

Tutorial 1

Author: Kia Kalani

Contributors: Ali Lezzaik, Alireza Teimoori

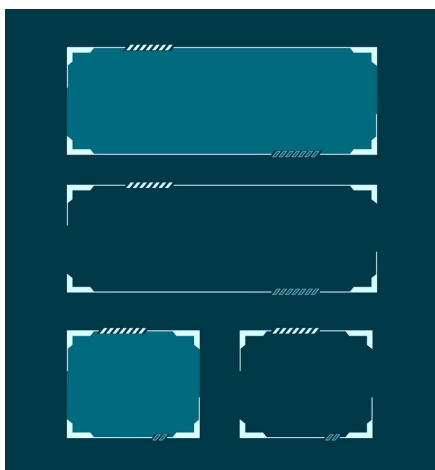
INTRODUCTION:

INTRODUCTION TO p5.js

The main focus of this tutorial is to familiarize you with `p5.js`.

WHAT IS P5JS?

`P5js` is a JavaScript library that is made for making nice looking user interfaces as well as making games. This library is based on `Processing` which is mainly used for designing and making games.



WHAT ARE THE ADVANTAGES OF USING P5JS?

There are many major advantages to using `p5js` for making user interfaces.

The first reason is its simplicity for making user interfaces. The built in functions of this library have the minimal requirements for creating items and rendering it to the webpage.

The second reason is due to being able to operate

Tutorial 1

with no necessity of using servers and yet being able to similar job to some other libraries made for servers.

The last and most important reason is because this library is really lightweight and capable of loading various components in itself including sounds and images.



REFERENCE:

For more information about this convenient JavaScript library please visit: <https://p5js.org/>

GETTING STARTED:

By the end of this tutorial you will be able to draw a circle with a specific diameter and make the circle follow the mouse cursor.

Depending on the programmer's taste, this library can be loaded in two different ways; Offline or Online. For this series of tutorial, the library would be loaded online; however, in case of interest of keeping everything functional while offline, please visit: <https://p5js.org/download/> and download the file that works the best with your preference.

MAKING THE PROJECT:

For making a `p5js` project, open a destination that would contain the project. In our case the directory is called `tutorial1`.

Tutorial 1

Let's get started with the HTML portion of the code. In our case we have no additional components rather than the `p5js` components and therefore our HTML file would only include the scripts loaded as well as a color that would avoid hurting eyes. In this case, our HTML file is called `index.html`. Inside this file, we need the normal skeleton of HTML with loading the scripts and setting the background color of the HTML file:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Tutorial 1</title>
</head>
<body>
  <style>
    body{
      background-color: grey;
    }
  </style>
</body>
<script src="sketch.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/0.10.2/p5.js">
</script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/0.10.2/addons/p5.sound.m
n.js"></script>
</html>
```



Note: It is best to load the scripts at the end of the body as by that point all of the elements inside the body are defined if any. For avoidance of making too many files the css styling is being written straightly inside the HTML.

Tutorial 1

There are two scripts that are needed for this project:

One of them is p5js itself which would be loaded with:

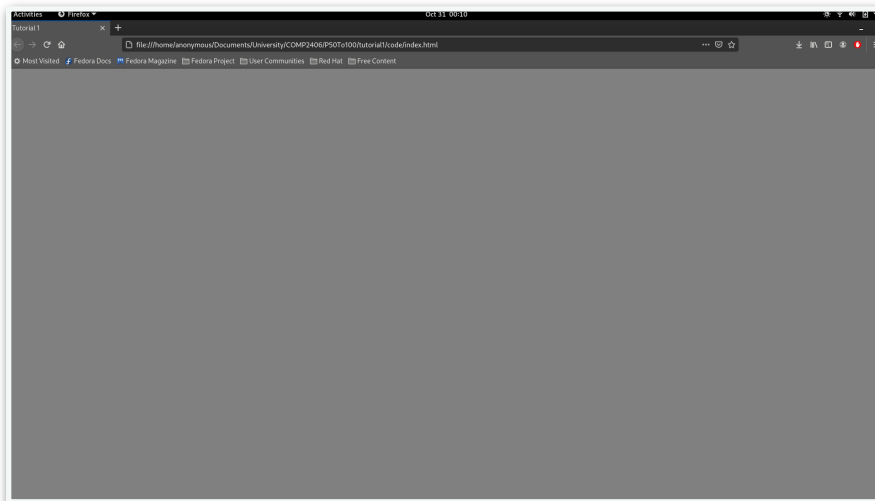
```
<script src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/0.10.2/p5.js"> </script>
```

and the sound p5.sound library which is loaded by:

```
<script src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/0.10.2/addons/p5.sound.min.js" ></script>
```

Finally a JavaScript file called sketch.js is needed to get started with making a p5js project.

The output by viewing the HTML file:



By opening the HTML file it is noticeable that excluding the color nothing happens. That is because the `sketch.js` file which is responsible for drawing the components is not defined yet.

Now let's start by making `sketch.js` file for getting started with the view itself. There are two functions necessary for getting the project to start working.

The first one is `function setup()` which would initialize the view for us and the second one is `function draw()` which is responsible for rendering the components for us. After making those two functions there is a built in method called `createCanvas(width: , height:)` which takes width and height of the canvas as parameters.

Finally, it is crucial to choose a background color within the `draw()` function. This can be done by calling the built in method `background(ColorValue:)`. Therefore in order

Tutorial 1

to make a canvas visible in the preview, we need to put all of these components together inside the `sketch.js` file. The file should look something like this:

```
const WIDTH = 1024;
const HEIGHT = 768;

function setup() {
  createCanvas(WIDTH, HEIGHT);
}

function draw() {
  background(100);
}
```

By reloading the `index.html` file you will notice that there is a darker canvas created on top of HTML with the provided width and height.

Now let's make a circle that would follow the mouse position. For doing so, there is a built in method called `circle(x: , y: , diameter:)` which would get x,y and the diameter of the circle as parameters and would render a circle in case of being called inside the `draw()` method.

For the purpose of getting the x and y position of the mouse, there are built in variables called `mouseX` and `mouseY` which represent the x position of the mouse and y position of the mouse inside the canvas.

Therefore, for the purpose of drawing a circle that would constantly change its position according to x and y position of the mouse, we put the x and y position of the mouse as x and y position of the circle and give some value to the diameter.

In code, it should look something like this: `circle(mouseX, mouseY, diameter)` where diameter should be set to a value preferred by the programmers. Therefore by putting everything together in `sketch.js` it should be similar to this:

Tutorial 1

```
const WIDTH = 1024;

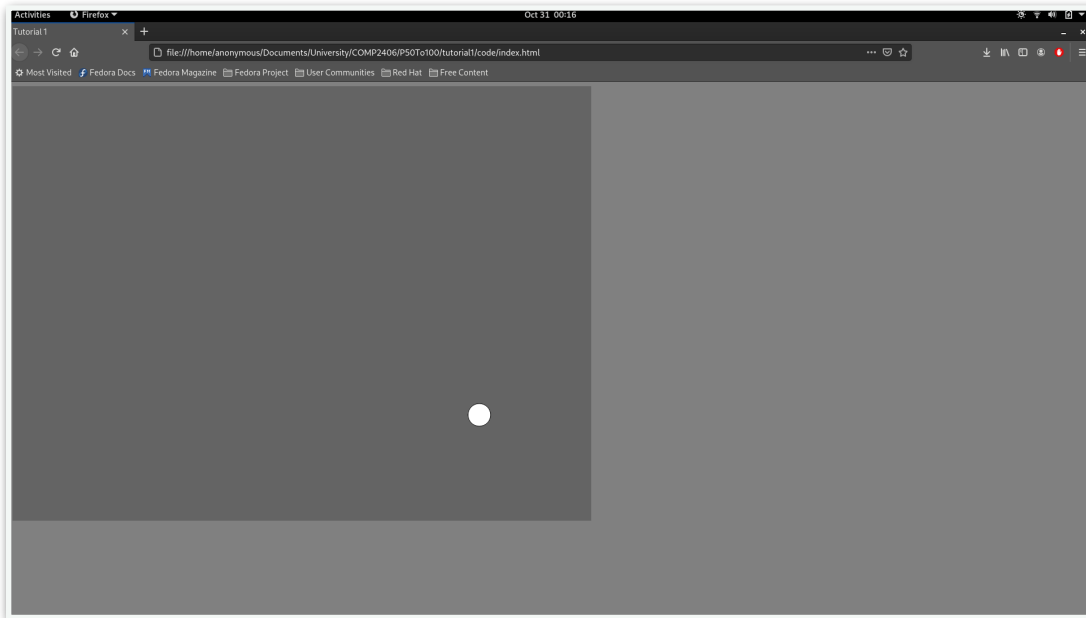
const HEIGHT = 768;

var radius = 40;

function setup() {
  createCanvas(WIDTH, HEIGHT);
}

function draw() {
  background(100);
  fill(255, 255, 255);
  circle(mouseX, mouseY, radius);
}
```

At last, by opening the `index.html` file you would notice that there would be a circle following the mouse.



This sums up the first tutorial!

For getting access to the source code please visit: <https://github.com/kiakalani/P5JSTutorial/tree/main/tutorial1/code>