Processing (Java)	Unity (JavaScript via Mono)
DATA	

int x = 70; // Initialize
x = 30; // Change value
	var x : int = 70; // Initialize
	x = 30; // Change value
	
	// without setting the type you are asking for trouble
	
	var x = 71;
	x -= 0.1; // x is now 70

float x = 70.0;
x = 30.0;
	var x : float = 70.0;
	x = 30.0;

int x = int(3.14); // explicit cast
// x is 3
	var x : int = 3.14; // dynamic cast
	// x is 3

// public variable
public int x = 20;
 	
// private, hidden variable
private int y = 30;
 	
// static variable belongs to class
static public int z = 14;
	// editable in Inspector
	var x : int = 20;
	
	// hidden from the Inspector.
	private var y : int = 30;
	
	// this is now a global variable
	static var theglobalvar : int = 14;
	
	// use the script file name to access
	MyScriptFileName.theglobalvar++;

int[] a = new int[3];
a[0] = 12; // Assign
int l = a.length;
	var a = int[3];
	a[0] = 12; // Assign
	var l = a.length;
	
	// JavaScript also has a unique Array Object
	var a = new Array();
	a.push(“hello”);
	print(a[0]); // prints hello
	
	// It allows for some interesting tricks
	var dog = {“color” : “brown”,
	           “size” : “large”};
	print(dog[“color”]); // prints brown
	print(dog[“size”]); // prints large

Processing (Java)	Unity (JavaScript via Mono)

ArrayList a = new ArrayList();
a.add(new Type(10));
SomeType t = (SomeType)a.get(i);
a.remove(i);
	var a = new ArrayList();
	a.Add(something);
	var t : SomeType = a[i];
	a.RemoveAt(i);

PVector temp = new PVector(1,2,3);
temp.x = 14;
temp.normalize();
	var temp : Vector3 = Vector3(1,2,3);
	temp.x = 14;
	temp[1] = -37;
	// same as temp.y = -37
	temp.Normalize();

CONTROL

for(int i = 20; i < 50; i ++){
}
	for(var i : int = 20; i < 50; i++){
	}

if(c == 1){
}
	if(c == 1){
	}

if((c >= 4) && (c <= 32)){
}
	if((c >= 4) && (c <= 32)){
	}

String s = “Test”
 	
if(s.equals(“Test”) == true){
 // it’s true
}
	var s : String = “Test”
	
	if(s == “Test”){
	 // it’s true
	}

STRUCTURE

// comment /* long comment*/	// comment /* long comment*/

void setup()
{
    // I run at the start
}
	function Start()
	{
	    // I run at the start
	}

Processing (Java)	Unity (JavaScript via Mono)

void draw()
{
    // I run every frame
}
	function Update()
	{
	  // I run every frame
	}
	
	function FixedUpdate()
	{
	   // I run at a fixed time step
	   // used for physics
	}

void myFunction(float f)
{
}
 	
myFunction(88888454);
	function myFunction(f : float)
	{
	}
	
	myFunction(88888454);

float getHalf(float num)
{
    return num / 2.0;
}
 	
float test = getHalf(354676.3);
	function getHalf(num : float) : float
	{
	   return num / 2.0;
	}
	
	var test : float = getHalf(354676.3);

void myFunction()
{
  {
    // brackets allow for scoping
    int x = 14;
  }
  // x no longer exists at this point
  // we can create a new x
  {
    int x = 7;
  }
}
	function myFunction()
	{
	  // scope exists at function level
	  // extra brackets will cause error
	  var x : int = 14;
	
	  // x only destroyed at end of function
	  // variables must be reused
	  x = 7;
	
	}

Processing (Java)

```
void setup()
{
  for(int i = 0; i < 100; i ++){
  }

  for(int i = -50; i < 50; i++){
  }
}
```

Unity (JavaScript via Mono)

```
function Start()
{
  for(var i : int = 0; i < 100; i ++){
  }

  for(var i : int = -50; i < 50; i++){
    // ERROR HERE
  }
}

/* as all variables are local the above
will throw an error on the second for loop.
Unity will already have a variable called i
in Start */

function Start()
{
  for(var i : int = 0; i < 100; i ++){

  }

  for(i = -50; i < 50; i++){
    // This will work
  }
}
```