

# Introduction to Processing

Class #7: 10/31/2007

Taught by Ms. Madsen, Ms. Huhn & Ms. Yen

Course website: >> [web.mit.edu/mish/www/processing](http://web.mit.edu/mish/www/processing)

Processing website: >> [www.processing.org](http://www.processing.org)

Email the teachers: [mish@mit.edu](mailto:mish@mit.edu), [ahuhn@mit.edu](mailto:ahuhn@mit.edu), [cyyen@mit.edu](mailto:cyyen@mit.edu)

On Page 2:

**Making colors partially transparent.**

**What if you want to have another shape moving from bottom to top?**

**What if you want to draw a line wherever you drag your mouse?**

**What if you want to draw shapes in random places?**

On Page 3:

**What if you want to turn *loop()* on and off?**

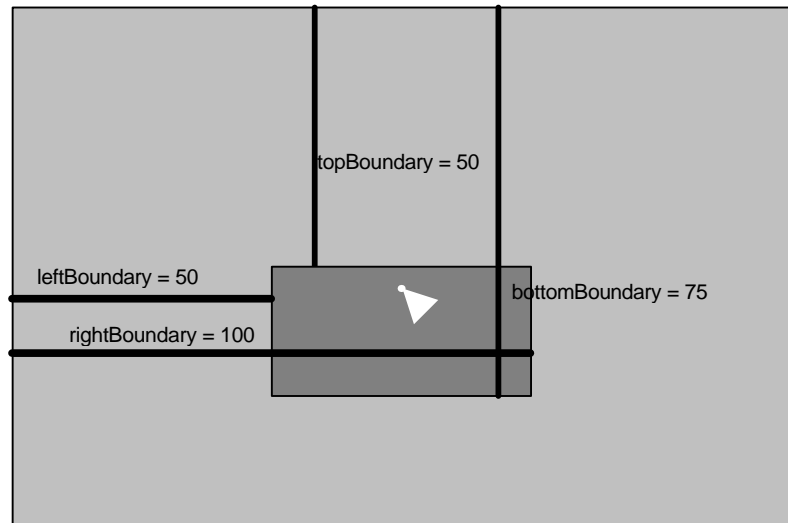
**What if you want to add text?**

**What if you want something to increase if you click a button & decrease if you click it again?**

Think of your button as something with a left boundary, a right boundary, a top boundary, and a bottom boundary.

You want:

```
mouseX > leftBoundary
mouseX < rightBoundary
mouseY > topBoundary
mouseY < bottomBoundary
```



```
int leftBoundary = 50, rightBoundary = 100, topBoundary = 50, bottomBoundary = 75;
// your button will have an xStart=50, width=50, yStart=100, and height=25

void draw(){
  fill(buttonColor); // use size() and specify some buttonColor first
  rect(leftBoundary, topBoundary, rightBoundary - leftBoundary, bottomBoundary - topBoundary);
}

void mousePressed(){
  if ((mouseX > leftBoundary) && (mouseX < rightBoundary) &&
      (mouseY > topBoundary) && (mouseY < bottomBoundary)) {
    runSomeCodeHere;
  }
}
```

### Making colors partially transparent:

You can make colors partially transparent!

When you're using the *fill* command, you give a *percentage* (0-100) of full color.

0 is totally transparent; 100 is totally opaque (opposite of transparent); 50 is halfway transparent.

```
color newColor = color(0,0,255);  
fill(newColor, 80); // this will also be 80% opaque, 20% transparent
```

### What if you want to have another shape moving from bottom to top?

You should make another integer variable outside of your methods, `distanceFromTop2`, and set that equal to the height of your window at the beginning. Then, add something like this to `draw()`, and don't forget to add `loop()` as well so that you're in loop mode:

```
rect(xStart, distanceFromTop2, width, height);  
distanceFromTop2--;
```

You can add another *if* statement (like the one we added before) to make sure it doesn't go off the top of the screen. (Why don't we have to worry about it going off the bottom of the screen?)

```
if (distanceFromTop2 < 0) {  
    distanceFromTop2 = height;  
}
```

### What if you want to draw a line wherever you drag your mouse?

You can add the following to your `draw()` method if you're in `loop()` mode. (If you want, you can change the `strokeColor` with an *if* statement by using a button, or just by something like `redValue++` added to your `draw()` method: `redValue` will increase during each loop.)

```
stroke(strokeColor); // make a strokeColor first  
if (mousePressed) {  
    line(mouseX, mouseY, pmouseX, pmouseY);  
}
```

### What if you want to draw shapes in random places?

Try adding this to your `draw()` method. (You'll want to have `noLoop()` in your `draw()` method – you can't have both `noLoop()` and `loop()`, and for this you can't be looping.) `random(x,y)` will give you a random number between `x` and `y`. In the following example, you can replace **0**, **width**, or **height** to get numbers in a smaller range, or you could get random colors with `color(random(0,255), 0, 0)`, for instance.

```
rect(random(0,width), random(0,height), 10,10);
```

If you want to use `random(x,y)` with `loop`, change `noLoop()` to `loop()` and add this to `draw()`:

```
background(random(0,255), random(0,255), random(0,255));
```

(Don't forget that you can use `frameRate` if you want to change the speed of your looping.)

### What if you want to turn `loop()` on and off?

Put a "boolean `inLoop = true;`" statement outside of your methods, and then add the following inside "`mousePressed()`" or inside a button within `mousePressed()`:

```
if (inLoop){
    noLoop();
    inLoop = false;
} else {
    loop();
    inLoop = true;
}
```

### What if you want to add text?

In the 'Tools' menu, use **Create Font** to make a font. Make the filename (at the bottom of the window) something like **newFont**. Use the following commands before you create any text, with `fontSize` being some number like 14 or 16:

```
PFont myFont = createFont("newFont", fontSize);
textFont(myFont);
```

At whatever integers `xStart` and `yStart` you want to put your text, use the command:

```
fill(textColor); // make a color textColor first
text("Hello world", xStart, yStart); // use integers for xStart and yStart
```

You can replace "hello world" with any string, such as: "Ms. Madsen is awesome!"

### What if you want to have something increase if you click a button, and decrease if you click it again?

The "something" that you're changing could be an RGB value in a color, or it could be the position of a shape, or it could be the size of a shape. Whatever it is, have some integer variables **changingVariable** and **amountToAdd** outside of your methods.

You should also create some *boolean* variable outside of your methods: you can call it **increaseTheVariable**, and you should set it equal to **true** initially.

```
int changingVariable = 50, amountToAdd = 1;
boolean increaseTheVariable = true;

void draw(){
    // use changingVariable here, in a color statement or a shape statement
    changingVariable = changingVariable + amountToAdd;
}
```

Then you can put the following code inside the *if* statement of some button:

```
if (increaseTheVariable){
    amountToAdd = -1;
    increaseTheVariable = false;
} else {
    amountToAdd = 1;
    increaseTheVariable = true;
}
```