


Topic	SETTING UP LOCAL ENVIRONMENT AND UPLOADING CODE TO GITHUB	
Class Description	<p>Students will explore the different files - HTML, CSS, JavaScript, other asset files - that go into making a web page which hosts a game. Students will learn about JavaScript libraries and how they can be used within our code.</p> <p>Students will set up their local environment to write and test code using JavaScript. Students will also learn to upload their code on GitHub.</p>	
Class	C9	
Class time	45 mins	
Goal	<ul style="list-style-type: none"> <li>Explore the role of HTML, CSS and JavaScript in the design for a web page that can host the game page.</li> <li>Learn about JavaScript libraries and how to use them within our code.</li> <li>Set up a local environment to write and test code using JavaScript.</li> <li>Upload the code on GitHub.</li> </ul>	
Resources Required	<ul style="list-style-type: none"> <li>Teacher Resources:               <ul style="list-style-type: none"> <li>VS code Editor</li> <li>Laptop with internet connectivity</li> <li>Earphones with mic</li> <li>Notebook and pen</li> </ul> </li> <li>Student Resources:               <ul style="list-style-type: none"> <li>Laptop with internet connectivity</li> <li>Earphones with mic</li> <li>Notebook and pen</li> </ul> </li> </ul>	
Class structure	<b>Warm-Up Slides</b> <b>Teacher - led Activity</b> <b>Student - led Activity</b> <b>Wrap-Up Slides</b>	<b>10 mins</b> <b>10 mins</b> <b>20 mins</b> <b>5 mins</b>

WARM-UP SESSION - 10 mins	
<div>  </div> <p><b>Teacher starts slideshow from slides 1 to 21</b></p> <p>Refer to speaker notes and follow the instructions on each slide.</p>	
Teacher Action	Student Action
<p><i>Hey &lt;student name&gt;. How are you doing today? We're going to have a great time today; we are going to learn how to make our game available to anyone. Are you excited?</i></p> <p><b>Run the presentation from slide 1 to slide 5.</b></p> <p><b>Following are the warm up session deliverables:</b></p> <ul style="list-style-type: none"> <li>Connecting students to the previous class.</li> <li>Help the student revise concepts covered so far.</li> </ul>	<p><b>ESR:</b> Hi, I'm good. Yes I am!</p>
Q&A Session	
Question	Answer
<p>Select the line of code that declares a variable score and initializes it to zero.</p> <p>A. <code>const score = 0;</code>            B. <code>var score = 0;</code>            C. <code>var score(0);</code>            D. <code>var score;</code></p>	<p><b>B.</b></p>
<p>Select the line of code to display the message "Game Over!" on the canvas at x position 160 and y position 160.</p> <p>A. <code>display("Game Over!",160,160);</code>            B. <code>show("Game Over!",160,160);</code>            C. <code>text("Game Over!",160,160);</code>            D. <code>message("Game Over!",160,160);</code></p>	<p><b>C.</b></p>
Continue the warm up session	
Activity details	Solution/Guidelines

Run the presentation from slide 6 to slide 21 to set the problem statement.

Following are the warm up session deliverables:

- Explain the coding environment with the current example of p5 editor.
- Local coding environment meaning and benefits.
- Concept of hosting a game online to make it available for others.
- Introduce Github to host a student's game.

Teacher ends slideshow



**TEACHER-LED ACTIVITY - 10 mins**

Teacher Initiates Screen Share

#### CHALLENGE

- Walk through the different files and their structure of p5.js code.
- Download the code from GitHub and extract it from the zip file.

Teacher Action

Student Action

**Step 1:**

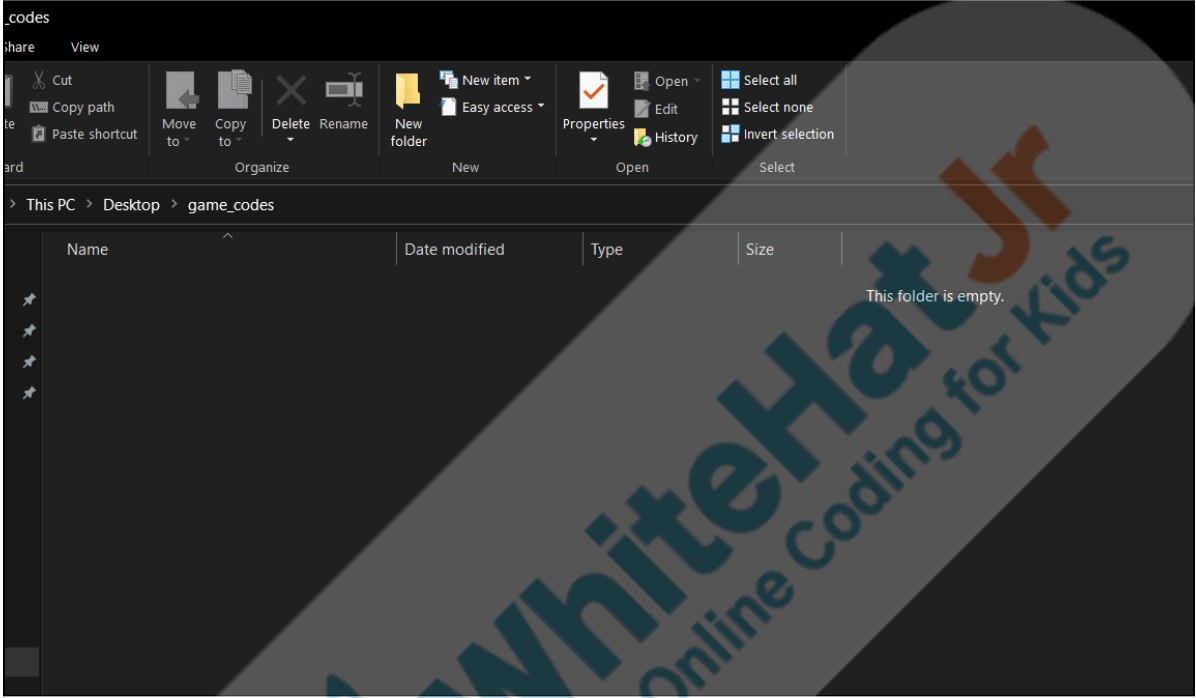
**Teacher-led Activity-1**  
(15 min)

*The teacher downloads the template code in VS code Editor from [Teacher Activity 1](#).*

In previous classes, we were using **code.org** to program the games. But from this class onwards, we are going to use a very popular coding editor—**Visual Studio Code**.

Code.org was running in our browser.

But VS Code editor will be installed on our computer and we can code in VS code even if we don't have the internet.

<p>Create a new folder named “game_code”.</p> <p>We will store our programs in this folder.</p>	<p><i>The student creates the local folder on their machine.</i></p>
	
<p>Before we install the VS code and start writing the code, let's first understand what different files are going to be used in our code.</p> <p>We are using p5.js library to write the code. We have been using it from past 8 classes, but on code.org it was in a much simpler way, but all the functions are going to remain the same.</p>	
<p>Let's see if we can see the different files p5.js library used to create our Trex game.</p> <p>Can you guess/identify these files?</p>	<p><i>The student identifies the image and sound files.</i></p> <p><i>Student also identifies the p5.play library, index.html and CSS file</i></p>

```
<> index.html
```

```
JS p5.js
```

```
JS p5.sound.min.js
```

```
JS sketch.js
```

```
# style.css
```

Remember, why do we use JavaScript libraries?

**Note:** However, the computer could already do certain things - like playSound(), createSprite(), etc. without teaching—codes for how to do these were defined in the library.

**ESR:**

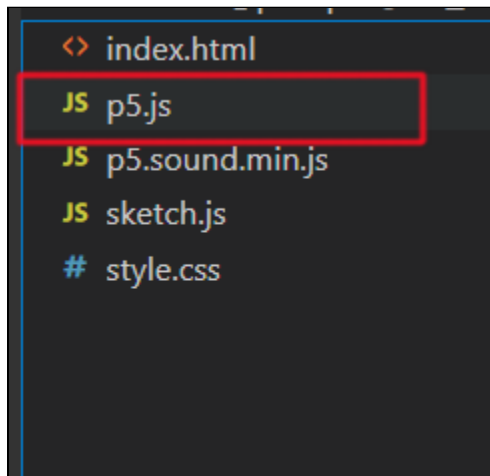
We taught the computer to do some things like - reset() etc. - by writing functions for them.

Yes. Someone has written code for these and created a library so that anyone can use them. Libraries are collections of code which can be reused by other programs in their code.

We are simply using several functions from this library into our code. The library we are using in our code is called p5.play.

Can you locate the library in our code?

*The student locates the library p5.play.js.*



It is common practice for developers to use code written by others into their program. It helps us create applications and programs faster. We will learn how to create our own libraries soon!

But how do we tell the computer to use the p5 library?

**ESR:**

Using `<script>` tags in index.html.

We tell the computer to use these different libraries in our index.html file.

index.html is the file which loads FIRST when you load your game. index.html file contains tags which can tell the computer what to display. It is similar to the markdown file we learned earlier. Here we use tags instead of symbols.

For example `<script>` is a tag.

Inside two `<script></script>` tags, we can tell the computer to load any JavaScript code.

Inside two `<body></body>` tags, we tell the computer what to display on the page.

*The student observes and asks questions.*

Inside two `<head></head>` tags, we tell the computer the different libraries we want to load.

```

<> index.html > ...
1  <!DOCTYPE html><html lang="en"><head>
2    <script src="p5.js"></script>
3    <script src="p5.sound.min.js"></script>
4    <link rel="stylesheet" type="text/css" href="style.css">
5    <meta charset="utf-8">
6
7  </head>
8  <body>
9    <script src="sketch.js"></script>
10
11
12 </body></html>
  
```

Can you identify the line of code where p5.play library is being uploaded in the HTML file?

*Student locates the line.*

We have seen the index.html file.  
 But we write our code in a sketch.js file. It looks very similar to the code on code.org.  
 Because code.org is also using the p5.js library in the backend.

```
JS sketch.js ×  
JS sketch.js > setup  
1 function setup() {  
2   createCanvas(400, 400);  
3 }  
4  
5 function draw() {  
6   background(220);  
7 }
```

Awesome! You are doing brilliantly!

If you notice, there are other libraries inside script tags which we are using in our code - p5.js, p5.dom.

So you know:

- the JavaScript library we are using in our game;
- the file where we are writing our game; and
- the HTML file which is responsible for displaying content and loading our libraries.

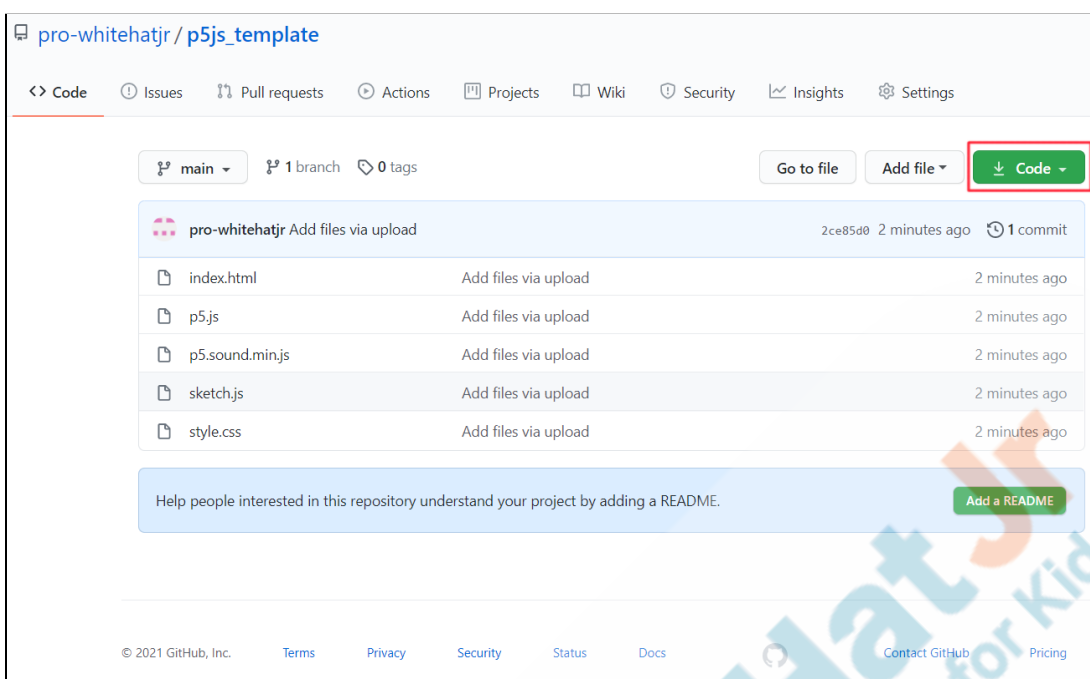
Which file is left, which we don't know about?

**ESR:**  
style.css

```
# style.css > html  
1 html, body {  
2   margin: 0;  
3   padding: 0;  
4 }  
5 canvas {  
6   display: block;  
7 }  
8
```

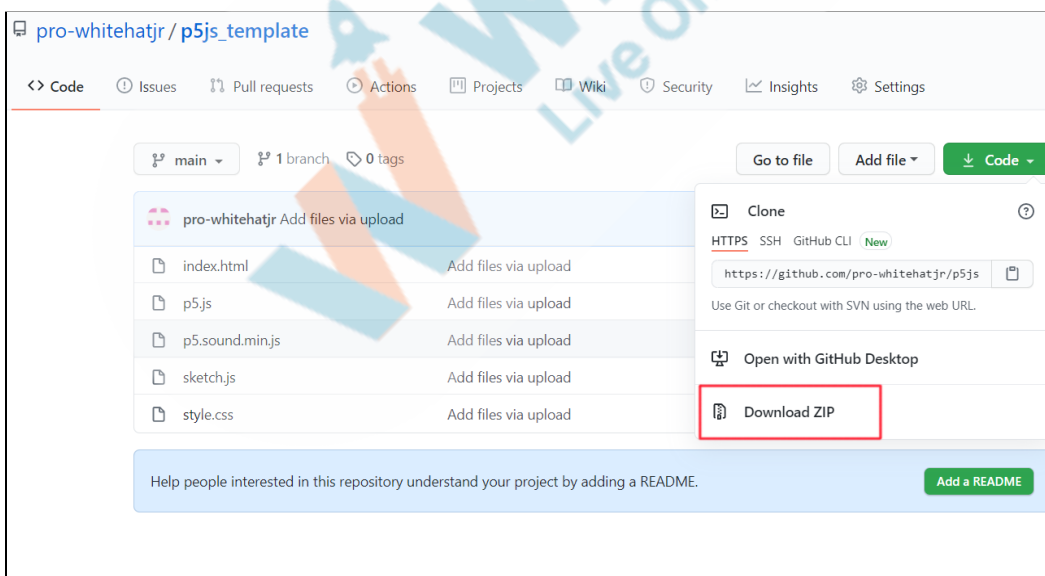


<p>style.css instructs the computer to apply formatting - changing style, adding margin, padding etc.</p> <p>We will look at them in detail later.</p>	
<p>We are going to recreate these files in our local machine and run them.</p> <p>We can download all the files on our local machine from p5 editor or GitHub OR we can create the same files on our system.</p> <p>For now, let's download all the files on our system from Github.</p> <p>What is GitHub?</p> <p>GitHub is a website where we can store our code and share that with others.</p> <p>More on that we are going to learn later in the class.</p>	<p><b>ESR:</b> Varied</p>
<p>The Code we download from GitHub will be in a zip file. We need to extract it so that we can run this in the VS Code editor.</p>	



The files are being downloaded in the **zip** format that means all the files are **compressed** together into one folder.

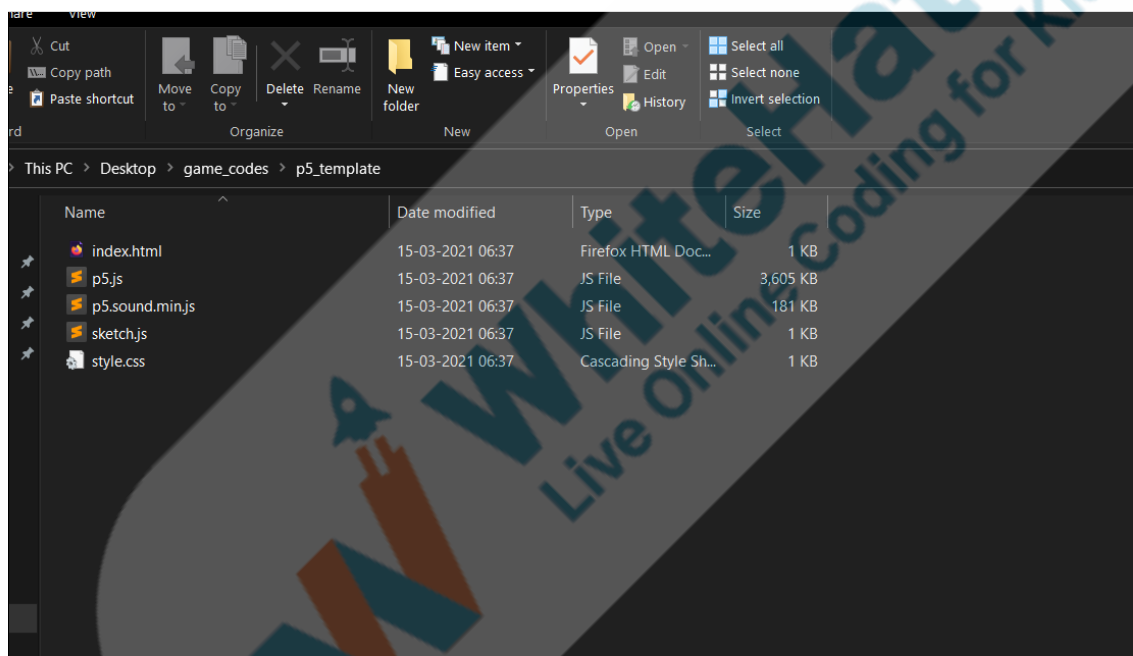
*The student downloads the zipped file in their system..*



To use or edit these we need to first **unzip** the files that mean we need to **extract all the files**.

To extract files:

- Copy and paste the zip folder into the “game\_codes” folder that we created instead of extracting the files in the download folder of the computer.
- Right on the folder and click extract all for Windows or double click on the folder for Mac OS.

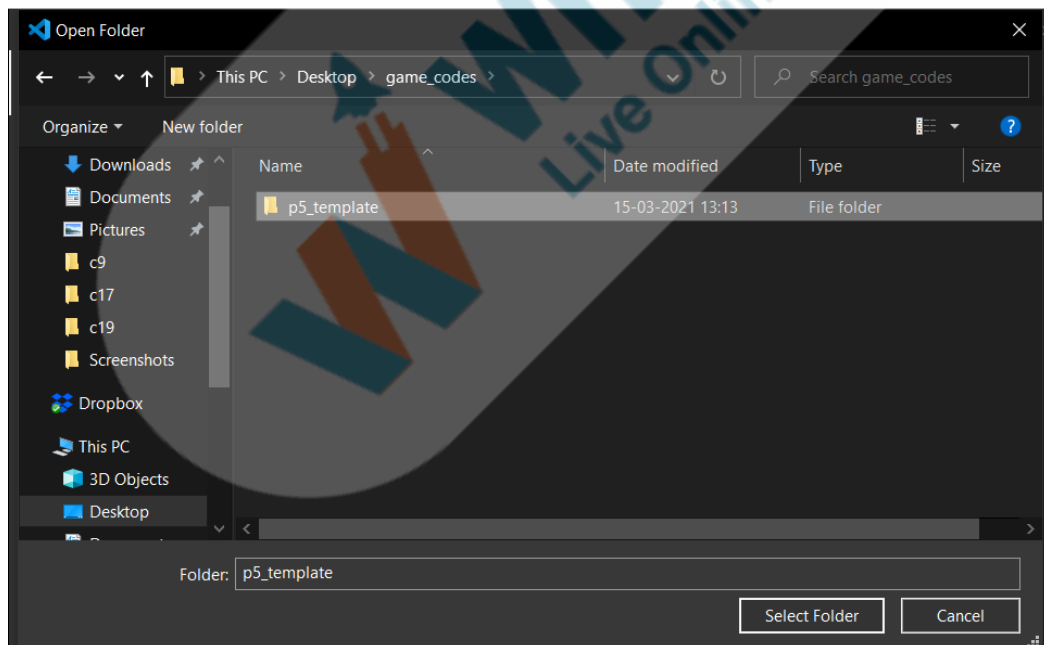
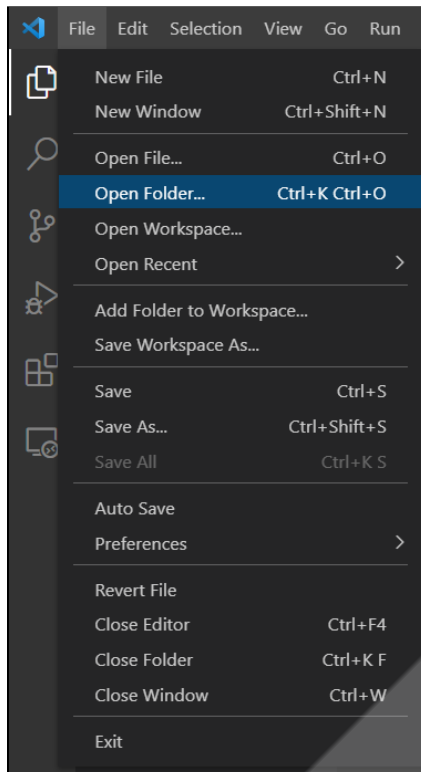


Now, we have all the files on our local machine.

Open the VS code and open the p5\_template folder in VS Code.

**Click on file>Go to Open folder option>select folder you want to open.**

*The teacher will show how to open the folder in vscode.*



Now we need a server to host our files so that we can run our game.

In code.org, the site had a server running in the background which was hosting all our files.

Our VS Code also has an extension which will help us run a live server.

The extension is called '**Live Server**'.

Steps to install the Live Server extension:

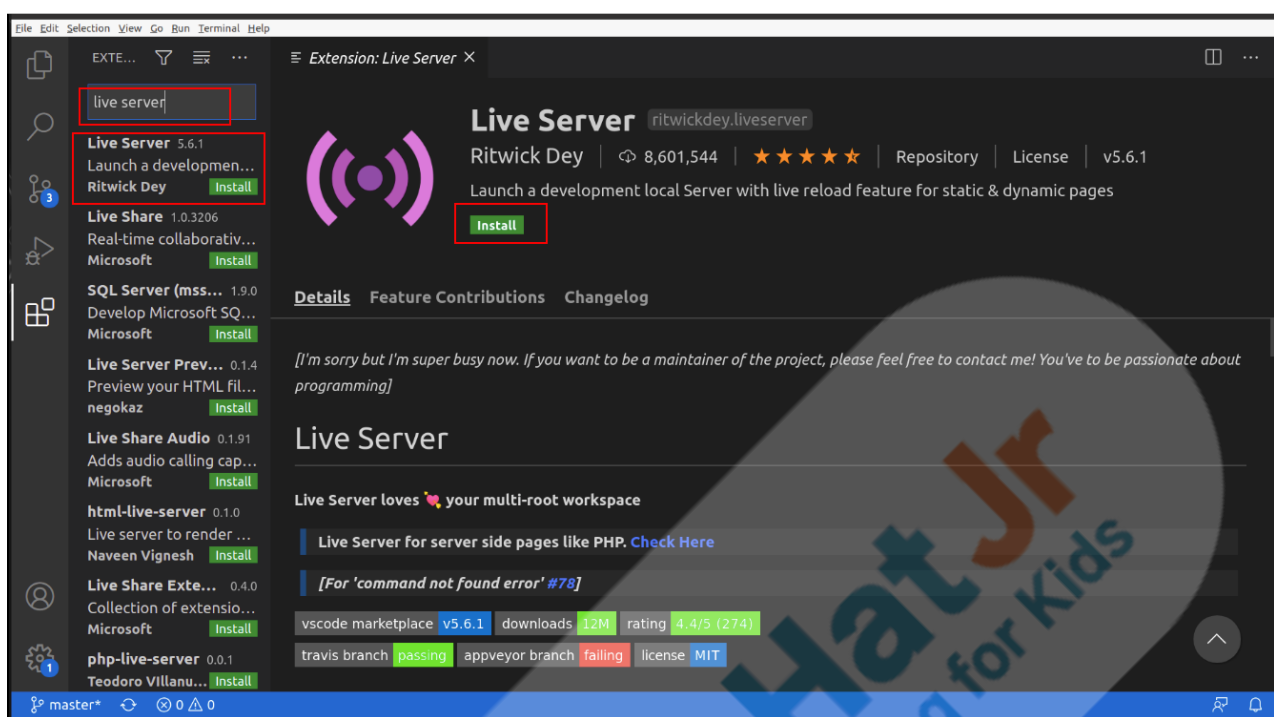
- 1) Click on View and then select 'Extensions' or press "Ctrl + Shift+x".
- 2) Search for "Live Server".
- 3) Click on install.

To start the Live Server:

- 1) Open the project Folder in the VS Code.
- 2) Click on **Go Live** in the status bar to turn the live server on and off. OR right click on index.html file and select **open with live server**.

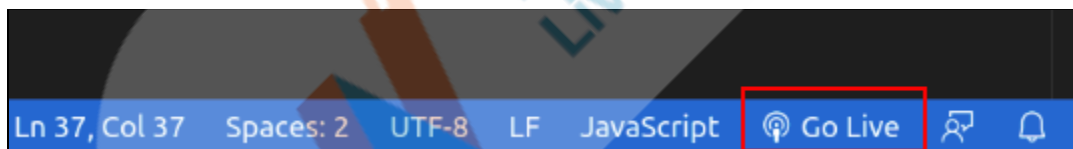
*The teacher shows how to install **Live Server Extension**.*

*The student installs Live Server extension*



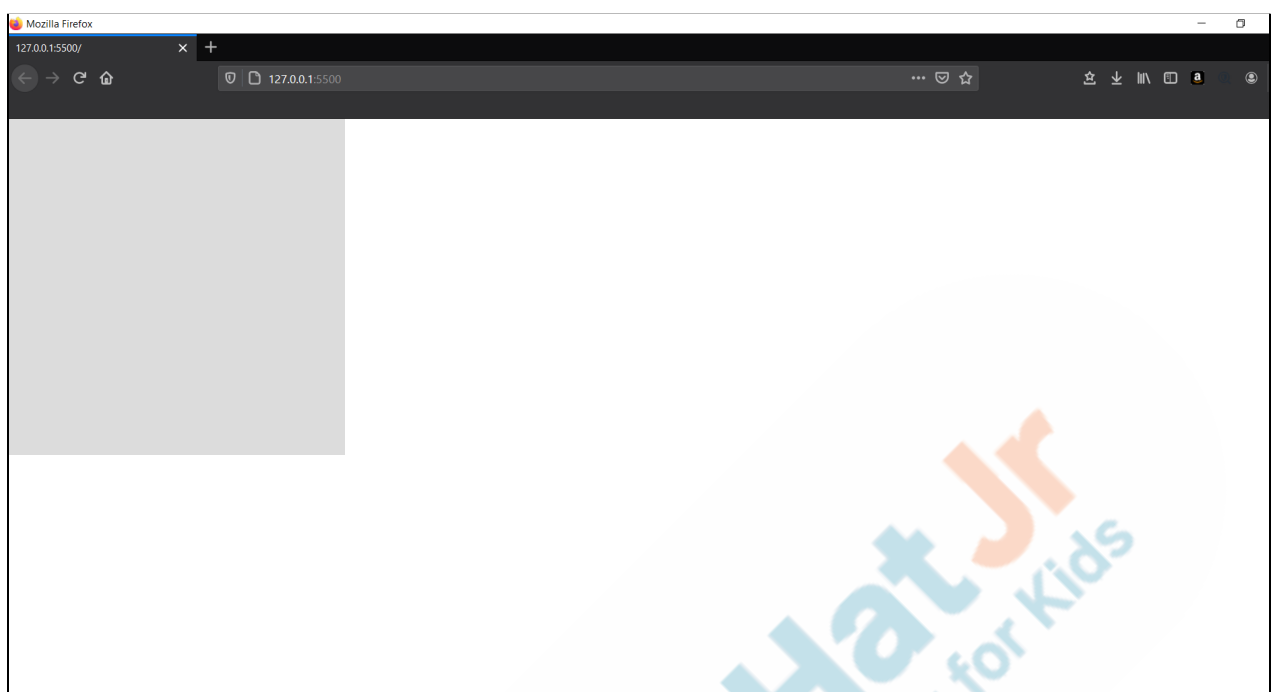
Let us launch the Live server.

*The teacher clicks on the Go Live option on the status bar to launch the server.*



Live Server opens the browser window showing the output of our code.

**Note:** This link will run on your local machine only. We cannot share this link with others.



Ok. Now we have all the files on our local machine and we can run the code on our local machine as well.

Now, I am going to teach you how to make changes to the code on your local system and run them.



### Teacher Stops Screen Share

### STUDENT-LED ACTIVITY - 20 mins




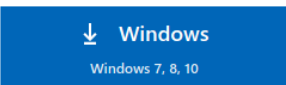
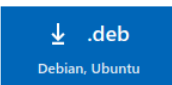

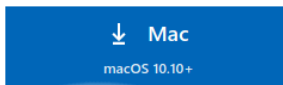
- Ask the student to press the ESC key to come back to the panel.
- Guide the student to start Screen Share.
- The teacher gets into Fullscreen.

### ACTIVITY

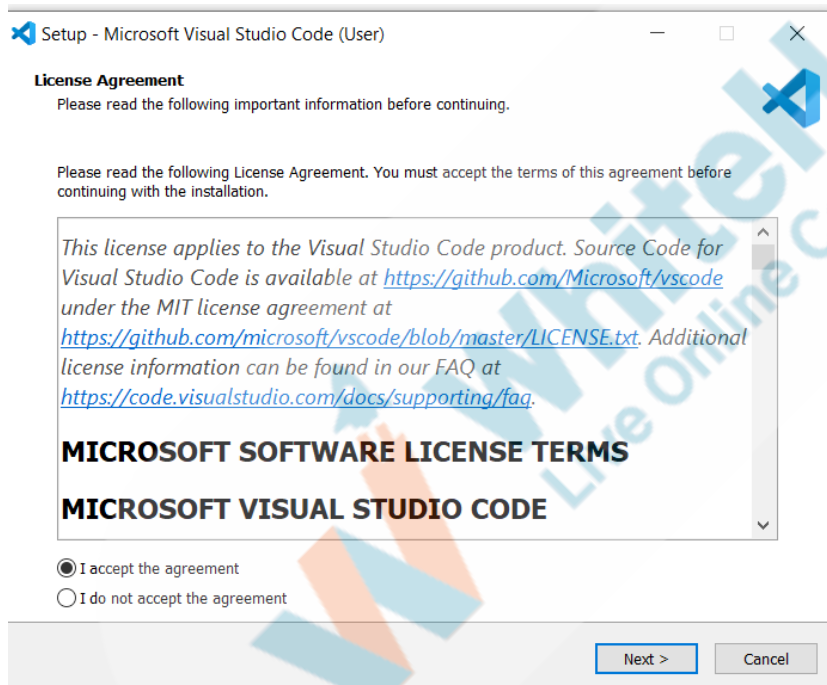
- Install Visual Studio Code Editor and learn to use it as a development environment.
- Make changes to the code on their local machine.
- Learn how to upload the code on github servers.

 <b>Teacher starts slideshow :Slide 22 to slide 31.</b>	
<b>Run the presentation for slide 22 to slide 31 set the student activity context.</b>	
Let's work locally on our systems to make a few changes to our code and finally host it online.	<i>Student listens carefully</i>
Go ahead and share your screen; let's get started.	<i>The student shares his/her screen, opens the Student Activity. Teacher guides the student to use Visual Studio Code and helps set up his/her github account.</i>
<b>Teacher ends slideshow</b> 	
Teacher Action	Student Action
<b>Step 3:</b> <b>Student-Led Activity</b> <b>(15 min)</b> We have all the files on our local machine. Now we need to install Visual Studio Code so that we can run them.	<i>The student installs Visual Studio Code on their local machine using <a href="#">Student Activity 1.</a></i>
Let's install Visual Code Editor on our system.  <i>The teacher guides the student to install visual code editor on their system.</i>	

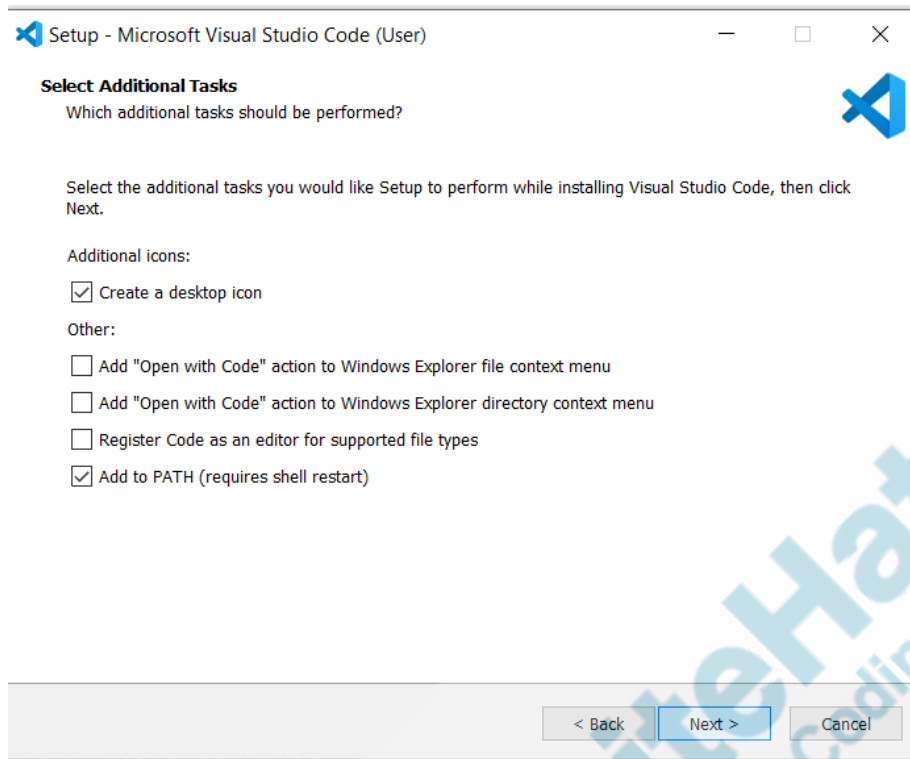


		
		
		
<p>User Installer    64 bit   32 bit</p> <p>System Installer   64 bit   32 bit</p> <p>.zip                64 bit   32 bit</p>	<p>.deb                64 bit</p> <p>.rpm                64 bit</p> <p>.tar.gz            64 bit</p> <p><a href="#">Snap Store</a></p>	

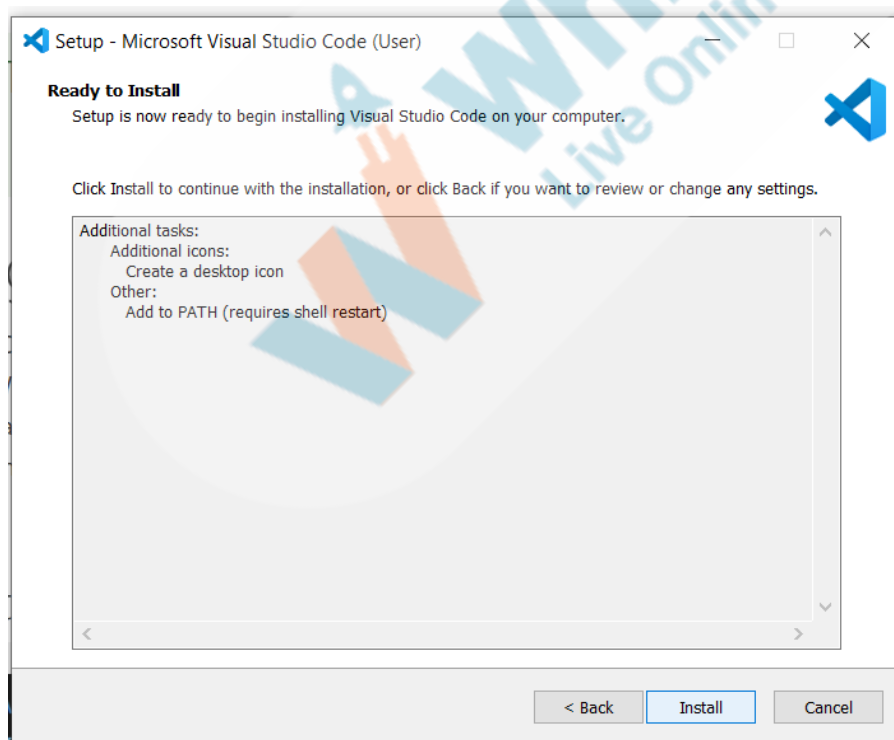
Double click on the downloaded file.  
Then click on the **next** button.



Click on **Next** again



Click on install, this will start the installation process.



<p>After installation, the student will open the code folder in the VS code editor and run the program using the live Server extension.</p> <p>We have our VS code editor running. Now let's create a small program to understand the process of writing the program and running with VScode Editor.</p>	
<p>We are going to write a code to move a sprite on the canvas using the <b>keydown()</b> function. You have created a similar program while creating World's Hardest Game.</p> <p>Can you tell me what we did in the C8 class?</p>	<p><i>The student downloads the code for <a href="#">Student Activity 2</a> and opens it in the VS Code Editor.</i></p> <p><b>ESR:</b> We created a sprite and controlled its motion using the arrow keys; there were other sprites on the canvas constantly moving up and down on the canvas. We had to navigate our sprite from one edge to the other edge of the canvas without colliding with the moving obstacles.</p>
<p>Very good!</p> <p>In this activity we are going to create a sprite and control its motion using arrow keys. But we are going to use our newly installed Visual Studio Code editor.</p> <p>First, we will define a variable for the sprite as <b>var box;</b> then we'll create a sprite at the center of the canvas. For that it will be having x and y position as 200.</p> <p>Then we will define the height and width of the sprite as</p>	

30. This sprite will be a square box.

In the **draw()** function we need to call 2 more functions, **background()** which will set the background color of the canvas and **drawSprites()** which will display our sprite.

**background()** function takes color value as an argument that ranges from 0 to 255. 0 means complete black whereas 255 represents white. We are going to choose 30 here.

You can choose any color value. We can also write the name of the colors such as 'blue', 'red' etc. like you did in the **code.org**.

```
var box;

function setup() {
  createCanvas(400,400);
  box = createSprite(200,200,30,30);
}

function draw()
{
  background(30);

  drawSprites();
}
```

*Guide the student to run the code using the Go Live button.*

We can see the output of the code as a square sprite at the center of the canvas.



**Output:**



Now we have our sprite. We can move forward to write the code to move this using the **keydown()** function. This function tells us whether a key is pressed or not.

We need to pass the key value as an argument such as **keydown (RIGHT\_ARROW)**.

We use this function along with the if condition.

if the right arrow key is pressed then move the sprite in the right direction.

To move the sprite we need to change its position and at first we are moving in the right direction. Can you tell me which position we need to change x or y?

**ESR:**  
Varied.

To move the sprite in the right direction we will change the x position.

Our coordinates start from the top left corner. As we move in the right direction the value of x will increase and when we move in the left direction its value decreases.

So in order to move the sprite in the right direction, we need to increase the position of the sprite by adding some value to it; higher the value, more will be the movement.

The code will be like:

```
if (keyIsdown(RIGH_ARROW)
{
  box.position.x = box.position. x+5;
}
```

*Guide the student to run the code using live Server.*

Let's run the code to see the output.

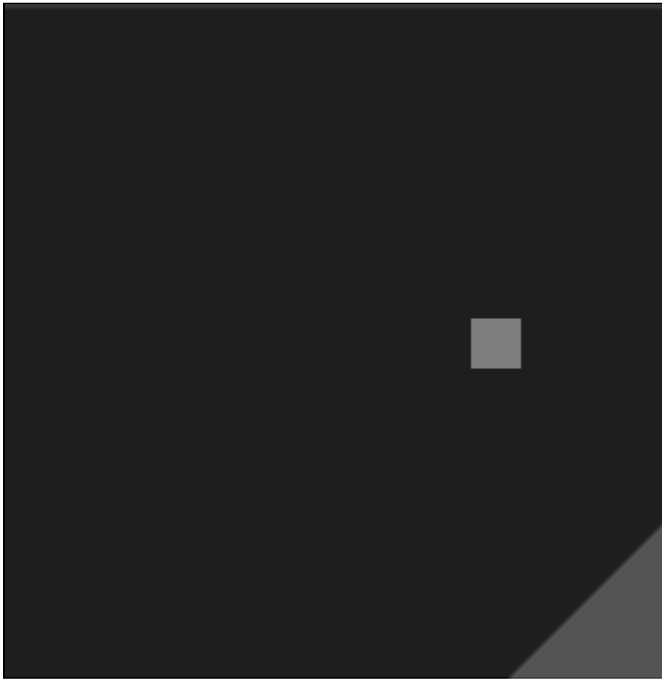
Click on the canvas and press the right arrow key to move the sprites towards the right side.

```
function draw()
{
  background(30);

  if (keyIsDown(RIGHT_ARROW))
  {
    box.position.x = box.position.x +5;
  }

  drawSprites();
}
```

**Output**



We are able to move the sprite in the right direction; now let's write the code for the left direction, which is going to be similar to what we just did.

Here we are going to use the **LEFT\_ARROW** and we need to subtract from the x position of the sprite.

*Run the code to see the output.*

```
function draw()
{
  background(30);

  if (keyIsDown(RIGHT_ARROW))
  {
    box.position.x = box.position.x +5;
  }

  if (keyIsDown(LEFT_ARROW))
  {
    box.position.x= box.position.x -5;
  }

  drawSprites();
}
```

Output:



Great! We can see that our sprite is now moving in the left



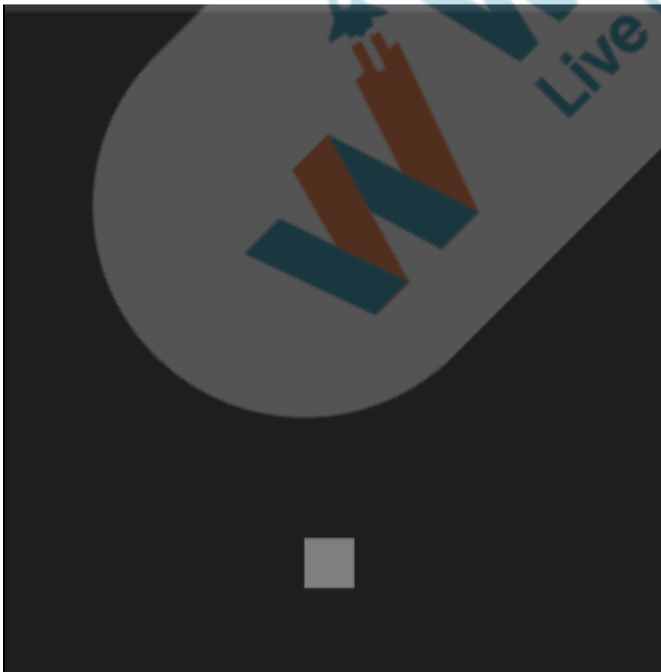
and right direction. The process is the same for moving the sprite upwards and downwards.

But instead of changing the x position we need to change the y position. To move the sprite up we need to add a value to the y position and to move it down. We will subtract some value.

*Guide the student to write code.*

```
if (keyIsDown(UP_ARROW))  
{  
    box.position.y = box.position.y - 5;  
}  
  
if (keyIsDown(DOWN_ARROW))  
{  
    box.position.y = box.position.y + 5;  
}
```

**Output:**



We can move the box sprite in all 4 directions. But if we keep moving the sprite, it goes out of the canvas.

This we will explore in the upcoming classes.

We are done with the code.

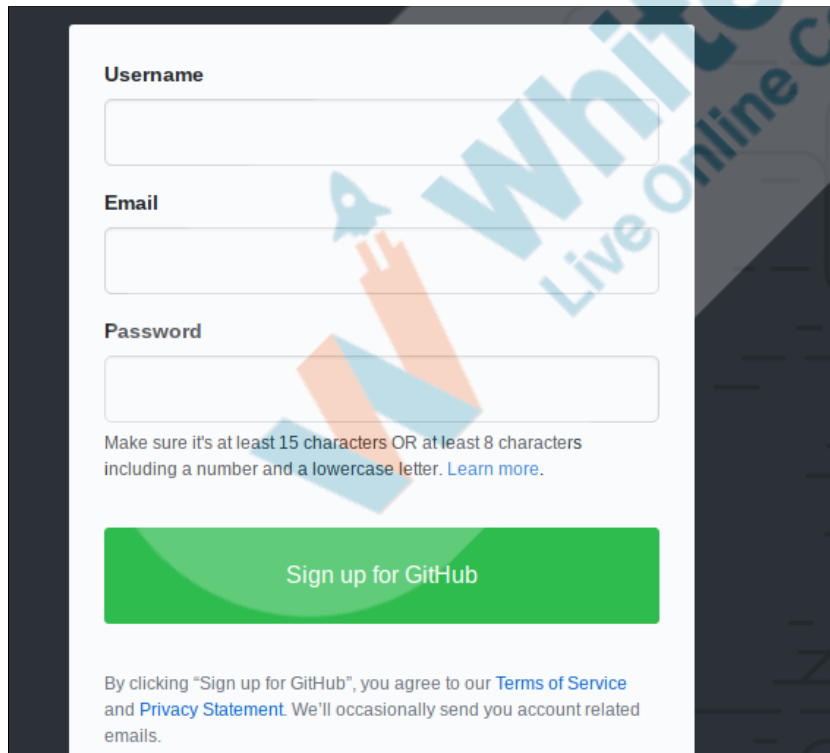
Now we will see how we can upload this code on GitHub.

To be able to upload the project on GitHub, we first need to have a GitHub account.

You can make a GitHub account using your email ID. First we need to go to the GitHub website.

*The teacher guides the student to setup a GitHub account.*

*The student will go to the GitHub website using [Student Activity 3](#).*



The screenshot shows the GitHub sign-up page. It has a dark background with a light gray sign-up form in the center. The form contains three input fields: 'Username', 'Email', and 'Password'. Below the 'Password' field, there is a note: 'Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter. [Learn more.](#)'. At the bottom of the form is a large green button that says 'Sign up for GitHub'. Below the button, there is a small text: 'By clicking "Sign up for GitHub", you agree to our [Terms of Service](#) and [Privacy Statement](#). We'll occasionally send you account related emails.'

*Guide the student to create a new project repository.*

*The student creates a new project repository by filling the form.*

## Learn Git and GitHub without any code!

Using the Hello World guide, you'll create a repository, start a branch, write comments, and open a pull request.

[Read the guide](#)

[Start a project](#)

## Discover interesting projects and people to

### Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository](#).

Owner

 whitehatjr

Repository name \*

trex 

Great repository names are short and memorable. Need inspiration? How about [super-umbrella](#)?

Description (optional)

Chrome's trex game created using p5.play



**Public**

Anyone can see this repository. You choose who can commit.



**Private**


You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.

☒ **Initialize this repository with a README**

This will let you immediately clone the repository to your computer.

Add .gitignore: **None**

Add a license: **None** 

[Create repository](#)

*Guide the student to upload the project files on github.*

*The student uploads the files to their Github project repository.*

*Student has to commit the change to save the files.*

Chrome's trex game created using p5.play

Edit

Manage topics

1 commit

1 branch

0 releases

1 contributor

Branch: master

New pull request

Create new file

Upload files

Find File

Clone or download

whitehatjr Initial commit

Latest commit 1b55a9a now

README.md

Initial commit

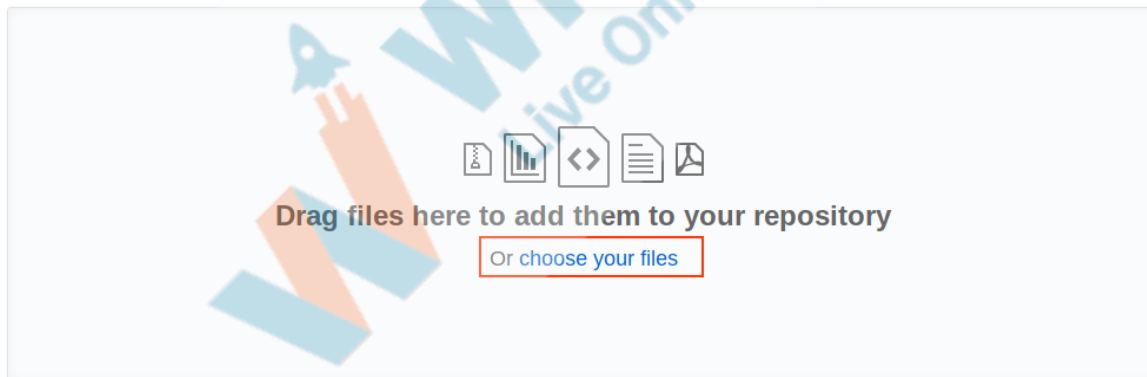
now

README.md

**trex**

Chrome's trex game created using p5.play

trex /



### Commit changes

Update Slingshot.js

Add an optional extended description...

☒ Commit directly to the `main` branch.

☐ Create a new branch for this commit and start a pull request. [Learn more about pull requests.](#)

Commit changes

Cancel

We have uploaded our code Files on GitHub.

Wow! Now you can share the link to your GitHub project with your friends and family so that they can see and download the code.


Our code is now hosted on the GitHub server - it can be accessed from anywhere.


Let's wrap-up today's class.

*The student visits the Github link to see the uploaded project.*

**Teacher Guides Student to Stop Screen Share**

**WRAP-UP SESSION - 5 Mins**

<div>  </div> Teacher starts slideshow from slide 32 to slide 42	
Activity details	Solution/Guidelines
<p>Run the presentation from slide 32 to slide 42</p> <p>Following are the warm up session deliverables:</p> <ul style="list-style-type: none"> <li>● Explain the facts and trivias</li> <li>● Next class challenge</li> <li>● Project for the day</li> <li>● Additional Activity</li> </ul>	<p><i>Guide the student to develop the project and share with us.</i></p>
Quiz time - Click on in-class quiz when on slide 40	
Question	Answer
<p>Which of these is the correct function to detect a key press?</p> <p>A. mouseOver(sprite);            B. keypressed() = true;            C. mousePressedOver();            D. keyIsdown()</p>	D.
<p>When we write our code on our own computer, what is it called?</p> <p>A. setting up a local environment            B. hosting a game            C. setting up a server            D. sharing the game online</p>	A.
<p>When we host a game on GitHub we create a ____.</p> <p>A. reservatory            B. reservoir            C. depository            D. repository</p>	D.

End the quiz panel	
<p>Teacher ends slideshow</p> 	
<p><b><u>FEEDBACK</u></b></p> <ul style="list-style-type: none"> <li>• Encourage the student to make reflection notes in markdown format.</li> <li>• Complement the student for her/his effort in the class.</li> <li>• Review the content of the lesson.</li> </ul>	
<p><b>Step 4:</b> <b>Wrap-Up</b> <b>(5 min)</b> Quickly, let's review what we studied in today's class.</p>	<p><b>ESR:</b></p> <ul style="list-style-type: none"> <li>• We learned about HTML, CSS and JS files and what they do in a webpage.</li> <li>• We learned to create a local environment to write code and run our program on our local machine.</li> <li>• We learned to host our game on github servers so that we can share our game with friends.</li> </ul>
<p>Awesome!</p> <p>In the next class, you will learn the concepts of how infinite runner games are created. You will be developing a program in VSC.</p> <p>We will also review what we have covered in the classes so far.</p> <p>Looking forward to the next class.</p>	

**\* This Project will take only 30 mins to complete.  
 Motivate students to try and finish it immediately after the class.**

## Project Overview

### BACKGROUND COLOR CHANGER

#### Goal of the Project:

In Class 9, you have learned how to move the sprite using the keyisdown() function. In this project you will learn how to change the color of canvas by pressing a keyboard key.

#### Story:

Johanna wants to create a game in which she wants to keep an option for changing background color while running the project. Can you help Johanna to create a background color changer?

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

Bye Bye!

*Students engage with the teacher over the project.*

You get Hats Off for your excellent work.

Make sure you have given at least 2 Hats Off during the class for:



**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?

*The student uses the markdown editor to write her/his reflections in the reflection journal.*

Activity	Activity Name	Links
Teacher Activity 1	p5js code template	<a href="https://github.com/pro-whitehatjr/p5js_template">https://github.com/pro-whitehatjr/p5js_template</a>
Teacher Activity 2	Github	<a href="https://github.com/">https://github.com/</a>
Teacher Activity 3	VS Code	<a href="https://code.visualstudio.com/download">https://code.visualstudio.com/download</a>
Teacher Reference	Reference code of Student Activity 2	<a href="https://github.com/pro-whitehatjr/Pro-c9-sa-reference">https://github.com/pro-whitehatjr/Pro-c9-sa-reference</a>
Student Activity 1	VS Code	<a href="https://code.visualstudio.com/download">https://code.visualstudio.com/download</a>
Student Activity 2	Template Code	<a href="https://github.com/pro-whitehatjr/p5play_template">https://github.com/pro-whitehatjr/p5play_template</a>
Student Activity 3	Github	<a href="https://github.com/">https://github.com/</a>
Teacher Reference visual aid link	Visual aid link	<a href="https://curriculum.whitehatjr.com/Visual+Project+Asset/PRO_VD/BJFC-PRO-V3-C9-withcues.html">https://curriculum.whitehatjr.com/Visual+Project+Asset/PRO_VD/BJFC-PRO-V3-C9-withcues.html</a>

Teacher Reference In-class quiz	In-class quiz	<a href="https://s3-whjr-curriculum-uploads.whjr.online/d5fab24a-fd7e-45ae-b6d5-67606bbb5030.pdf">https://s3-whjr-curriculum-uploads.whjr.online/d5fab24a-fd7e-45ae-b6d5-67606bbb5030.pdf</a>
Project Solution	Background Color Change	<a href="https://github.com/pro-whitehatjr/c9_project">https://github.com/pro-whitehatjr/c9_project</a>

