编程,Processing&绘图
PROGRAMMING, PROCESSING & DRAWING

Day_01_4

编程 | PROGRAMMING

语言｜HUMAN LANGUAGES

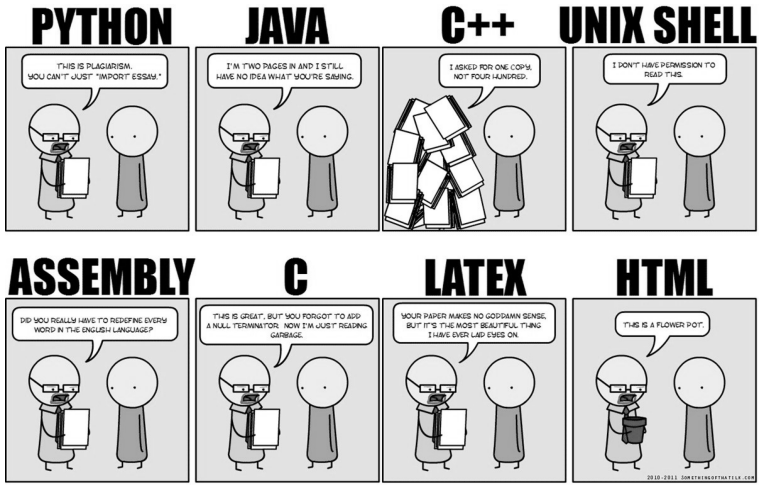
Human languages are the combination of sounds, gestures or symbols designed to communicate abstract ideas through symbolic representation.

人类语言由声音、手势以及符号去表达抽象的意义

程序语言｜PROGRAMMING LANGUAGES

Programming languages are a type of written language designed to communicate instructions to a computer.

程序语言指的是一种文字语言用以人和计算机沟通



逻辑｜LOGIC

Logic, a system of strict principles used to determine the validity of a statement or the outcome of an argument, provides the foundation for many computational and programming methodologies.

逻辑是计算机程序实现的基础

语法 | SYNTAX

The correct structure of a language is determined by establishing rules, called syntax.Syntax is the arrangement of words, phrases, and punctuation that, when used appropriately, create meaning.

Each programming language has unique rules of syntax, but there are also many common rules shared between various languages.

语法指特定语言的结构规则，它指定了词汇顺序，标点用法与位置，只有语法正确是表述的基础。不同的程序语言由着自己的特定语法。

```
def add5(x):
   return x+5

def dotwrite(ast):
   nodename = getNodename()
   label=symbol.sym_name.get(int(ast[0]),ast[0])
   print '    %s [label="%s' % (nodename, label),
   if isinstance(ast[1], str):
      if ast[1].strip():
         print '= %s"];' % ast[1]
      else:
         print '"]'
   else:
      print '"];'
      children = []
      for  in n, childenumerate(ast[1:]):
         children.append(dotwrite(child))
      print ,'    %s -> {' % nodename
      for  in :namechildren
         print '%s' % name,
```

标点｜PUNCTUATION

Punctuation is critically important in programming.

The following punctuation is commonly used:

- ( ) (parentheses)
- , (comma)
- . (dot or period)
- ; (semicolon)
- = (equals)
- [ ] (square braces)
- { } (curly braces)

序列｜SEQUENCE

Computer programs have linear and non- linear characteristics.
For the time being, think of a program as linear or sequential, meaning that the computer will execute each line of code in sequence, from the beginning until the end. In time you will learn to take advantage of non-linear techniques.

PROCESSING

PROCESSING

Processing is a programming language, development environment, and online community.

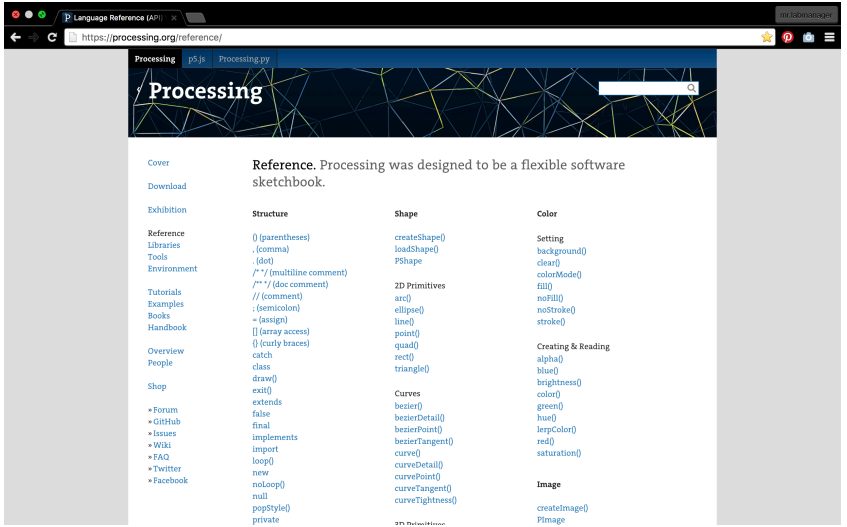
Processing是一种开源编程语言，专门为电子艺术和视觉交互设计而创建，其目的是通过可视化的方式辅助编程教学，并在此基础之上表达数字创,Processing也指Processing语言的集成开发环境(IDE)
2001年，MIT媒体实验室的 Casey Reas 和Benjamin Fry发起了此计划。其固定目标之一便是作为一个有效的工具，通过激励性的可视化反馈帮助非程序员进行编程的入门学习。Processing语言建立在Java语言的基础之上，但使用简化的语法和图形编程模型

参考文档｜PROCESSING REFERENCES

The processing IDE has a built-in language reference where you can explore the features and syntax of the processing language.

The language reference is also available online:

- processing.org/reference/

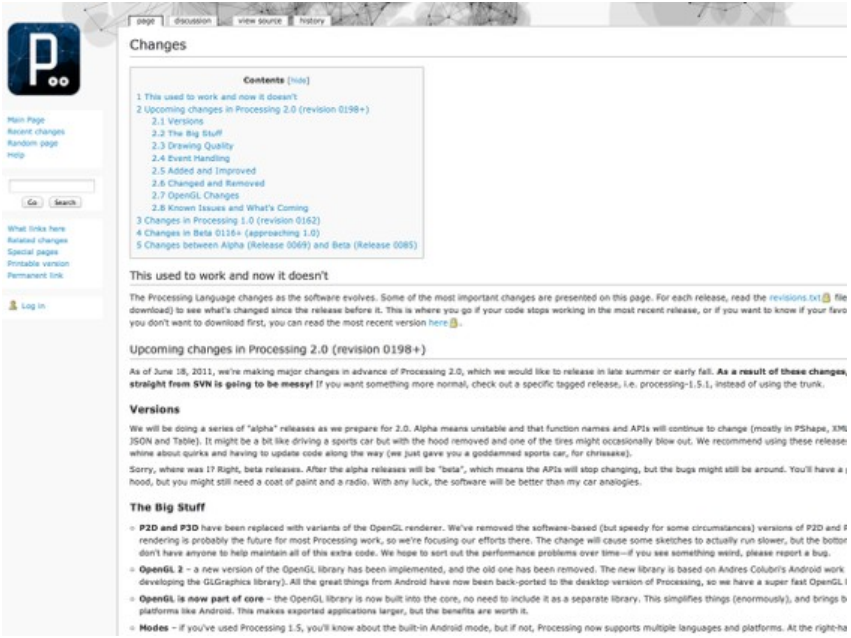


3.X更新｜PROCESSING 3.X CHANGES

The books we are using were written for Processing 1.x however we will be working with processing 3.x.

Changes between Processing versions have been documented on the Processing Wiki:

- wiki.processing.org/w/Changes



SCREENSHOT OF PROCESSING.ORG

JAVA

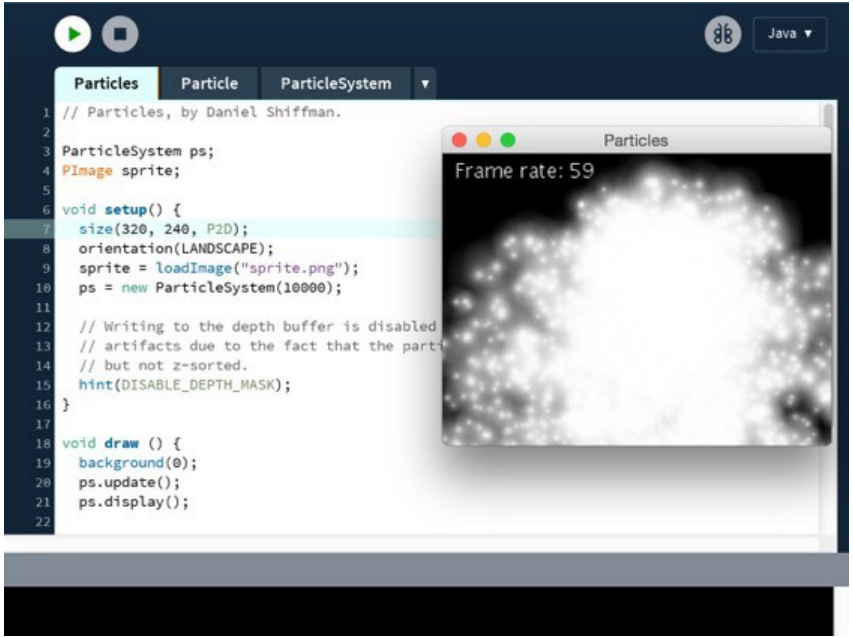
Java is a programming language, developed by Sun Microsystems and currently owned by Oracle, designed to run on any computing platform through a Java Virtual Machine (JVM) regardless of computing architecture.

Java是一种计算机编程语言，拥有跨平台、面向对象、泛型编程的特性，广泛应用于企业级Web应用开发和移动应用开发

开发环境｜PROCESSING IDE

The Processing IDE is designed for writing, playing, testing and debugging, as well as publishing programs written in the Processing language.

It has menus, a toolbar, a text ditor, message area and console.



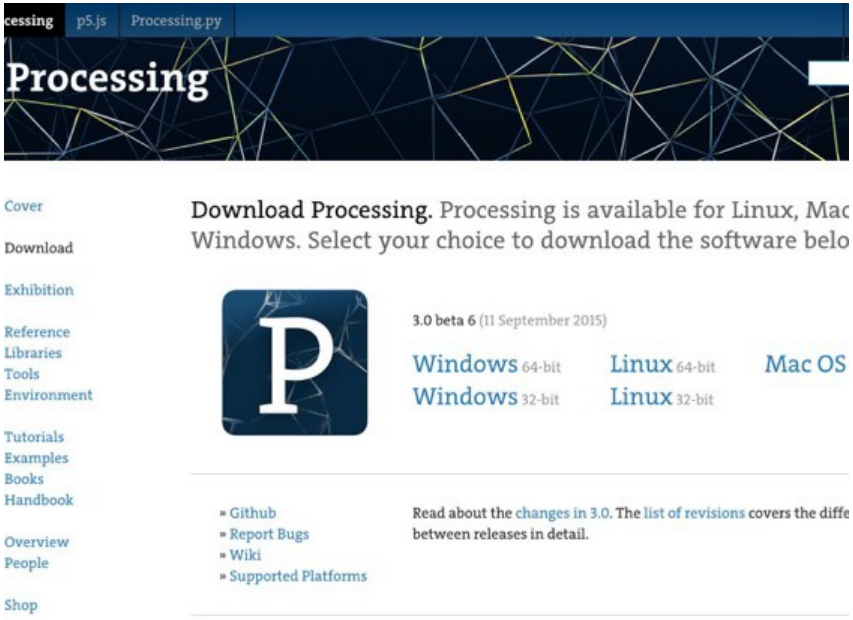
安装开发环境
INSTALLING THE PROCESSING IDE

Visit the following URL and download Processing 3.X :

- processing.org/download/

Depending on the configuration of your operating system, you may also need to install the Java Runtime Environment (JRE) from one of these websites:

- oracle.com
- support.apple.com/kb/DL1572



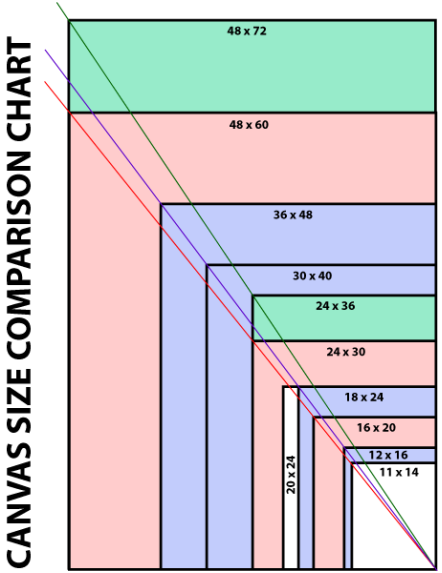
编写｜WRITING SKETCHES

Programs written in Processing are called sketches.

The word sketch calls to mind a quick and loose way of working.

You type your sketches in the Processing IDE.

基本函数丨SIZE(100，100)；



The first thing you'll want to do when creating a new sketch is to establish the dimensions of the window your sketch will playback in.

You can accomplish this by using the size() function.
Size takes 2 values, a width and a height.

在一切开始之前先确定窗口尺寸：
要求连个变量，默认值是100，100。

运行｜RUNNING SKETCHES

To play your sketch press the Run button in the toolbar. Don’t wait until you think you are finished to run your software.

It is useful to try your sketch out repeatedly during the development process.

运行Processing Sketch，点击左上角运行按钮，或者快捷键Command ＋ R（control ＋ R / windows）

备注｜COMMENTING SKETCHES

Comments are not only a way for you to provide personal notes within your
program, they can also be used to exclude portions of your code from running.

Single line comments begin with double slashes (//).

Multiline comments begin with a slash then an asterisk (/*), and end the opposite way with an asterisk then a slash (*/).

发布｜PUBLISHING SKETCHES

There are several ways to publish your sketches:

- Creating a Java executable is the ideal means of playback.
- Sketches can also be deployed to Android phones and tablets.
- Sketches can also be converted to Javascript for deployment on the web, however not all functions are supported with this method.

DRAWING

绘图｜DRAWING

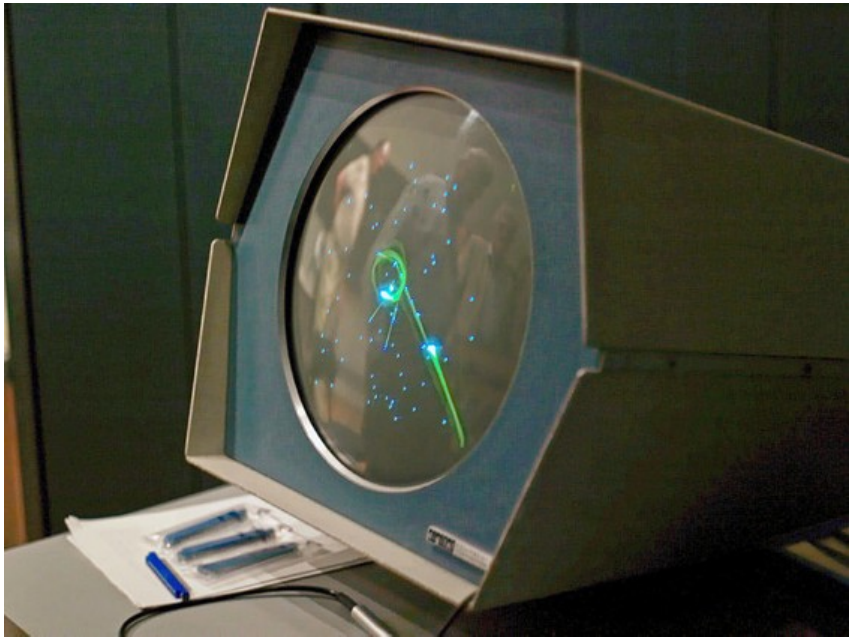
Drawing is a visual art that relies on the use of lines and tones to create an illusion (realistic or abstract picture) on a two- dimensional surface.

计算机图形｜COMPUTER GRAPHICS

Computer graphics are the generation and presentation of images on a computer display.

There are two methods for efining computer graphics:

Raster
Vector



位图｜RASTER GRAPHICS

Raster graphics, sometimes referred to as bitmaps, are defined as a grid (matrix) of pixels on a computer display or dots on a printed page.

Rasters are best suited for images that are photographic in nature.

However, rasters suffer from loss of quality when enlarged.

矢量图｜VECTOR GRAPHICS

Vector graphics define images as a set of points, lines and geometric shapes.

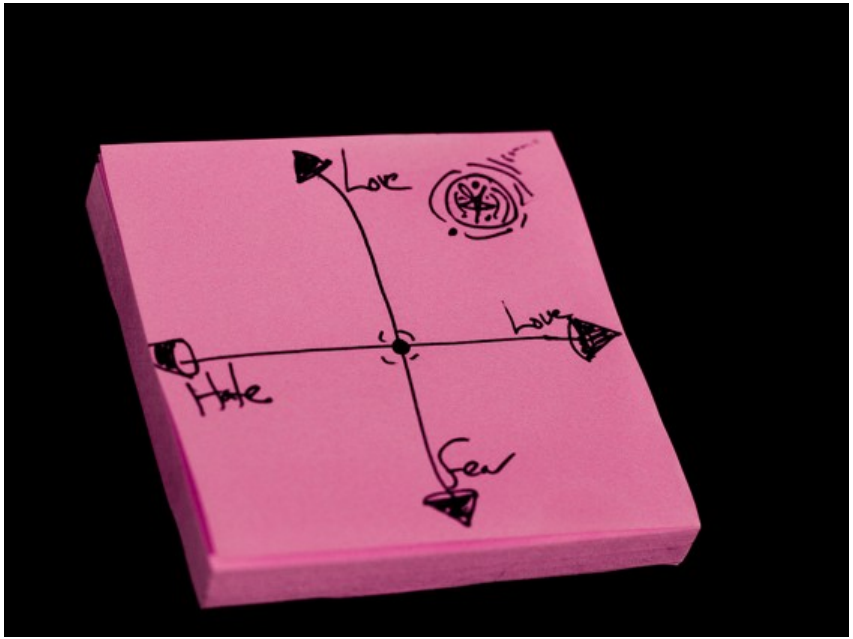
Vectors can scale cleanly to any size without suffering a loss of quality.
For this reason, vectors are best suited for illustrations including logos and typography.

坐标系｜COORDINATE SYSTEMS

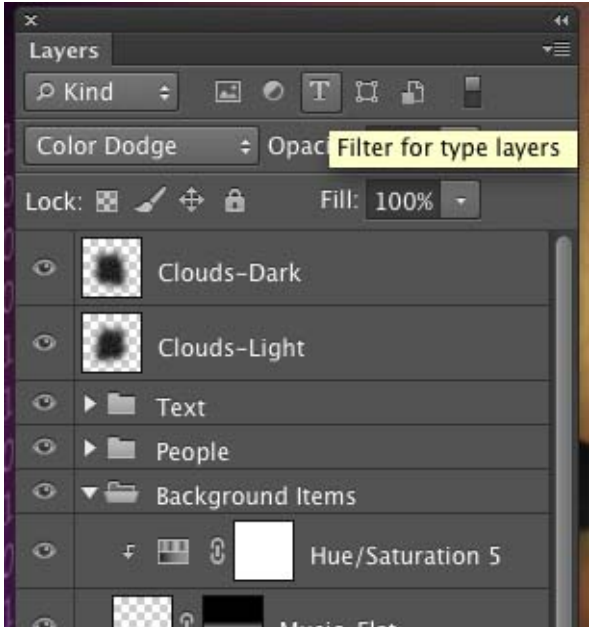
Coordinate systems are used to uniquely determine the position of something within a space.

The coordinate system used in computer graphics differs from the Cartesian
coordinate system used in mathematics.

The Computer graphics coordinate system begin (0,0) at the top left corner instead of in the center.



绘图顺序&图层
DRAWING ORDER & LAYERING



Graphic elements like shapes, images, and text are created one at a time in your program.

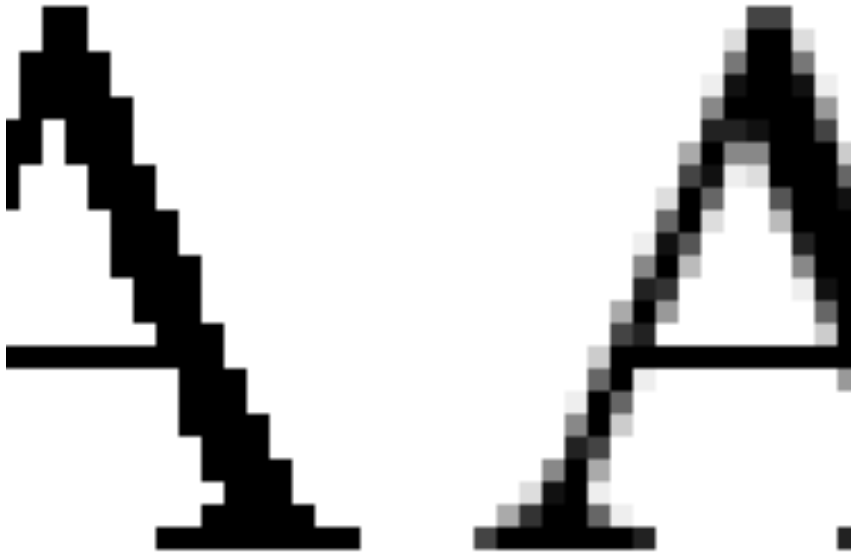
A graphic element created later in your program will appear on top of an element created earlier in your program if they occupy the same coordinates.

抗锯齿｜SMOOTHING

Computer graphics systems use a process called anti-aliasing to smooth jagged transitions between graphic elements that would otherwise be visible on a low resolution computer display.

This smoothing can be turned on or off in Processing.

Having smoothing on generally looks better, but it can have negative consequences for performance.



图元｜2D PRIMITIVES

Processing has commands for drawing simple shapes called 2D Primitives.

To draw a circle you would use the ellipse() function, passing in the x and y coordinates as well as the width and height.

The drawing functions require that different values be provided, corresponding to various attributes of the shape, such as position or size.

图元｜2D PRIMITIVES

The types of shapes that can be drawn using Processing’s built-in 2d Primitive Shape drawing functions are:

- Points
- Lines
- Curves
- Triangles
- Rectangles
- Quadrilaterals
- Ellipses
- Arcs

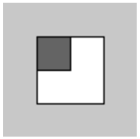
绘图模式｜DRAWING MODE

You can use the rectMode() or ellipseMode() functions to control where the point of origin will be for shapes that you draw.

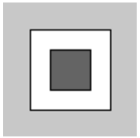
The options for rectMode() and ellipseMode() are CORNER, CORNERS, CENTER, or RADIUS.

Name rectMode()

Examples



```
rectMode(CORNER);  // Default rectMode is CORNER
fill(255);  // Set fill to white
rect(25, 25, 50, 50);  // Draw white rect using CORNER mode

rectMode(CORNERS);  // Set rectMode to CORNERS
fill(100);  // Set fill to gray
rect(25, 25, 50, 50);  // Draw gray rect using CORNERS mode
```



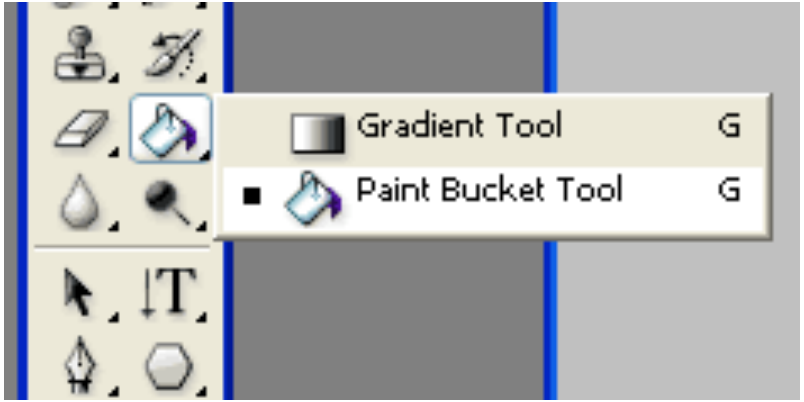
```
rectMode(RADIUS);  // Set rectMode to RADIUS
fill(255);  // Set fill to white
rect(50, 50, 30, 30);  // Draw white rect using RADIUS mode

rectMode(CENTER);  // Set rectMode to CENTER
fill(100);  // Set fill to gray
rect(50, 50, 30, 30);  // Draw gray rect using CENTER mode
```

描边&填充｜STROKE & FILL

Vector shapes can be defined with a stroke, sometimes called a path or outline, and a fill, or interior color.

To set the stroke color use the stroke() function.

To set the fill color use the fill() function.



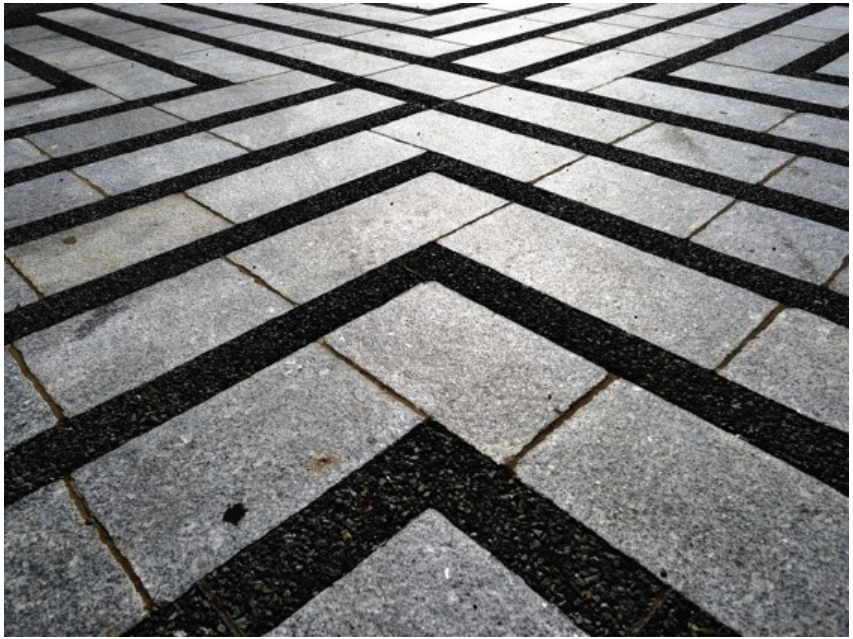
描边宽度｜STROKE WEIGHT

The stroke, or outline of a shape, can be thin, thick or anything in between.

Strokes can also be hidden completely.

Set the stroke weight by using the strokeWeight() function.

Use the noStroke() function to remove the stroke altogether.



COLOR

Color can be set in Processing by several different methods using different color systems and forms of notation:

- Grayscale
- RGB Color
- Hexadecimal Color

PHOTO BY
NATHAN RUPERT

灰度图｜GRAYSCALE

Grayscale is a color system used in computer graphics which defines images based on a range of gray shades from white to black.

Grayscale colors are defined using a single value from 0 to 255.

RGB COLOR

The RGB color system is an additive color system in which three primary colors (red, green and blue) are mixed to create all available colors.

RGB colors can be defined using three values, one each for red, green, and blue between 0 and 255, or by using a hexadecimal triplet.

HEXADECIMAL COLOR

Color can also be expressed using hexadecimal numbers, from 00 to FF.

Hexadecimal triplets are six-digit three-byte values prefixed by a hash mark (#) or 0x.

Some hexadecimal color values:

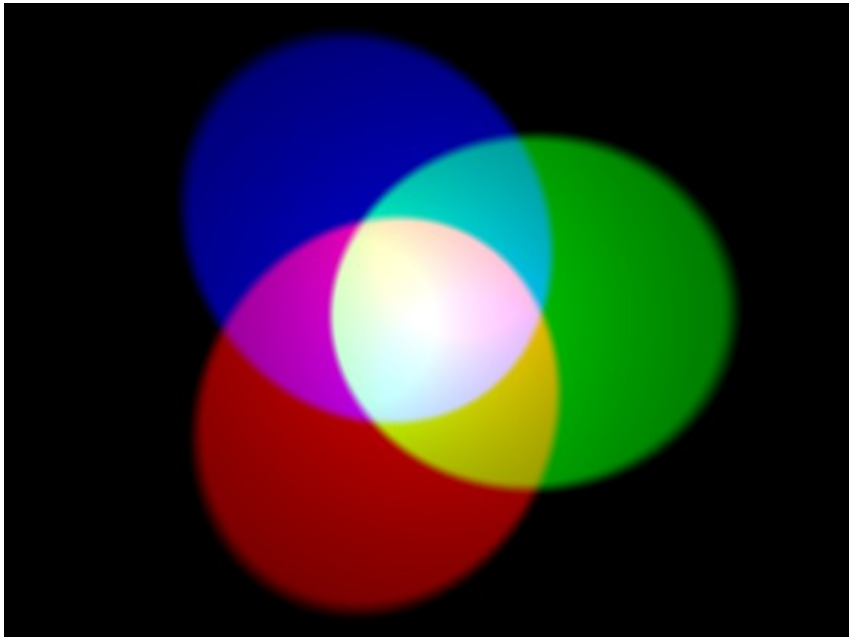
- #000000
- #0000FF
- #FFCC00

COLOR MIXING

Color mixing involves adding the three primary colors, red, green, and blue in the case of light (additive color) and cyan, magenta, and yellow in the case of pigment (subtractive color), to create secondary
and tertiary colors.

Refer to a color wheel or use a color picker if you need a visual aid:

- color.adobe.com



BACKGROUND COLOR

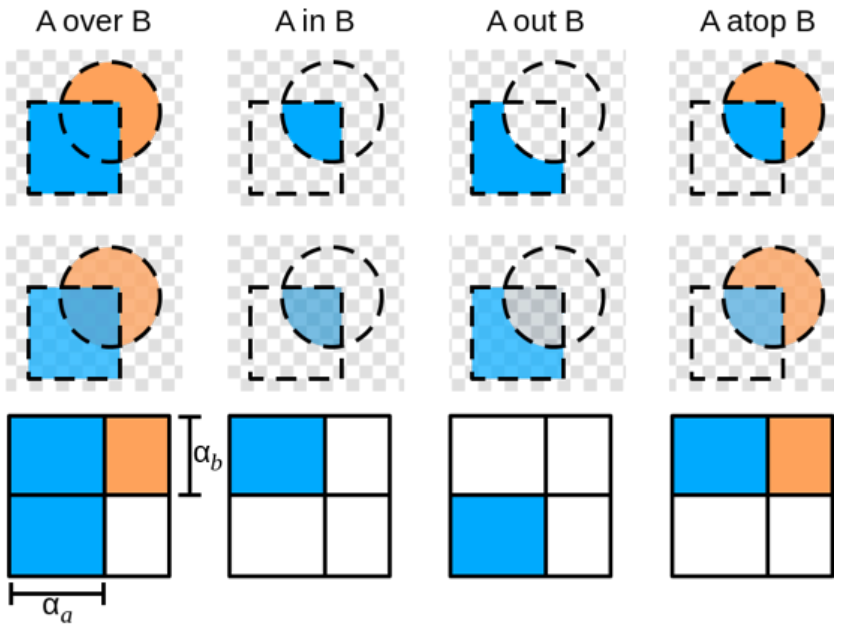
You can define the background color for your sketch by using the background()
function.

Keep in mind that the background() function clears the entire window and fills it with the background color specified.

TRANSPARENCY / OPACITY / ALPHA

Transparency and opacity are inverse qualities that express the ability to see through something to reveal whatever lies behind.

This property can be controlled by specifying the alpha value after the color itself.



COLOR EXAMPLE

```
size(200, 200);
background(150, 0, 150);
stroke(150);
fill(255, 0, 255);
rect(10, 10, 55, 55);
noStroke();
fill(0, 255, 0, 150);
rect(100, 10, 55, 55);
```