

Processing

Data Visualization Programming Language

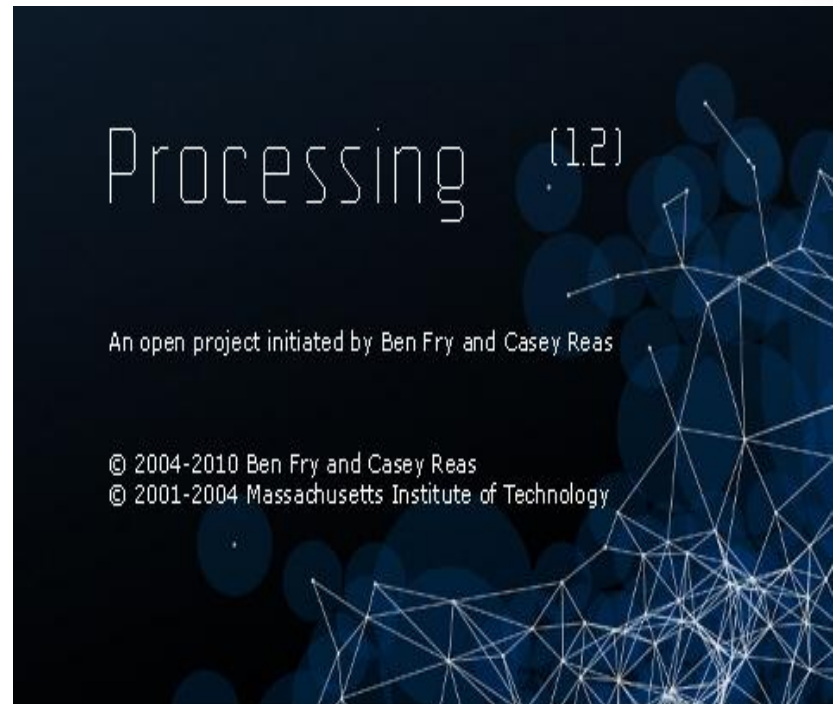
By Rutvi Joshi

What is Processing ?

- A Graphical Sketch Book and Environment, to graphically teach the fundamentals of computer science
- But it has evolved into a tool for generating professional art work
- As of today it is used by many students, artists, designers, researchers and hobbyists
- Being a Free and Open Source software tool, has made it accessible and thus very popular

The Beginning...

- Born in the MIT Media Labs in 2001
- Created by Ben Fry and Casey Reas
- With aids from:
 - ✓ Carnegie Mellon
 - ✓ Miami University
 - ✓ University of California – Los Angeles
 - ✓ And others



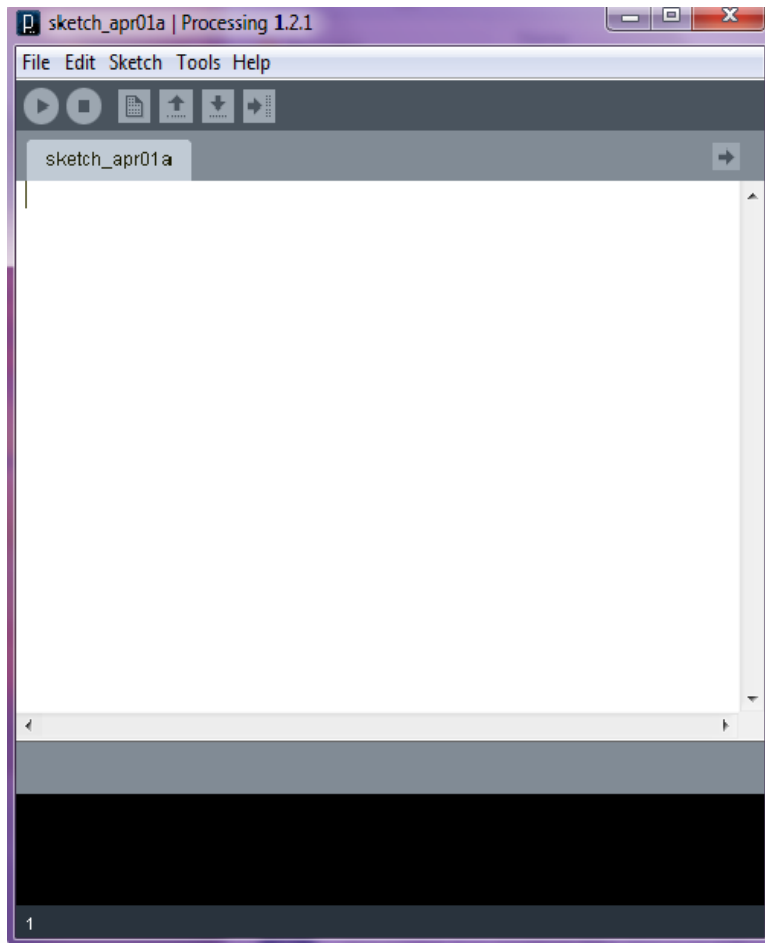
The Growth...

- Taught in: NYU's graduate ITP Program
UCLA's Design and Media Arts Program
Schools to help supplement algebra and geometry
- Projects: Design Firms to create motion graphics
Bands to create animation complementing music
Publications with information graphics
- Public visualization of pollution levels in Helsinki by a group HeHe
- Visualization of the marine eco-system at University of Washington for the NSF RISE project

The Language...

- Strictly typed
- Very similar to Java
- Includes globally accessible functions, Classes, Inheritance, Polymorphism
- Does not include the advanced features of Java, but integrates them, making it **easier** to learn
- Uses plenty of third party libraries like Open GL, JavaScript, PDF, Databases, ..., to name a few

The Sketch Book...




- Sketches run as Java Applets
- Thus, they can be put in Web pages like Flash Movies
- **Export** as software application on Windows, Linux and MAC

Bored ?

Enough prattling about the language ??

Lets get to business...

But first [Download Processing](#)

Then Click on the application file - 

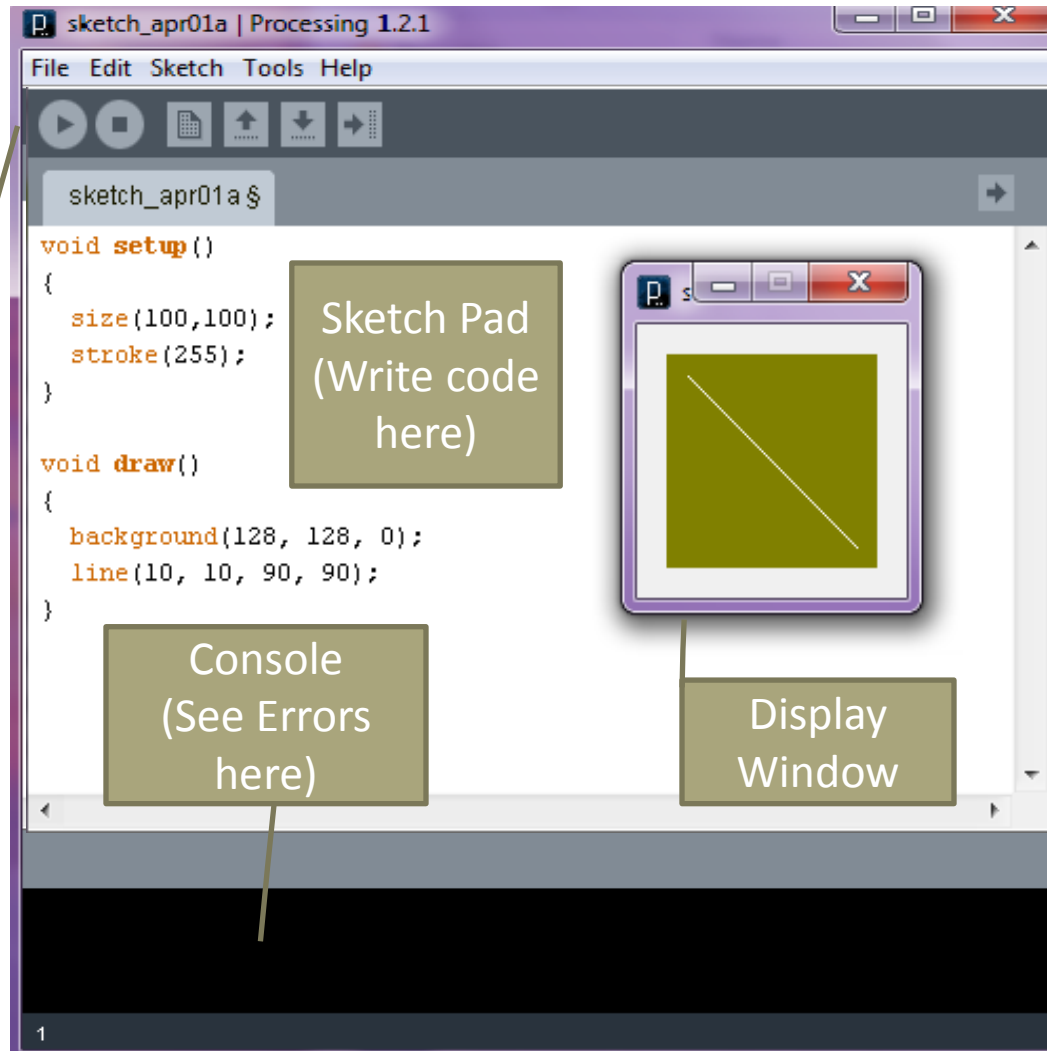
And You are all set!!

Lets get familiar...

From left to right:

- Run
- Stop
- New File
- Save
- Save As
- Export

(Function as name suggests)



The A-B-C's...

There are two core methods:

- `void setup() {}`
 - it initializes the drawing area
 - it is always called first and only once

- `void draw() {}`
 - it handles the animation
 - it is called repeatedly

... D-E-F's ...

- Size(width, height):
 - always the first function in setup()
 - specifies the width and height of the drawing area
- Background(255,255,255):
 - specifies the color of the drawing area in RGB
 - If included in draw(), it will keep clearing the screen, by overwriting on the earlier screen
 - If included in setup(), it will set screen color once, thus gives a continuous effect

... G-H-I's ...

- `Stroke(255):`
 - Set color of drawing / drawing outline(in case of polygons) in RGB
- `Fill(255):`
 - Sets color of the inside of the drawing (polygons)
- `Line(x1,y1,x2,y2):`
 - Draws a line from co-ordinates x_1, y_1 to x_2, y_2 of the drawing area
 - The color of the line will be as specified by the stroke function

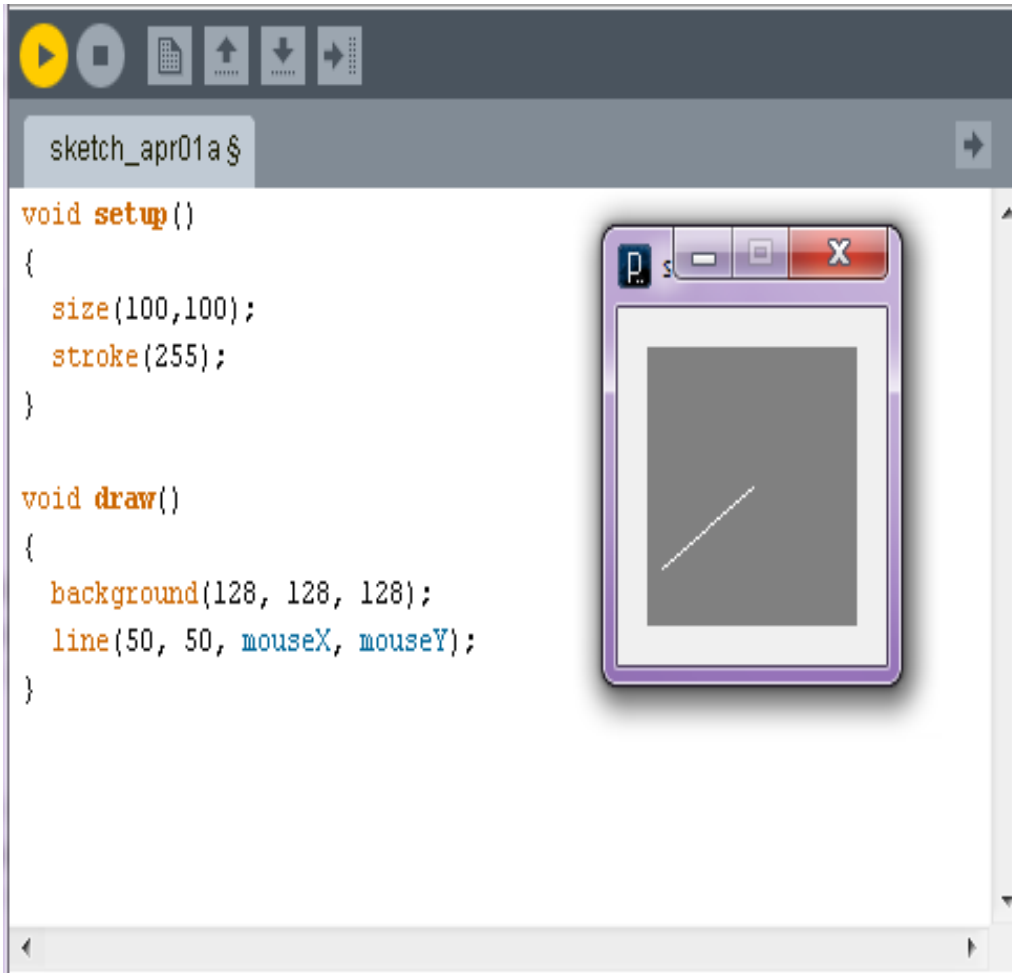
... J-K-L's ...

- `mousePressed()`:
 - is a function and called every time the mouse is pressed
- `mousePressed`:
 - variable storing true if mouse is pressed and false if mouse not pressed
- `mouseX`:
 - variable storing the current X co-ordinate of the mouse as on the drawing area
- `mouseY`:
 - variable storing the current Y co-ordinate of the mouse as on the drawing area

1+1=2...

- Now that you are familiar with the general syntax
- Lets put a code together

Example 1...



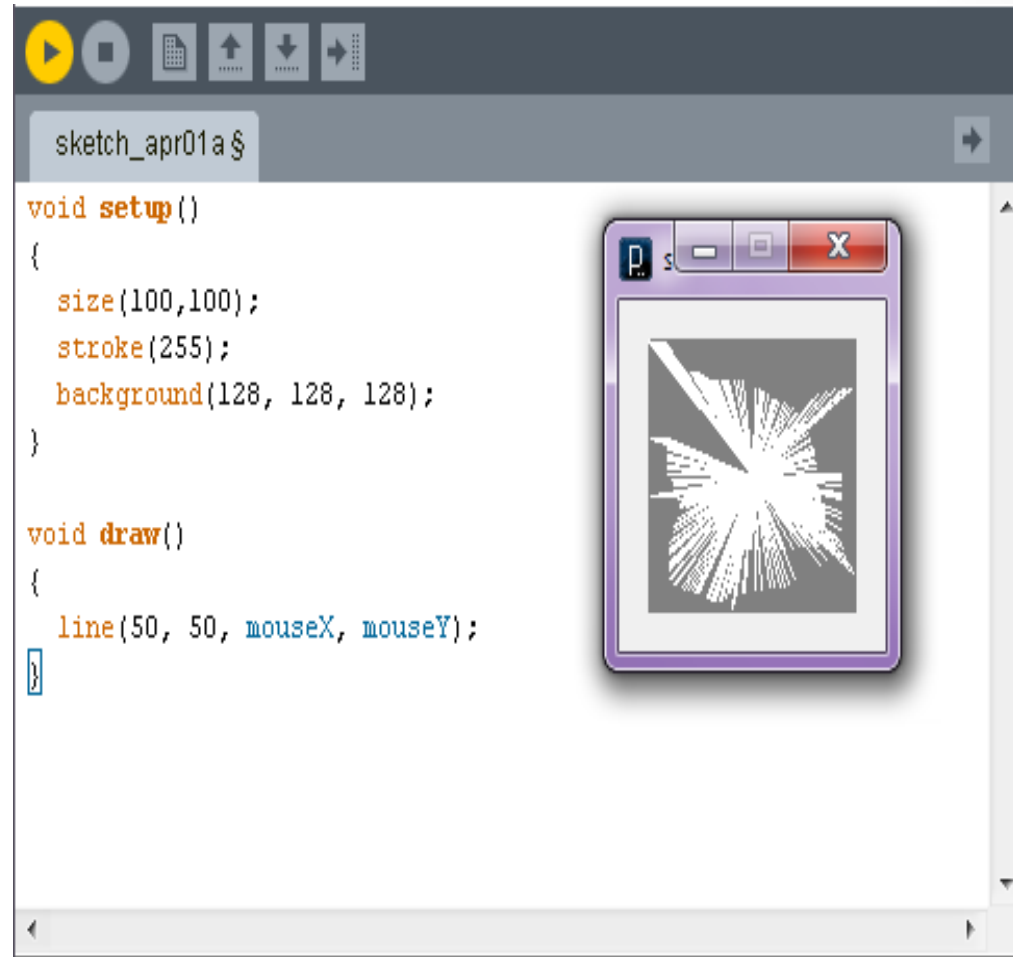
```
void setup()
{
  size(100,100);
  stroke(255);
}

void draw()
{
  background(128, 128, 128);
  line(50, 50, mouseX, mouseY);
}
```

- This code draws a line from the center of the drawing area to the position of the mouse
- Note where the background function is written.

Example 2...

- This is the same as Example 1 except the background is not cleared
- Hence we get a continuous movement track of the mouse



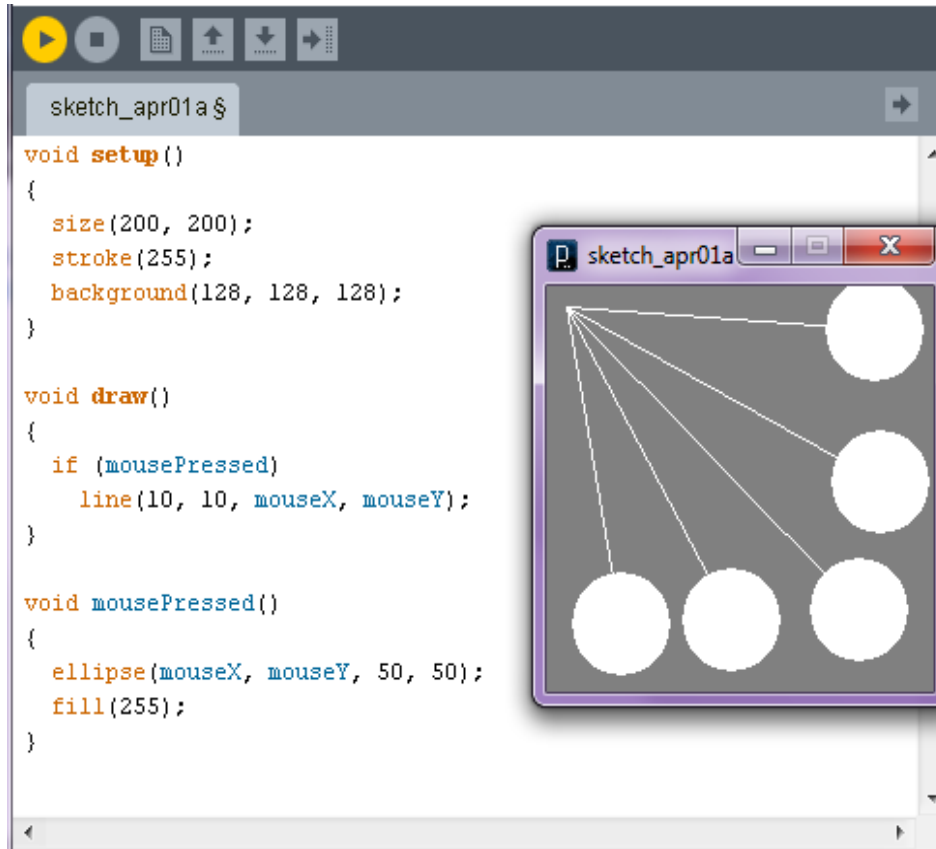
The screenshot shows a Processing IDE window titled "sketch_apr01a \$". The code in the editor is as follows:

```
void setup()
{
  size(100,100);
  stroke(255);
  background(128, 128, 128);
}

void draw()
{
  line(50, 50, mouseX, mouseY);
}
```

On the right side of the IDE, a small window displays the visualization. It shows a 100x100 pixel canvas with a dark gray background. A series of white lines radiate from the center point (50, 50) to the current mouse position, creating a starburst or movement track effect. The window has standard OS window controls (minimize, maximize, close) and a title bar.

Example 3...



- This should not be a task to understand
- Every time the mouse is pressed, the pendulum is drawn

Tip of the Iceberg...

- That was fun !
- There are many more functions which are categorized according to their feature
- Structure, Color, Shape, Environment, Image, Input, Data, Typography, Math, Output, Control, Lights and Camera, Rendering
- Here's the [Index](#) of commands you can further explore!

Expanding Horizons...

- There is so much one can do with this language given its horizons...
 - Object Oriented Processing
 - Image Processing
 - Image Merging
 - 3D graphics
 - Physics Examples like projectile simulations
 - Networking

Deep end of the pool...

- Since Processing is Open Source there is a lot of data available
- [Tutorial](#) – one out of many
- [Forum](#) – one out of many
- And our age old friends, [Books](#)
- You can also [Share your Sketches](#) !

Use your imagination...

- [Processing.js](#) is a recent addition to this family
- It makes your application that you developed run using web standards without any plug ins!
- So now you simply write a code using Processing Language and include it in your web page
- Processing.js does the rest... like [Magic](#) !

Its not a competition...

- Here are some really well made examples...
- [Abstract.js](#)
- [Bit-Torrent Visualizations](#)
- And many [More...](#)

Take it up as a hobby...

- In conclusion, we can say that the goal of Processing Language is successfully being achieved
- All you have to do is a little bit of coding...
- And lot of Imagination !
- Thank You !

References...

- www.processing.org
- <http://www.ibm.com/developerworks/opensource/library/>
- <http://www.slideshare.net/fredrikb/processing-presentation-presentation>
- <http://www.slideshare.net/jeresig/processing-and-processingjs>
- <http://www.slideshare.net/chilibeeto/artdm-170-week-14-introduction-to-processing>