

# mouseY

A `Number` system variable that tracks the mouse's vertical position.

`mouseY` keeps track of the mouse's position relative to the top-left corner of the canvas. For example, if the mouse is 50 pixels from the top edge of the canvas, then `mouseY` will be 50.

If touch is used instead of the mouse, then `mouseY` will hold the y-coordinate of the most recent touch point.

## Reference

### Mouse

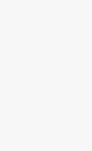
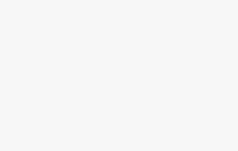
`doubleClicked()`  
`exitPointerLock()`  
`mouseButton`  
`mouseClicked()`  
`mouseDragged()`  
`mouseIsPressed()`  
`mouseMoved()`  
`mousePressed()`  
`mouseReleased()`  
`mouseWheel()`  
`mouseX`  
`mouseY`  
`movedX`  
`movedY`  
`pmouseX`  
`pmouseY`  
`pwinMouseX`  
`pwinMouseY`  
`requestPointerLock()`

## Examples



```
function setup() {
  createCanvas(100, 100);

  describe("A horizontal black line moves up and down
following the mouse's y-position.");
}
```



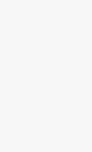
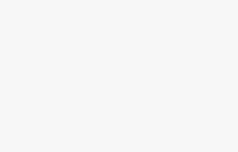
```
function draw() {
  background(200);

  // Draw a horizontal line that follows the mouse's y-
  coordinate.
  line(0, mouseY, 100, mouseY);
}
```



```
function setup() {
  createCanvas(100, 100);

  describe("A gray square. The mouse's x- and y-coordinates
are displayed as the user moves the mouse.");
}
```



```
function draw() {
  background(200);

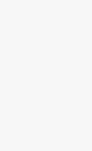
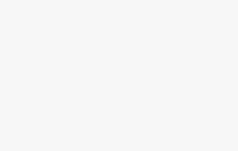
  // Style the text.
  textAlign(CENTER);
  textSize(16);

  // Display the mouse's coordinates.
  text(`x: ${mouseX} y: ${mouseY}`, 50, 50);
}
```



```
function setup() {
  createCanvas(100, 100, WEBGL);

  describe("A horizontal black line moves up and down
following the mouse's y-position.");
}
```



```
function draw() {
  background(200);

  // Adjust coordinates for WebGL mode.
  // The origin (0, 0) is at the center of the canvas.
  let my = mouseY - 50;
}
```

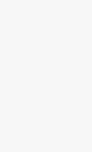
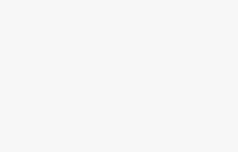


```
// Draw the line.
line(-50, my, 50, my);
```



```
let font;

// Load a font for WebGL mode.
function preload() {
  font = loadFont('/assets/inconsolata.otf');
}
```



```
function setup() {
  createCanvas(100, 100, WEBGL);

  describe(
    "A gray square. The mouse's x- and y-coordinates are
    displayed as the user moves the mouse."
  );
}
```



```
function draw() {
  background(200);

  // Style the text.
  textAlign(CENTER);
  textSize(16);
  textFont(font);
  fill(0);

  // Display the mouse's coordinates.
  text(`x: ${mouseX} y: ${mouseY}`, 0, 0);
}
```

This page is generated from the comments in [src/events/mouse.js](#). Please feel free to edit it and submit a pull request!

## Related References

### doubleClicked

A function that's called once when a mouse button is clicked twice quickly.

### exitPointerLock

Exits a pointer lock started with `requestPointerLock`.

### mouseButton

A String system variable that contains the value of the last mouse button pressed.

### mouseClicked

A function that's called once after mouse button is pressed and released.

## p5.js

### Resources

[Reference](#)

[Tutorials](#)

[Examples](#)

[Contribute](#)

[Community](#)

[About](#)

[Start Coding](#)

[Donate](#)

### Information

[Download](#)

[Contact](#)

[Copyright](#)

[Privacy Policy](#)

[Terms of Use](#)

### Socials

[GitHub](#)

[Instagram](#)

[X](#)

[YouTube](#)

[Discord](#)

[Forum](#)

