

The background of the slide is a dense, abstract pattern of overlapping circles. The circles vary in size and color, including shades of dark blue, light blue, grey, brown, and yellow. The overall effect is a textured, bokeh-like background.

Processing

Basic introduction & Imaging

Hello!

I'm the TA for this class

Contact: Bryan Leow 81813134

Tips for the class

- Don't worry if you're new to programming
- Programming is a mindset
- Free resources are everywhere!
- Have fun and keep experimenting!

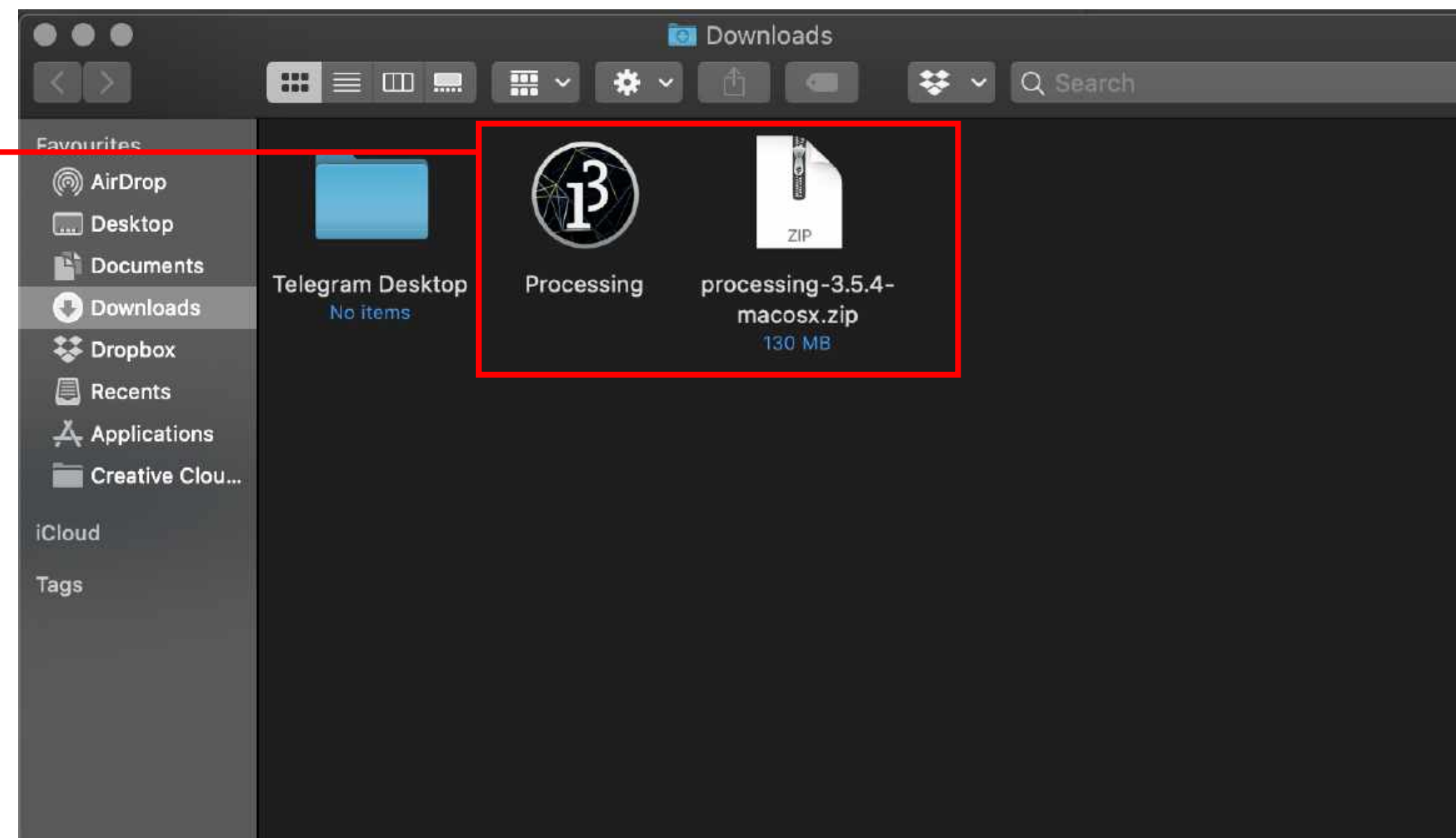
Download Processing

<https://processing.org/download/>

1. Download with the link



2. Extract the zipped file to retrieve icon. You can now use processing! You can also drop it in your Applications



Windows users: Same thing, extract and it will be a folder with Processing.exe file

Interface and basics

The screenshot displays the Arduino IDE interface. At the top, there are two buttons: a play button (Run code) and a square button (Stop code). Below these is a dropdown menu showing the current sketch name, "sketch_200818b". The main area contains a code editor with the following code:

```
1 void setup(){
2   size(400,800); //this is a comment
3   background(0);
4   println("Hello!");
5 }
6
7 void draw(){
8   //background(0);
9   rectMode(CENTER);
10  fill(255);
11  stroke(255);
12  rect(200, 400, 100,100);
13  delay(100);
14 }
15
16
17
18
19
```

At the bottom, there is a console window showing the output "Hello!".

- Play button: Run code
- Stop button: Stop code

Sets up the initial conditions, runs only once

```
void setup(){
  size(400,800); //this is a comment
  background(0);
  println("Hello!");
}
```

- Code runs within squiggly brackets { }
- Each line of code is separated by semicolon ;
- size(x-axis, y-axis) controls the size of the window

Continuously looping after set up is done, forever

```
void draw(){
  //background(0);
  rectMode(CENTER);
  fill(255);
  stroke(255);
  rect(200, 400, 100,100);
  delay(100);
}
```

- default commands like "fill", "rect", "delay" etc

println("Hello!") on void setup: the text is only printed once on console VS if it's on void draw

Hello!

Interface and basics

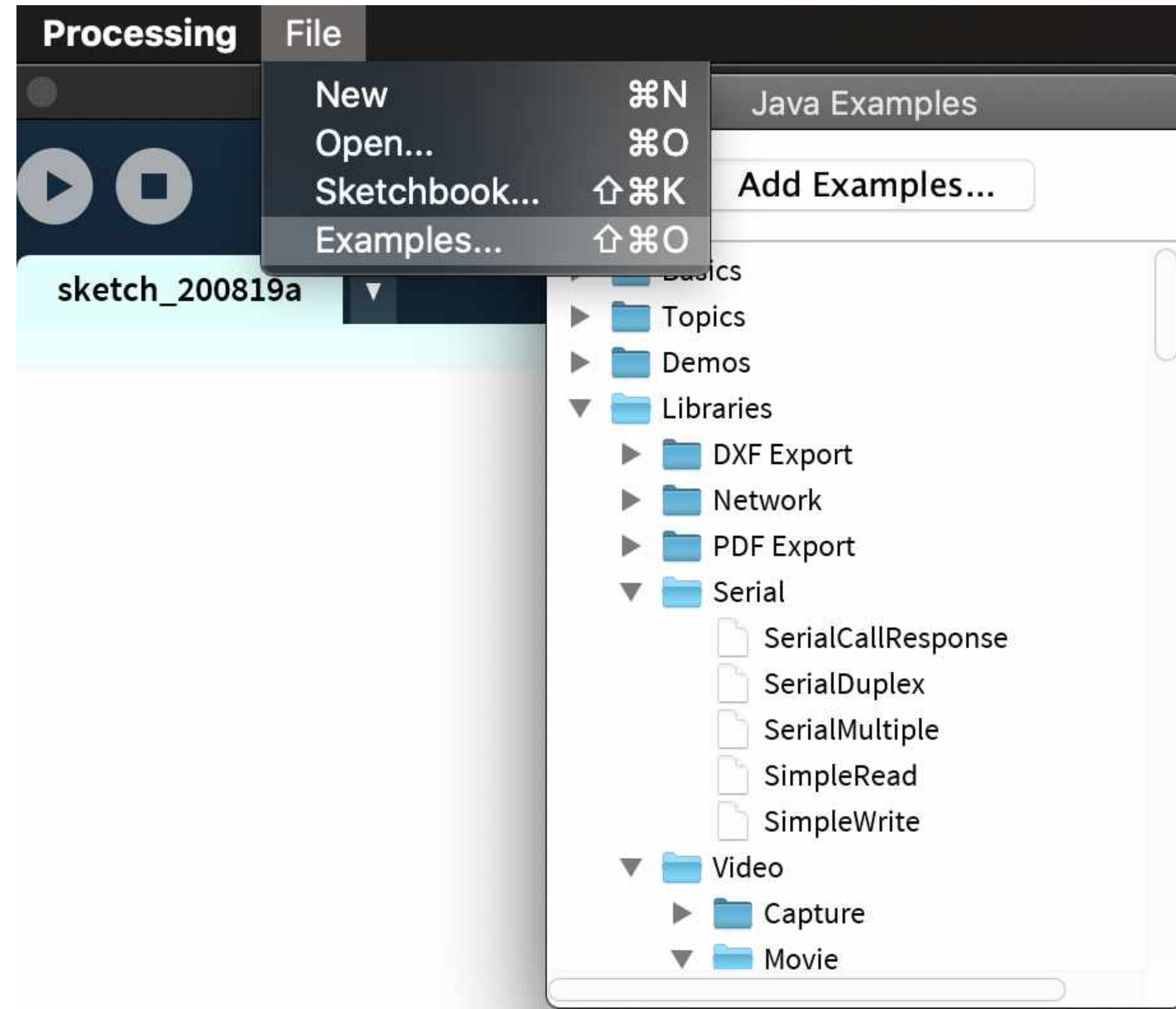
- white fill and stroke (255)
- Colours: (R,G,B), or greyscale from 0 to 255
 - i.e. fill(255,255,255) is white, same with fill(255)
- rectangle's anchor point is in the center
- rectangle is placed in the middle of the window
- rectangle has 100 by 100 pixels dimension

```
sketch_200818b
1 void setup(){
2 size(400,800); //this is a comment
3 background(0);
4 println("Hello!");
5 }
6
7 void draw(){
8 //background(0);
9 rectMode(CENTER);
10 fill(255);
11 stroke(255);
12 rect(200, 400, 100, 100);
13 delay(100);
14 }
15
16
17
18
19
```

Hello!

Console Errors

References: Examples



References: Online Resources

<https://processing.org/reference/>

Processing p5.js Processing.py Processing for Android Processing for Pi Processing Foundation

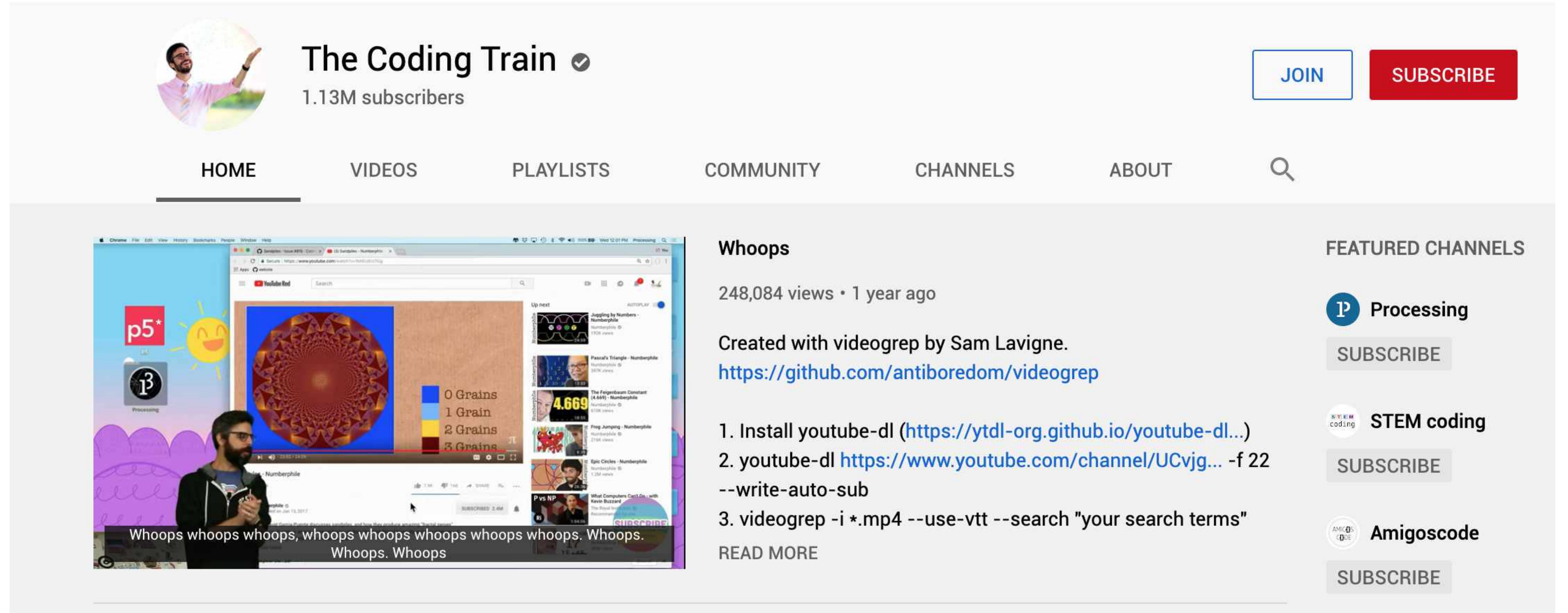
Processing

Reference. Processing was designed to be a flexible software sketchbook.

Structure	Shape	Color
<code>()</code> (parentheses)	<code>createShape()</code>	Setting
<code>,</code> (comma)	<code>loadShape()</code>	<code>background()</code>
<code>.</code> (dot)	<code>PShape</code>	<code>clear()</code>
<code>/**/</code> (multiline comment)	2D Primitives	<code>colorMode()</code>
<code>/**/</code> (doc comment)	<code>arc()</code>	<code>fill()</code>
<code>//</code> (comment)	<code>circle()</code>	<code>noFill()</code>
<code>;</code> (semicolon)	<code>ellipse()</code>	<code>noStroke()</code>
<code>=</code> (assign)	<code>line()</code>	<code>stroke()</code>
<code>[]</code> (array access)	<code>point()</code>	Creating & Reading
<code>{}</code> (curly braces)	<code>quad()</code>	<code>alpha()</code>
<code>catch</code>	<code>rect()</code>	<code>blue()</code>
<code>class</code>	<code>square()</code>	<code>brightness()</code>
<code>draw()</code>	<code>triangle()</code>	<code>color()</code>
<code>exit()</code>	Curves	<code>green()</code>
<code>extends</code>	<code>bezier()</code>	<code>hue()</code>
<code>false</code>	<code>bezierDetail()</code>	<code>lerpColor()</code>
<code>final</code>	<code>bezierPoint()</code>	<code>red()</code>
<code>implements</code>	<code>bezierTangent()</code>	<code>saturation()</code>
<code>import</code>	<code>curve()</code>	Image
<code>loop()</code>	<code>curveDetail()</code>	<code>createImage()</code>
<code>new</code>	<code>curvePoint()</code>	<code>PImage</code>
<code>noLoop()</code>	<code>curveTangent()</code>	Loading & Displaying
<code>null</code>	<code>curveTightness()</code>	
<code>pop()</code>		
<code>popStyle()</code>		
<code>private</code>		
<code>public</code>		

References: Online Resources

<https://www.youtube.com/thecodingtrain>



The screenshot shows the YouTube channel page for 'The Coding Train', which has 1.13M subscribers. The channel navigation menu includes HOME, VIDEOS, PLAYLISTS, COMMUNITY, CHANNELS, and ABOUT. A search icon is also present. The featured video is 'Whoops' by Numberphile, with 248,084 views and posted 1 year ago. The video description states it was created with videogrep by Sam Lavigne and provides a GitHub link. A list of terminal commands is provided for downloading the video and generating subtitles. The 'FEATURED CHANNELS' section lists Processing, STEM coding, and Amigoscode, each with a 'SUBSCRIBE' button.

The Coding Train ✓
1.13M subscribers

JOIN SUBSCRIBE

HOME VIDEOS PLAYLISTS COMMUNITY CHANNELS ABOUT

Whoops
248,084 views • 1 year ago

Created with videogrep by Sam Lavigne.
<https://github.com/antiboredom/videogrep>

1. Install youtube-dl (<https://ytdl-org.github.io/youtube-dl...>)
2. youtube-dl <https://www.youtube.com/channel/UCvjg...> -f 22 --write-auto-sub
3. videogrep -i *.mp4 --use-vtt --search "your search terms"

READ MORE

FEATURED CHANNELS

- Processing**
SUBSCRIBE
- STEM coding**
SUBSCRIBE
- Amigoscode**
SUBSCRIBE

Basic commands and functions on Processing

Let's try some examples

Conditionals1
Keyboard Functions
Mouse Signal

Generating graphics and images on Processing

Let's try some examples

Form: Bezier

Image: Pointilism

Lights: Directional

Math: Double Random

Shape: ScaleShape

Image Processing: Blending

Next week: Videos & Animations

Self enrichment:

- Try exploring the examples given by Processing
 - Watch TheCodingTrain on x1.5 speed

Thanks!

Bryan Leow 81813134