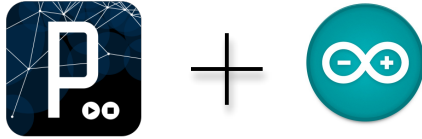


# Programming basics in Processing

Arduino + Processing workshop 1

Esteban Garcia



Before we start using the Arduino inputs as variables in Processing, let's review some basic programming elements of Processing. In processing the syntax is almost identical to Java, but processing adds special features related to **graphics** and **interaction**.

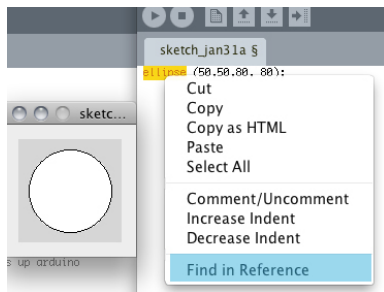
Let's look at the following expression:

*name of function*

`ellipse (50, 50, 80, 80);`

*parameters*

This processing function draws an ellipse on the screen using the points (50,50) as center and a dimension of 80 by 80. You don't need to memorize all the processing functions, but you will become familiar with them as you use them. The important part is that you can find the meaning of the syntax in the processing reference:



to find the syntax of a function select it and right click. Choose "Find in reference."

This will open the processing reference website and explain the function in detail.

From the reference we learn that:  
`ellipse(x, y, width, height);`

Functions are compounds of commands that run when they are called by their name. For example "ellipse" however we can also create our own compounds of procedures. For example:

*defines function*

```
void lots() {
  ellipse (50,50,80, 80);
  ellipse (20,20,80, 80);
}
```

`lots();` *calls function*

Let's try the following code in Processing:

```
void lots() {
  ellipse (50,50,80, 80);
  ellipse (20,20,80, 80);
}

void draw() {
  lots();
}
```

The code calls the function "lots" and draws 2 ellipses in the order described. Notice that "draw" is also a function, and so is "setup."

## Objects

Another of the elements that is important in programming is the concept of "object." Think about an object for example a pencil:



The pencil:

- Is yellow
- It's 7 inches
- Has an eraser

In the same way, every "object" in programming has different properties. Objects in processing can be images, movies, sounds, etc.

In programming we can "call" the characteristics (or properties) of the objects and use them as variables. For example the following code:

```
PImage b; //creates the object
b = loadImage("mar.png"); // assigns a file to that image object

image(b, 0, 0); // the function image draws "b" in the screen point
(0,0)
```

We can also explore more in the reference to learn about the properties of each object. In the case of the image, the syntax explains:

```
image(img, x, y, width, height)
```

To access the object's properties we use "." To refer to that object for example the width of image b is noted as "**b.width**"

Experiment in processing bringing images and modifying their properties using the following code. Remember to save image in the "data" folder of the same sketch.

```
PImage a; //creates the object a

void setup() {
  size(800, 600);
  a = loadImage("name of your image.png");
}

void draw() {
  image(a, 400, 300, a.width/mouseX, a.height/mouseY);

}
```