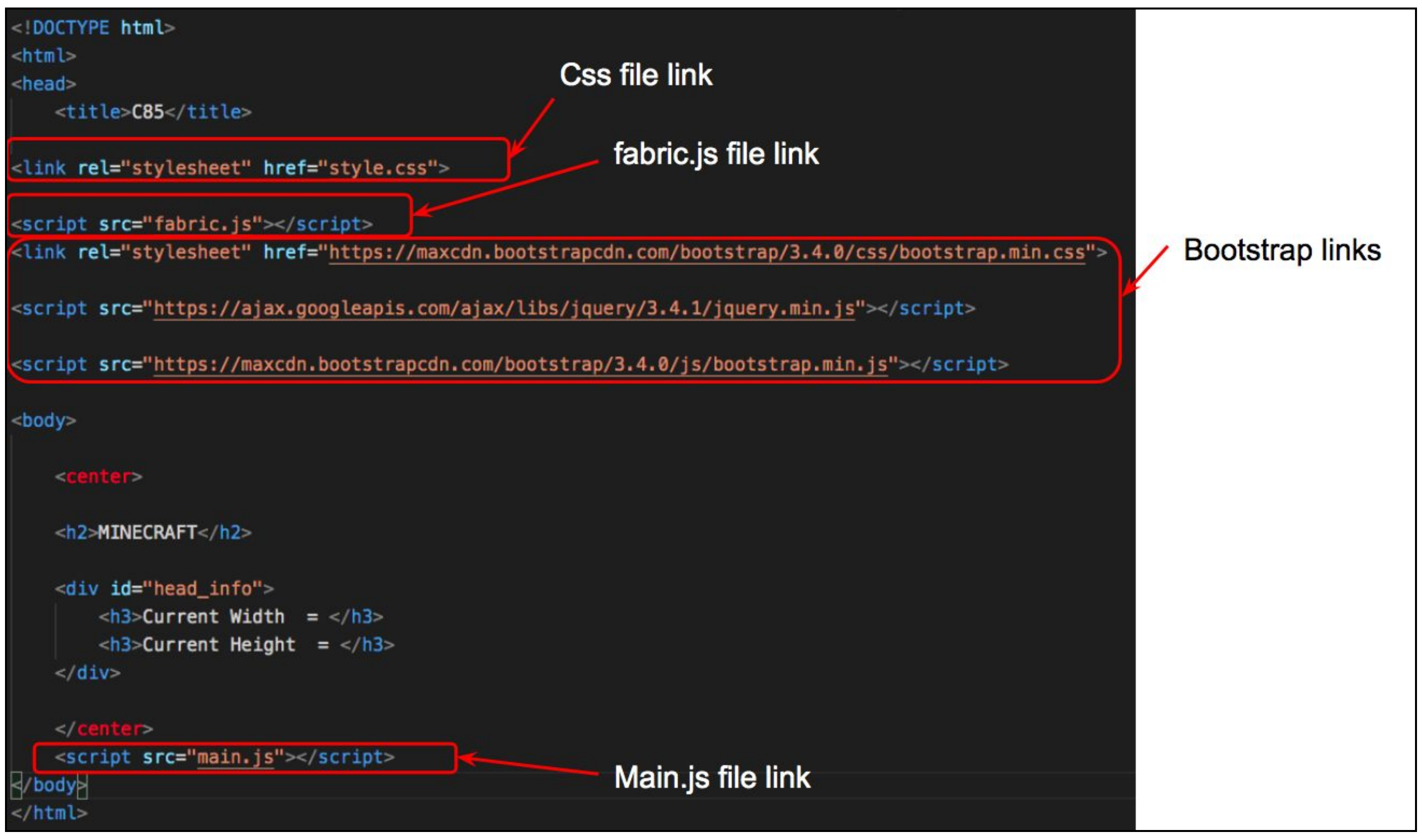
- **Prewritten HTML code**



```html
<!DOCTYPE html>
<html>
<head>
    <title>C85</title>

<link rel="stylesheet" href="style.css">

<script src="fabric.js"></script>
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/css/bootstrap.min.css">

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/js/bootstrap.min.js"></script>

<body>

    <center>

    <h2>MINECRAFT</h2>

    <div id="head_info">
        <h3>Current Width  = </h3>
        <h3>Current Height  = </h3>
    </div>

    </center>
    <script src="main.js"></script>
</body>
</html>
```

The above HTML code had -
- Link for the stylesheet
- Link for the fabric library
- Link for the bootstrap classes
- Body tag
- Center tag
- Some heading tags and div tag
- Link for the main js file

- **Adding class to the body tag -**

```html
<body class="body_background">
```

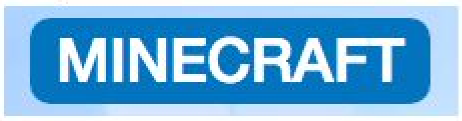
So first you need to add a class to the body tag, so this class will help to get the background image to the body. The CSS is already defined for this class in style.css, you just need to add the class.

- **Adding bootstrap classes to the h2 tag -**

```html
<h2 class="btn-primary">MINECRAFT</h2>
```

btn-primary will add a blue background color and font color white.

Output -

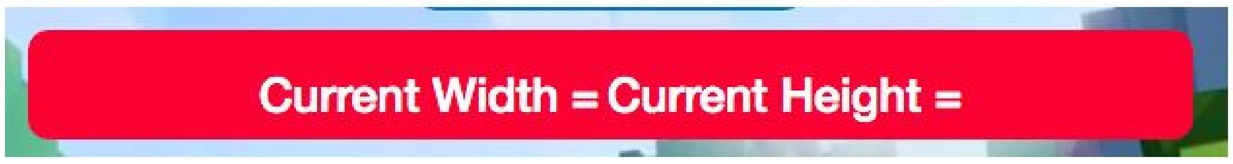


- **Adding bootstrap classes to the h3 tag -**

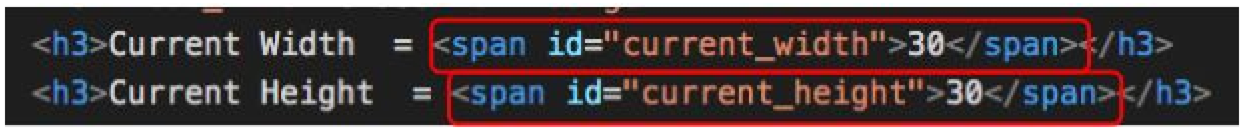
```html
<div id="head_info" class="btn-danger">
```

btn-danger will add a red background color and font color white.

Output -



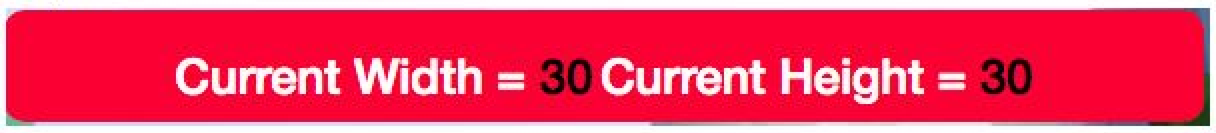
- **Adding 2 span tags inside the h3 tag -**



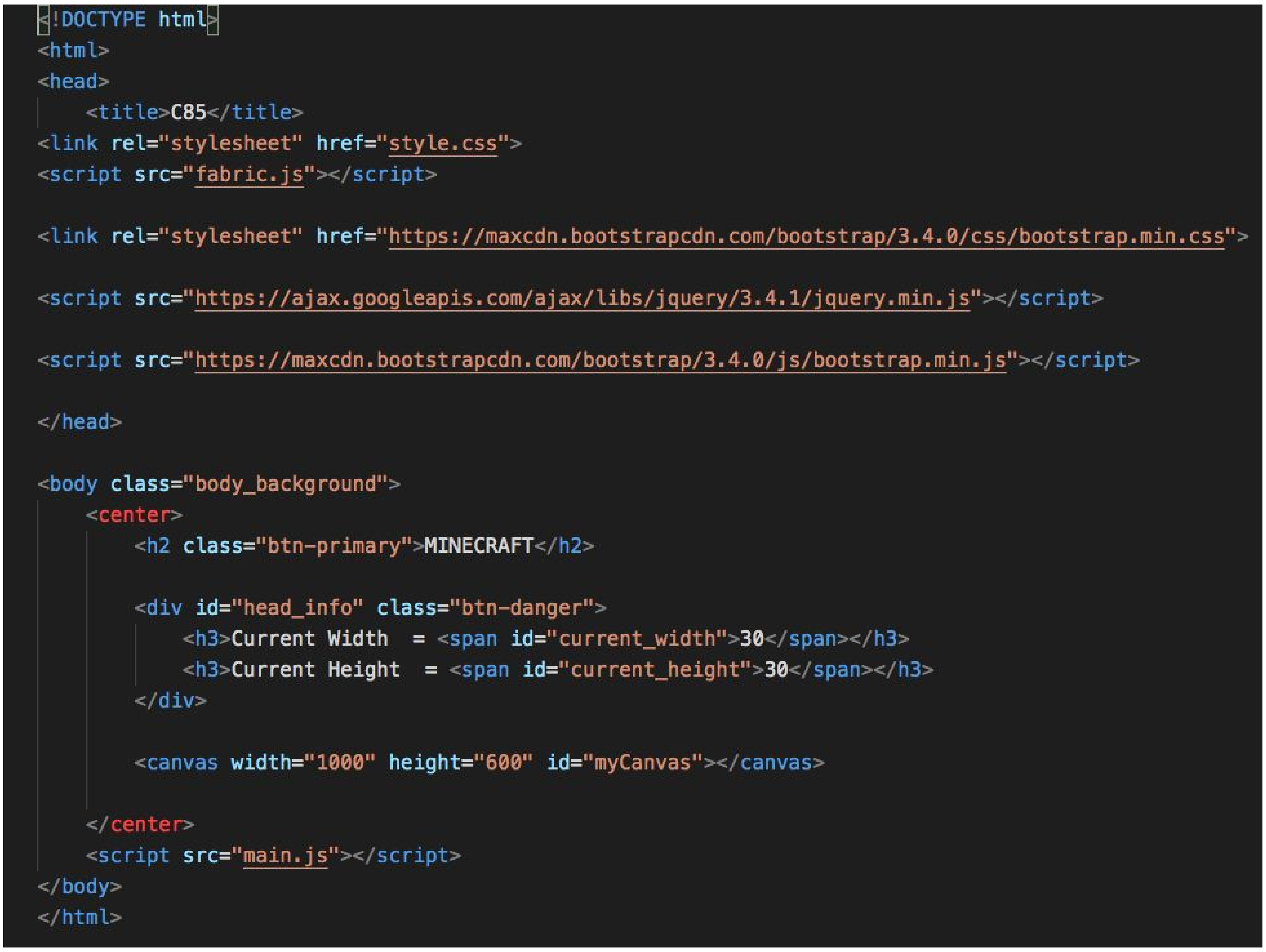
```
<h3>Current Width  = <span id="current_width">30</span></h3>
<h3>Current Height  = <span id="current_height">30</span></h3>
```

We are adding 2 span tags inside the h3 tags and give them id, so this span tag will be used to show on the screen the width and height of the blocks(like wall, trunk, and others), as we increase or decrease them. And set the text as 30 to both of the span tag as we will have the start we will set the height and width of the blocks as 30px

Output -

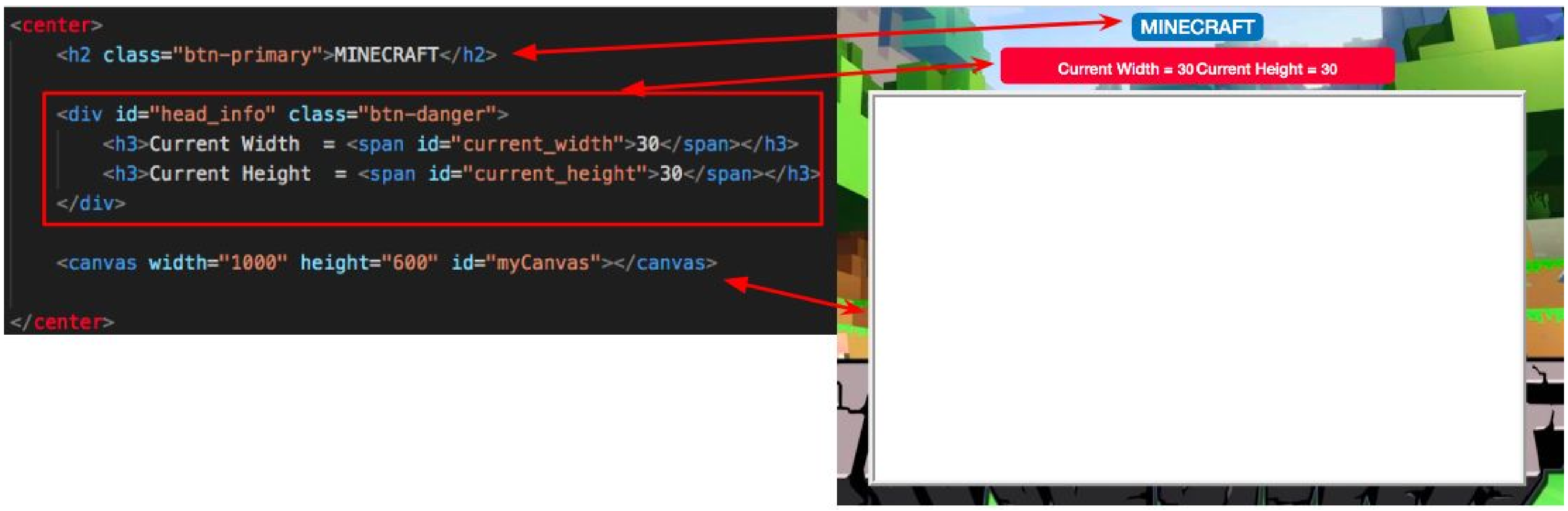


After adding all the HTML elements -



```
<!DOCTYPE html>
<html>
<head>
    <title>C85</title>
<link rel="stylesheet" href="style.css">
<script src="fabric.js"></script>

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/css/bootstrap.min.css">

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/js/bootstrap.min.js"></script>

</head>

<body class="body_background">
    <center>
        <h2 class="btn-primary">MINECRAFT</h2>

        <div id="head_info" class="btn-danger">
            <h3>Current Width  = <span id="current_width">30</span></h3>
            <h3>Current Height  = <span id="current_height">30</span></h3>
        </div>

        <canvas width="1000" height="600" id="myCanvas"></canvas>

    </center>
    <script src="main.js"></script>
</body>
</html>
```

Output -



```html
<center>
    <h2 class="btn-primary">MINECRAFT</h2>

    <div id="head_info" class="btn-danger">
        <h3>Current Width  = <span id="current_width">30</span></h3>
        <h3>Current Height  = <span id="current_height">30</span></h3>
    </div>

    <canvas width="1000" height="600" id="myCanvas"></canvas>

</center>
```

Main.js code -

- **First we will get the reference of the canvas and store it inside a variable**

```js
var canvas = new fabric.Canvas('myCanvas');
```

- **Define the width and height of block image**

```js
block_image_width = 30;
block_image_height = 30;
```

- **Define x and y coordinates of player image**

```js
player_x = 10;
player_y = 10;
```

- **Define a variable to store player object**

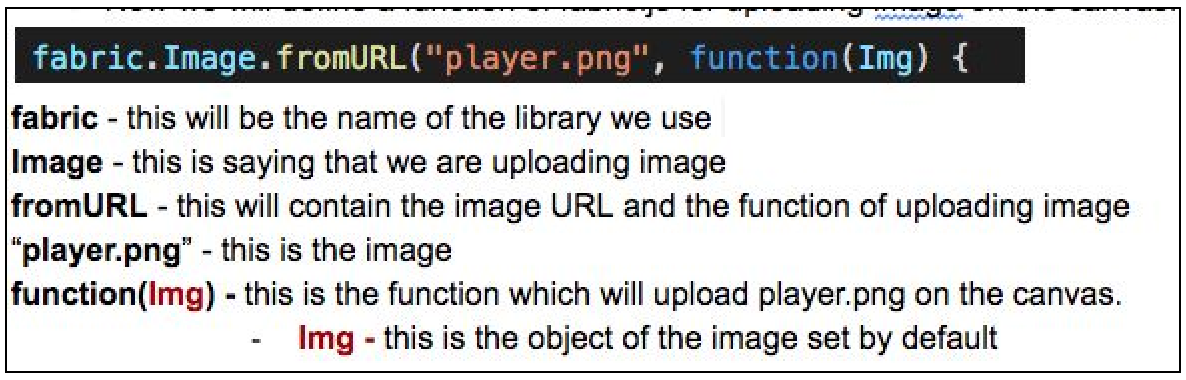
So in fabric.js images are stored in canvas as objects, so when we add an object, we also can delete the same object.
Simply it means we can add and delete images using objects

```js
var player_object= "";
```

- **Function for adding player image**

```js
function player_update()
{
    fabric.Image.fromURL("player.png", function(Img) {
    player_object = Img;

    player_object.scaleToWidth(150);
    player_object.scaleToHeight(140);
    player_object.set({
    top:player_y,
    left:player_x
    });
    canvas.add(player_object);

    });
}
```

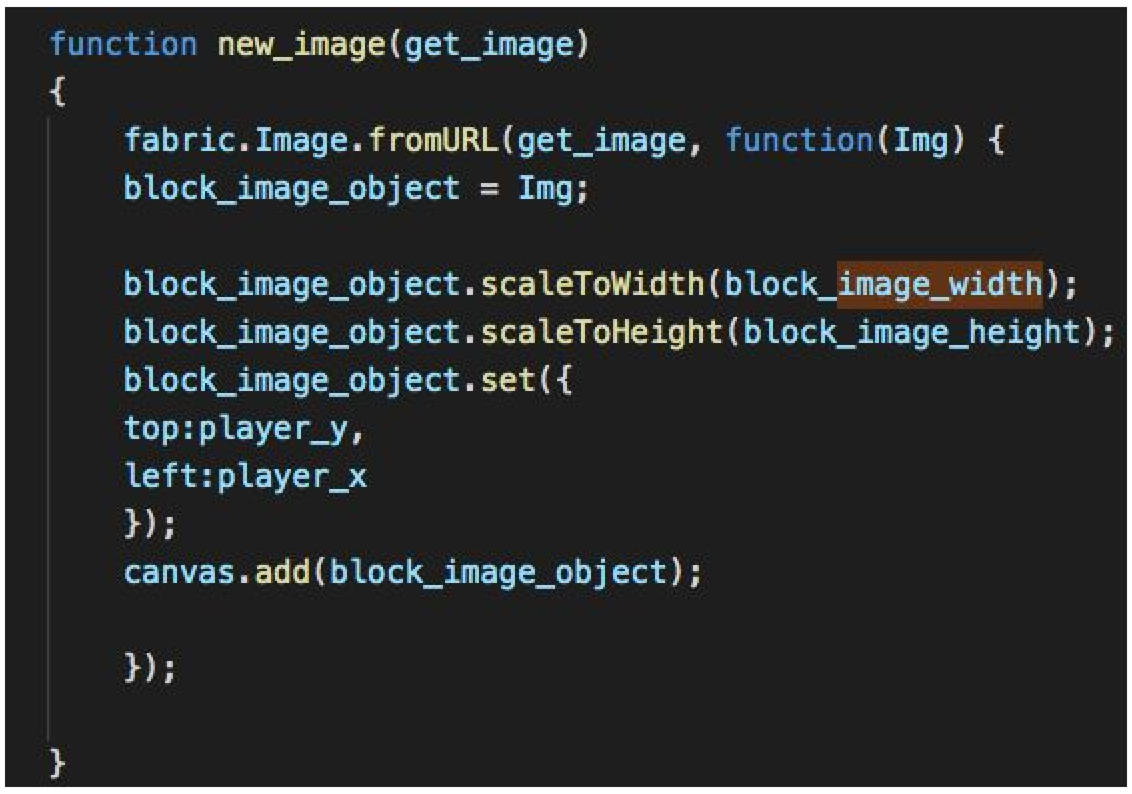
- **Explaining adding image using fabric**



```
fabric.Image.fromURL("player.png", function(Img) {
```

fabric - this will be the name of the library we use
Image - this is saying that we are uploading image
fromURL - this will contain the image URL and the function of uploading image
"**player.png**" - this is the image
function(Img) - this is the function which will upload player.png on the canvas.
- **Img** - this is the object of the image set by default

- **Function for adding block images**



```
function new_image(get_image)
{
    fabric.Image.fromURL(get_image, function(Img) {
    block_image_object = Img;

    block_image_object.scaleToWidth(block_image_width);
    block_image_object.scaleToHeight(block_image_height);
    block_image_object.set({
    top:player_y,
    left:player_x
    });
    canvas.add(block_image_object);

    });

}
```