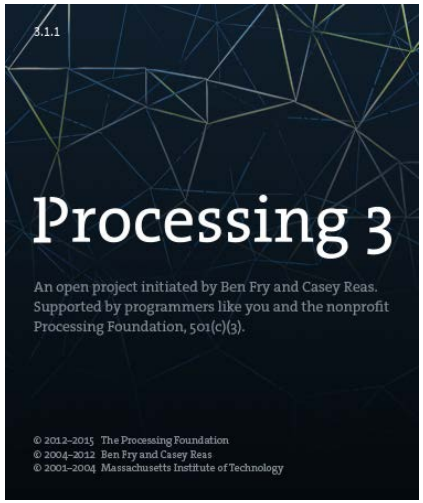


创意建模工具 Part IV

Processing简介

Processing概况

PROCESSING?

- 一个基于JAVA的开源环境和编程语言
- 一个用于创意概念开发的数字草图本
- http://processing.org/

定位

- 一个编程教学工具
- 一个艺术与设计的创作介质
- 数据表达工具

使用环境

- 基于动效呈现数据的丰富意义
- 进行基于网页的内容创作
- 一个有助于编程初学者快速入门的教学工具

优势

- 免费
- 不需要很深入的编程技巧就能较为容易创造出复杂的视觉呈现
- 基于JAVA，继承了JAVA的优势（非常普及的使用-从软件到硬件、从桌面应用到移动应用；异常庞大的开发者社群；丰富的开源资源）

有待优化的不足

- 一直处于开发中，稳定性不足
- 出错信息可用性不大
- 对于复杂的项目而言，需要调用JAVA的资源，processing开发环境对复杂项目的支持没有eclipse更丰富强大

竞争优势

- Cinder（基于C++）和openFramework（基于C++）
- Adobe Illustrator（静态2维图形）
- Flash（动态2维图形）
- 3D程序，但是通常需要更复杂的编程技术

开源协议

- Development environment released under GPL (GNU General Public License)
- Core (export) libraries released under LGPL
- Sketches and Projects made using can be distributed without opening source code (and commercially) and export libraries can be linked into any project under LGPL.
- Changes to core libraries or development environment source code **must** be released as open source

小结

- 广为设计师、艺术工作者、教育工作者、科技工作者、和专业编程人员所使用
- 自2001以来得到了越来越多的关注和部署
- Processing是面向艺术与设计领域的基于开源精神的创新项目

EXAMPLES

- Obama / McCain money graph
(http://www.pitchinteractive.com/election2008/)
- Courbes
(http://phpbb.sans.entraves.free.fr/forum/jeux/applet/index.html)
- Homo sapiens chromosome 20 - first 140,000 bases
(http://www.flickr.com/photos/thirteen/2738353101/)
- Proximity of Needs
(http://www.proximityofneeds.org/)

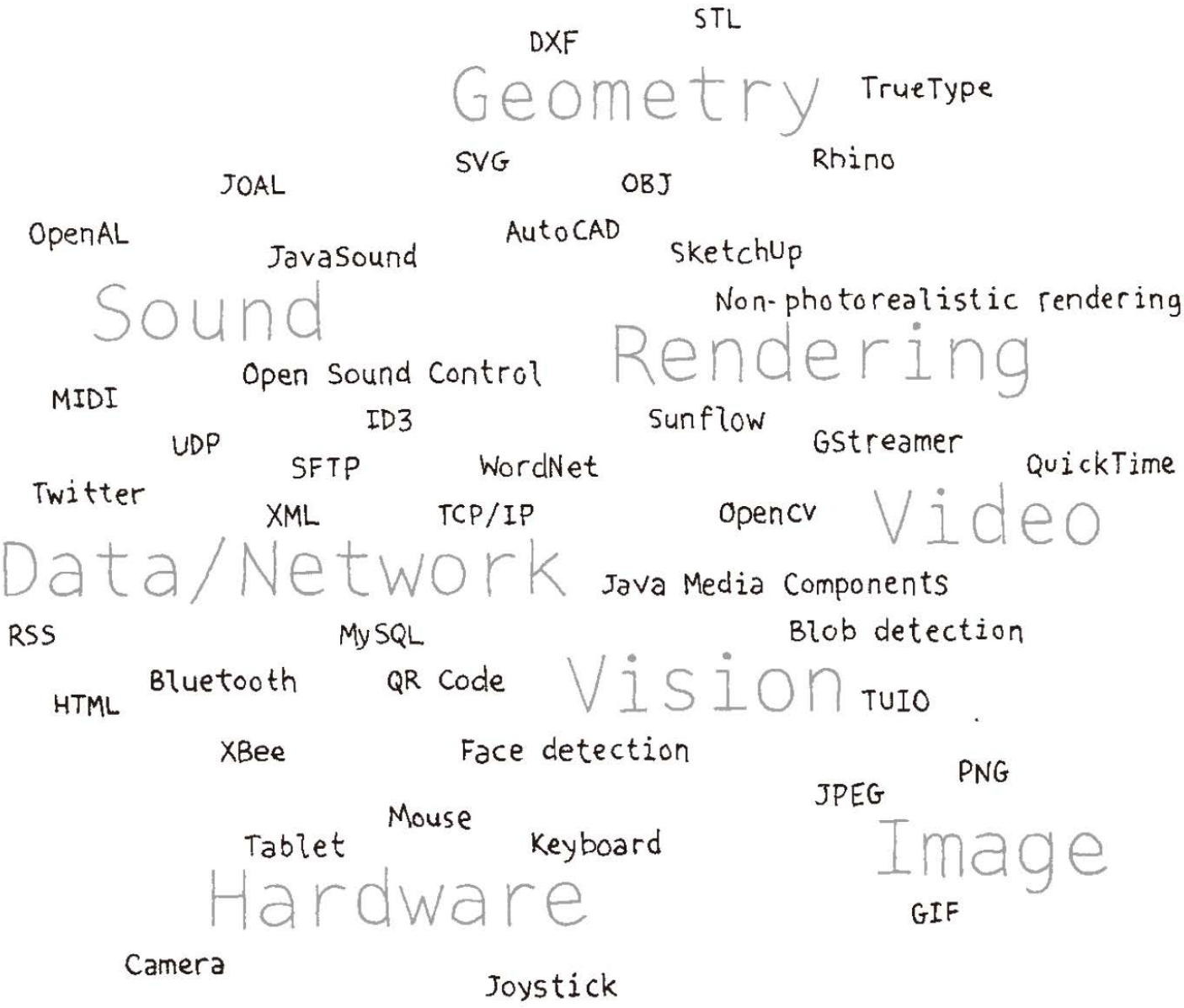
INTERACTIVE

- The God of Music
(http://www.artisopensource.net/GoM/)
- Last FM Graph
(http://www.swcp.com/~atomboy/lastfmgraph/)
- tigtag
(http://tictag.cc/)
- ChakraPuter
(http://www.artisopensource.net/Chakraputer/index.html)
- Shooting with the Burdening Game
(http://gromina.googlepages.com/shootingwiththeburdeninggame)

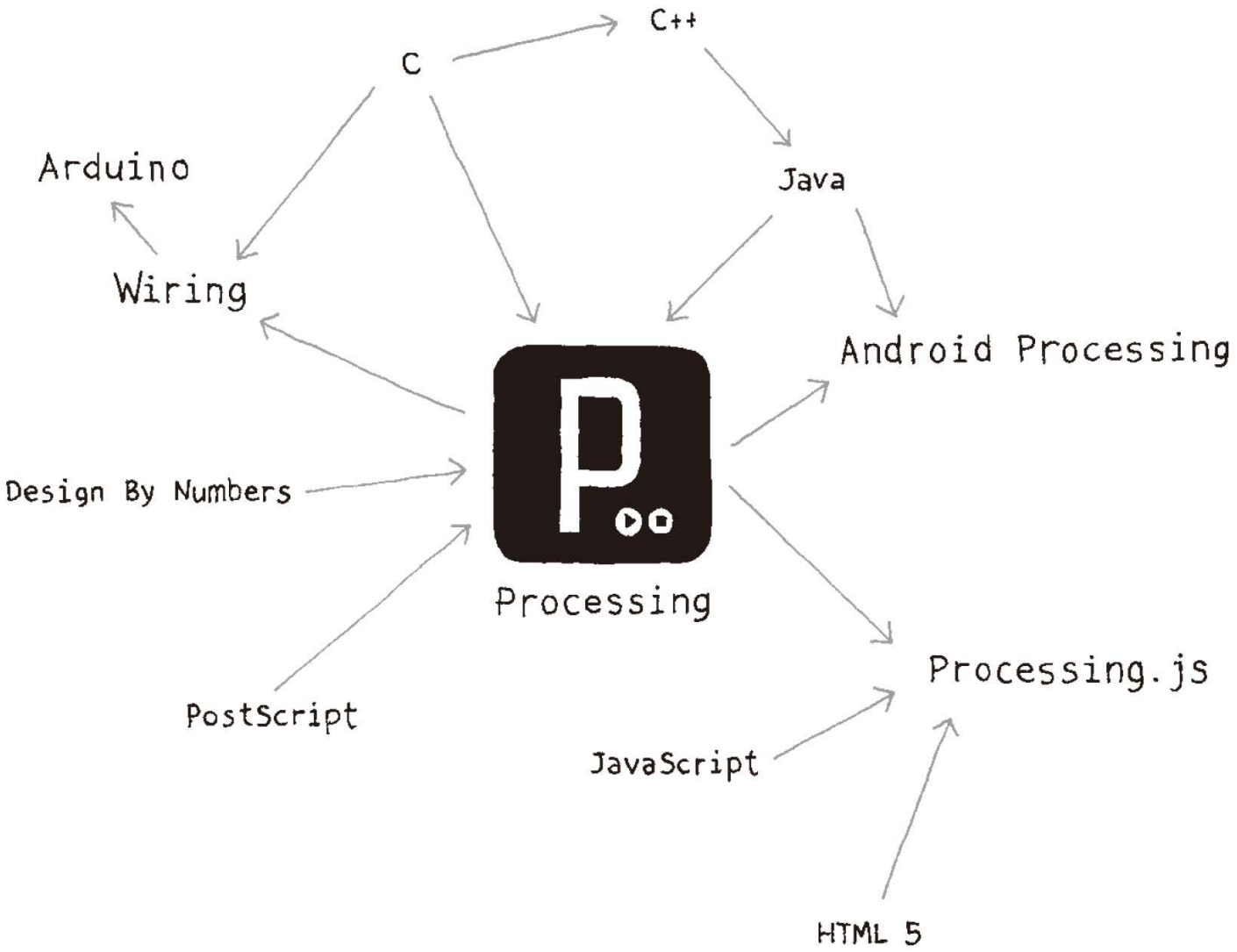
REFERENCES

- http://processing.org/
- Processing, A Programming Handbook for Visual Designers and Artists, Casey Reas and Ben Fry
- GPL Licensing
(http://www.opensource.org/licenses/gpl-2.0.php)
- LGPL Licensing
(http://www.opensource.org/licenses/lgpl-2.1.php)

Casey Reas and Ben Fry



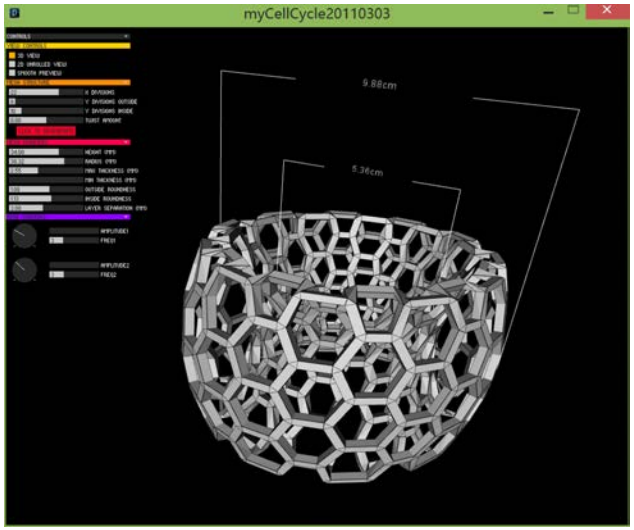
Processing支持许多种数据类型的输入输出

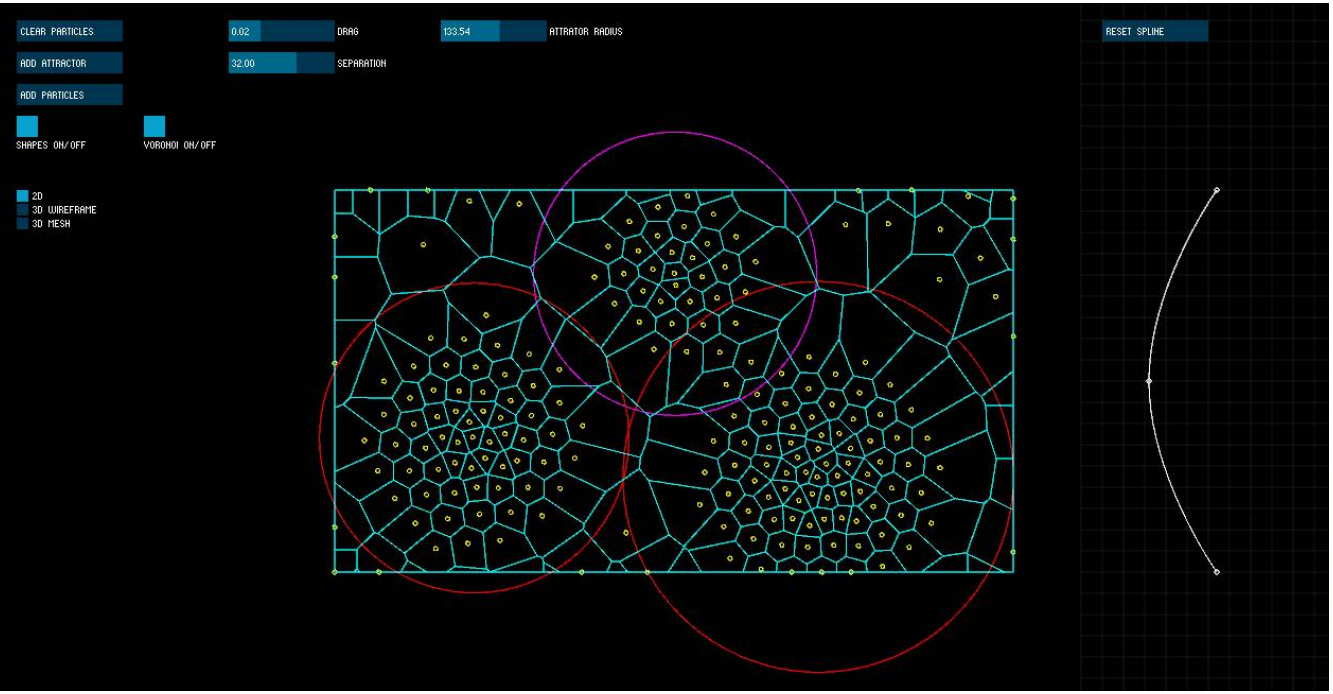


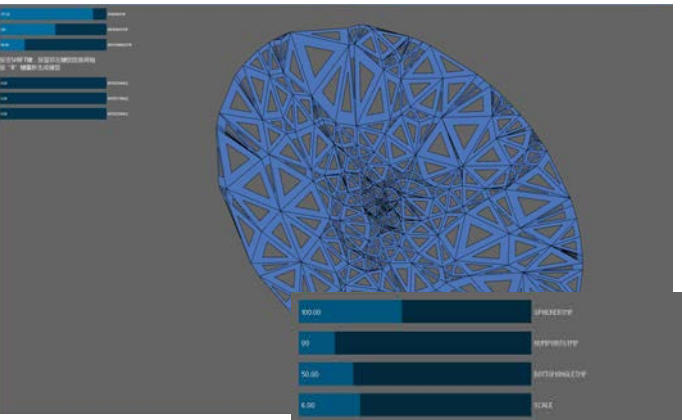
Processing与许多编程环境和语言形成了关联

Processing应用案例

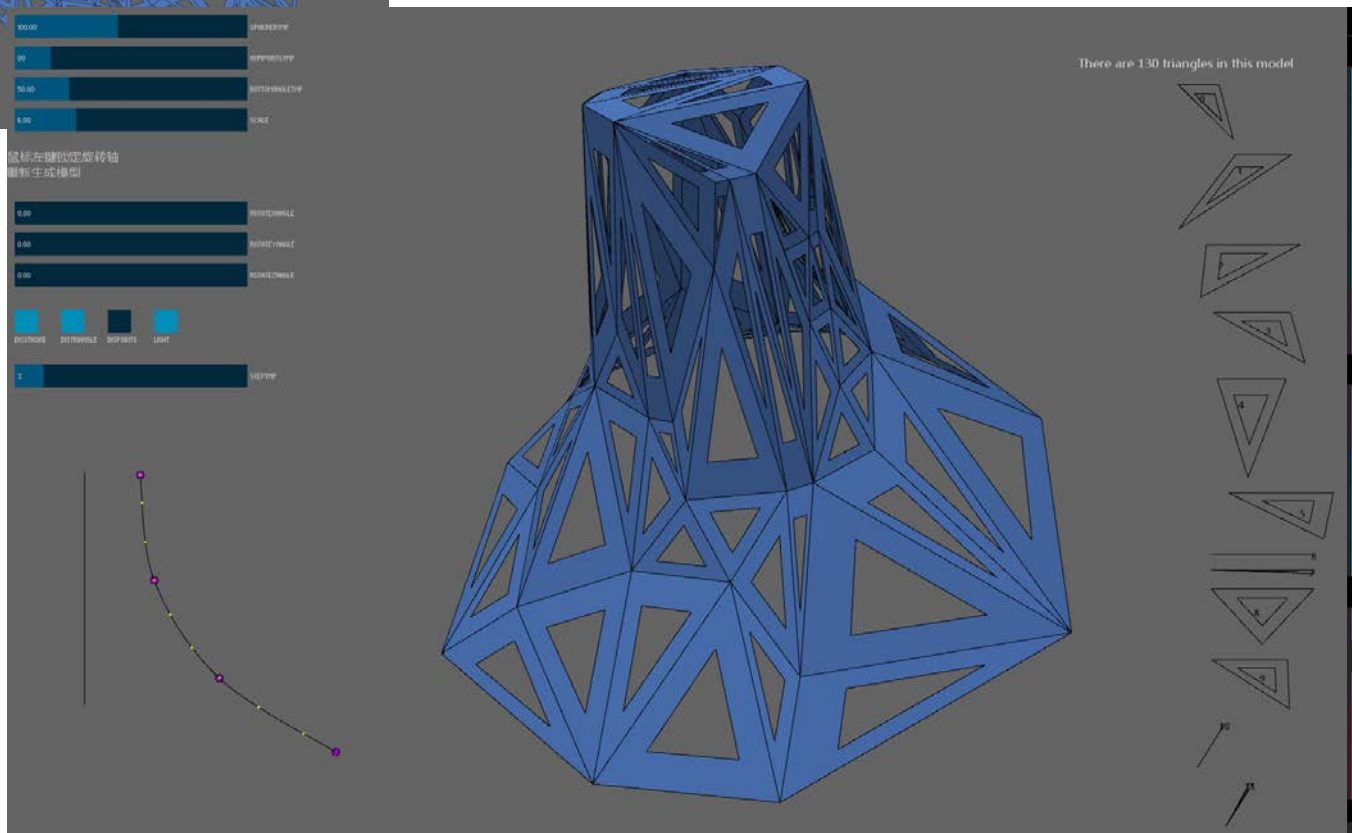
基于开源集成开发环境的创意建模基础– 概念原型工作坊
Creative Conceptual Prototype Building



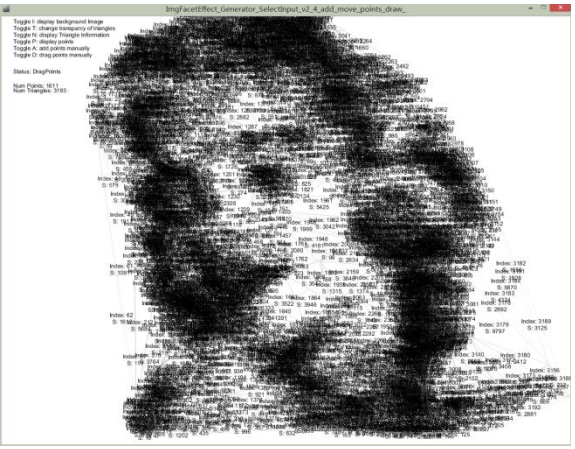




- 运行的系统基于web，采用HTML5、CSS、Javascript、WebGL等相关技术，有助于快速部署；
- 造型的生成基于特定的算法，通过对系统变量和用户变量的定义与约束，支持用户通过调节相关变量生成自己满意的造型；
- 系统生成的数据包将根据协议发送到加工制造节点完成制造过程。



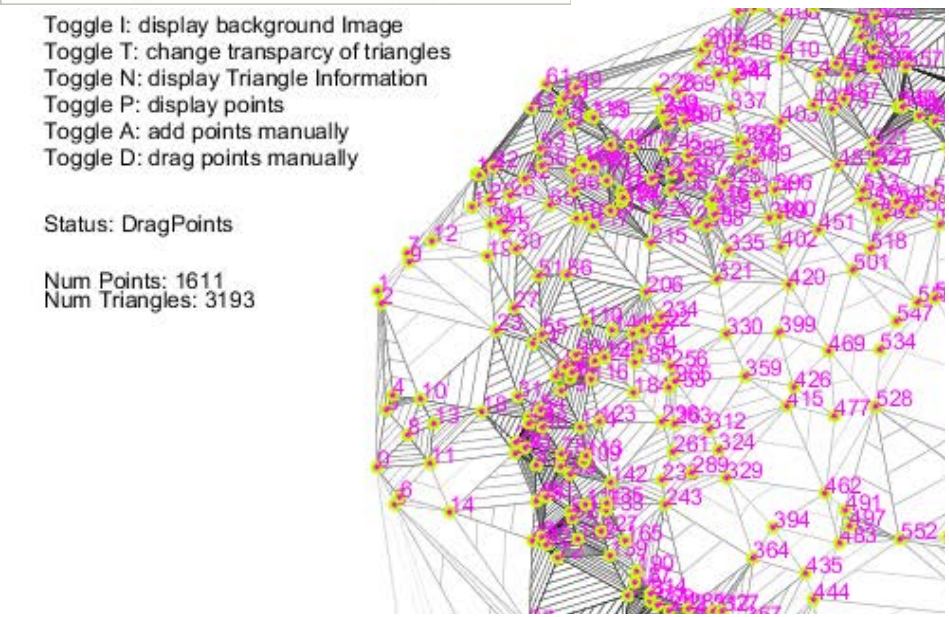
一种生成规则（构型法规则、功能定义规则、美学特征定义规则）集合，根据不同的功能定位，采用不同的参数控制，可以生成多种不同的造型。

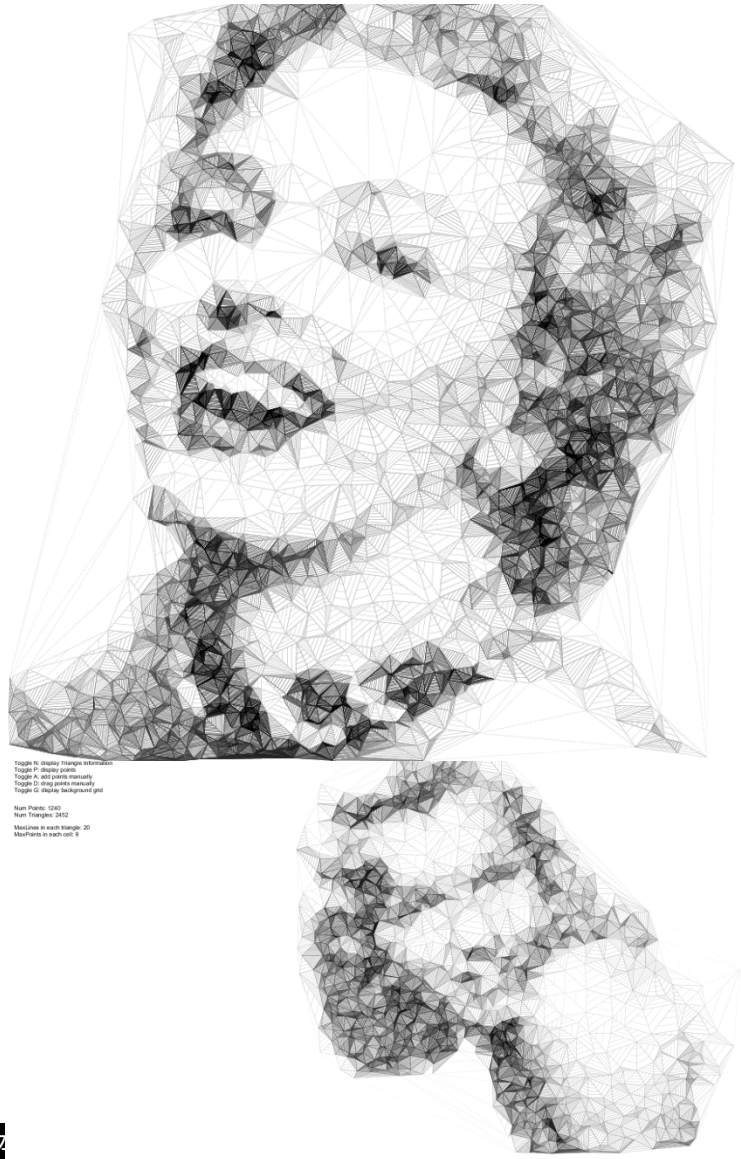


Toggle I: display background Image
Toggle B: change transparcy of triangles
Toggle T: display Triangles
Toggle N: display Triangle Information
Toggle P: display points
Toggle A: add points manually
Toggle D: drag points manually
Toggle G: display background grid

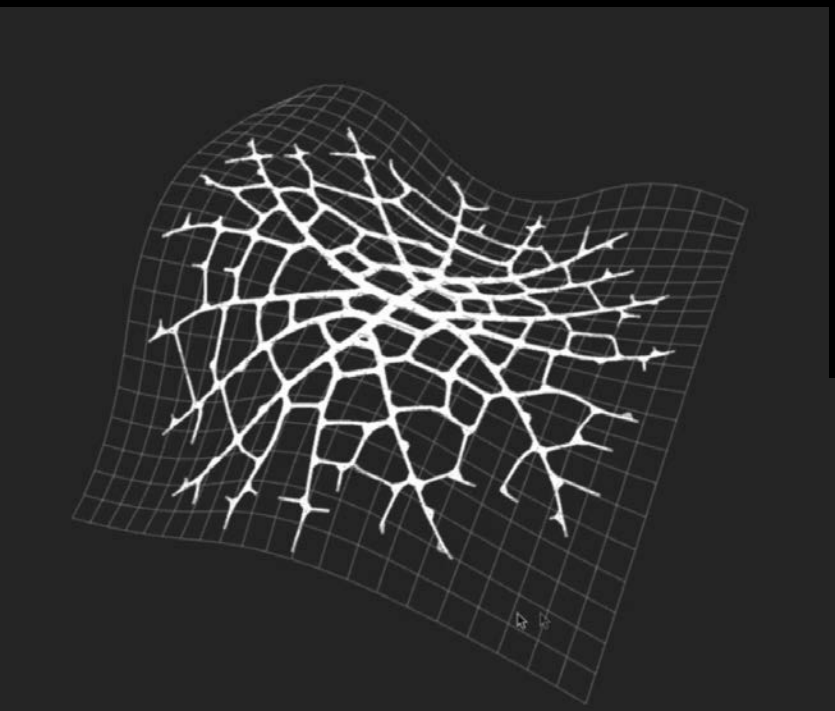
Num Points: 2136
Num Triangles: 4243

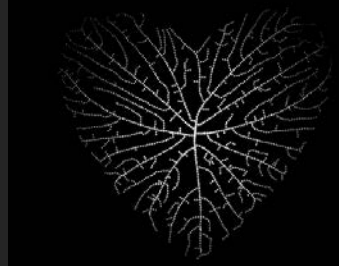
MaxLines in each triangle: 20
MaxPoints in each cell: 9

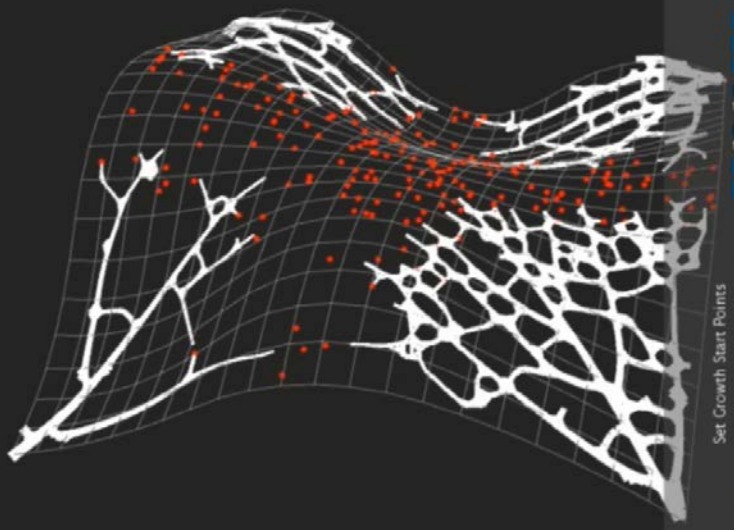


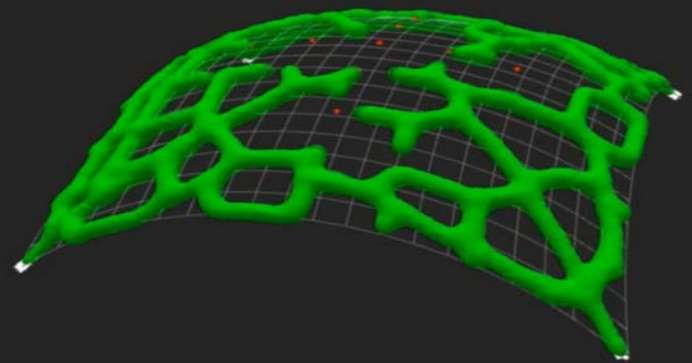


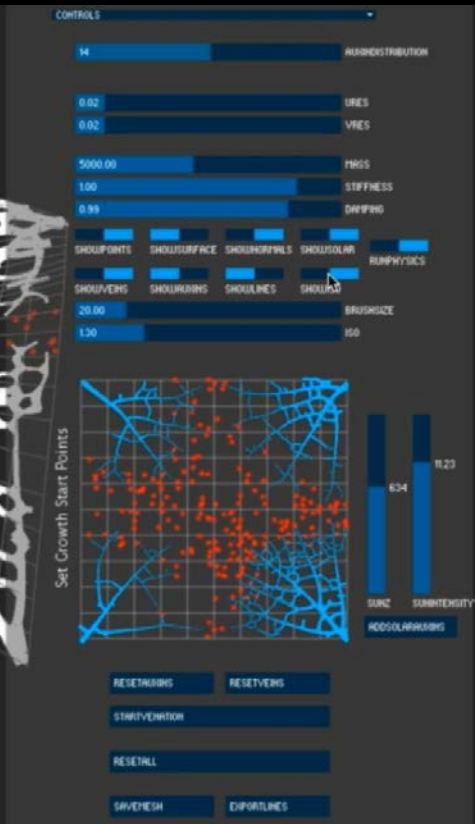
基于开源集成开发环境的创意建模基础– 概念原型工作坊
Creative Conceptual Prototype Building



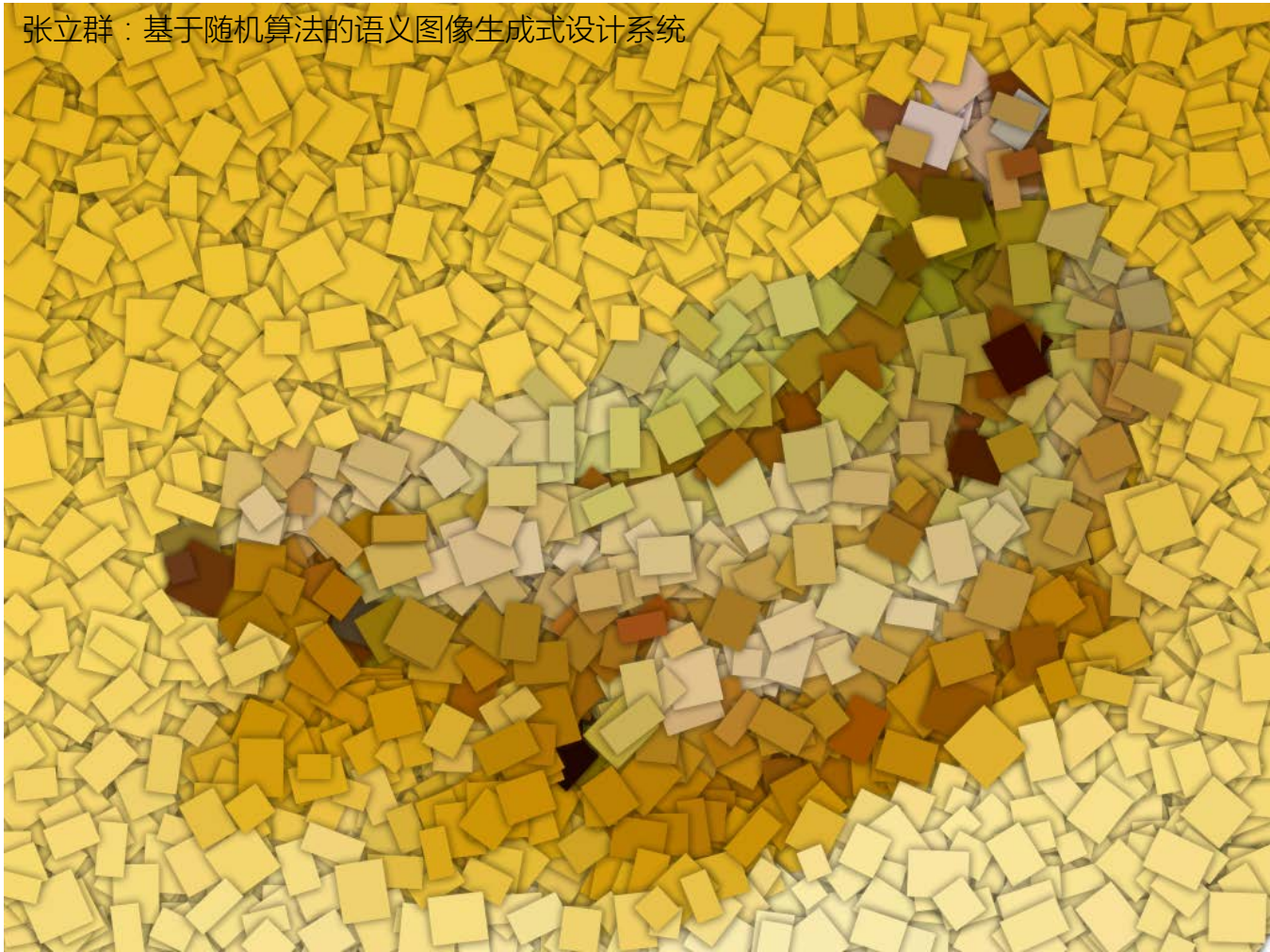






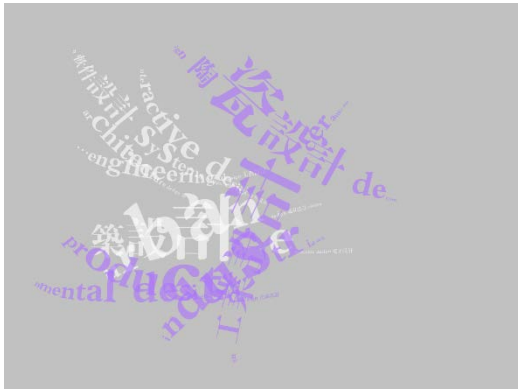


产品属性生成规则 - 形态结构生成算法的开发与代码实现

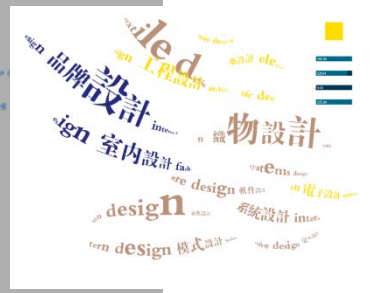


张立群：基于随机算法的语义图像生成式设计系统

本例中开放给用户的参数：
1-布撒的单元型类型及大小
2-阴影扩散阈值与色彩属性



张立群：FreeTexty-基于wacom压感笔的文本笔刷绘图设计工具



本例中开放给用户的参数：
1-字体类型与颜色
2-文本内容
3-字体大小阈值与属于设备的映射比

Processing安装与运行

安装

- 下载(http://processing.org/download/index.html)
- 运行

Processing

Cover

Download

Exhibition

Reference
Libraries
Tools
Environment

Tutorials
Examples
Books
Handbook

Overview
People

Shop

» Forum
» GitHub
» Issues
» Wiki
» FAQ
» Twitter
» Facebook

Download Processing. Processing is available for Linux, Mac OS X, and Windows. Select your choice to download the software below.

2.2.1 (19 May 2014)

Windows 64-bit | Linux 64-bit | Mac OS X
Windows 32-bit | Linux 32-bit

» Github
» Report Bugs
» Wiki
» Supported Platforms

The list of revisions covers the differences between releases in detail. Please read the changes if you're new to the 2.0 series.

Stable Releases

2.2.1 (19 May 2014) Win 32 / Win 64 / Linux 32 / Linux 64 / Mac OS X

1.5.1 (15 May 2011) Win (standard) / Win (no Java) / Linux x86 / Mac OS X

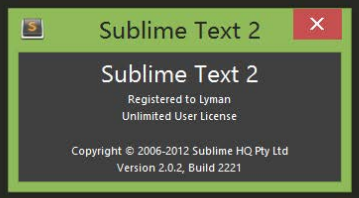
Earlier releases have been removed because we can only support the current versions of the software. To update old code, read the changes page. Per-release changes can be found in revisions.txt. If you have problems with the current release, please file a bug so that we can fix it. Older releases can also be built from the source. Read More about the releases and their numbering. Note that Android mode no longer works in Processing 1.5, you'll need to use a Processing 2 release to do Android development.

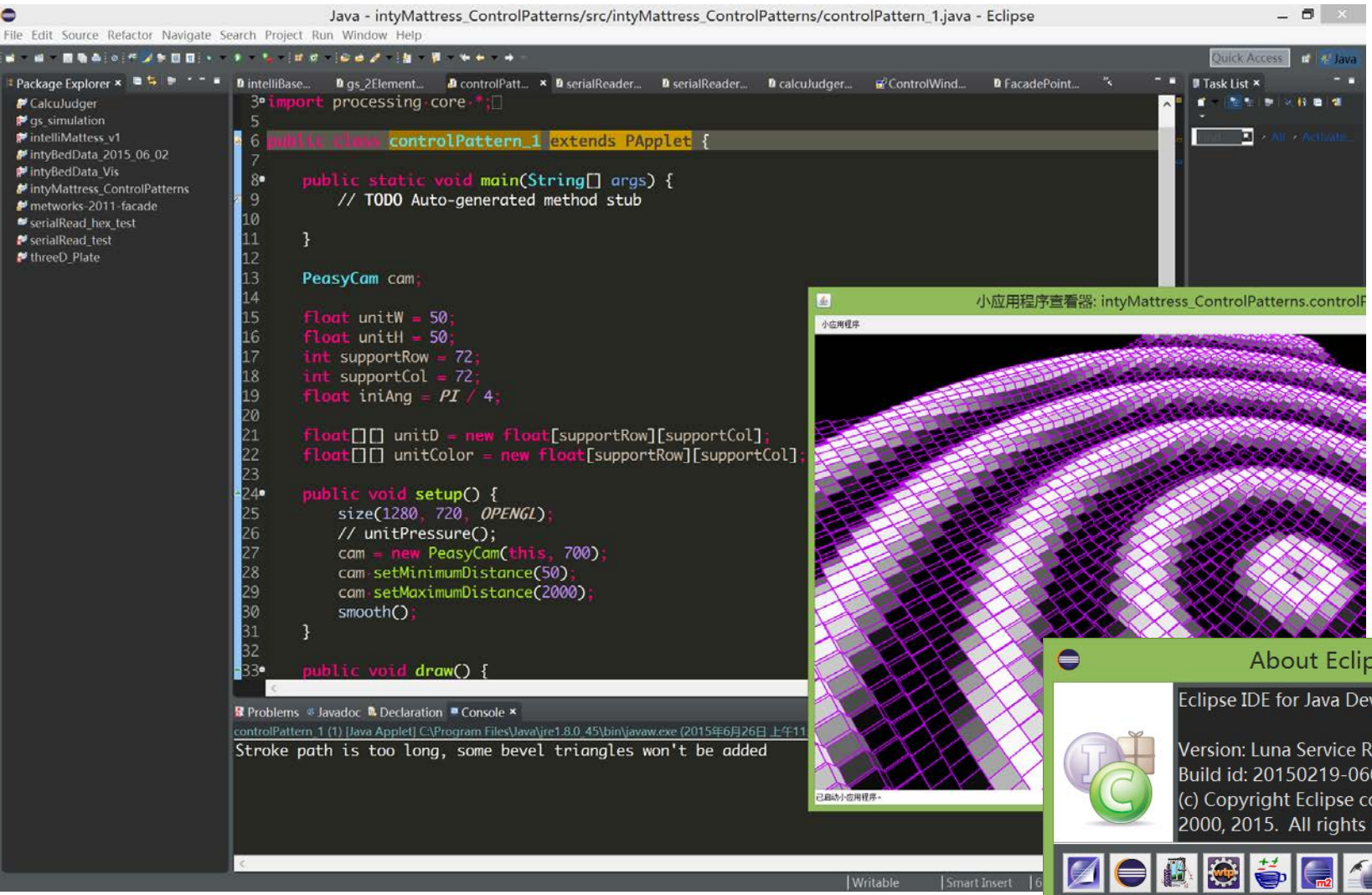
其他有用的IDE：

1-Sublime Text 2/3

```
(function e(t,n,r){function s(o,u){if(!n[o]){if(!t[o]){var a=typeof require=="func
//import modules
var voronoi = require('./voronoi.js');

var bookshelf = exports;

bookshelf.width = 1200; //bookshelf width in mm
bookshelf.height = 1800; //bookshelf height in mm
bookshelf.depth = 254; //bookshelf depth in mm

bookshelf.woodWidth = 12.2; //thickness of the wood sheets in mm



/*
sets the bookshelf width

callback for width slider

@param float val
  the width of the bookshelf in mm
*/
function setWidth(val) {
  bookshelf.width = val;
  voronoi.setDimensions(bookshelf.width,bookshelf.height);


}
/*
sets the bookshelf height

callback for height slider
```

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

void draw(){
  noStroke();
  fill(0, 10);
  rect(0, 0, width, height);

  //draw something
  noStroke();
  fill(random(255), random(255), random(255));
  ellipse(random(width), random(height), 50, 50);
}
```

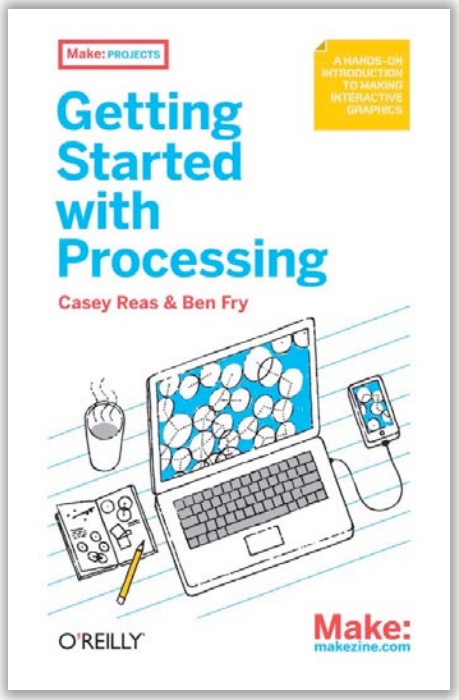


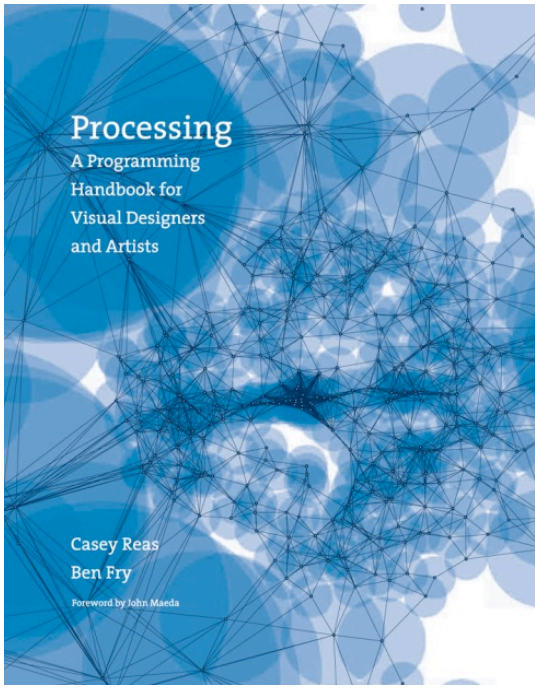


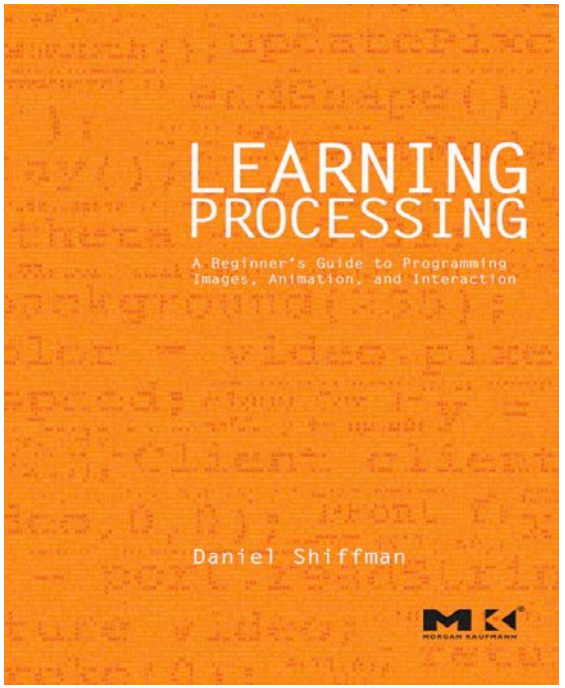
其他有用的IDE：

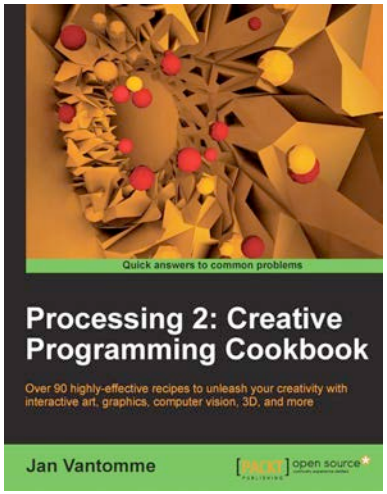
1-Eclipse JAVA IDE

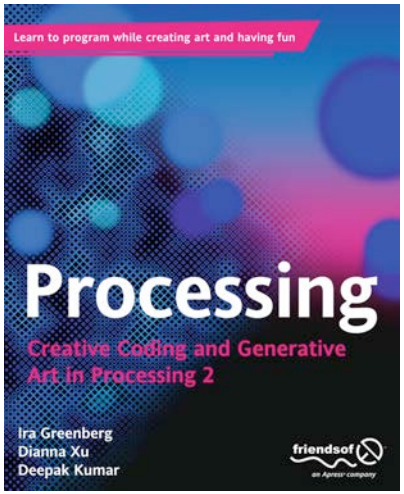
Processing学习资源

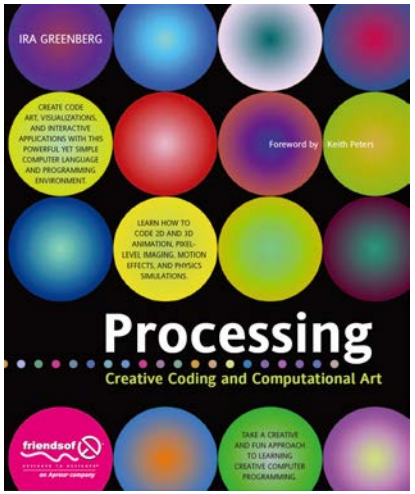

Make: PROJECTS
A HANDS-ON INTRODUCTION TO MAKING INTERACTIVE GRAPHICS
Getting Started with Processing
Casey Reas & Ben Fry
O'REILLY
Make:
makezine.com

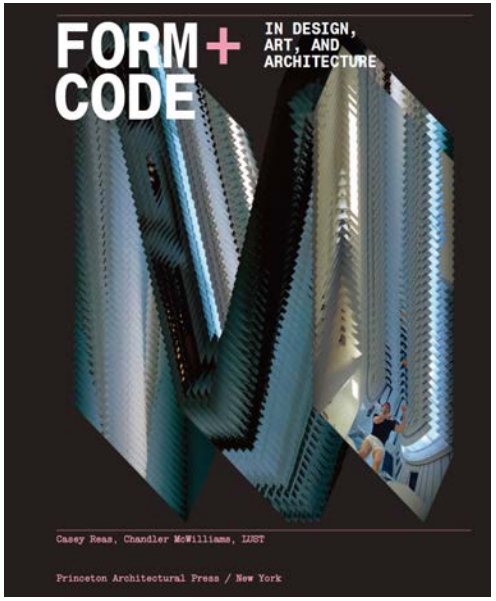

Processing
A Programming Handbook for Visual Designers and Artists
Casey Reas
Ben Fry
Foreword by John Maeda

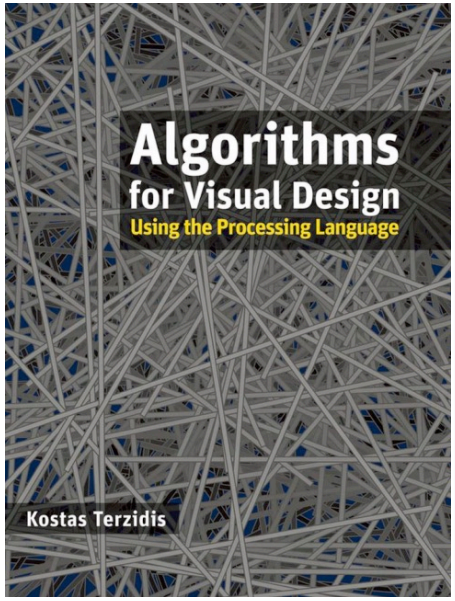

LEARNING PROCESSING
A Beginner's Guide to Programming Images, Animation, and Interaction
Daniel Shiffman
MORGAN KAUFMANN

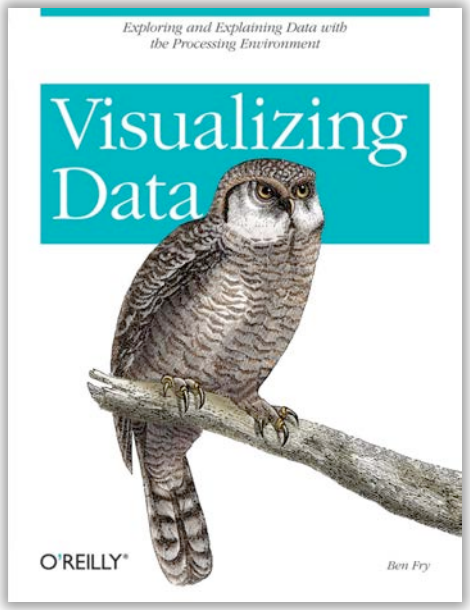

Quick answers to common problems
Processing 2: Creative Programming Cookbook
Over 90 highly-effective recipes to unleash your creativity with interactive art, graphics, computer vision, 3D, and more
Jan Vantomme
open source

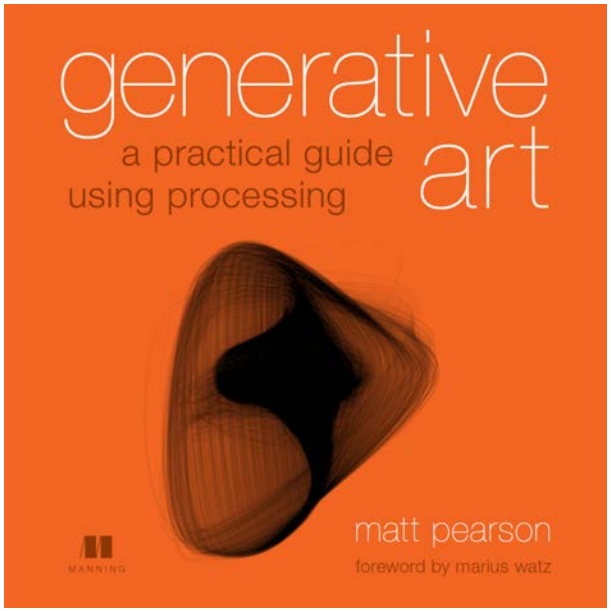

Learn to program while creating art and having fun
Processing
Creative Coding and Generative Art in Processing 2
Ira Greenberg
Dianna Xu
Deepak Kumar
friendsof

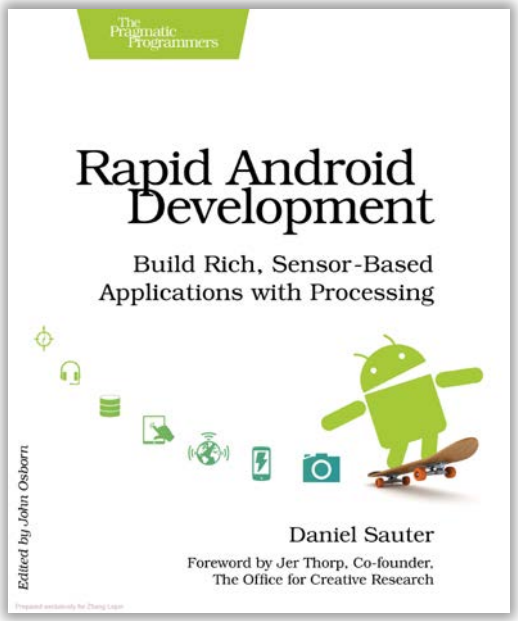

IRA GREENBERG
Foreword by Keith Peters
Processing
Creative Coding and Computational Art
friendsof

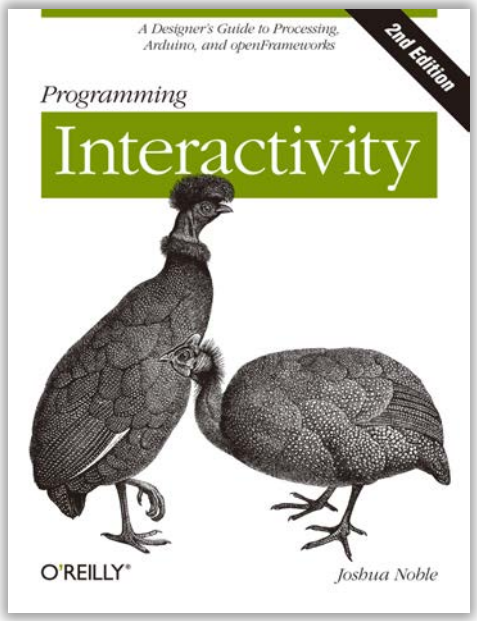

FORM+
CODE
IN DESIGN, ART, AND ARCHITECTURE
Casey Reas, Chandler McWilliams, LUST
Princeton Architectural Press / New York


Algorithms
for Visual Design
Using the Processing Language
Kostas Terzidis

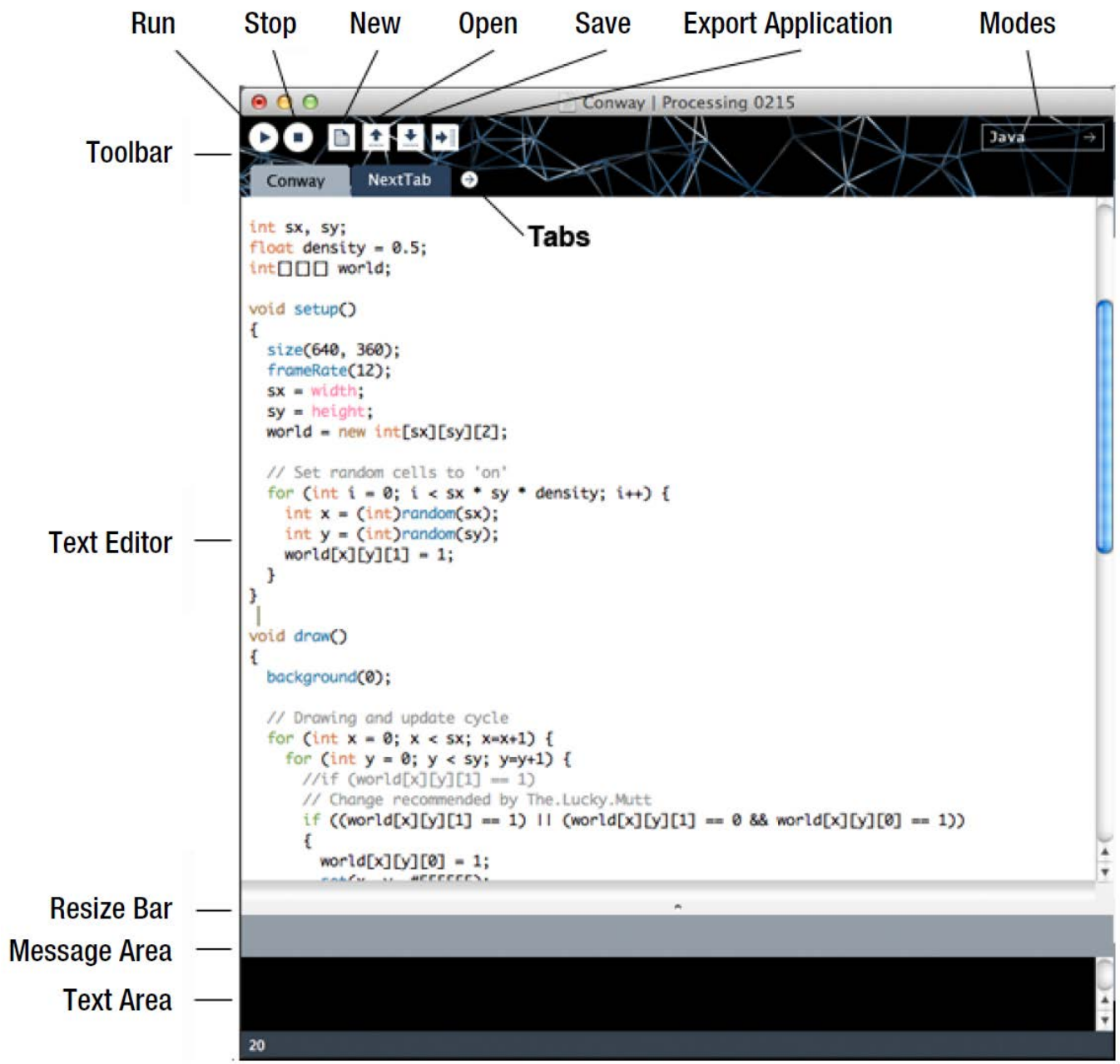

Exploring and Explaining Data with the Processing Environment
Visualizing Data
O'REILLY
Ben Fry


generative art
a practical guide using processing
matt pearson
foreword by marius watz
MANNING

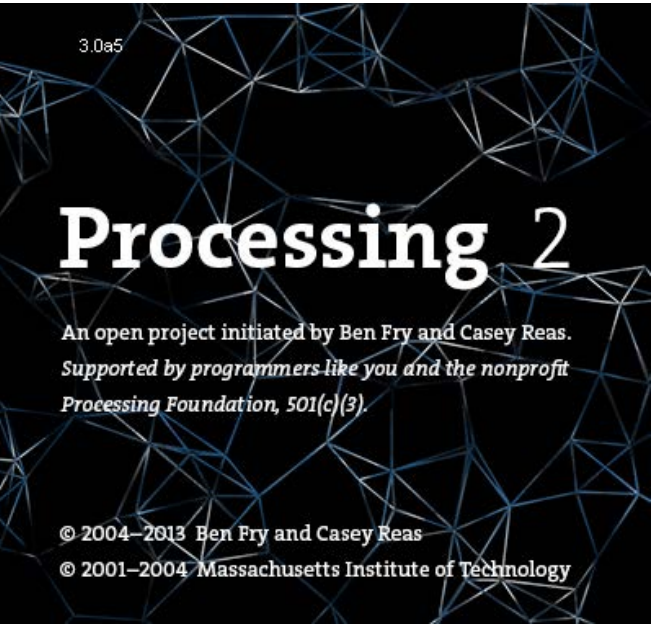

The Pragmatic Programmers
Rapid Android Development
Build Rich, Sensor-Based Applications with Processing
Daniel Sauter
Foreword by Jer Thorp, Co-founder, The Office for Creative Research
Edited by John Osborn


A Designer's Guide to Processing, Arduino, and openFrameworks
2nd Edition
Programming
Interactivity
O'REILLY
Joshua Noble

Processing IDE介绍




```
int sx, sy;
float density = 0.5;
int[][][] world;

void setup()
{
  size(640, 360);
  frameRate(12);
  sx = width;
  sy = height;
  world = new int[sx][sy][2];

  // Set random cells to 'on'
  for (int i = 0; i < sx * sy * density; i++) {
    int x = (int)random(sx);
    int y = (int)random(sy);
    world[x][y][1] = 1;
  }
}

void draw()
{
  background(0);

  // Drawing and update cycle
  for (int x = 0; x < sx; x=x+1) {
    for (int y = 0; y < sy; y=y+1) {
      //if (world[x][y][1] == 1)
      // Change recommended by The.Lucky.Mutt
      if ((world[x][y][1] == 1) || (world[x][y][1] == 0 && world[x][y][0] == 1))
      {
        world[x][y][0] = 1;
```



OOP_14_array_objects | Processing 3.0a5

File Edit Sketch Tools Help

OOP_14_array_objects | Car | Java (2.0)

```
// Daniel Shiffman
// <http://www.shiffman.net>

// Array of objects
// Note how the Car class hasn't changed!
// Also note how we are using multiple tabs to organize our code

// An array of "Cars" with 20 spots
Car[] cars = new Car[20];

void setup() {
  size(200,200);
  for (int i = 0; i < cars.length; i++) {
    // Creating the objects by call the constructor
    // Note the use of 4 parameters -- color, x position, y position, speed, size
    cars[i] = new Car(color(i*10),0,i*10,i*0.2+1,i+2);
  }
}

void draw() {
  framerate(30);
  background(100);
  // Calling functions "render" and "drive" for each car
  for (int i = 0; i < cars.length; i++) {
     cars[i].drive();
     cars[i].render();
  }
}
```

3.0 alpha 3.0a10

Processing 3

An open project initiated by Ben Fry and Casey Reas.
Supported by programmers like you and the nonprofit Processing Foundation, 501(c)(3).

OOP_16_array_objects | Processing 3.0a10

File Edit Sketch Debug Tools Help

Java ▾

OOP_16_array_objects | Stripe ▾

```
// Daniel Shiffman
// <http://www.shiffman.net>

// Another array of objects example

Stripe[] stripes = new Stripe[50];

void setup() {
  size(200,200);
  frameRate(30);
  // Initialize all "stripes"
  for (int i = 0; i < stripes.length; i++) {
    stripes[i] = new Stripe();
  }
}

void draw() {
  background(100);
  // move and display all "stripes"
  for (int i = 0; i < stripes.length; i++) {
    stripes[i].move();
    stripes[i].render();
  }
```

Console Errors

Processing开发资源

Language
Libraries
Tools
Environment

Reference. The Processing Language was designed to facilitate the creation of sophisticated visual structures.

Structure

() (parentheses)
, (comma)
. (dot)
/* */ (multiline comment)
/** */ (doc comment)
// (comment)
; (semicolon)
= (assign)
[] (array access)
{} (curly braces)
catch
class
draw()
exit()
extends
false
final
implements
import
loop()
new
noLoop()
null
popStyle()
private
public
pushStyle()
redraw()
return

Shape

createShape()
loadShape()
PShape

2D Primitives
arc()
ellipse()
line()
point()
quad()
rect()
triangle()

Curves
bezier()
bezierDetail()
bezierPoint()
bezierTangent()
curve()
curveDetail()
curvePoint()
curveTangent()
curveTightness()

3D Primitives
box()
sphere()
sphereDetail()

Color

Setting
background()
clear()
colorMode()
fill()
noFill()
noStroke()
stroke()

Creating & Reading
alpha()
blue()
brightness()
color()
green()
hue()
lerpColor()
red()
saturation()

Image

createImage()
PImage

Loading & Displaying
image()
imageMode()

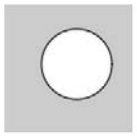
Structure
Environment
Data
Control
Shape
Input
Output
Transform
Lights
Camera
Color
Image
Rendering
Typography
Math
Constants

- **Name** —The name of the function.
- **Examples** —Example code (and visual result, if applicable).
- **Description** —A friendly description of what the function does.
- **Syntax** —Exact syntax of how to write the function.
- **Parameters** —These are the elements that go inside the parentheses. It tells you what kind of data you put in (a number, character, etc.) and what that element stands for. (This will become clearer as we explore more in future chapters.) These are also sometimes referred to as " arguments. "
- **Returns** —Sometimes a function sends something back to you when you call it (e.g., instead of asking a function to perform a task such as draw a circle, you could ask a function to add two numbers and *return* the answer to you). Again, this will become more clear later.
- **Usage** —Certain functions will be available for *Processing* applets that you publish online (" Web ") and some will only be available as you run *Processing* locally on your machine (" Application ").
- **Related Methods** —A list of functions often called in connection with the current function. Note that " functions " in Java are often referred to as " methods. "

This reference is for Processing 2.0+. If you have a previous version, use the reference included with your software. If you see any errors or have suggestions, please let us know. If you prefer a more technical reference, visit the Processing Javadoc.

Name ellipse()

Examples



```
ellipse(56, 46, 55, 55);
```

Description Draws an ellipse (oval) to the screen. An ellipse with equal width and height is a circle. By default, the first two parameters set the location, and the third and fourth parameters set the shape's width and height. The origin may be changed with the `ellipseMode()` function.

Syntax

```
ellipse(a, b, c, d)
```

Parameters

a	float: x-coordinate of the ellipse
b	float: y-coordinate of the ellipse
c	float: width of the ellipse by default
d	float: height of the ellipse by default

Returns void

Related ellipseMode()
arc()

Language
Libraries
Tools
Environment

Libraries. Extend Processing beyond graphics and images into audio, video, and communication with other devices.

The following libraries are included with the Processing software. To include a library, select its name from the "Import Library..." option in the Sketch menu. These libraries are open source; the code is distributed with Processing.

Video
Read images from a camera, play movie files, and create movies.

DXF Export
Create DXF files to save geometry for loading into other programs. It works with triangle-based graphics including polygons, boxes, and spheres.

Serial
Send data between Processing and external hardware through serial communication (RS-232).

PDF Export
Create PDF files. These vector graphics files can be scaled to any size and printed at high resolutions.

Network
Send and receive data over the Internet through simple clients and servers.

» Minim
Uses JavaSound to provide an easy-to-use audio library while still providing flexibility for more advanced users.

Contributions

Contributed Libraries must be downloaded individually. Select "Add Library..." from the "Import Library..." submenu within the Sketch menu. Not all available libraries have been converted to show up in "Add Library...". If a library isn't there, it will need to be installed manually. Follow the How to Install a Contributed Library instructions on the Processing Wiki for more information.

Contributed libraries are developed, documented, and maintained by members of the Processing community. For feedback and support, please post to the Forum. For development discussions post to the Create & Announce Libraries topic. Instructions for creating your own library are on the Processing GitHub site.

A complete list of libraries can be found online.

Contributions

Contributed Libraries must be downloaded individually. Select "Add Library..." from the "Import Library..." submenu within the Sketch menu. Not all available libraries have been converted to show up in "Add Library...". If a library isn't there, it will need to be installed manually. Follow the How to Install a Contributed Library instructions on the Processing Wiki for more information.

Contributed libraries are developed, documented, and maintained by members of the Processing community. For feedback and support, please post to the Forum. For development discussions post to the Create & Announce Libraries topic. Instructions for creating your own library are on the Processing GitHub site.

3D
Animation
Compilation
Data
GUI
Geometry
Hardware
I/O
Language
Math
Other
Simulation
Sound
Typography
Utilities
Video & Vision

3D

» proscene
by Jean Pierre Charalambos
Library that eases the creation of interactive scenes.

» HE_Mesh 2014
by Frederik Vanhoutte
HE_Mesh is an implementation of a half-edge datastructure for manipulating 3D meshes.

» PeasyCam
by Jonathan Feinberg
A mouse driven camera-control library for 3D sketches.

» Picking
by Nicolas Clavaud
Pick an object in a 3D scene easily.

» OCD: Obsessive Camera Direction
by Kristian Damkjer
The Obsessive Camera Direction (OCD) library allows intuitive control and creation of Processing viewport Cameras.

» Patchy
by Jonathan Feinberg
Patchy provides an easy-to-use bicubic patch for 3D Processing sketches.

» OBJLoader
by Tatsuya Saito and Matt Ditton
.OBJ 3D model file loader

» Shapes 3D
by Peter Lager
3D Shape creation and display made easy.

» SimpleOpenNI
by Max Rheiner
A simple wrapper for OpenNI(Kinect-Library). Before you can use SimpleOpenNI you have to install OpenNI.

» iGeo
by Satoru Sugihara
3D geometry library with packages of NURBS geometry, polygon mesh geometry, vector math, 3D display and navigation, 3D data file I/O and agent-based 3D geometry modeling.

» planetarium
by Andres Colubri
This library provides a renderer to project 3D scenes on a full dome.

Tools. Extend the capability of the Processing Environment (PDE) beyond the core functionality.

The following tools are included with the Processing software. Select the tool you want to use from the Tools menu of the Processing Environment. These tools are open source; the code is distributed with Processing.

Color Selector
Simple interface for selecting colors as RGB, HSB, and Hex values.

Movie Maker
Creates a QuickTime movie from a sequence of images. Options include setting the size, frame rate, and compression, as well as an audio file.

Archive Sketch
Archives a copy of the current sketch in .zip format. The archive is placed in the same directory as the sketch.

Create Font
Converts fonts into the Processing font format and adds to the current sketch. Opens a dialog box which give options for setting the font, its size, if it is anti-aliased, and if all characters should be generated.

Contributions

Contributed tools must be downloaded individually. Select "Add Tool..." from the Tools menu to select a Tool to download.

Contributed tools are developed, documented, and maintained by members of the Processing community. For feedback and support, please post to the Forum. For development discussions post to the Libraries and Tool Development topic. Instructions for creating your own tool are on the Processing GitHub site.

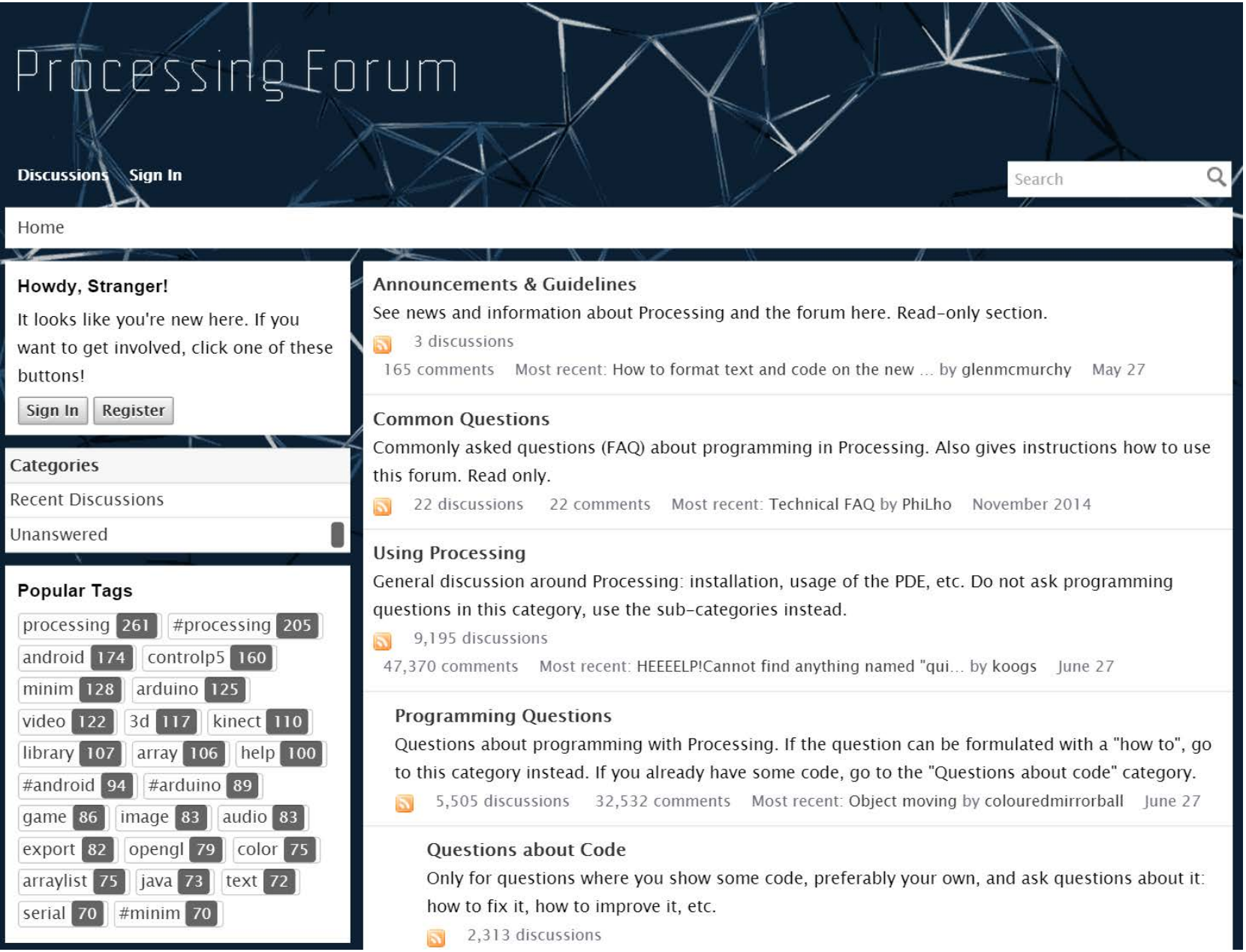
» Color Selector Plus
by Manindra Moharana
A feature rich color selector tool.

» TabManager
by Thomas Diewald

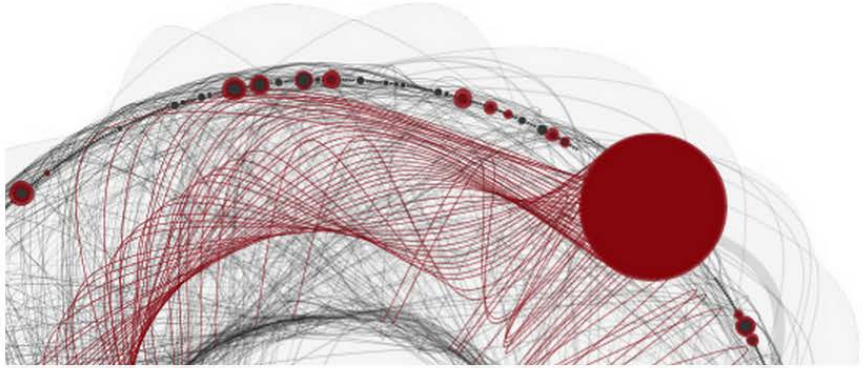
» AppletMaker
by Peter Lager
Applet exporter and signer.

» Templater
by Jonathan Acosta

» Bézier Editor
by Devon Rifkin
Draw Bézier curves visually and then automatically generate Processing code to include in your sketches.


Processing Forum
Discussions
Sign In
Search
Home
Howdy, Stranger!
It looks like you're new here. If you want to get involved, click one of these buttons!
Sign In
Register
Categories
Recent Discussions
Unanswered
Popular Tags
processing 261
#processing 205
android 174
controlp5 160
minim 128
arduino 125
video 122
3d 117
kinect 110
library 107
array 106
help 100
#android 94
#arduino 89
game 86
image 83
audio 83
export 82
opengl 79
color 75
arraylist 75
java 73
text 72
serial 70
#minim 70
Announcements & Guidelines
See news and information about Processing and the forum here. Read-only section.
3 discussions
165 comments Most recent: How to format text and code on the new ... by glenmcmurchy May 27
Common Questions
Commonly asked questions (FAQ) about programming in Processing. Also gives instructions how to use this forum. Read only.
22 discussions 22 comments Most recent: Technical FAQ by PhiLho November 2014
Using Processing
General discussion around Processing: installation, usage of the PDE, etc. Do not ask programming questions in this category, use the sub-categories instead.
9,195 discussions
47,370 comments Most recent: HEEEELP!Cannot find anything named "qui... by koogs June 27
Programming Questions
Questions about programming with Processing. If the question can be formulated with a "how to", go to this category instead. If you already have some code, go to the "Questions about code" category.
5,505 discussions 32,532 comments Most recent: Object moving by colouredmirrorball June 27
Questions about Code
Only for questions where you show some code, preferably your own, and ask questions about it: how to fix it, how to improve it, etc.
2,313 discussions

login - sign up



featured 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

Processing is an open source programming language and environment for people who want to program images, animation, and interactions. It is an open project initiated by Ben Fry and Casey Reas. It can be downloaded from processing.org.

Books are available for learning Processing at all levels.

Introducing Plus+ Membership!

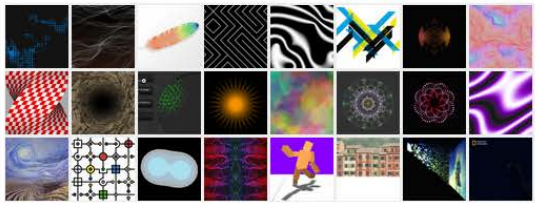
Enjoy the next level for your sketches while supporting OpenProcessing

Bigger uploads, no ads, custom license & private sketches

create new sketch

upload from processing

favorites of the last 7 days browse all-time favorites



recent likes

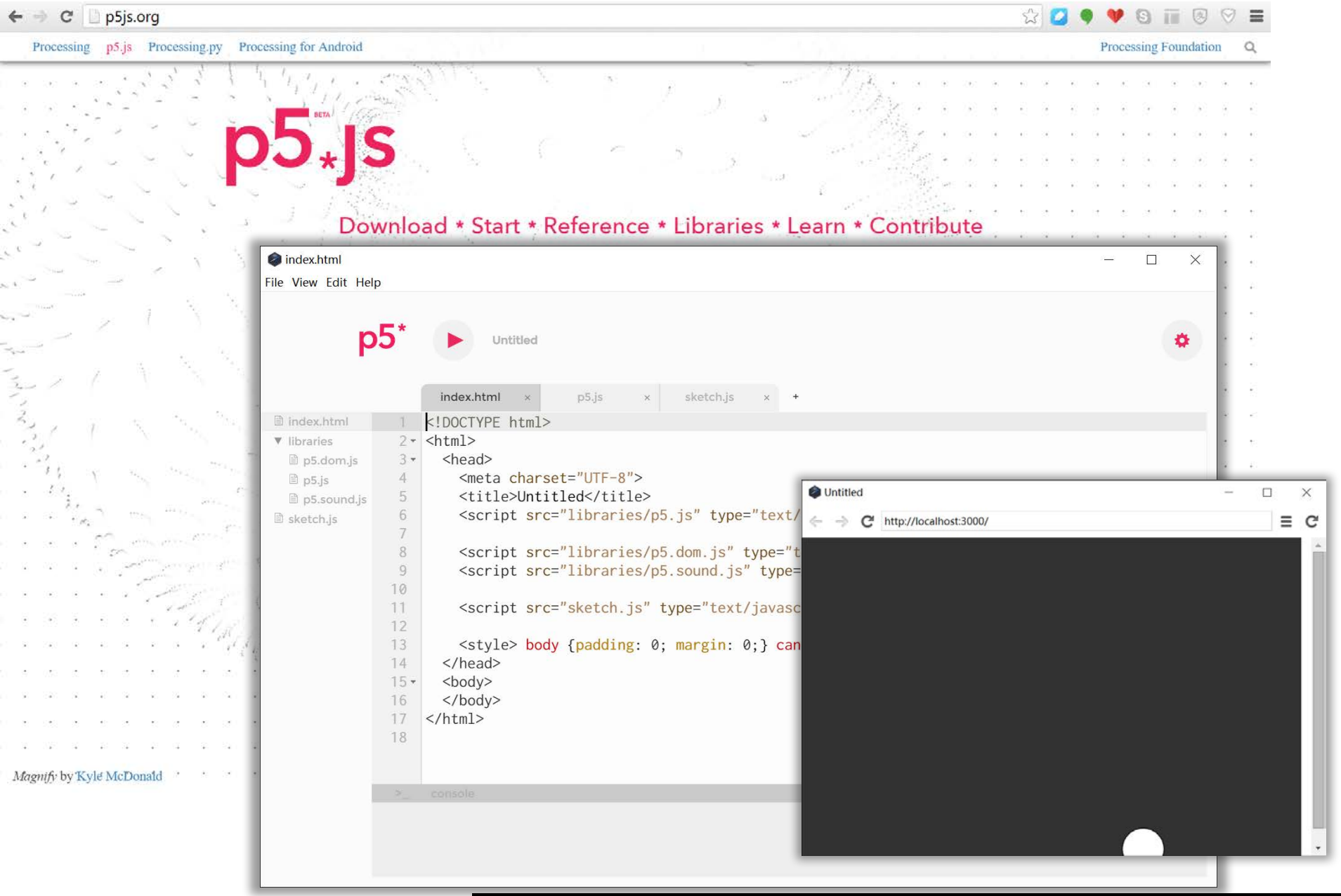
http://www.openprocessing.org/

recent blog post: Visualizing OpenProcessing

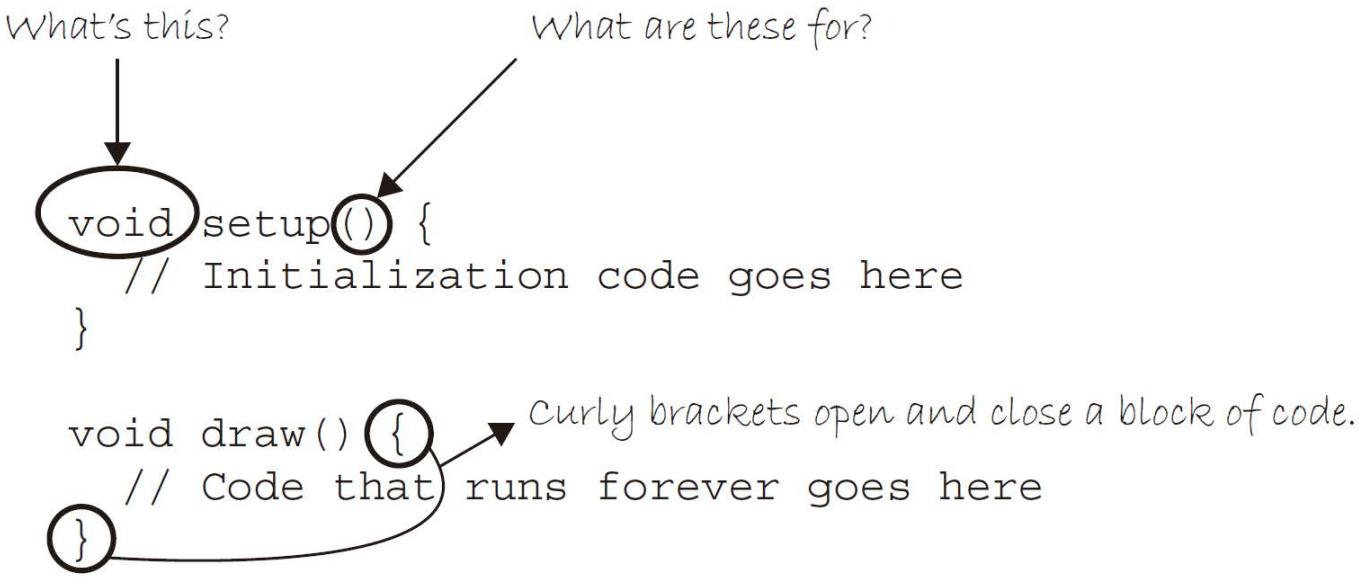
Here is a quick link to a data visualization I

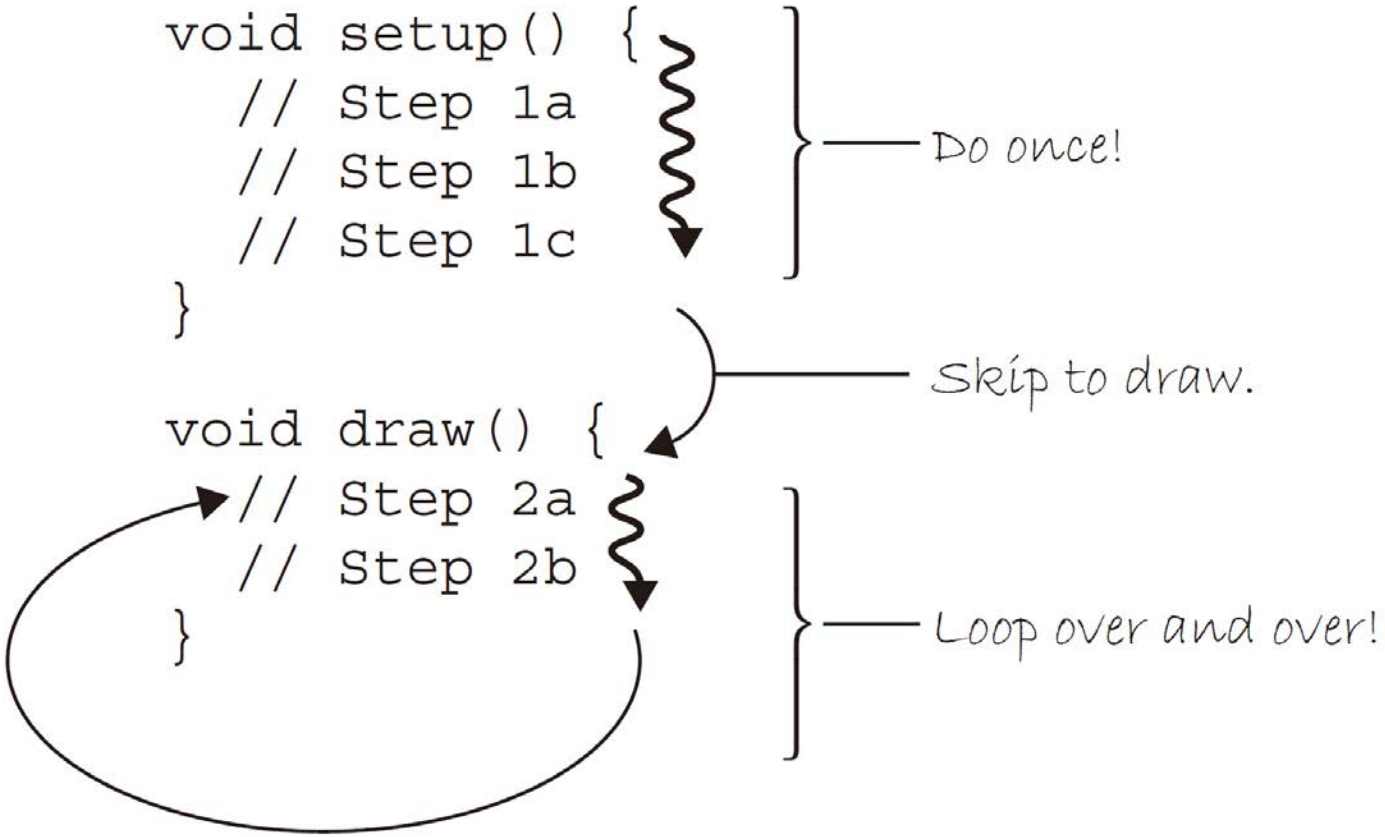
p5*JS开发环境

1-java Applet小程序替代解决方案
2-基于web的创意编程具有更大的发展空间


p5js.org
Processing
p5.js
Processing.py
Processing for Android
Processing Foundation
p5.js
Download * Start * Reference * Libraries * Learn * Contribute
index.html
File View Edit Help
p5*
Untitled
index.html
p5.js
sketch.js
libraries
p5.dom.js
p5.sound.js
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Untitled</title>
<body>
</body>
</html>
console
Untitled
http://localhost:3000/
Magnify by Kyle McDonald

Processing程序编写


What's this?
What are these for?
void setup() {
// Initialization code goes here
}
void draw() {
// Code that runs forever goes here
}
Curly brackets open and close a block of code.

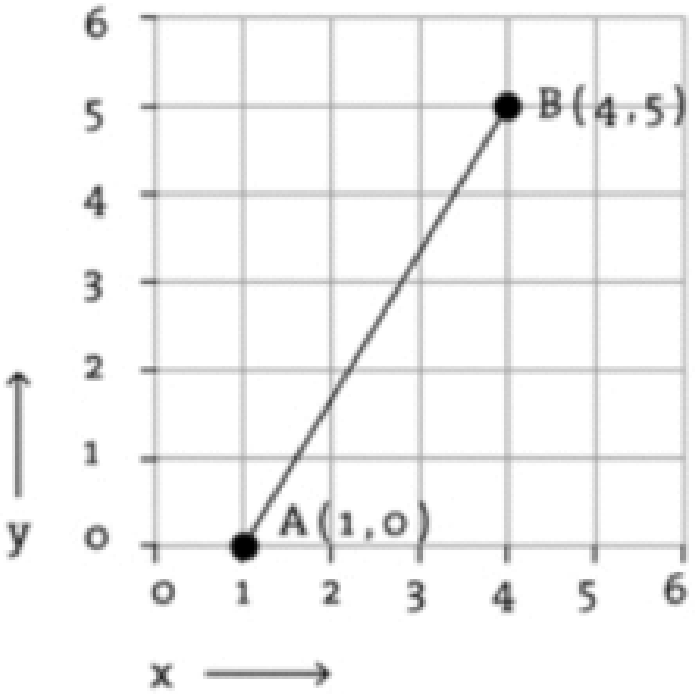

void setup() {
// Step 1a
// Step 1b
// Step 1c
}
Do once!
Skip to draw.
void draw() {
// Step 2a
// Step 2b
}
Loop over and over!

All Primitive Types

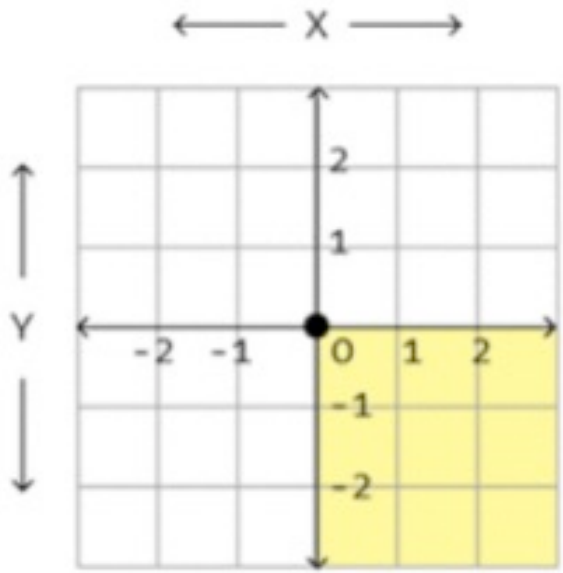
- *boolean* : true or false
- *char* : a character, ‘ a ’ ,‘b ’ ,‘c ’ , etc.
- *byte* : a small number, –128 to 127
- *short* : a larger number, –32768 to 32767
- *int* : a big number, –2147483648 to 2147483647
- *long* : a really huge number
- *fl oat* : a decimal number , such as 3.14159
- *double* : a decimal number with a lot

```
void setup(){

}
```
起始设定(只有第一次执行)

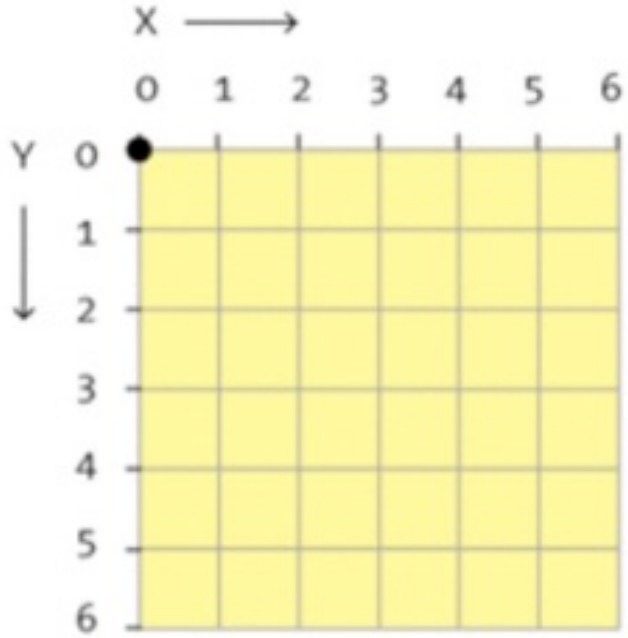
```
void draw(){

}
```
画图执行(重复连续不断)




```
line(1,0,4,5);
```

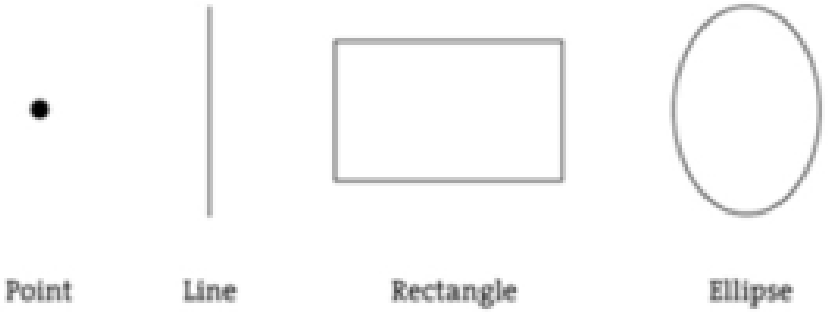


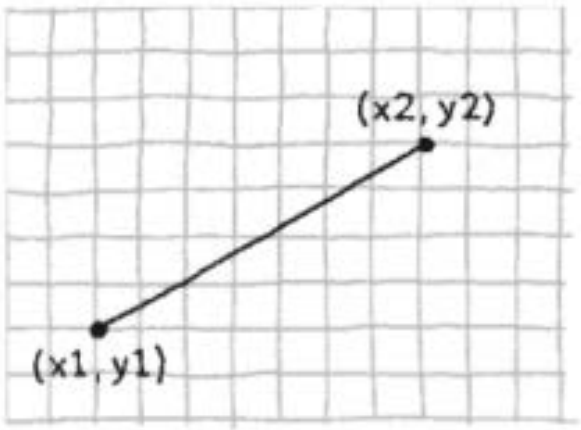
Eighth Grade



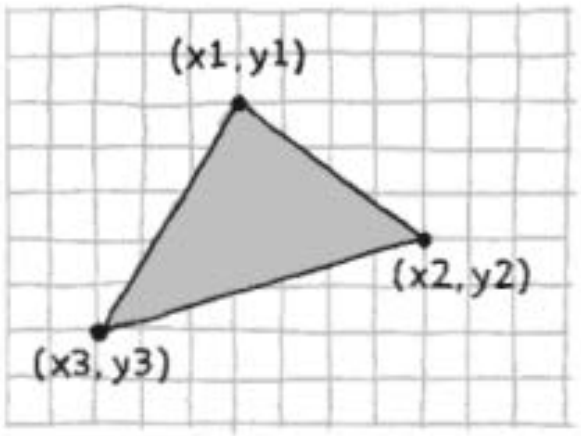
Computer

point (点)
line (直线)
triangle (三角型)
quad (四边形)
rect (矩形)
ellipse (椭圆、圆)
arc (弧形、弧形区域)

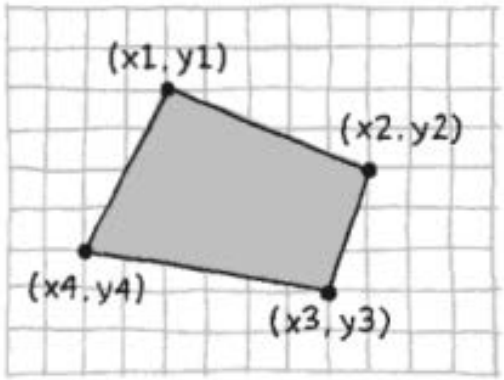




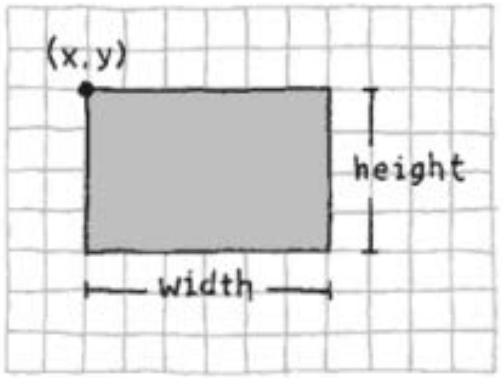

```
line(x1, y1, x2, y2)
```



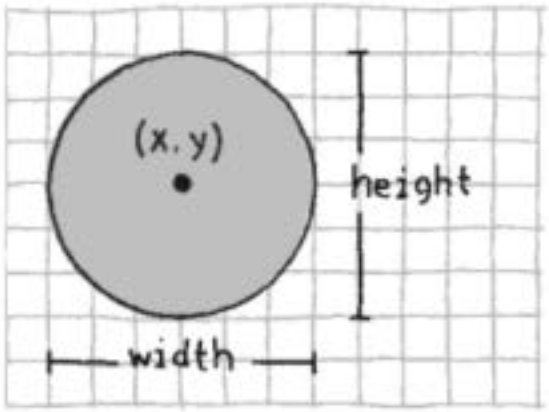

```
triangle(x1, y1, x2, y2, x3, y3)
```



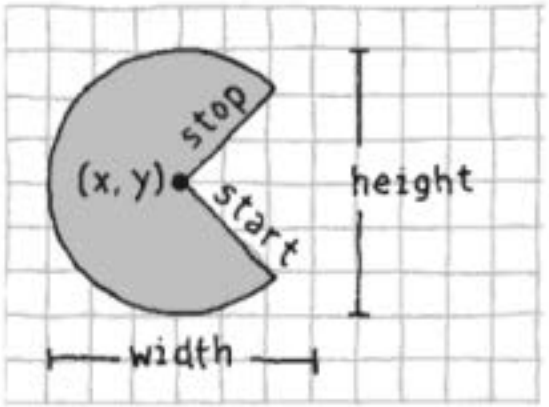

```
quad(x1, y1, x2, y2, x3, y3, x4, y4)
```



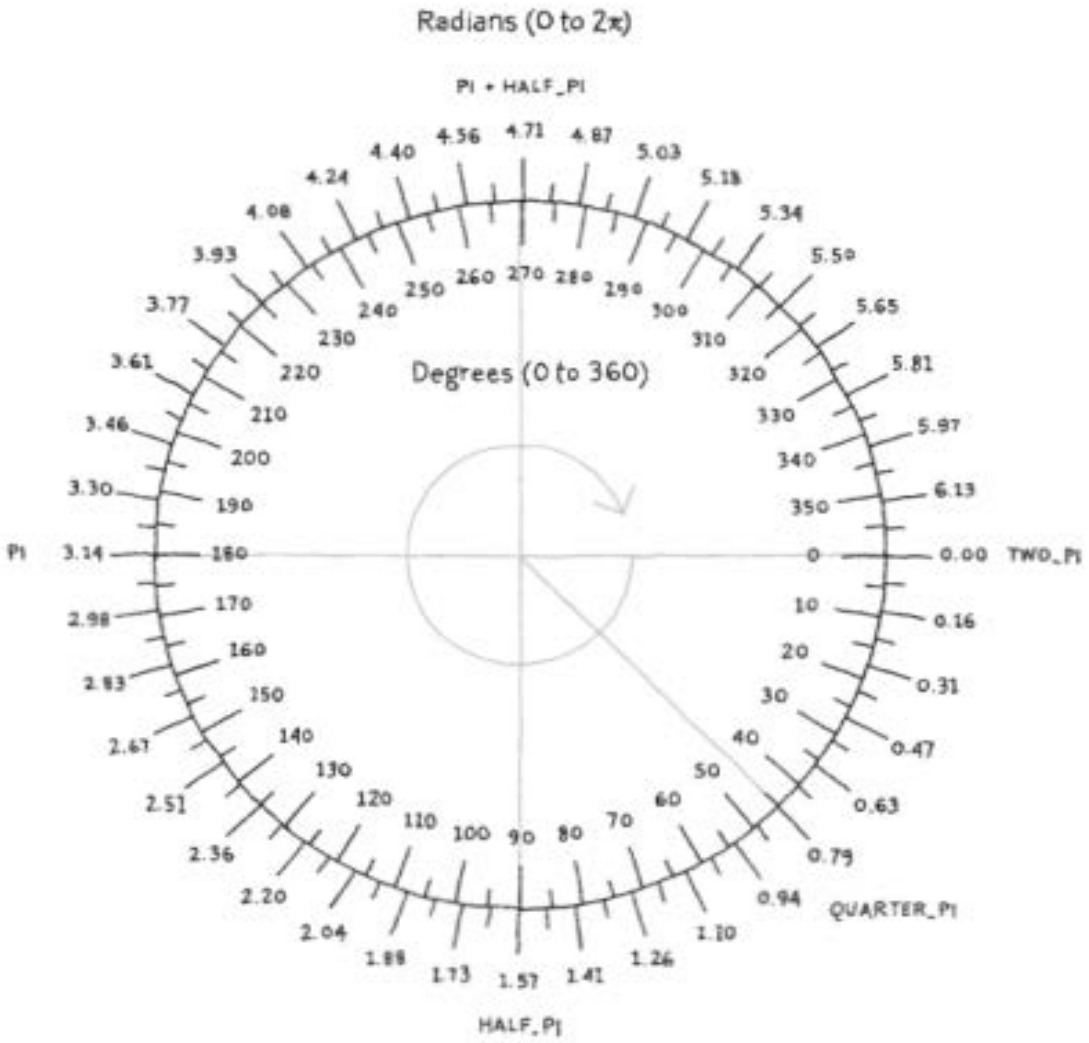

```
rect(x, y, width, height)
```



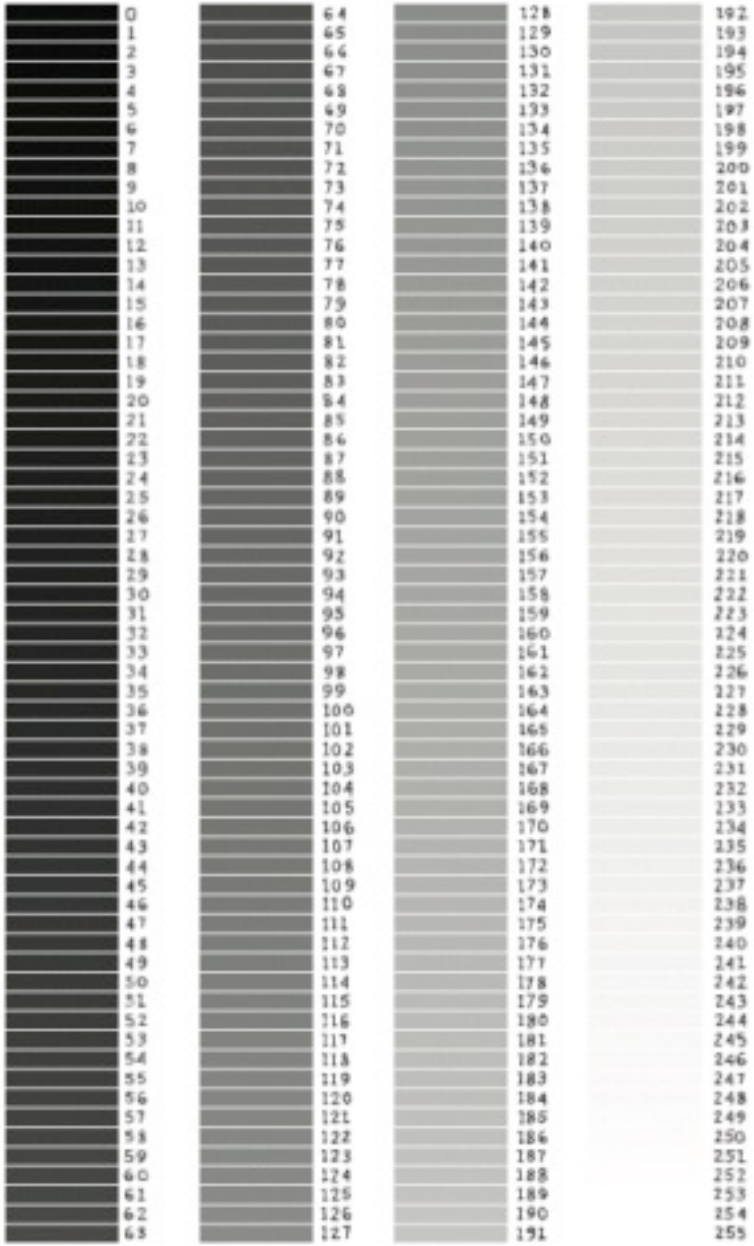

```
ellipse(x, y, width, height)
```

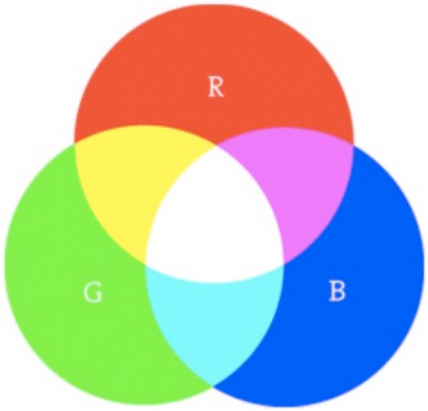


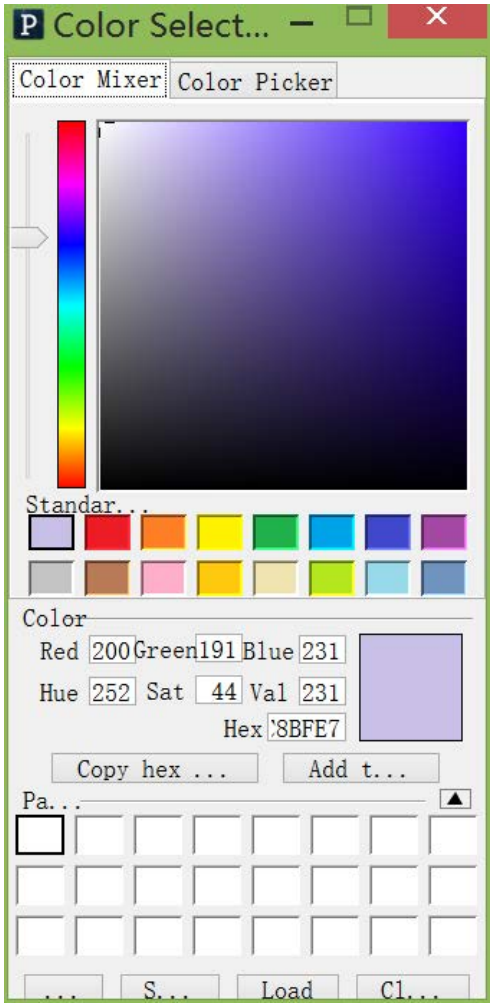

```
arc(x, y, width, height, start, stop)
```

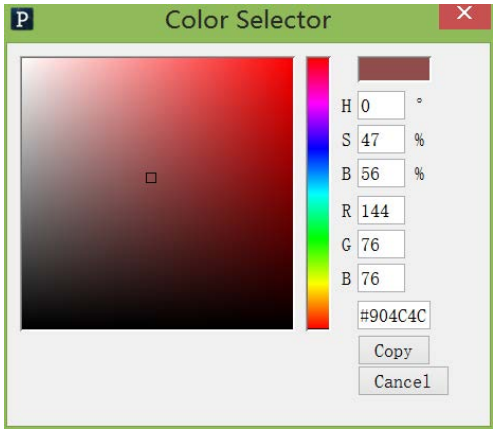

Radians (0 to 2π)
PI + HALF_PI
4.56
4.71
4.87
4.40
5.03
4.24
5.18
4.08
5.34
3.93
5.50
3.77
5.65
3.61
5.81
Degrees (0 to 360)
3.46
5.97
3.30
6.13
PI
3.14
0.00
TWO_PI
2.98
0.16
2.83
0.31
2.67
0.47
2.51
0.63
2.36
0.79
2.20
0.94
QUARTER_PI
2.04
1.10
1.88
1.26
1.73
1.41
1.57
HALF_PI
180
190
200
210
220
230
240
250
260
270
280
290
300
310
320
330
340
350
0
10
20
30
40
50
60
70
80
90
100
110
120
130
140
150
160
170

background()
stroke()
noStroke()
fill()
noFill()

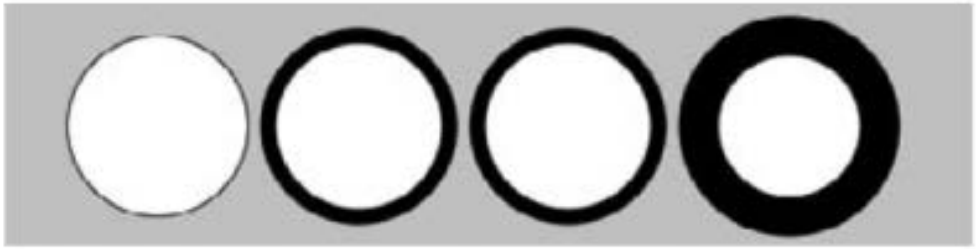



R
G
B

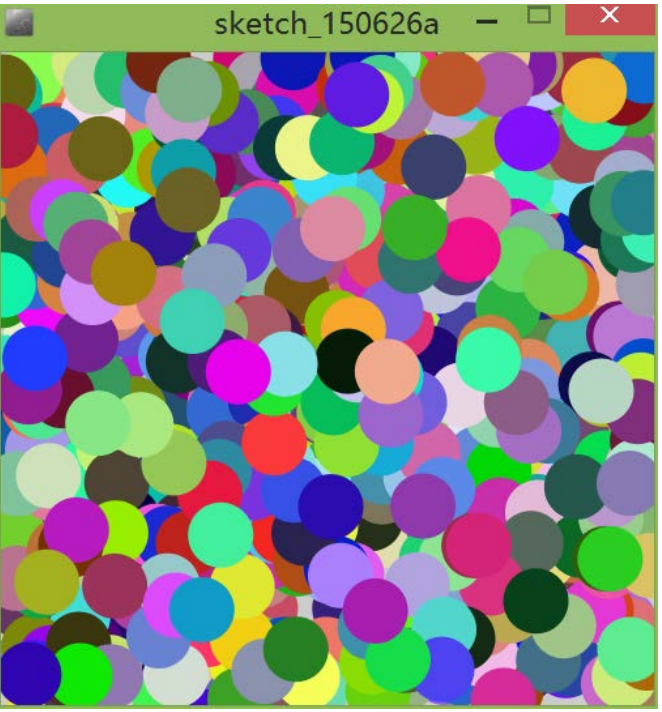

Color Select...
Color Mixer
Color Picker
Standar...
Color
Red 200 Green 191 Blue 231
Hue 252 Sat 44 Val 231
Hex 8BFE7
Copy hex ...
Add t...
Pa...
...
S...
Load
Cl...


Color Selector
H 0 °
S 47 %
B 56 %
R 144
G 76
B 76
#904C4C
Copy
Cancel

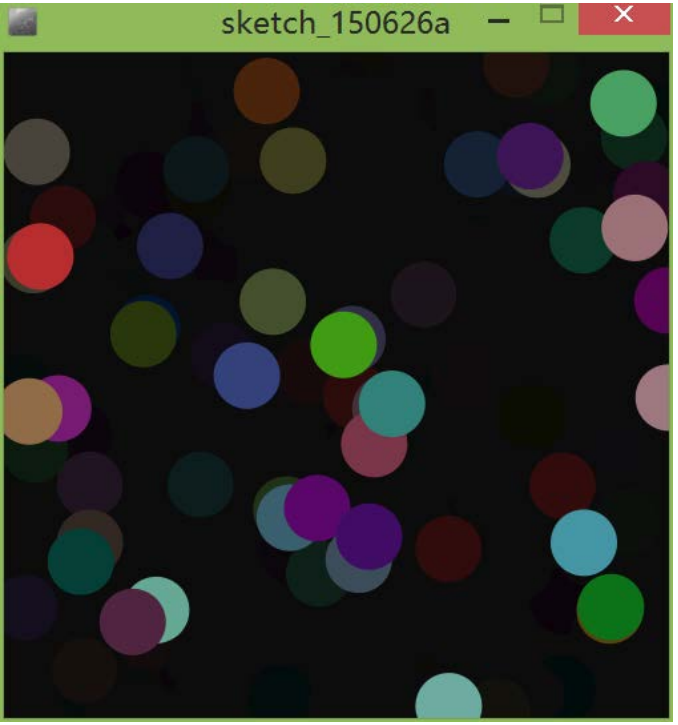
strokeWeight()



```
size(480, 120);
smooth();
ellipse(75, 60, 90, 90);
strokeWeight(8);  // Stroke weight to 8 pixels
ellipse(175, 60, 90, 90);
ellipse(279, 60, 90, 90);
strokeWeight(20);  // Stroke weight to 20 pixels
ellipse(389, 60, 90, 90);
```




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

void draw(){

  //draw something
  noStroke();
  fill(random(255), random(255), random(255));
  ellipse(random(width), random(height), 50, 50);
}
```




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

void draw(){
  noStroke();
  fill(0, 10);
  rect(0, 0, width, height);

  //draw something
  noStroke();
  fill(random(255), random(255), random(255));
  ellipse(random(width), random(height), 50, 50);
}
```

作业要求

利用绘图函数创作，自行设计造型图案

作业提交内容包含：

- word档案（题目、说明、运行画面截图）
- 源程序（*.pde）

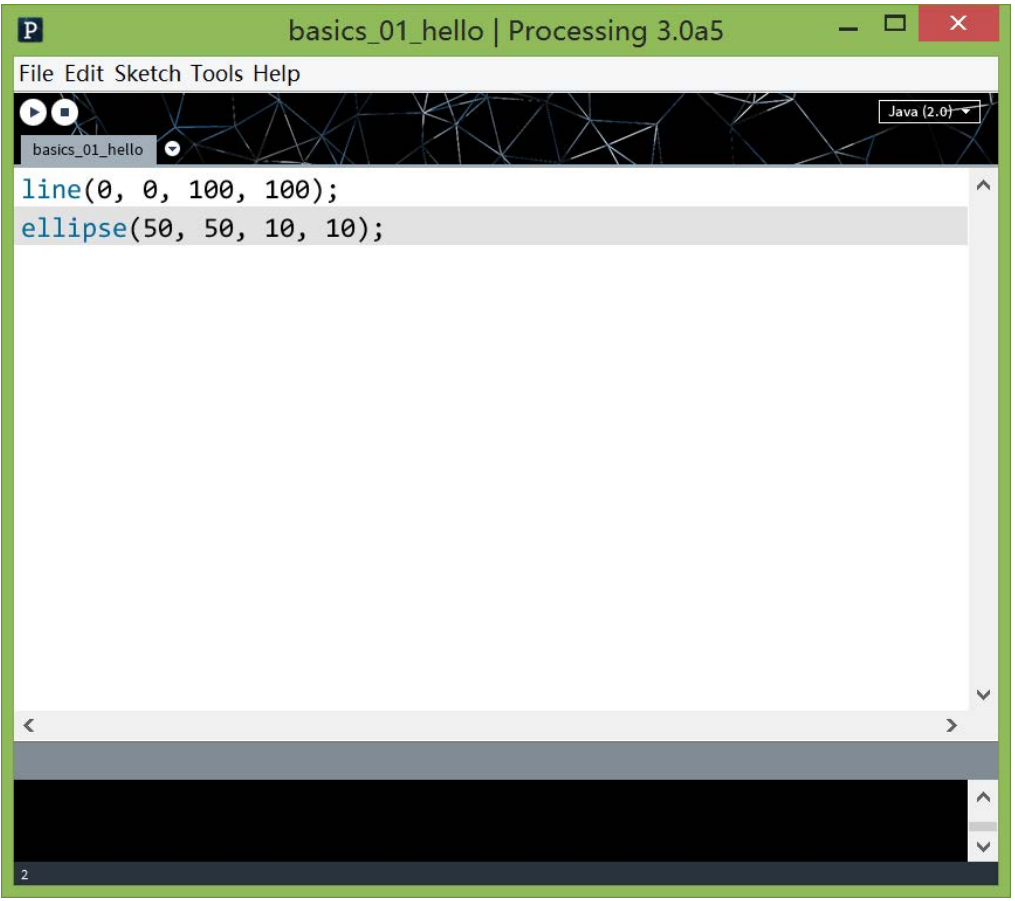
作业缴交

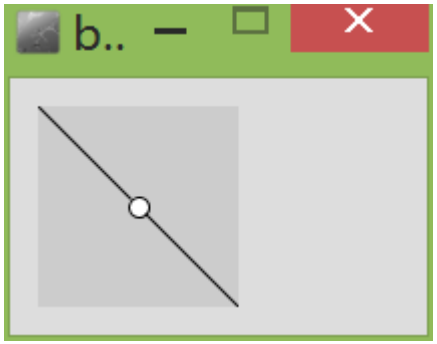
email: lyman988@gmail.com

提交材料命名：

processingWorkshop2015_学号_姓名.zip

basics_01_hello





basics_02_shapes | Processing 3.0a5

File Edit Sketch Tools Help

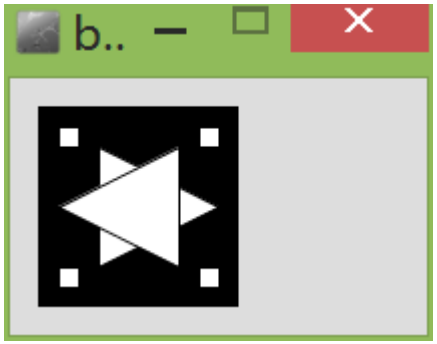
Run

Java (2.0)

basics_02_shapes

```
background(0);
triangle(30, 80, 90, 50, 30, 20);
triangle(100-30, 80, 100-90, 50, 100-30, 20);

rectMode(CENTER);
rect(15, 15, 10, 10);
rect(100-15, 15, 10, 10);
rect(100-15, 100-15, 10, 10);
rect(15, 100-15, 10, 10);
```

1



basics_02b_shapes | Processing 3.0a5

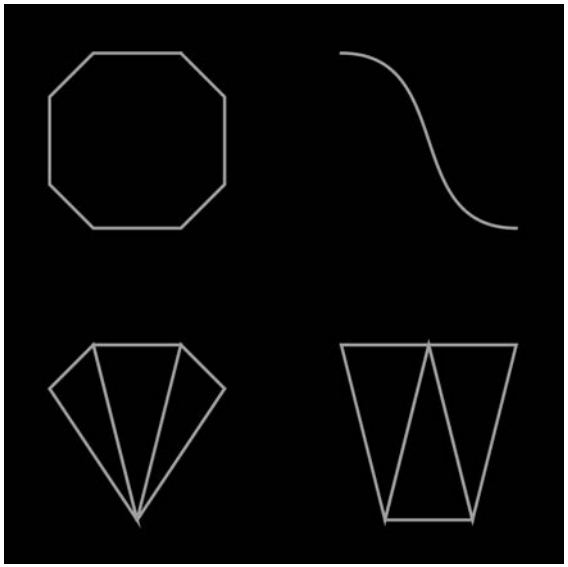
File Edit Sketch Tools Help

Java (2.0)

basics_02b_shapes

```
import processing.pdf.*;
size(200,200);
smooth();

beginRecord(PDF, "shapes.pdf");

background(0);
noFill();
stroke(160);

beginShape();
vertex(65,20);
vertex(80,35);
vertex(80,65);
vertex(65,80);
vertex(35,80);
vertex(20,65);
vertex(20,35);
vertex(35,20);
vertex(65,20);
```

13



basics_03_drawing | Processing 3.0a5

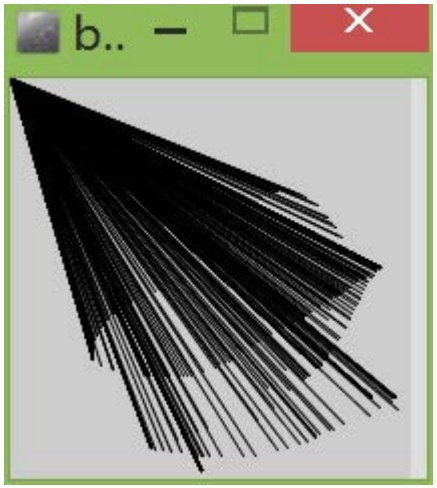
File Edit Sketch Tools Help

Java (2.0)

basics_03_drawing

```
void setup() {
  size(200, 200);
}

void draw() {
  // how can we avoid this big black lump and add some shades?
  stroke(0,0,0);
  line(0, 0, mouseX, mouseY);
}
```

1



basics_04_doodling | Processing 3.0a5

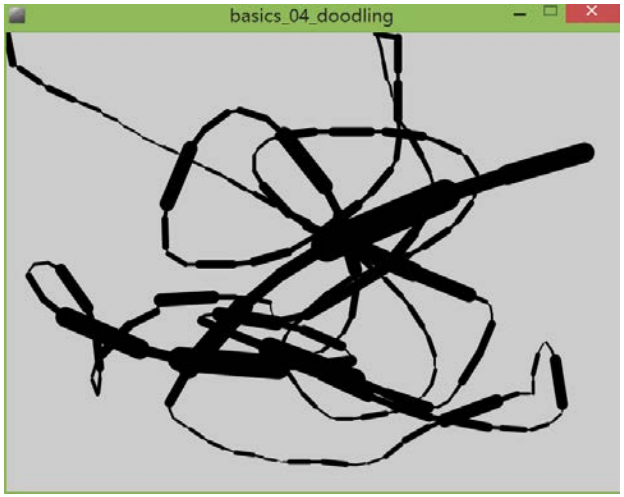
File Edit Sketch Tools Help

Java (2.0)

basics_04_doodling

```
void setup() {
  size(800, 600);
}

void draw() {
  // mouseX & mouseY store the CURRENT mouse position
  // pmouseX & pmouseY keep hold of the PREVIOUS mouse position
  float distance=dist(pmouseX, pmouseY, mouseX, mouseY);
  strokeWeight(distance*0.25);
  line(pmouseX, pmouseY, mouseX, mouseY);
  // uncomment the line below to export each frame
  //saveFrame("doodle-####.tga");
}
```

1



basics_05_movingline | Processing 3.0a5

File Edit Sketch Tools Help

Java (2.0)

basics_05_movingline

```
// this is a variable, it can (only) store numbers
int x=0;

void setup() {
  size(200, 200);
}

void draw() {
  background(255);
  // how about more than a single line?
  line(x, 0, x, height);
  // move it!
  x = x + 1;
  // ...but how to stop it moving outside the screen?
}
```

1

b..

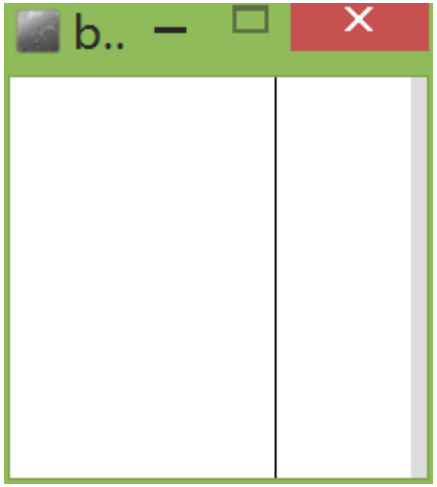
basics_06_conditional_line | Processing 3.0a5

File Edit Sketch Tools Help

Java (2.0)

basics_06_conditional_line

```
// this is a variable, it can (only) store numbers
int x = 0;

void setup() {
  size(200, 200);
}

void draw() {
  background(255);
  // how about more than a single line?
  line(x, 0, x, height);

  // move it! (x++ is shorthand for x=x+1)
  x++;
  // or use the vertical mouse position to control speed
  //x += mouseY;

  // here we check if we have reached the right side of the screen
  // and if so subtract the screen width from x to start again
  // from the left
  if (x > width) {
    x = x - width;
  }
}
```



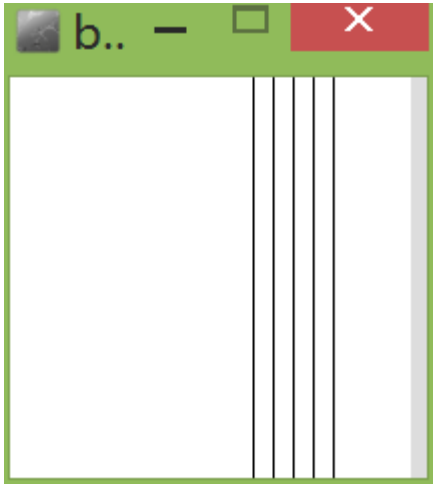
basics_07_loops_n_lines | Processing 3.0a5

File Edit Sketch Tools Help

Java (2.0)

basics_07_loops_n_lines

```
// this is a variable, it can store numbers
int x = 0;

// here we define the desired gap between each line
int gap = 10;

void setup() {
  size(200, 200);
}

void draw() {
  background(255);
  // a loop is a loop...
  // i is a counter variable being increased after each iteration
  // all code within the loop is executed for as long as i is less than 5
  for(int i = 0; i < 5; i++) {
    line(x + i * gap, 0, x + i * gap, height);
  }
  x++;
  if (x > width) {
    x = x - width;
  }
}
```



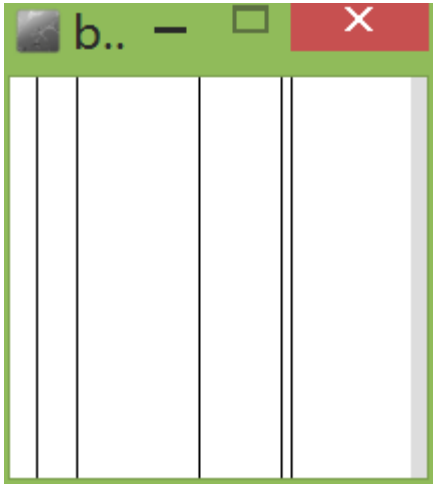
basics_08_arrays | Processing 3.0a5

File Edit Sketch Tools Help

Java (2.0)

basics_08_arrays

```
// now x is an array variable, it can store several numbers
// in this case the x position of each line
// you can recognize arrays from the [] syntax
float[] x = new float[5];

// we also want each line to move at different speeds
// we make the size of this array the same as that of x
float[] speed = new float[x.length];

void setup() {
  size(200, 200);
  // pick random speeds for each line
  for(int i = 0; i < speed.length; i++) {
    // the [] is used to refer to a certain slot/index of the array
    speed[i] = random(0.5, 5);
  }
}

void draw() {
  background(255);
  // now we loop over all array entries
  for(int i = 0; i < x.length; i++) {
    // add the related speed to each position
    x[i] += speed[i];
    // the % operator produces the remainder of a division
    // and in this case can be used instead of the previous if() statement
    // to keep the lines within the defined screen region
    x[i] = x[i] % width;
    // now draw the line at the new position
    line(x[i], 0, x[i], height);
  }
}
```



basics_09_objects | Processing 3.0a5

File Edit Sketch Tools Help

Java (2.0)

basics_09_objects | line

```
// now our array is not storing numbers but lines directly
// we can define new custom types like this via classes
Line[] lines = new Line[5];

void setup() {
  size(200, 200);
  // create instances of our new Line class
  for(int i = 0; i < lines.length; i++) {
    lines[i] = new Line();
  }
}

void draw() {
  background(255);
  // now we loop over all array entries
  for(int i = 0; i < lines.length; i++) {
    lines[i].update();
    lines[i].draw();
  }
}
```

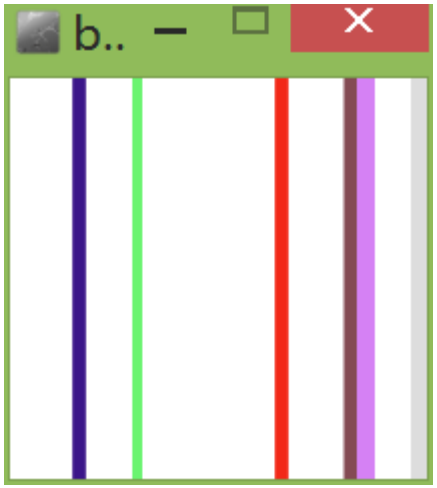
basics_09_objects | Processing 3.0a5

File Edit Sketch Tools Help

Java (2.0)

basics_09_objects | line

```
class Line {
  float position;
  float weight;
  float speed;
  color col;

  Line() {
    weight = random(1,10);
    speed = random(0.5, 5);
    col = color(random(255), random(255), random(255));
  }

  void update() {
    position += speed;
    position %= width;
  }

  void draw() {
    strokeWeight(weight);
    stroke(col);
    line(position, 0, position, height);
  }
}
```



basics_09b_objects | Processing 3.0a5

File Edit Sketch Tools Help

Java (2.0)

basics_09b_objects | line

```
// now our array is not storing numbers but lines directly
// we can define new custom types like this via classes
Line[] lines = new Line[10];

void setup() {
  size(200, 200);
  // create instances of our new Line class
  for(int i = 0; i < lines.length; i++) {
    lines[i] = new Line();
  }
}

void draw() {
  background(255);
  // now we loop over all array entries
  for(int i = 0; i < lines.length; i++) {
    lines[i].update();
    lines[i].draw();
  }
}
```

19

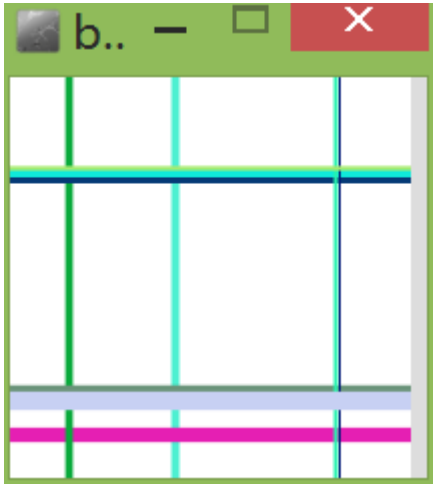
basics_09b_objects | Processing 3.0a5

File Edit Sketch Tools Help

basics_09b_objects | line

```
class Line {
  float position;
  float weight;
  float speed;
  color col;

  // a boolean value can only either be true or false
  // it's equivalent to a switch/flag
  boolean isVertical;

  Line() {
    weight = random(1,10);
    speed = random(0.5, 5);
    col = color(random(255), random(255), random(255));
    // here we decide if the line is vertical or horizontal
    // based on the outcome of the condition below
    // we are basically asking this:
    // is the random number picked less than 50? true or false...
    isVertical = (random(100) < 50);
  }

  void update() {
    position += speed;
    if (isVertical) {
      position %= width;
    }
    else {
      position %= height;
    }
  }

  void draw() {
    strokeWeight(weight);
    stroke(col);
    if (isVertical) {
      line(position, 0, position, height);
    }
    else {
      line(0, position, width, position);
    }
  }
}
```

33



Shapes3D | Processing 3.0

File Edit Sketch Tools Help

Shapes3D Shape Triangle

```
import toxi.util.datatypes.*;
import toxi.math.noise.*;
import toxi.math.waves.*;
import toxi.geom.*;
import toxi.math.*;
import toxi.math.conversion.*;

Shape[] lines=new Shape[1000];

void setup() {
  size(800,600,OPENGL);
  noStroke();
  for(int i=0; i<lines.length; i++) {
    if (random(100)<50) {
      lines[i]=new Triangle();
    }
    else {
      lines[i]=new Shape();
    }
  }
}

void draw() {
  background(255);
  translate(width/2,height/2,0);
  rotateX(mouseY*0.01);
  rotateY(mouseX*0.01);
  scale(0.25);
  for(int i=0; i<lines.length; i++) {
    lines[i].update();
    lines[i].render();
  }
}
```

Shapes3D | Processing 3.0a5

File Edit Sketch Tools Help

Shapes3D Shape Triangle

```
class Shape {
  Vec3D position;
  Vec3D speed;
  color col;
  int age,maxAge;

  Shape() {
    position=new Vec3D();
    speed=Vec3D.randomVector().scale(random(1,5));
    col=color(random(255),random(255),random(255));
    maxAge=(int)random(500,1000);
  }

  void update() {
    position.addSelf(speed);
    age++;
    if (age==maxAge) {
      position.clear();
      age=0;
    }
  }

  void render() {
    fill(col);
    pushMatrix();
    translate(position.x,position.y,position.z);
    rect(0,0,50,50);
    popMatrix();
  }
}
```

Shapes3D | Processing 3.0a5

File Edit Sketch Tools Help

Shapes3D Shape Triangle Java (2.0)

```
class Triangle extends Shape {

  Vec3D a,b,c;

  Triangle() {
    a=Vec3D.randomVector().scale(50);
    b=Vec3D.randomVector().scale(50);
    c=Vec3D.randomVector().scale(50);
  }

  void render() {
    fill(col);
    pushMatrix();
    translate(position.x,position.y,position.z);
    beginShape(TRIANGLES);
    vertex(a.x,a.y,a.z);
    vertex(b.x,b.y,b.z);
    vertex(c.x,c.y,c.z);
    endShape();
    popMatrix();
  }
}
```