SETTING UP LOCAL ENVIRONMENT AND UPLOADING CODE TO GITHUB

What is our GOAL for this MODULE?

Exploring HTML, CSS, JavaScript, other asset files - that go into making a web page which hosts a game.

What did we ACHIEVE in the class TODAY?

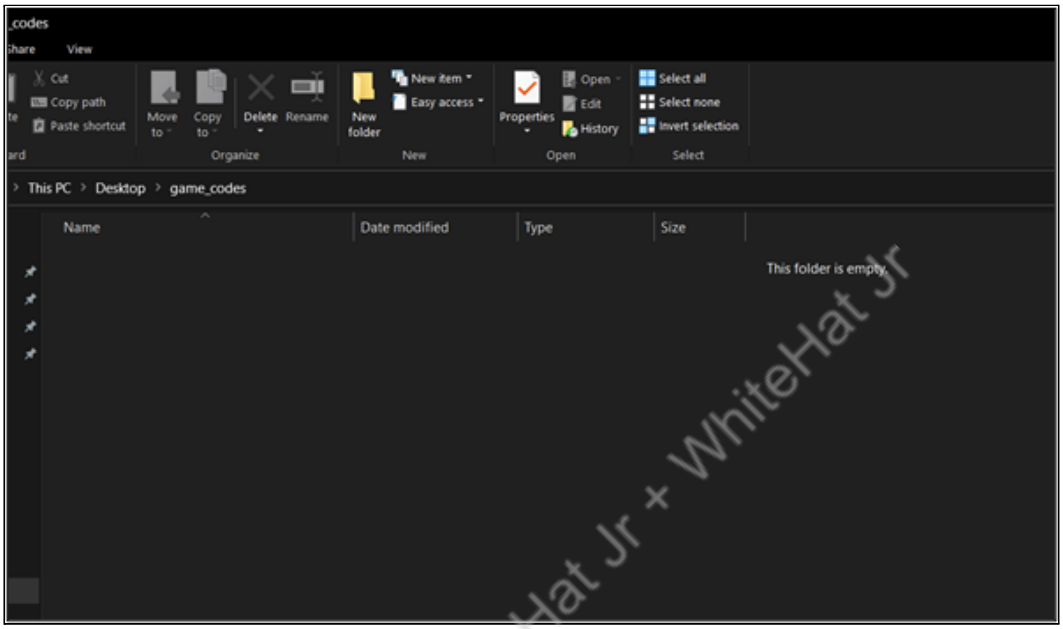
- Learned the role of HTML,CSS and JavaScript in the design for a web page which can host the game page.
- Learned about javascript libraries and how to use them within our code.
- Learned how to set up a local environment to write and test code using javascript.
- Uploaded the code on GitHub.

Which CONCEPTS/ CODING BLOCKS did we cover today?

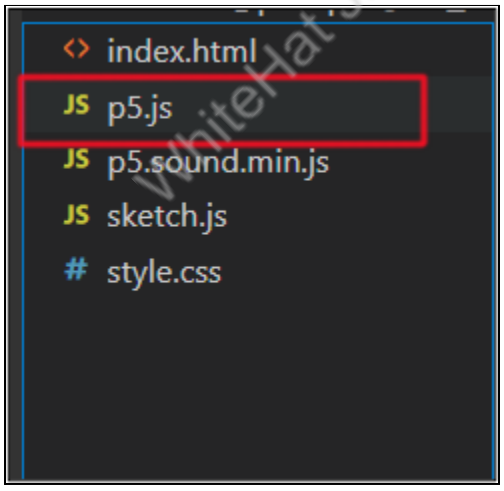
- Role of JavaScript library files.
- Concept of online hosting platform GitHub.
- VSC local environment.

How did we DO the activities?

1. Create a folder game_codes in your machine. Going ahead, keep a separate folder to save code for each class work and project work.



2. Use the predefined libraries to perform sprite-related functionalities is p5.play.js. JavaScript programs include many libraries; libraries are collections of code which can be reused by other programs in their code.



3. Connect sketch.js & support files (libraries) by including all the files in index.html.
    - JavaScript code can run in the browser with the help of index.html.
    - Inside a pair of **<head></head>** tags, we tell the computer the different libraries we want to load.
    - Inside a pair of **<body></body>** tags, we tell the computer what to display on the webpage. Sketch.js is added in the body tag.
    - Inside a pair of **<script></script>** tags, we provide the location of any JavaScript code.

```
ex.html > ...
<!DOCTYPE html><html lang="en"><head>
    <script src="p5.js"></script>
    <script src="p5.sound.min.js"></script>
    <link rel="stylesheet" type="text/css" href="style.css">
    <meta charset="utf-8">

  </head>
  <body>
    <script src="sketch.js"></script>


</body></html>
```

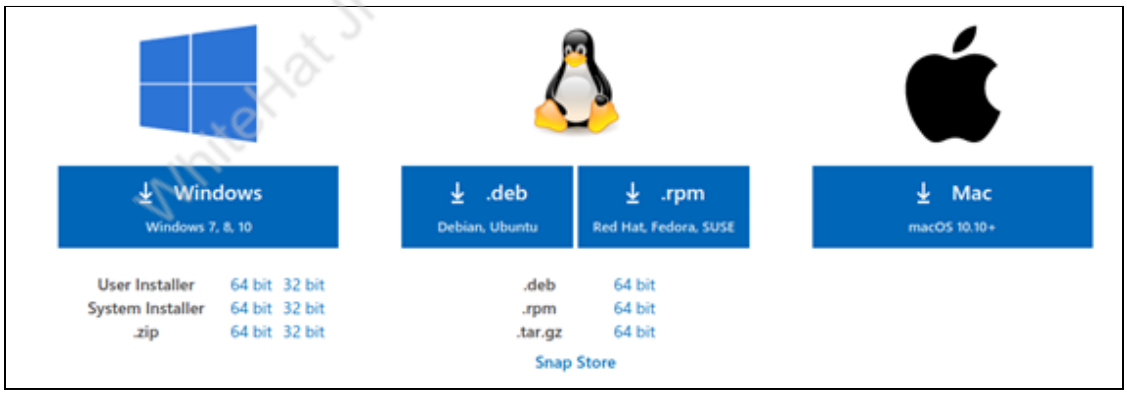
- Sketch.js includes the executable code.

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

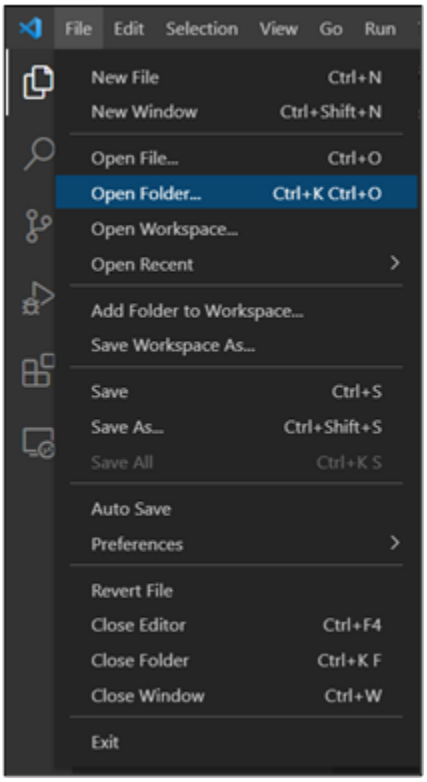
- Style.css file is used to change/add style to the webpage such as font color type, background color, position of elements such as button and images.

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

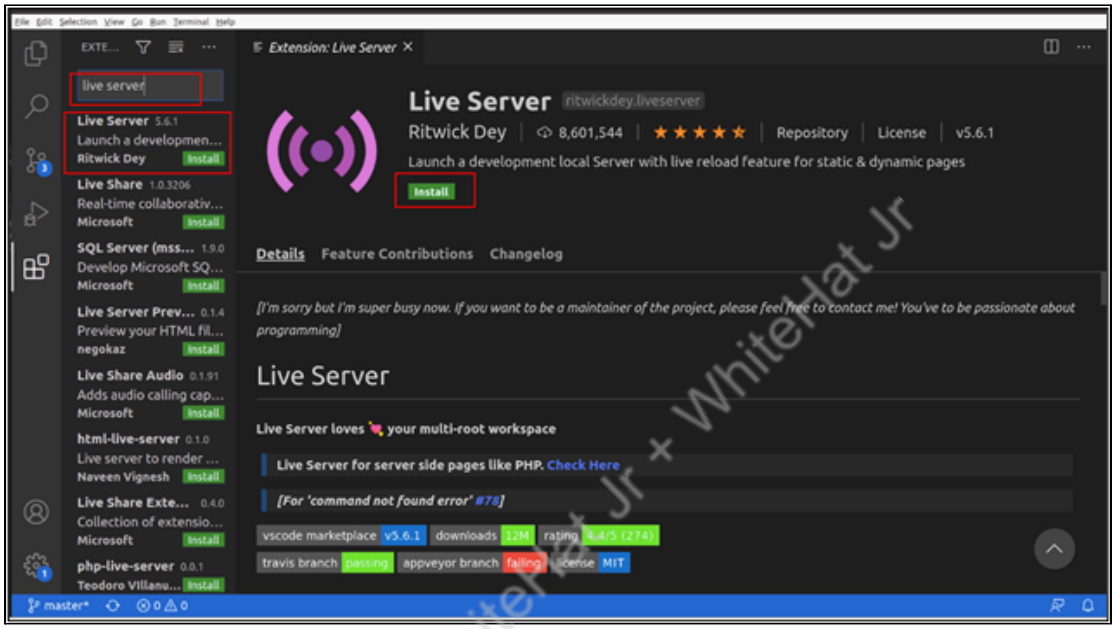
4. Visual Studio Code Editor allows us to write the code on our computer. It has various benefits over using Code.org.
    - No dependency on the Internet.
    - The canvas size can be changed as per requirement, In code. Org the canvas size is fixed at 400 x 400
    - While checking output, the player will not get an option to view code
    - The entire code is saved onto our machine



5. Download and save the code in the local folder. Then from the Open Folder option under File Tab you can choose the folder to edit the files.



6. Run the code; to see the result we need a local server. VSC comes with an extension to install Live Server. Install the Liver Server with the following steps:
    - Click on View and then select 'Extensions' or press "Ctrl + Shift+x".
    - Search for "Live Server".
    - Click on install.



7. Start the Live Server.
    - Open the project Folder in the VS Code.
    - 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.**

8. Use the template we downloaded in the game_cades folder. We wrote the code to create a calculator in VSCode and ran the code using Live Server.

9. Create the Input Box using createInput(). Two boxes created for num1 and num2

variables where the input provided through the textbox will be stored.

```
var n1,n2;

function setup() {

  createCanvas(400, 400);

  num1 = createInput();
  num1.position(5, 60);

  num2 = createInput();
  num2.position(200, 60);
```

10. Define b1, b2, b3, b4 variables to create buttons. Use **createButton()** to create the buttons. Also, attach the buttons to the respective functions like for add button the button will be attached to add function. The **mousePressed()** event will be triggered once the button will be pressed.

```
var n1,n2;
var b1,b2,b3,b4;
function setup() {

  createCanvas(400, 400);

  num1 = createInput();
  num1.position(5, 60);

  num2 = createInput();
  num2.position(200, 60);

  b1 = createButton("ADD");
  b1.position(10, 200);
  b1.mousePressed(add);

  b2 = createButton("Multiply");
  b2.position(100, 200);
  b2.mousePressed(mul);
```

```
b3 = createButton("Subtract");
b3.position(200, 200);
b3.mousePressed(sub);

b4 = createButton("Divide");
b4.position(300, 200);
b4.mousePressed(division);

textAlign(CENTER);
textSize(15);
```

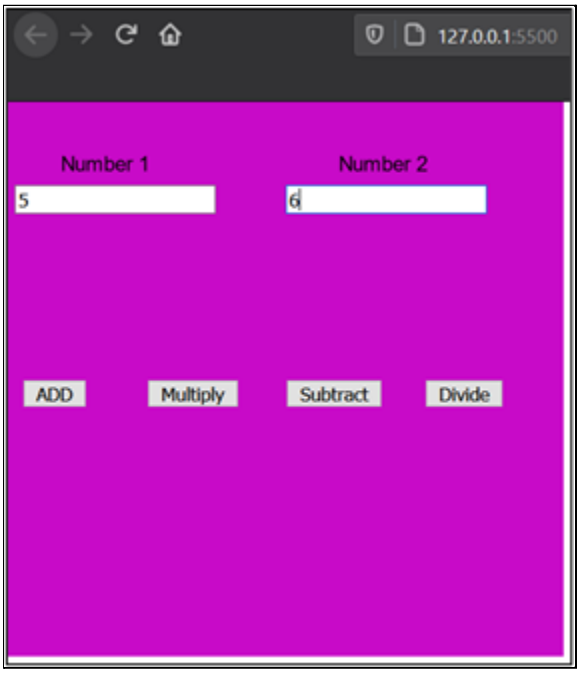
11. Convert the input by the user into integers. By default, the text typed is in string format. To convert to integer use parseInt().

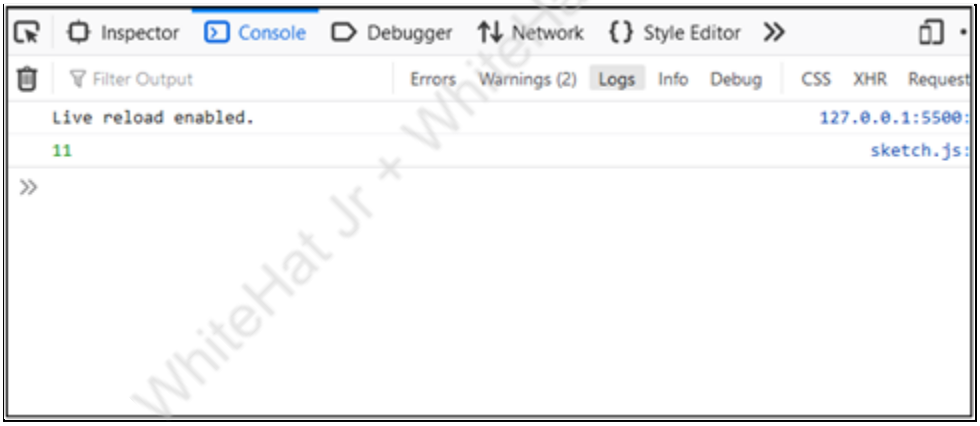
```
function draw() {
  background(200,10,200)
  text("Number 1",70,50)
  text("Number 2",270,50)

  n1 = parseInt(num1.value());
  n2 = parseInt(num2.value());

}
```

12. Create functions: add(), multiply(), subtract() and divide().

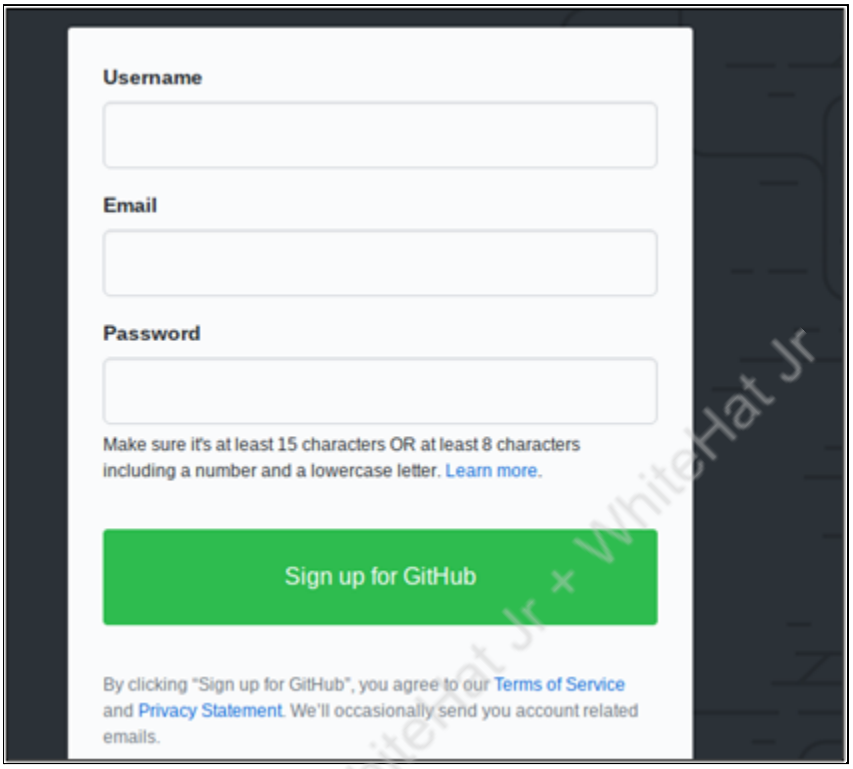
```
function add()
{

   console.log(n1+n2);

}
function sub()
{

   console.log(n1-n2);

}
function mul()
{

   console.log(n1*n2);

}

function division()
{
   console.log(n1/n2);
}
```

13. Run the code.



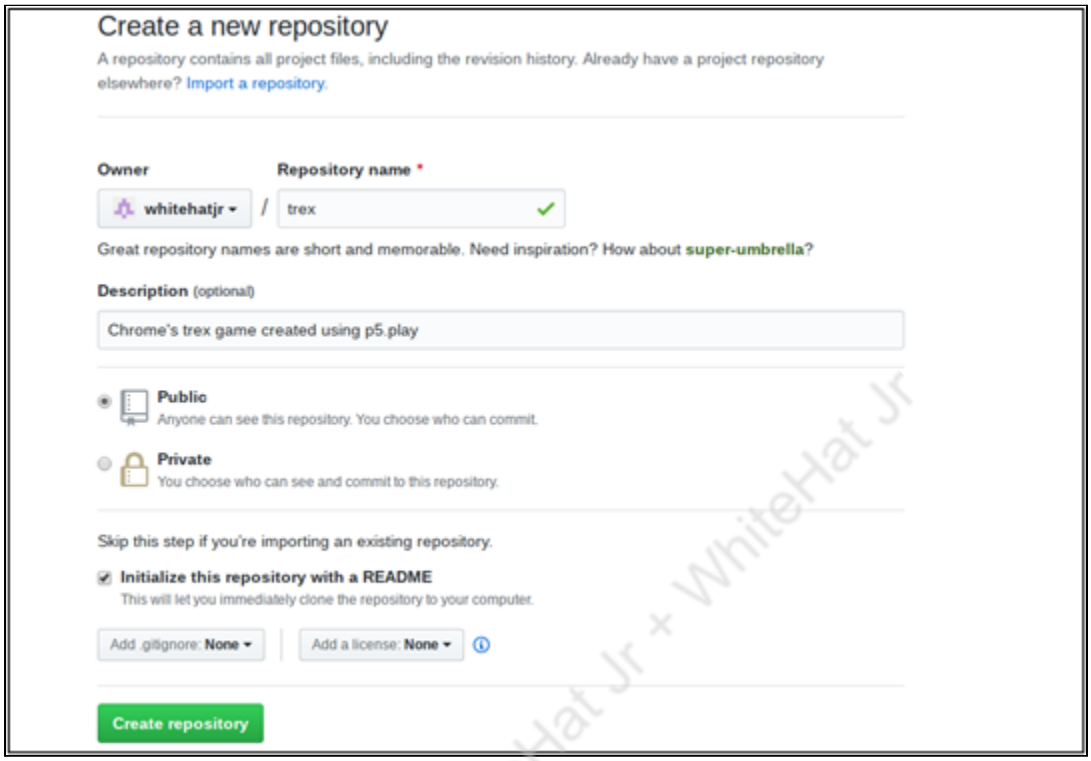


14. Upload the code in GitHub.
    - On the GitHub website, make your account using your email ID.

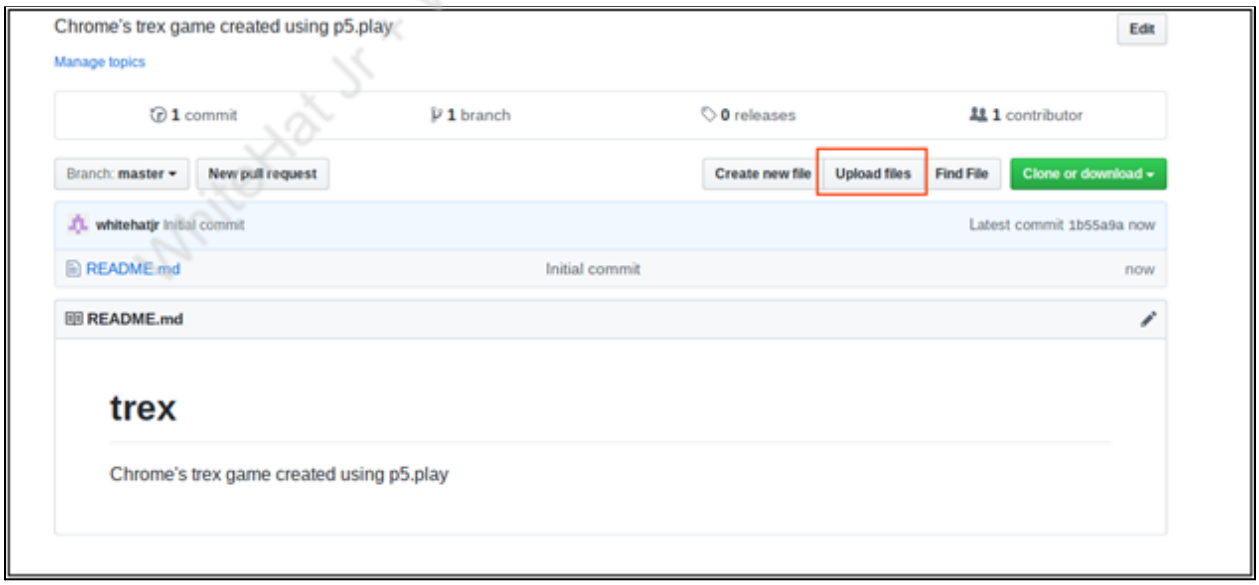


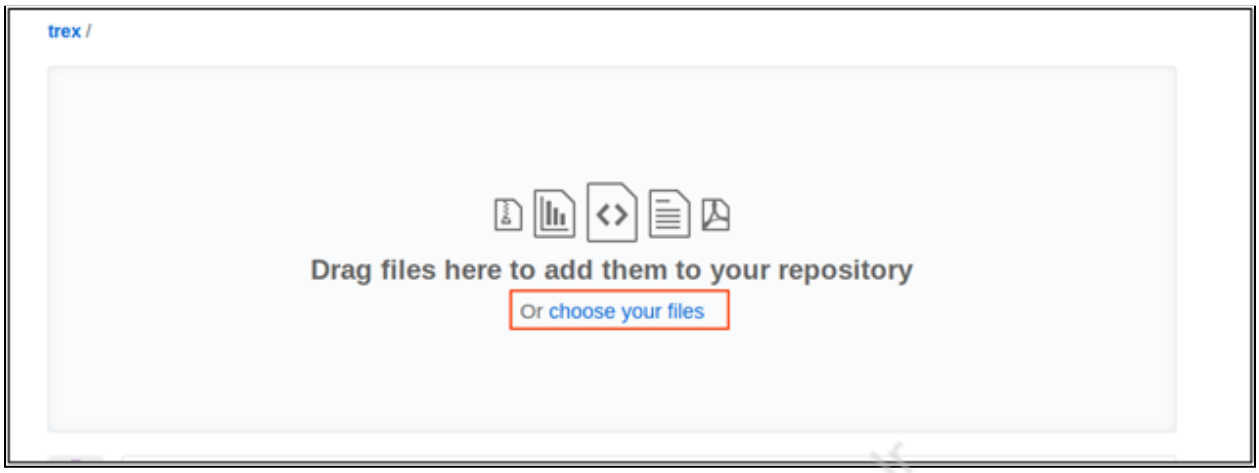
    - Click on start a project.

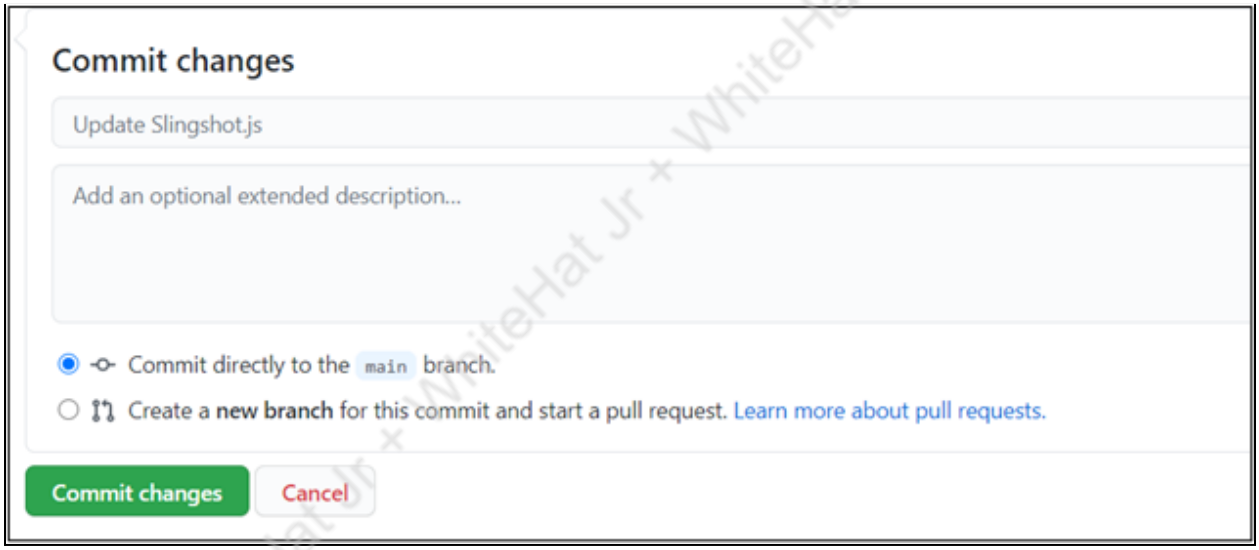
- Create a new repository.



- Upload project files in GitHub.







- We have uploaded the code in GitHub.

What's next?

In the next class, you will learn the concepts of how Infinite Runner Games are created. You will be developing a program in VSC.

EXTEND YOUR KNOWLEDGE:

1. GitHub is very powerful tool for developers; know more about it: https://kinsta.com/knowledgebase/what-is-github/