MUSI3620/40 Music Technology

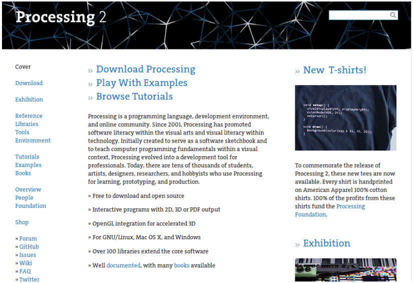
Processing 1 - Introduction

Alex McLean

Processing

- Free/open source project
- Initiated by Casey Reas and Ben Fry in 2005
- For learning programming in visual context
- Based on Java, but simplified
- Sketchbook metaphor

On-line documentation



Processing is a programming language, development environment, and online community. Since 2001, Processing has promoted software literacy within the visual arts and visual literacy within technology. Initially created to serve as a software sketchbook and to teach computer programming fundamentals within a visual context, Processing evolved into a development tool for professionals. Today, there are tens of thousands of students, artists, designers, researchers, and hobbyists who use Processing for learning, prototyping, and production.

» Free to download and open source

» Interactive programs with 2D, 3D or PDF output

» OpenGL integration for accelerated 3D

» For GNU/Linux, Mac OS X, and Windows

» Over 100 libraries extend the core software

» Well documented, with many books available

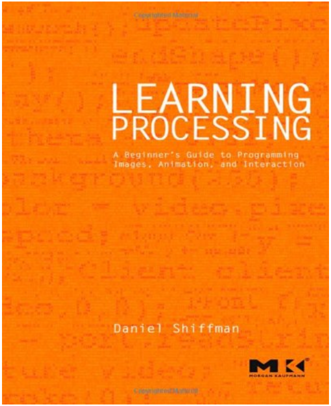
```
void setup() {
  size(displayWidth, displayHeight);
  colorMode(HSB, 26);
  noCursor();
}

void draw() {
  background(color(key & 31, 26, 26));
}
```

To commemorate the release of Processing 2, these new tees are now available. Every shirt is handprinted on American Apparel 100% cotton shirts. 100% of the profits from these shirts fund the Processing Foundation.

`https://processing.org/`

- Reference, Tutorials, Forum
- Off-line from UI:
  - Help -> reference
  - Right click on code -> find in reference

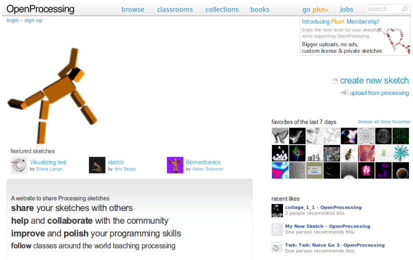
Further reading


LEARNING PROCESSING
A Beginner's Guide to Programming Images, Animation, and Interaction
Daniel Shiffman
MK MORGAN KAUFMANN

Video lectures: `http://icm.shiffman.net/`

More books: `http://processing.org/books/`

Open processing



Introducing Plus+ Membership!
Enjoy the next level for your sketches while supporting OpenProcessing
Bigger uploads, no ads, custom license & private sketches

Visualizing text by Diana Lange

sketch by Aris Bezas

Biomechanics by Asher Salomon

A website to share Processing sketches

share your sketches with others

help and **collaborate** with the community

improve and **polish** your programming skills

follow classes around the world teaching processing

collage_1_1 - OpenProcessing
2 people recommend this.

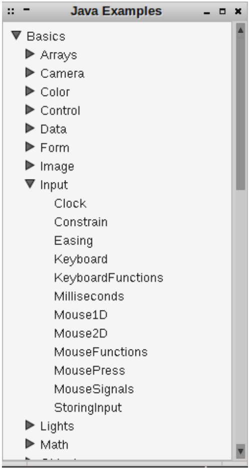
My New Sketch - OpenProcessing
One person recommends this.

Twk: Twk: Naive Go 3- OpenProcessing
One person recommends this.

`https://openprocessing.org/`

Examples

File -> Examples



Lets get programming

- Draw some shapes
- Play a sound

Draw a circle

Use the ellipse function call

```
// x, y, width, height
ellipse(10,10,10,10);
```

Draw a square

Use the rect function call

```
// x, y, width, height
rect(30,30,10,10);
```

Change the brush colour

Specify colour as red, green and blue components, from 0 to 255.

```
stroke(0,0,0);

fill(255,0,0);
ellipse(10,10,10,10);

fill(0,255,0);
rect(30,30,10,10);

fill(0,0,255);
// x1, y1, x2, y2, x3, y3
triangle(40,40,50,40,55,45);

stroke(0,0,255);
line(0,0,50,50);
```

Exercise 1

Draw a face (or something) using fill, stroke, ellipse, rect and line. For more see “2D primitives” here: `http://processing.org/reference`

Reminder:

```
// red, green, blue component from 0 to 255
fill(0, 255, 255);
// Same, but for line colour (e.g. around a shape)
stroke(255, 0, 255);

// x, y, width, height in pixels
ellipse(10, 10, 10, 10);

// same as ellipse
rect(20, 20, 10, 10);

// fromX, fromY, toX, toY
line(40, 40, 50, 50);
```

Loops - Draw ten squares

```
int count = 0;
while (count < 10) {
  fill(255 * (count/10.0), 255, 0);
  rect(count * 10, 10, 10, 10);
  // add 1 to count
  count = count + 1;
}
```

or

```
for (int count = 0; count < 10; count++) {
  fill(255 * (count/10.0), 255, 0);
  rect(count * 10, 10, 10, 10);
}
```

Exercise 2: Make a sound

Add the minim library to your sketch: (Sketch -> Import library -> minim)

Download sound from `http://is.gd/kickdrum`

Add the sound to your sketch with Sketch -> Add file

```
// Initialise audio
Minim minim = new Minim(this);

// Prepare a sound
AudioSample kick = minim.loadSample("kick.wav");

// Trigger the sound
kick.trigger();
```

Animation the Processing way

Follows code in `setup()` once, and then in `draw()` every frame.

```
// global variables
AudioSample kick;

void setup() {
  // make the canvas a bit bigger
  size(300,300);
  Minim minim = new Minim(this);
  kick = minim.loadSample("kick.wav");
  // follow draw() two times every second
  frameRate(2);
}

void draw() {
  ellipse(random(width),random(height),10,10);
  kick.trigger();
}
```

Exercise 3 - Movement

```
// global variables
float bally = 0;
float ballx = 150;

void setup() {
  // make the canvas a bit bigger
  size(300,300);
}

void draw() {
  ellipse(ballx,bally,10,10);
}
```

1. Make the shape move
   - Add another global variable that stores the speed of the shape
   - Add `speed` to `bally` every frame (i.e. within `draw()`)

Exercise 3 - Movement

2. Make the shape bounce off the edges of the sketch and make a sound
   - have an `if` statement that tests whether `bally` $>$ `height`
   - when that is true, play a sound and invert `speed`, (i.e. set `speed = 0 - speed`)
   - do something similar for the top of the sketch (i.e. when `bally` $<$ 0)
   - add the code to load a sound in `setup()`, and have a bounce trigger it in `draw()`
3. Challenges: have multiple shapes, or add gravity effect.
4. Challenge question: Could we replace the `float` variables with `int` variables? What is the advantage of either?