



Mouse Interaction

LO: IWBAT to use P5-JS to create a program that requires mouse input.

Notice and Wonder

```
1 function setup(){  
2   createCanvas(400, 300)  
3 }  
4  
5 function draw(){  
6   background(0);  
7  
8   noStroke();  
9   fill(255);  
10  ellipse(mouseX, 150, 75, 75)  
11 }
```

Enter the following code into P5-JS and save as "Mouse".

What do you notice and wonder about this program?

Best Guess

Take a look at the following code without putting it into your editor.

What do you think will happen?

```
1 ▼ function setup(){  
2   createCanvas(400, 300);  
3   background(0);  
4 }  
5  
6 ▼ function draw(){  
7  
8   noStroke();  
9   fill(255);  
10  circle(mouseX, mouseY, 25)  
11 }
```

mouseX and mouseY



mouseX

The system variable `mouseX` always contains the current horizontal position of the mouse, relative to (0, 0) of the canvas. The value at the top-left corner is (0, 0) for 2-D and (-width/2, -height/2) for WebGL. If touch is used instead of mouse input, `mouseX` will hold the x value of the most recent touch point.

mouseY

The system variable `mouseY` always contains the current vertical position of the mouse, relative to (0, 0) of the canvas. The value at the top-left corner is (0, 0) for 2-D and (-width/2, -height/2) for WebGL. If touch is used instead of mouse input, `mouseY` will hold the y value of the most recent touch point.

Try This

```
1 ▼ function setup() {  
2   createCanvas(400, 400);  
3   noStroke();  
4 }  
5 ▼ function draw() {  
6   background(126);  
7   ellipse(mouseX, 100, 100, 100);    // Top  
circle  
8   ellipse(mouseX+20, 200, 100, 100); //  
Middle circle  
9   ellipse(mouseX-20, 300, 100, 100); //  
Bottom circle  
10 }
```

Run the following program and see what it does. When done, create a program where:

- Bottom circle is fastest
- Middle circle is slowest

Exit Ticket

Create a program where when you drag the mouse across horizontally, 2 circles will cross each other perpendicularly.