Processing Cheat Sheet – INTERACTION!

setup() and draw() functions

Almost every Processing program has two main functions: setup() and draw(). A function is a chunk of code that does a specific job. The code is contained in curly brackets under the function name.

setup() runs once at the start of the program. Use this to set the screen size, set the background color, etc.

draw() loops endlessly while the program is running. Use this to make changes to values, animate, respond to user input, etc.

```
void setup(){
// Put code here to run at the start of the program
}

void draw(){
//Put code here to run continuously throughout the program
}
```

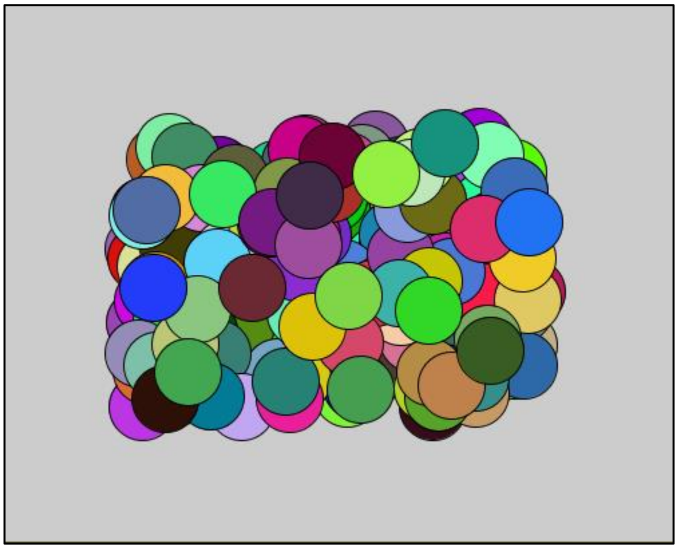
random()

random(x) – generates a random number between 0 and x.
random(min, max) – generates a random number between min and max

These can make things move, appear, change color, etc. randomly and wildly, when used in the draw() function.

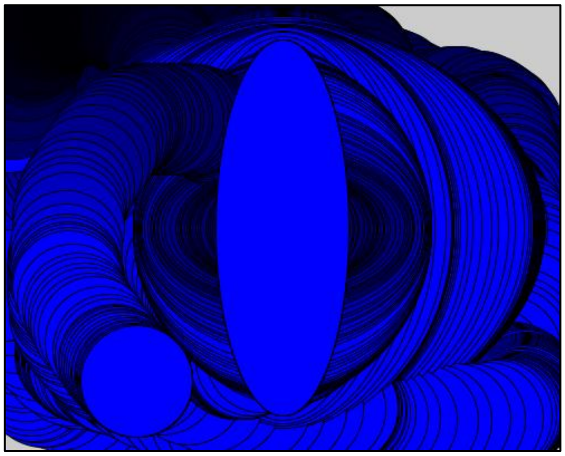
They can be used anywhere instead of a number.

For example:



```
void setup(){
size(500, 400);
}

void draw(){

fill(random(255), random(255), random(255));     //fills the next shape with a
                                                 //totally random RGB color

ellipse(random(100, 400), random(100, 300), 50, 50);   //draws a (50, 50) ellipse
                                                       //at a random x-position
                                                       //between 100 and 400, and
                                                       //a random y-position
                                                       //between 100 and 300
}
```

Mouse Position



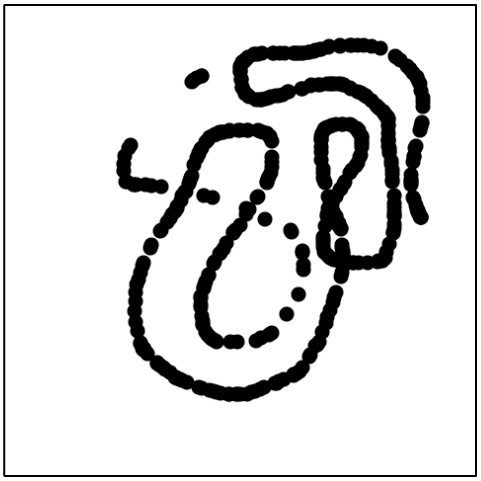
mouseX – returns the mouse's horizontal position
mouseY – returns the mouse's vertical position

These are typically used to make things follow the mouse, but you can use them anywhere instead of a number.

For example:

```
void setup(){
size(500, 400);
}

void draw(){

ellipse(250,  200,  100, 100);         //Draws a 100x100 circle at position
                                       //(250, 200)

ellipse(mouseX, mouseY, 100, 100);     //Draws a 100x100 circle wherever the mouse
                                       //is

ellipse(250, 200, mouseX, mouseY);     //Draws an ellipse at position (250, 200),
                                       //which grows fatter as the mouse moves
                                       //right, and taller as the mouse moves down

fill(0, 0, mouseY);                    //Changes the fill of the next shape to go
                                       //from black to blue, as the mouse moves
                                       //down
}
```

Mouse Click



You can detect whether the mouse button is down using a simple 'if statement,' **if (mousePressed)** , followed by curly brackets encasing what will happen when the mouse button is down.

For example:

```
void setup()
{
  size(500, 500);
  background(255);
}

void draw()
{
  if (mousePressed) {                  //If the mouse button is pressed…
    fill(0);                           //…then pick up the black pen
    noStroke();                        //…with no outline…
    ellipse(mouseX, mouseY, 15, 15);   //…and draw a little black circle wherever
                                       //the mouse is.
  }
}
```