# 01 Introduction

## Learning Processing Resources

Language Reference: <http://processing.org/reference/>

Online Tutorials: <http://processing.org/tutorials/>

Online Examples: <http://processing.org/examples/>

My own examples: <https://github.com/mmayo888/ProcessingExamples>

Where to download: <http://processing.org/download/>

## Basic Processing Concepts

**Sketch** – a window or screen where your program displays graphics and images with its drawing commands.

**Frame** – an update to the sketch, usually occurs every 1/60th or 1/30th of a second.

**Setup()** – a special function called once at the start of the program to set everything up.

**Draw()** – another special function called once per frame, used to update the screen.

**Coordinate System** – (0,0) is defined at the top left of the sketch, *x* increases as you move right, *y* increases as you move down.

**Built-In Variables** – Processing defines several very useful built-in variables, e.g. width, height, frameRate, mouseX, mouseY etc.

## Basic Programming Concepts

**Instruction Sequence –** Processing is a Java-dialect, very similar to C#; semi-colons separate statements and curly braces define code blocks.

**Primitive Data Types** – int, float, char, String, color.

**Color Data Type** – specifies an RGB color, created using either the color() function, e.g. color(231, 67, 88); or hexadecimal notation, e.g. #88A2FA

**Arrays** – must be created with the new keyword and use square bracket notation for assignment, e.g. sizes[index]=45;.

**Scope** – variables declared inside functions are local, variables outside all functions are global.

**Conditional Statements** – if/then/else, switch.

**Iteration** – for loops, while/do loops, do/while loops.

**Functions** – Processing supports functions that either don’t return a value (void functions) or do return a value.

## Principles of Good Programming

**Do not duplicate code** – use loops and functions to eliminate unnecessary duplication.

**Use proper naming convention** – lowerCamelCase for variable and function names; UpperCamelCase for class names; FULL\_UNDERSCORED\_CAPS for constants. Use lengthy, descriptive names.

**Comment your code** – Processing supports // for single line comments and /\*…\*/ for multi-line comments. Write descriptive comments so that your peers can understand your program.

--