**Name: gravity(g)**

**Examples:**

float x;

float y;

float diam;

float v;

void setup(){

size(800,600);

background(0);

x = width/2;

y = height/2;

diam = 50;

v = random(-7,7);

}

void draw(){

background(0);

stroke(20,50,20);

fill(20,50,20);

gravity(.1);

ellipse(x,y,diam,diam);

if(y + diam/2 > height){

y = height - diam/2;

v = -v;

}

}

void gravity(float g){

//add gravity to velocity

v += g;

//add velocity to the ball's y-position

y += v;

}

**Description:**

Creates a simple gravity formula for use on moving objects with a velocity (v) previously defined in the function. Each frame, the gravity is added to the velocity, and the velocity determines the change in the height of the object relative to the bottom of the canvas; this creates the illusion of gravity acting on the object, as it accelerates the father it falls and begins to decelerate if its velocity reverses due to a “bounce” off of a floor.

**Syntax:**

gravity()

**Parameters:**

g = gravity (the factor by which the velocity is modified with each frame)

**Returns:**

void

**Other notes:**

The function requires a velocity for an object (given the variable “v” in the above example)

**Name: crosshair()**

**Examples:**

float x;

float y;

float diam;

float v;

void setup(){

size(800,600);

void draw(){

background(0);

crosshair(50,mouseX,mouseY,15,5,25,150,200);

}

void crosshair(float d, float xi, float yi, float l, float w, float r, float g, float b){

stroke(r,g,b);

fill(r,g,b);

ellipse(xi,yi,d/4,d/4);

rectMode(CENTER);

rect(xi, yi + l + w, w, l);

rect(xi, yi - l - w, w, l);

rect(xi + l + w, yi, l, w);

rect(xi - l - w, yi, l, w);

}

**Description:**

The crosshair() function is used to create a set of crosshairs centered on a specified location, usually at the location of the cursor. It allows one to specify the dimensions of both the central reticle of the crosshair and the bars, including allowing one to completely remove either.

**Syntax:**

crosshair()

crosshair(d,xi,vi,l,w,r,g,b)

**Parameters:**

d = diameter (of central ellipse)  
xi = x position of crosshair

yi = y position of crosshair

l = length (the crosshair’s bars)

w = width (of crosshair’s bars)

r = red value (of color)

g = green value (of color)

b = blue value (of color)

**Returns:**

void

**Name: fortyTwo()**

**Examples:**

void setup(){

}

void draw(){

fortyTwo(900);

}

Void fortyTwo(){

println("What do you get when you multiply six by nine?");

num /= num;

num = num \* 60;

num = num \* 7;

num = num / 10;

println(num);

println("'I always thought there was something fundamentaly wrong with the universe...'");

}

**Description:**

The fortyTwo function is used to change any number—via a series of multiplications and divisions—into forty-two, then to print that number into the console. The other phrases printed to the console in the example are not necessary, but are highly recommended to the gimmicky nature of this function.

**Syntax:**

fortyTwo(num);

**Parameters:**

num = number (to be changed to forty-two via the function)

**Returns:**

void