



-0

Topic	Code Debugging and Code Indentation	
Class Description	Students learn to indent code to make it more readable. Students use the method of logging on the console to debug the program.	
Class	PRO-C10	
Class time	45 mins	
Goal	<ul style="list-style-type: none"> • Indent the code correctly to make it more readable. • Identify an additional condition needed in the program to stop the trex from jumping again while it is in the air. • Create an invisible ground sprite to make the trex run below the ground. 	
Resources Required	<ul style="list-style-type: none"> • Teacher Resources <ul style="list-style-type: none"> ○ P5 editor login ○ Laptop with internet connectivity ○ Earphones with mic ○ Notebook and pen • Student Resources <ul style="list-style-type: none"> ○ P5 editor login ○ Laptop with internet connectivity ○ Earphones with mic ○ Notebook and pen 	
Class structure	Warm Up Teacher-led Activity Student-led Activity Wrap up	5 mins 10 mins 20 mins 5 mins
CONTEXT		
• Code Readability		
Class Steps	Teacher Action	Student Action
Step 1: Warm Up (5 mins)	"How was your day today?"	ESR: varied

<p>Optional (If student has invited his/her friend to the class)</p>	<p>Did you get your friend along for this class? How is your friend doing?</p> <p>“Hey, what’s your name?”</p> <p>Hi <Friend’s name>. Welcome to this class. I will quickly tell you why I asked <student name> to call you in the class today.</p> <p><Student name> has been learning to code with us since the last 9 classes. He/she has been coding to create an endless runner type of game where the character keeps running and keeps overcoming several obstacles on the way. In these classes he/she, in fact, is creating an endless runner game similar to trex-runner - the game which you see on a chrome browser when you are not connected to the internet.</p> <p>After the end of this class, <student name> will embark on a project of his/her own where he/she will be creating his/her own endless runner game - that is - they can choose their character, their game environment, the type of obstacles, they have to overcome etc.</p> <p>You can help <student name> in two ways: You can help him/her brainstorm on the ideas for the endless runner</p>	<p>ESR: <Friend name></p>
---	---	--

	<p>game. The game ideas become more clear and better when you have someone to talk to.</p> <p>Second and most importantly, you can hold <student name> accountable for creating this game. You see coding is a skill where one has to stick it out!</p> <p>Creating this game is going to span over several days and there are going to be a number of problems on the way. The longer one works on a problem, the higher is their chances of learning and being able to build what they want.</p> <p>Students in our course who have held themselves accountable to their friends for making a game have ended up creating beautiful games - because they were committed to making it.</p> <p>Additionally, you can also test out the game built by your friend later and provide him/her valuable feedback.</p> <p>How are you feeling about your role?</p> <p>Great...we will reach the project in a while. Till then you can either wait and be an observer in the class or participate if you want!</p>	<p>ESR: excited/responsible</p>
--	--	--

	<p>"Could you work on correcting the two problems in the game that we discussed in the last class?"</p> <p>"Let's work on it together today."</p>	<p>ESR: varied</p>
	<p>I have an exciting quiz question for you! Are you ready to answer this question?</p>  <p>Teacher click on the button on the bottom right corner of your screen to start the In-Class Quiz.</p> <p>A quiz will be visible to both you and the student.</p> <p>Encourage the student to answer the quiz question.</p> <p>The student may choose the wrong option, help the student to think correctly about the question and then answer again.</p> <p>After the student selects the correct option, the  button will start appearing on your screen.</p> <p>Click the End quiz to close the quiz pop-up and continue the class.</p>	<p>ESR: Yes</p>
	<p>"Before we can do that we will look a little into making codes more readable by giving proper spaces inside the code. We call these spaces CODE INDENTATION"</p>	<p>-</p>

Teacher Initiates Screen Share		
<p style="text-align: center;"><u>CHALLENGE</u></p> <ul style="list-style-type: none"> Identify and correct the two problems in the current code: <ul style="list-style-type: none"> - the Trex jumps midway in the air when the space key is pressed. - the Trex is running above the ground. 		
Step 2: Teacher-led Activity (10 mins)	<p>Teacher opens <u>Teacher Activity 1.</u></p> <p>"This is the code from the last class. Do you see any problems with this code?"</p>	<p>ESR:</p> <p>There is no spacing between the lines of the code. They are all together and difficult to read.</p>
	<p>"Yes, computers don't mind or read spaces. But it is important to give spaces in your code to make them easily readable. Remember, you want other programmers to easily read your code. Giving proper spaces in your code makes your code easily readable. Let's try to add some space in your code."</p>	<p><i>The student listens and learns.</i></p>
	<p>"We try to leave a line after a meaningful block of code."</p> <p>Can you tell which code lines together make a meaningful block of code?</p> <p>Teacher leaves lines after each meaningful block of instruction.</p>	<p><i>The student observes, comments and learns.</i></p>

```

11▼ function setup() {
12   createCanvas(600, 200);
13   
14   //create a trex sprite
15   trex = createSprite(50,160,20,50);
16   trex.addAnimation("running", trex_running);
17   trex.scale = 0.5;
18   
19   //create a ground sprite
20   ground = createSprite(200,180,400,20);
21   ground.addImage("ground",groundImage);
22   ground.x = ground.width /2;
23   ground.velocityX = -4;
24
25 }
26
27▼ function draw() {
28   background(220);
29   
30   //jump when the space button is pressed
31▼ if (keyDown("space")) {
32     trex.velocityY = -10;
33   }
34   
35   trex.velocityY = trex.velocityY + 0.8
36   
37▼ if (ground.x < 0) {
38   ground.x = ground.width / 2;
39 }
40 
41 trex.collide(ground);
42 drawSprites();

```

There is no fixed rule to leaving these spaces. It is just like leaving spaces between paragraphs when you write a story. But do you see how easy it is to read and understand the code now?

ESR:
Yes

We also add some code indentation to lines to show that they are contained inside another block of code.

The student observes and learns.

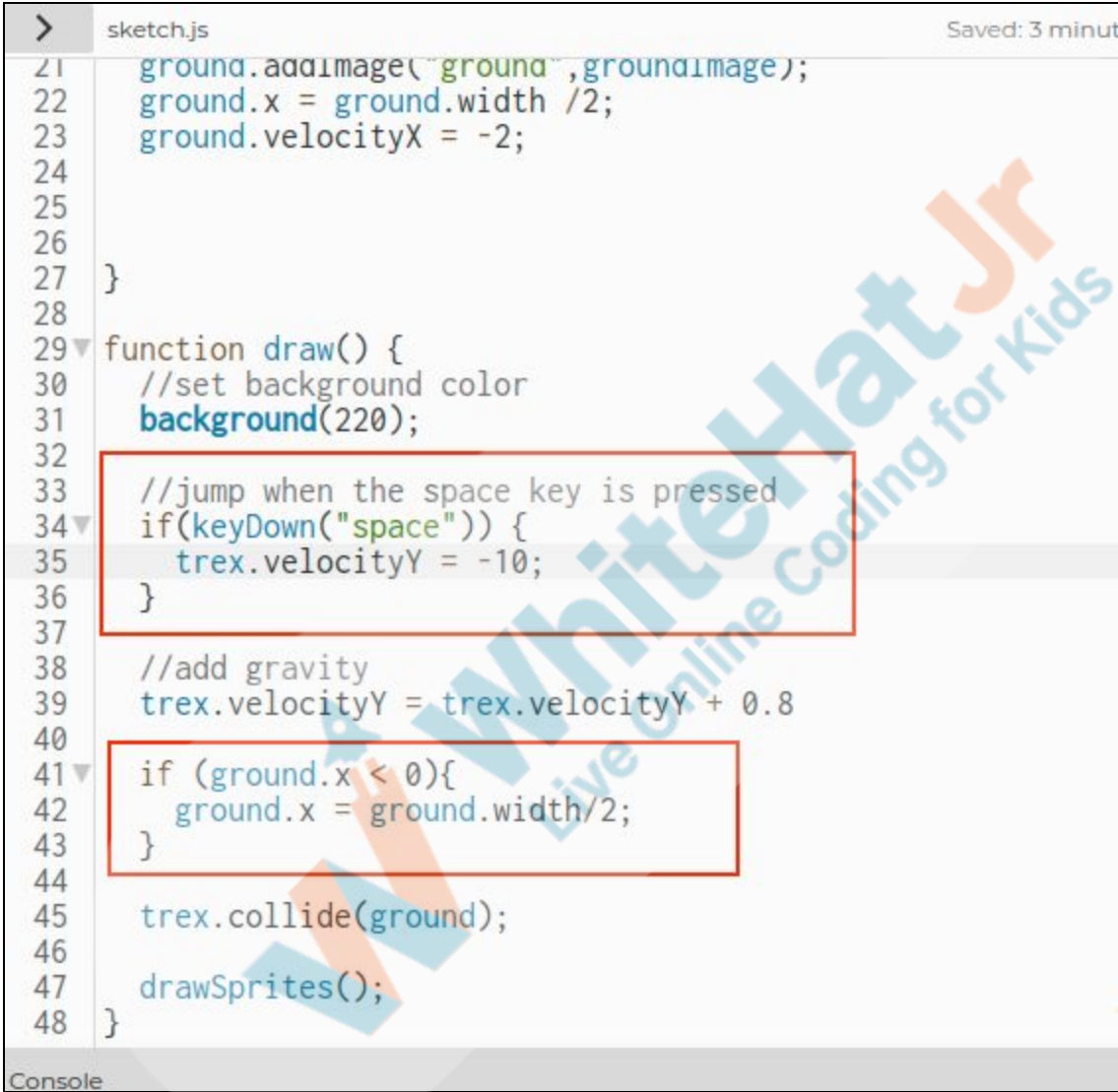
Let me show you how.
For example: lines 30 to 47 are contained inside the function draw(). We show it by indenting these lines by adding some space in front of these lines. This spacing should be consistent.

Teacher selects lines 30 to 47 and presses TAB.

```

> sketch.js Saved:
21 ground.addImage( ground,groundImage);
22 ground.x = ground.width /2;
23 ground.velocityX = -2;
24
25
26
27 }
28
29 function draw() {
30 //set background color
31 background(220);
32
33 //jump when the space key is pressed
34 if(keyDown("space")) {
35     trex.velocityY = -10;
36 }
37
38 //add gravity
39 trex.velocityY = trex.velocityY + 0.8
40
41 if (ground.x < 0){
42     ground.x = ground.width/2;
43 }
44
45 trex.collide(ground);
46
47 drawSprites();
48 }

```

	<p>This is called code indentation. Similarly, the lines of code inside if blocks or for blocks need to be indented as well.</p>	<p>The student observes and learns.</p>
 <pre> 21 ground.addImage(ground,groundImage); 22 ground.x = ground.width /2; 23 ground.velocityX = -2; 24 25 26 27 } 28 29 function draw() { 30 //set background color 31 background(220); 32 33 //jump when the space key is pressed 34 if(keyDown("space")) { 35 trex.velocityY = -10; 36 } 37 38 //add gravity 39 trex.velocityY = trex.velocityY + 0.8 40 41 if (ground.x < 0){ 42 ground.x = ground.width/2; 43 } 44 45 trex.collide(ground); 46 47 drawSprites(); 48 } </pre>		
	<p>Code indentation helps us understand the program structure easily. It also makes us less likely to make mistakes while typing out text - like missing out on closing curly brackets.</p>	<p>The student listens and learns.</p>

	<p>Ok now it's time to work on solving the two bugs we had in our program. Bugs are parts of the program that do not work as we want. We have two such bugs that we will solve for today.</p> <ul style="list-style-type: none"> - the dinosaur jumping in mid air when the space key is pressed. <p>What do we want instead?</p>	<p>ESR:</p> <p>We want the dinosaur to be able to jump only when it is touching the ground. It should be able to jump again only when it falls back on the ground.</p>
	<p>Great! Our second bug is that our dinosaur is running a little over the ground. What do we want instead?</p>	<p>ESR:</p> <p>We want it to run a little below the ground.</p>
	<p>Ok let's fix these.</p>	
Teacher Stops Screen Share		
	<p>Now it's your turn. Please share your screen with me.</p>	
<ul style="list-style-type: none"> • Ask Student to press ESC key to come back to panel • Guide Student to start Screen Share • Teacher gets into Fullscreen 		
<p align="center"><u>ACTIVITY</u></p> <ul style="list-style-type: none"> • Identify and add an additional condition so that the T rex jumps only when in contact with the ground. • Create an invisible ground sprite which is below the ground and on which the T rex dinosaur is supported. 		

<p>Step 3: Student-Led Activity (10 mins)</p>	<p>Ok. Quickly fire up your activity and indent your code.</p> <p><i>Teacher helps the student to properly indent their code.</i></p>	<p><i>Student opens Student Activity 1 and presses File> Duplicate.</i></p>
	<p>Let us first fix the second bug. The Trex right now is supported by the ground sprite. Collision with the ground sprite is not letting the Trex fall off the ground.</p> <p>Let us create an invisible ground sprite just below this ground. We want to do this so that rather than being supported by the ground and being above the ground- Trex gets supported by an invisible ground just below the actual ground.</p> <p>Can you create another ground sprite just below the first ground and make it cover the entire width of the screen.</p> <p>For fun's sake, let us call it invisibleGround.</p> <p><i>Guide the student to create an invisible ground Sprite.</i></p>	<p><i>Student creates an invisible ground Sprite.</i></p>

```

14
15 //create a trex sprite
16 trex = createSprite(50,160,20,50);
17 trex.addAnimation("running", trex_running);
18 trex.scale = 0.5;
19
20 //create a ground sprite
21 ground = createSprite(200,180,400,20);
22 ground.addImage("ground",groundImage);
23 ground.x = ground.width /2;
24 ground.velocityX = -4;
25
26 //creating invisible ground
27 invisibleGround = createSprite(200,190,400,10);
28 invisibleGround.visible = false;
29
30 }
31
32 function draw() {
33 //set background color
34 background(220);
35

```

Instead of supporting the trex on the ground, let us collide it with the invisible ground.

*Student modifies
trex.collide(ground) to
trex.collide(invisibleGround)*

```

31
32 ▼ function draw() {
33     //set background color
34     background(220);
35
36     console.log(trex.y)
37
38     //jump when the space key is pressed
39 ▼ if(keyDown("space")) {
40     trex.velocityY = -10;
41 }
42
43     //add gravity
44     trex.velocityY = trex.velocityY + 0.8
45
46 ▼ if (ground.x < 0){
47     ground.x = ground.width/2;
48 }
49
50     //stop trex from falling down
51     trex.collide(invisibleGround);
52
53     drawSprites();
54 }

```

Now for the magic! Let us make this ground sprite invisible. There is an instruction in the toolbox sprite.visible. You need to make it false to make the ground invisible.

By saying 'sprite.visible = false' , we are asking the computer to NOT make this sprite visible.

Student writes **invisibleGround.visible = false** to make the ground invisible.

Student runs the code to see the output.

Add the following line of code anywhere outside the function draw() and after creating the invisibleGround Sprite:

invisibleGround.visible = false;

We have the trex running on the ground now!!

Student amazed :)



Let us try to fix the other bug which is

ESR:
Make the dinosaur jump only when it is on the ground.

When does the dinosaur jump now?

ESR:
When the space key is pressed.

	We want to make it jump when the space key is pressed and when it is on the ground. Right?	ESR: Yes
	What do we need to do ?	ESR: Add an additional if condition?
	<p>Right. Let us try to log the current y position of the Trex when it is running on the ground.</p> <p><i>Guide the student to log trex.y on the console window.</i></p>	<p><i>Student writes code to log trex.y on the console window.</i></p>


```

32 function draw() {
33   //set background color
34   background(220);
35
36   console.log(trex.y)
37
38   //jump when the space key is pressed
39   if(keyDown["space"]) {
40     trex.velocityY = -10;
41   }
42
43   //add gravity
44   trex.velocityY = trex.velocityY + 0.8
45
46   if (ground.x < 0){
47     ground.x = ground.width/2;
48   }
49
50   //stop trex from falling down
51   trex.collide(invisibleGround);
52
53   drawSprites();
54 }

```

Console

```

161.5
162.3
161.5
162.3
161.5

```

What do you see?

ESR:

The y position of the trex changes between 362 and 362.8

	Right. and when it jumps what will happen to trex.y?	<p><i>Student runs the code and makes the trex jump to see the change in the trex.y in the console.</i></p> <p>ESR: trex.y reduces when the trex jumps.</p>
	So we want the trex to jump only when it is on the ground , that is, only when trex.y >= 100. How can we do that?	<p>ESR: By adding an additional condition inside the if block where we make the trex jump.</p>
	<p>Let's do that.</p> <p><i>Teacher guides the student to write the additional condition inside 'if block'</i></p>	<p><i>Student writes the code and runs the program.</i></p> <p><i>The student can press space repeatedly to see if the program works as expected.</i></p>

```

31
32 ▼ function draw() {
33   //set background color
34   background(220);
35
36   console.log(trex.y)
37
38   //jump when the space key is pressed
39 ▼ if(keyDown("space") && trex.y >= 100) {
40     trex.velocityY = -10;
41   }
42
43   //add gravity
44   trex.velocityY = trex.velocityY + 0.8
45
46 ▼ if (ground.x < 0){
47     ground.x = ground.width/2;
48   }
49
50   //stop trex from falling down
51   trex.collide(invisibleGround);
52
53   drawSprites();
54 }

```

Teacher Guides Student to Stop Screen Share

FEEDBACK

- Appreciate the student for their efforts in the class.
- Review the content of the class.
- Ask the student to make notes for the reflection journal along with the code they wrote in today's class.

Step 4: Wrap-Up (15 mins)	Doesn't it feel powerful to be able to fix these bugs. How are you feeling after today's class?	ESR: varied
	You can always find ways to do things exactly the way you want them. Programming in this respect is like	<i>Make sure you have given at least 2 Hats Off during the class for:</i>

	building something with the tools, instructions and resources available to you.	<div>Creatively Solved Activities +10</div> <div>Great Question +10</div> <div>Strong Concentration +10</div>
	<p>Now, as I said - your project for this class is about creating your own endless runner style of game. You can choose your own character, your own game environment, any obstacle the character has to overcome or anything the character has to collect !</p> <p>Brainstorm with your friend about the game you will be building.</p>	<i>Student and the friend exchange ideas to come up with the game.</i>
	<p>Alright. Great ideas for a game. Now that you have a solid idea for what you want in your endless runner game -your friend can hold you accountable such that you are continuously working on your game idea.</p> <p>Your friend can also test the game when it is ready!</p>	-
	<p>Next class we will look at how to create floating clouds at different heights.</p> <p>Thank you <Friend name> for joining us in this class. I hope to see you again sometime.</p>	-

	Looking forward to our next class.	
Project Overview	<p>BALLOON BUSTER - 2</p> <p>Goal of the Project: In this class we have learned how to indent code and use the console display live position of an object. Using visible properties of ground we made ground invisible.</p> <p>In this project, you have to create a function to make arrows for balloons and call it when the space key is pressed.</p> <p>Story: As you saw in the last project, Meera has created a bow and arrow game she saw in the town fair. You have been helping her build that game. So let's continue with it.</p> <p>You have created balloons and bows in the last project, And now you have to create a function for arrows.</p> <p>I am very excited to see your project solution and I know you will do really well.</p> <p>Bye Bye!</p>	<p><i>Students engage with the teacher over the project.</i></p>
<div> <div>Teacher Clicks</div> <div>✕ End Class</div> </div>		
Additional Activities	<p>There are several other ways the console can be used.</p> <p>For example:</p>	<p><i>Student practices using console.count() in their own code.</i></p>

We can use `console.count()` to count how many times a particular program is called.

Let's say, we want to know how many times the `draw()` function is called. We can use `console.count()` inside the `draw` function.

Teacher shows how to use `console.count()` inside `draw()`.

This is the same as counting the number of frames!

```

32 function draw() {
33   //set background color
34   background(220);
35
36   console.count("Draw frame is called:");
37
38   //jump when the space key is pressed
39   if(keyDown("space") && trex.y >= 100) {
40     trex.velocityY = -10;
41   }
42 }

```

Console

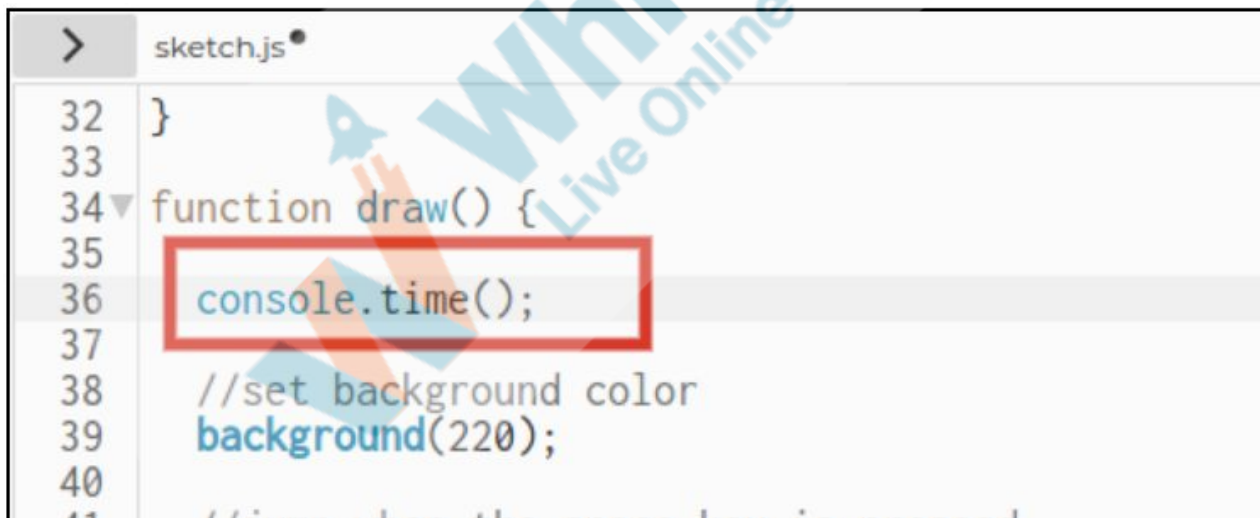
```

Draw frame is called:: 4
Draw frame is called:: 5
Draw frame is called:: 6
Draw frame is called:: 7
Draw frame is called:: 8
Draw frame is called:: 9

```


	<p>You can also use the console to find out how much time it takes your program to run.</p> <p>We use <code>console.time()</code> to start keeping log of the time and <code>console.timeEnd()</code> to stop and print the time on the console.</p> <p>This is used by programmers when they want to optimize and reduce the time taken by their program to run.</p> <p><i>Teacher shows how to use <code>console.time()</code> and <code>console.timeEnd()</code> to measure the time taken by the draw function to run.</i></p>	<p><i>The student learns how to use <code>console.time</code> and <code>console.timeEnd</code> to log the time taken by the program to run.</i></p>
--	--	---

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   //time when the space key is pressed
  
```

console.timeEnd() when the draw function ends

```

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

```

Similarly, you can also find out how long it takes for function setup or function preload to run before your game can start.

Note: Observe how it takes different time each time the draw function runs. The variation is because your computer's processing speed depends on a lot of factors like - how heated your computer chips are, what are the other things your computer is

Student writes a simple for-loop inside the function draw() and observes the change in execution time of the draw function.

The student observes the lag in the game - where every character slows down and gives an impression of the game hanging.

	<p>doing - for example what else is happening on your browser etc.</p> <p>Let us write a simple for loop inside function draw. Log anything inside the for loop and check if the execution time of the draw function changes. Also, observe the effect of this on your game.</p> <p><i>Ask the student to explain the reason behind the lag in the game.</i></p>	<p>ESR: Every frame in the game is rendered (drawn) each time the draw function gets called.</p> <p>The lag in the game is because it takes longer for the next frame to render.</p>
--	--	---

```

32 function draw() {
33   //set background color
34   background(220);
35
36   console.time();
37
38   for (var i=0; i<100; i++){
39     console.log("Running Loop")
40   }
41

```

Console

```

1406unning Loop
1494unning Loop
1400unning Loop
1500unning Loop
1420unning Loop

```

Good! Our goal should always be to write programs which run in the least time possible.

There are other ways you can use the console.

console.log() is used to print a simple message.

You can use console.warn() to print a warning. The warning message is formatted differently.

Student experiments with different types of console messages.

Similarly, you can use `console.error()` to print an error(). the error message is formatted differently.

You can also use `console.info()` to print any information.

Teacher shows how to print information, errors and warnings on the console.

These are especially helpful when you are working on a big project with several other developers. You want your program to be meaningful for them.

```
32 function draw() {
33   //set background color
34   background(220);
35
36   console.info("Start of the draw function")
37   console.error("This is how error appears")
38   console.warn("A warning")
39
40
41   //jump when the space key is pressed
42   if(keyDown("space") && trex.y >= 100) {
```

Console

✖ This is how error appears

⚠ A warning

ℹ Start of the draw function

✖ This is how error appears

⚠ A warning

ℹ Start of the draw function

	<p>Encourage the student to write reflection notes in their reflection journal using markdown.</p> <p>Use these as guiding questions:</p> <ul style="list-style-type: none"> • What happened today? • Describe what happened. • 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? 	<p><i>Student uses the markdown editor to write her/his reflection as a reflection journal.</i></p>
--	---	---

Activity	Activity Name	Links
Teacher Activity 1	Unindented code	https://editor.p5js.org/Abhijeet/sketches/zbK8rf0uz
Teacher Activity 2	Finished code for reference	https://editor.p5js.org/Abhijeet/sketches/NNVga4nFL
Student Activity 1	Unindented code	https://editor.p5js.org/Abhijeet/sketches/zbK8rf0uz

