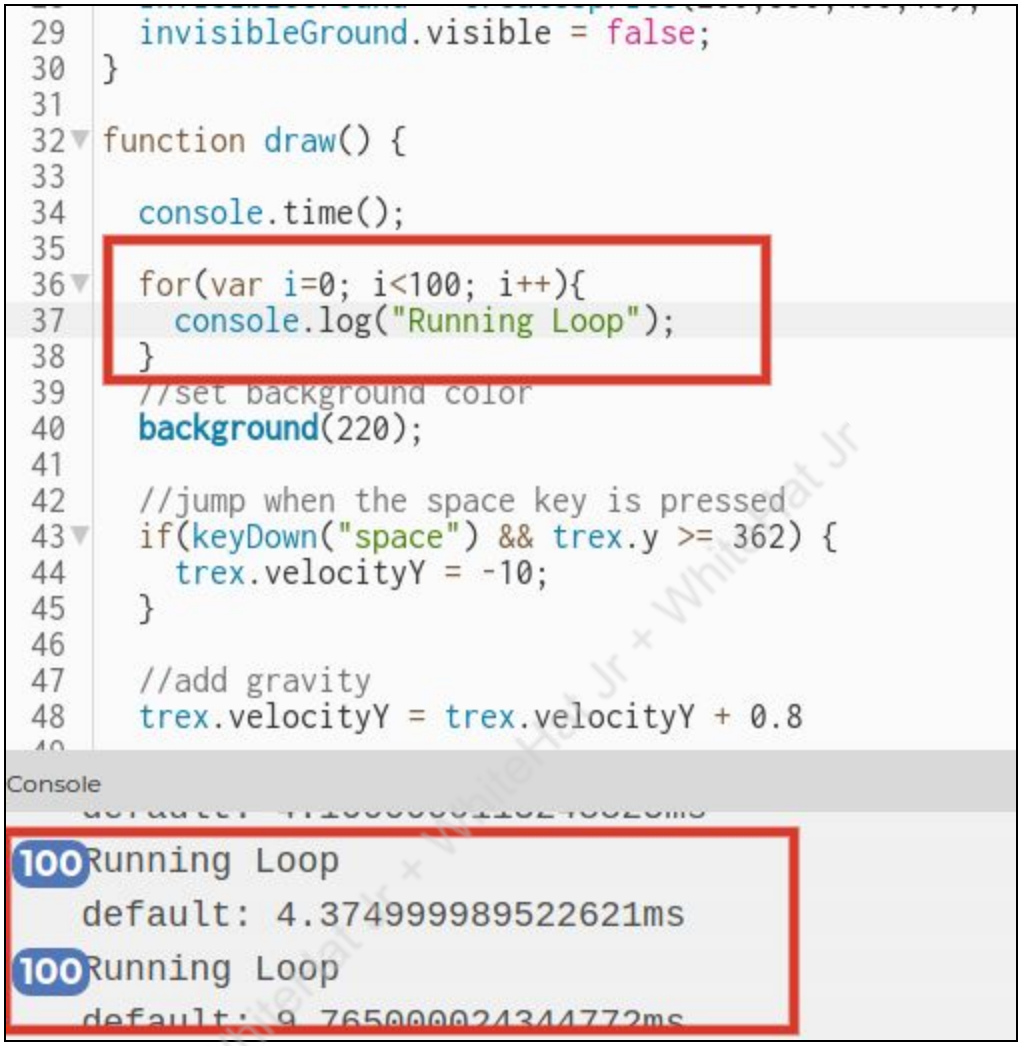
10. Use **console.time()** to keep a log of the time and **console.timeEnd()** to stop and print the time on the console.
    - **console.time()** when the **draw()** function starts:

```
sketch.js
32  }
33
34  function draw() {
35
36    console.time();
37
38    //set background color
39    background(220);
40
41    //jump when the space key is pressed
42    if(keyDown("space") && trex.y >= 362) {
43      trex.velocityY = -10;
44    }
45
46    //add gravity
47    trex.velocityY = trex.velocityY + 0.8
48
49    if (ground.x < 0){
```

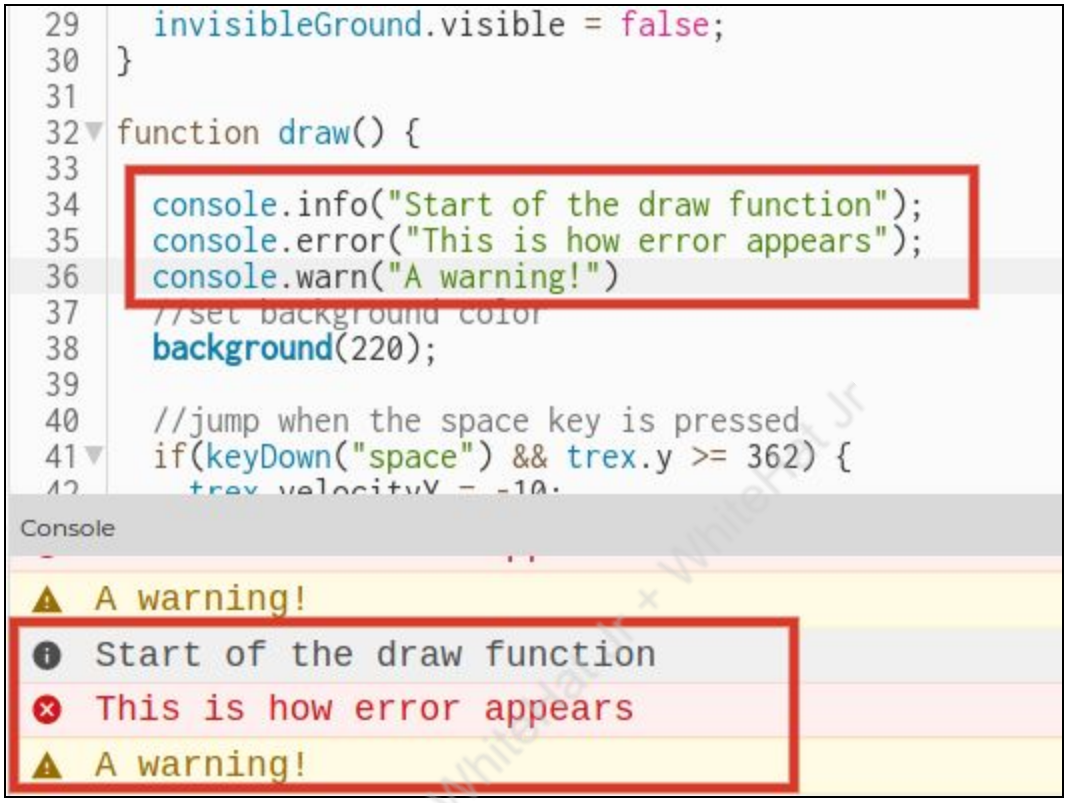
    - **console.timeEnd()** when the **draw()** function ends:

```
41    //jump when the space key is pressed
42    if(keyDown("space") && trex.y >= 362) {
43      trex.velocityY = -10;
44    }
45
46    //add gravity
47    trex.velocityY = trex.velocityY + 0.8
48
49    if (ground.x < 0){
50      ground.x = ground.width/2;
51    }
52
53    //stop trex from falling down
54    trex.collide(invisibleGround);
55
56    drawSprites();
57    console.timeEnd();
58
59  }
Console
default: 0.2800000074785203ms
default: 0.09499999578110874ms
default: 0.14999997802078724ms
default: 0.3549999964889139ms
```

11. Find the time taken by functions **setup()** or **preload()** to run before the game starts.



12. **Use console.log()** to print a simple message; **console.warn()** to print a warning; **console.error()** to print errors; and **console.info()** to print any information.



What's next?

We will start creating floating clouds on different heights.

Extend Your Knowledge:

1. Console in Javascript: Read more about the console.