README.md

# 5 Basic Input

## 5.1 Mouse Coordinates

Just like p5.js automatically runs `draw()` over and over again, it automatically creates and updates several variables for you. One of the most useful and simple to use of these variables are the mouse position variables, as seen below. Any of these names/labels can be used in place of a number in a formula or in the place of a parameter for a shape.

| Name | Description |
|---|---|
| `mouseX` | Current horizontal position for mouse (pixels between mouse and left edge) |
| `mouseY` | Current vertical position for mouse (pixels between mouse and top edge) |
| `pmouseX` | Previous horizontal position for mouse |
| `pmouseY` | Current vertical position for mouse |

Try it!

- `ellipse(mouseX, mouseY, 50, 50); // draw a circle where the mouse is`
- `line(mouseX, mouseY, pmouseX, pmouseY); // draw a line from your previous position to your current position`

## 5.2 Booleans and Events

When the user performs an action that occurs at one discrete moment, that action is called an **event**. Some common input events are clicking the mouse, pressing a key, releasing a key, etc. There are two ways to detect events in p5.js: a boolean variable and an event function. For the following explanations, we'll be using the variables and event functions for the mouse. We'll cover keys later.

### 5.2.1 Boolean Variables

A boolean variable is a variable that holds a value that is only either `true` or `false`. For input, p5.js tracks and updates several variables for you, the way that is updates mouse coordinates. One such variable is `mouseIsPressed`. `mouseIsPressed` will contain `true` if a mouse button is pressed down and `false` if no mouse buttons are pressed.

These variables are most commonly used with if-statements, as you can see with the basic example below.

```
if (mouseIsPressed) {
  background(0, 255, 0);
} else {
  background(255, 0, 0);
}
```

This is a really good time to use the ! boolean operator. Writing `!mouseIsPressed` will return true when the mouse is not pressed down. This will be useful, I promise, you'll see!

### 5.2.2 Event Functions

An event function is a function like `setup()` and `draw()`. They're written in a very similar format and only fire when their name occurs. For example:

```
function mousePressed() {
  // code only runs the moment the mouse is pressed
  // only once as well, will not loop
}
```

These functions are very useful and interesting because they happen immediately, regardless of where in `draw()` your code is currently executing.

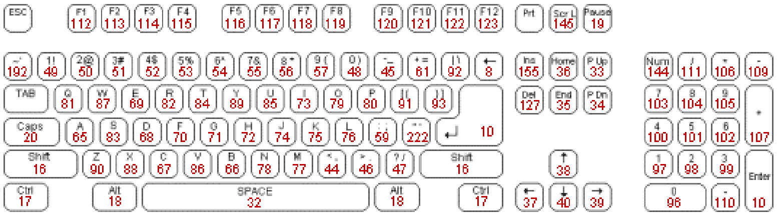
## 5.3 Key Presses

The details:

| Type | Syntax |
|---|---|
| Boolean variable | `keyIsPressed` |
| Event function | `function keyPressed() {` |

### 5.3.1 key and keyCode

When you press keys on the keyboard, p5.js saves the most recent letter to `key` and the most recent key code to `keyCode` .



Here is a sketch to reference what you're looking for: keyCode printer.