Uni Studies 3:
Intro to Processing

Assoc. Professor Donald J. Patterson
Uni Stu 3 Fall 2012

Intro to Processing

http://processing.org/

Intro to Processing

http://processing.org/

Intro to Processing

- What the heck is Processing?
    - A programming language
    - An environment for running the programs

http://processing.org/

Intro to Processing

- What the heck is Processing?
  - A programming language
  - An environment for running the programs
- What is it for?
  - It is for people who want to create
    - images
    - animations
    - interactions

http://processing.org/

Intro to Processing

- What the heck is Processing?
    - A programming language
    - An environment for running the programs
- What is it for?
    - It is for people who want to create
        - images
        - animations
        - interactions
- Who is it for?
        - students
        - artists
        - designers
        - researchers
        - hobbyists

http://processing.org/

Intro to Processing

http://processing.org/

Intro to Processing

- Free to download

http://processing.org/

Intro to Processing

- Free to download
- Open source

http://processing.org/

Intro to Processing

- Free to download
- Open source
- Programs output in 2D, 3D or pdf

http://processing.org/

Intro to Processing

- Free to download
- Open source
- Programs output in 2D, 3D or pdf
- For Windows, Mac, Linux

http://processing.org/

Intro to Processing

- Free to download
- Open source
- Programs output in 2D, 3D or pdf
- For Windows, Mac, Linux
- Programs can be put in web pages

http://processing.org/

Intro to Processing

- Free to download
- Open source
- Programs output in 2D, 3D or pdf
- For Windows, Mac, Linux
- Programs can be put in web pages
- Programs can be run as applications

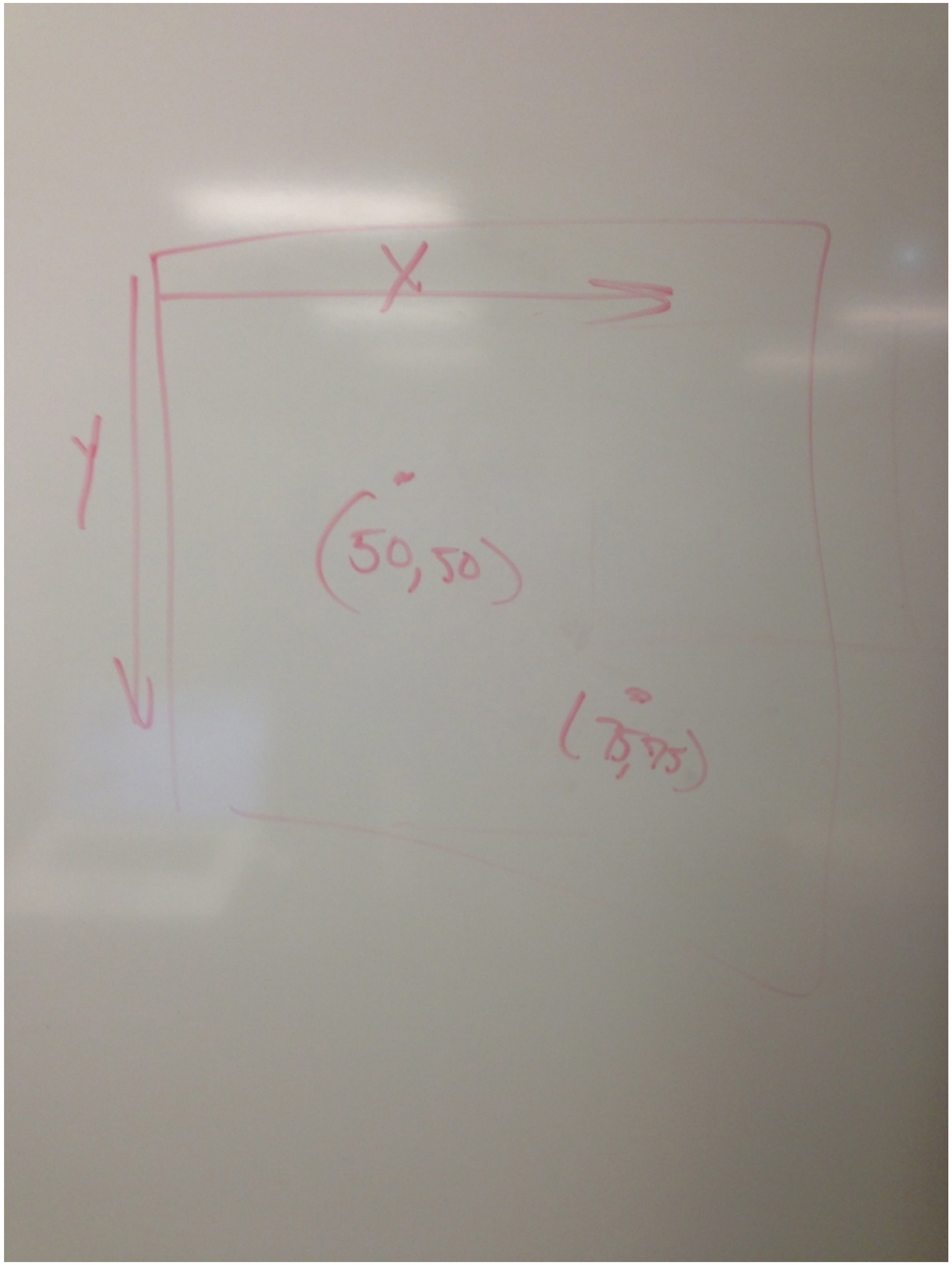
http://processing.org/

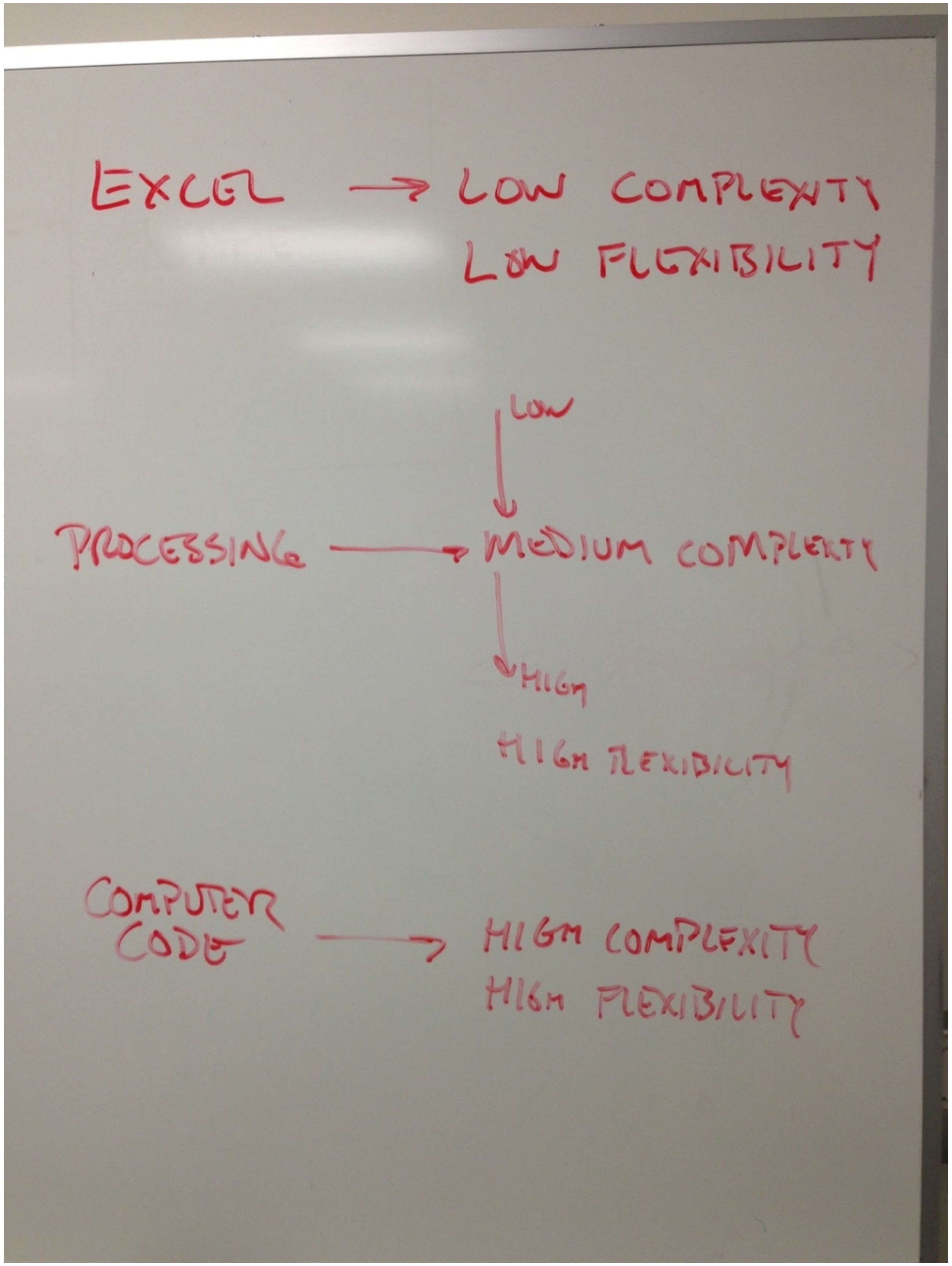
Intro to Processing

- Free to download
- Open source
- Programs output in 2D, 3D or pdf
- For Windows, Mac, Linux
- Programs can be put in web pages
- Programs can be run as applications
- Lots of documentation and books available

http://processing.org/

Intro to Processing





EXCEL → LOW COMPLEXITY
LOW FLEXIBILITY

LOW

PROCESSING → MEDIUM COMPLEXTY

HIGH

HIGH FLEXIBILITY

COMPUTER CODE → HIGH COMPLEXITY
HIGH FLEXIBILITY

Intro to Processing

http://processing.org/

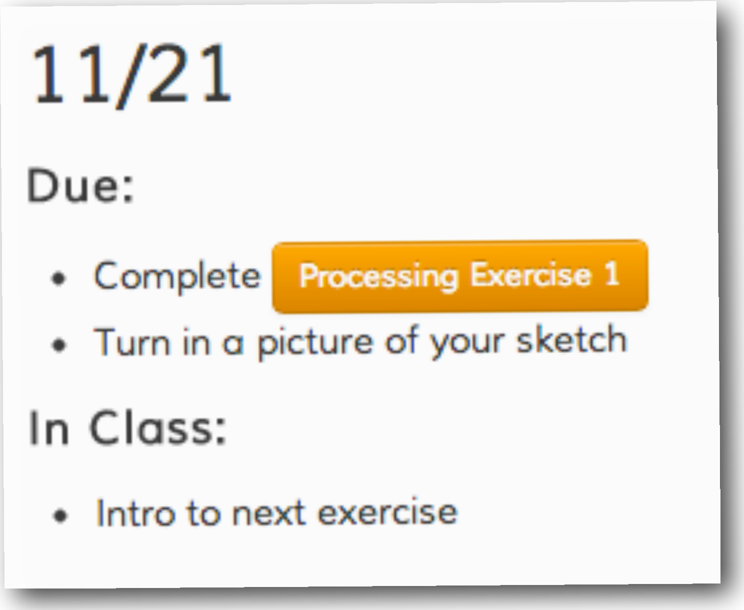
Intro to Processing

- Your assignment for next week
  - Complete the lab:
    - "Getting Started. Welcome to Processing"

http://processing.org/

Intro to Processing

- Your assignment for next week
  - Complete the lab:
    - "Getting Started. Welcome to Processing"



11/21

Due:

- Complete Processing Exercise 1
- Turn in a picture of your sketch

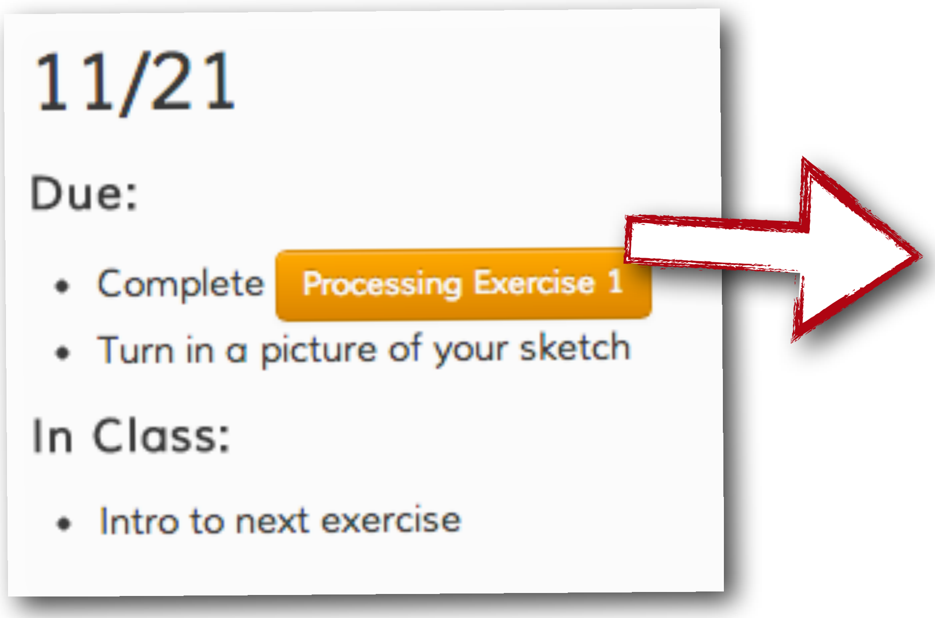
In Class:

- Intro to next exercise

http://processing.org/

Intro to Processing

- Your assignment for next week
  - Complete the lab:
    - "Getting Started. Welcome to Processing"



11/21

Due:

- Complete
- Turn in a picture of your sketch

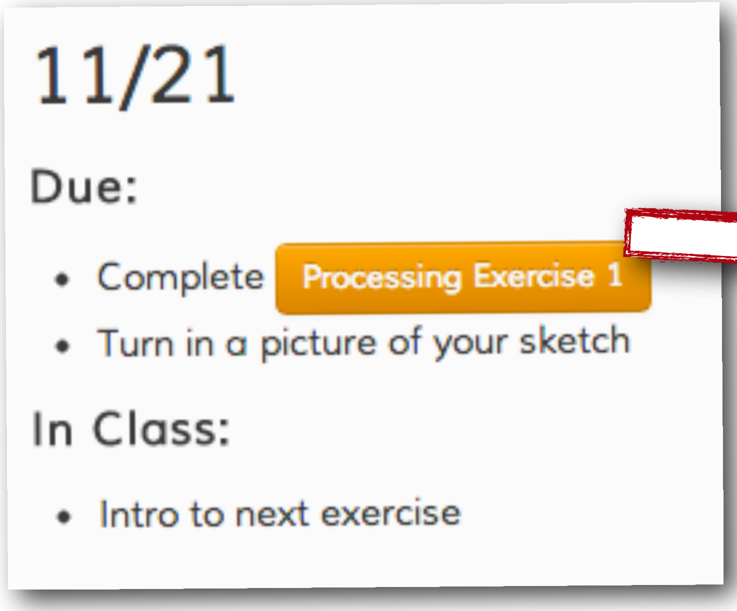
In Class:

- Intro to next exercise

http://processing.org/

Intro to Processing

- Your assignment for next week
  - Complete the lab:
    - "Getting Started. Welcome to Processing"



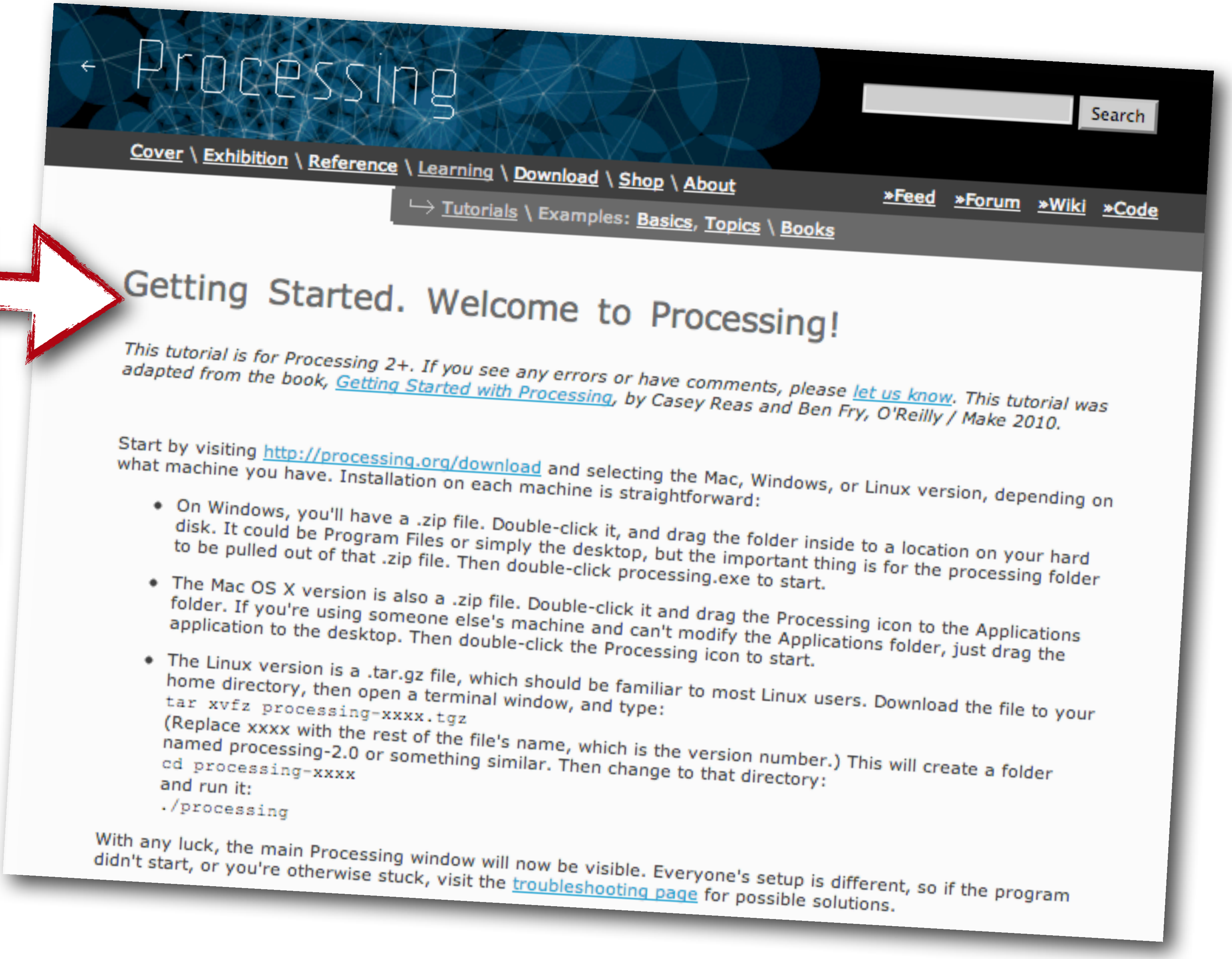
11/21

Due:

- Complete Processing Exercise 1
- Turn in a picture of your sketch

In Class:

- Intro to next exercise



Processing
Search
Cover \ Exhibition \ Reference \ Learning \ Download \ Shop \ About
»Feed »Forum »Wiki »Code
Tutorials \ Examples: Basics, Topics \ Books

Getting Started. Welcome to Processing!

This tutorial is for Processing 2+. If you see any errors or have comments, please let us know. This tutorial was adapted from the book, Getting Started with Processing, by Casey Reas and Ben Fry, O'Reilly / Make 2010.

Start by visiting http://processing.org/download and selecting the Mac, Windows, or Linux version, depending on what machine you have. Installation on each machine is straightforward:

- On Windows, you'll have a .zip file. Double-click it, and drag the folder inside to a location on your hard disk. It could be Program Files or simply the desktop, but the important thing is for the processing folder to be pulled out of that .zip file. Then double-click processing.exe to start.
- The Mac OS X version is also a .zip file. Double-click it and drag the Processing icon to the Applications folder. If you're using someone else's machine and can't modify the Applications folder, just drag the application to the desktop. Then double-click the Processing icon to start.
- The Linux version is a .tar.gz file, which should be familiar to most Linux users. Download the file to your home directory, then open a terminal window, and type:
  ```
  tar xvfz processing-xxxx.tgz
  ```
  (Replace xxxx with the rest of the file's name, which is the version number.) This will create a folder named processing-2.0 or something similar. Then change to that directory:
  ```
  cd processing-xxxx
  ```
  and run it:
  ```
  ./processing
  ```

With any luck, the main Processing window will now be visible. Everyone's setup is different, so if the program didn't start, or you're otherwise stuck, visit the troubleshooting page for possible solutions.

http://processing.org/

Intro to Processing

http://processing.org/

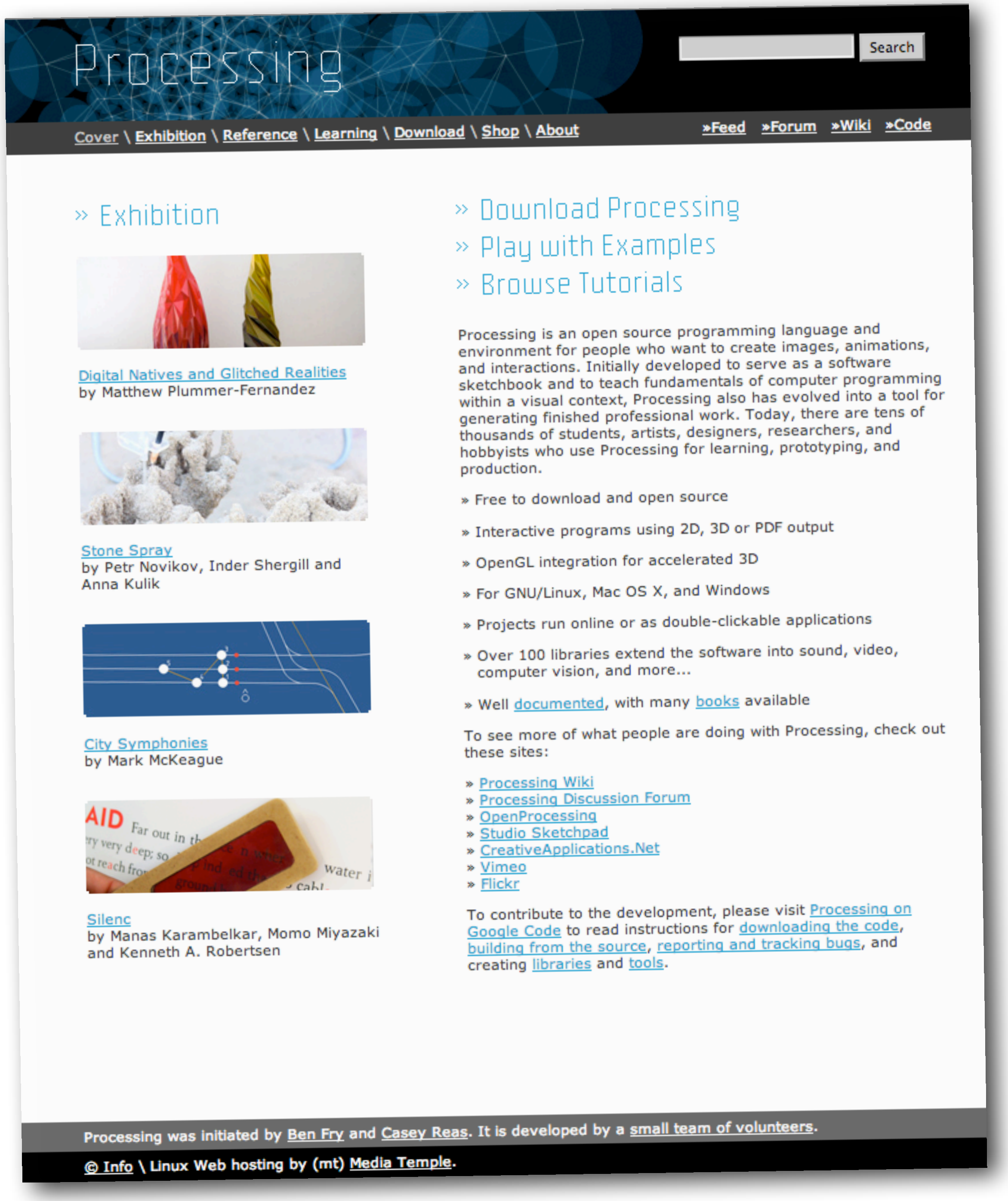
Intro to Processing

- Download the software v 2.0 beta 6

http://processing.org/

Intro to Processing

- Download the software v 2.0 beta 6


Processing
Search
Cover \ Exhibition \ Reference \ Learning \ Download \ Shop \ About
»Feed »Forum »Wiki »Code

» Exhibition

Digital Natives and Glitched Realities
by Matthew Plummer-Fernandez

Stone Spray
by Petr Novikov, Inder Shergill and Anna Kulik

City Symphonies
by Mark McKeague

Silenc
by Manas Karambelkar, Momo Miyazaki and Kenneth A. Robertsen

» Download Processing
» Play with Examples
» Browse Tutorials

Processing is an open source programming language and environment for people who want to create images, animations, and interactions. Initially developed to serve as a software sketchbook and to teach fundamentals of computer programming within a visual context, Processing also has evolved into a tool for generating finished professional work. 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 using 2D, 3D or PDF output

» OpenGL integration for accelerated 3D

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

» Projects run online or as double-clickable applications

» Over 100 libraries extend the software into sound, video, computer vision, and more...

» Well documented, with many books available

To see more of what people are doing with Processing, check out these sites:

» Processing Wiki
» Processing Discussion Forum
» OpenProcessing
» Studio Sketchpad
» CreativeApplications.Net
» Vimeo
» Flickr

To contribute to the development, please visit Processing on Google Code to read instructions for downloading the code, building from the source, reporting and tracking bugs, and creating libraries and tools.

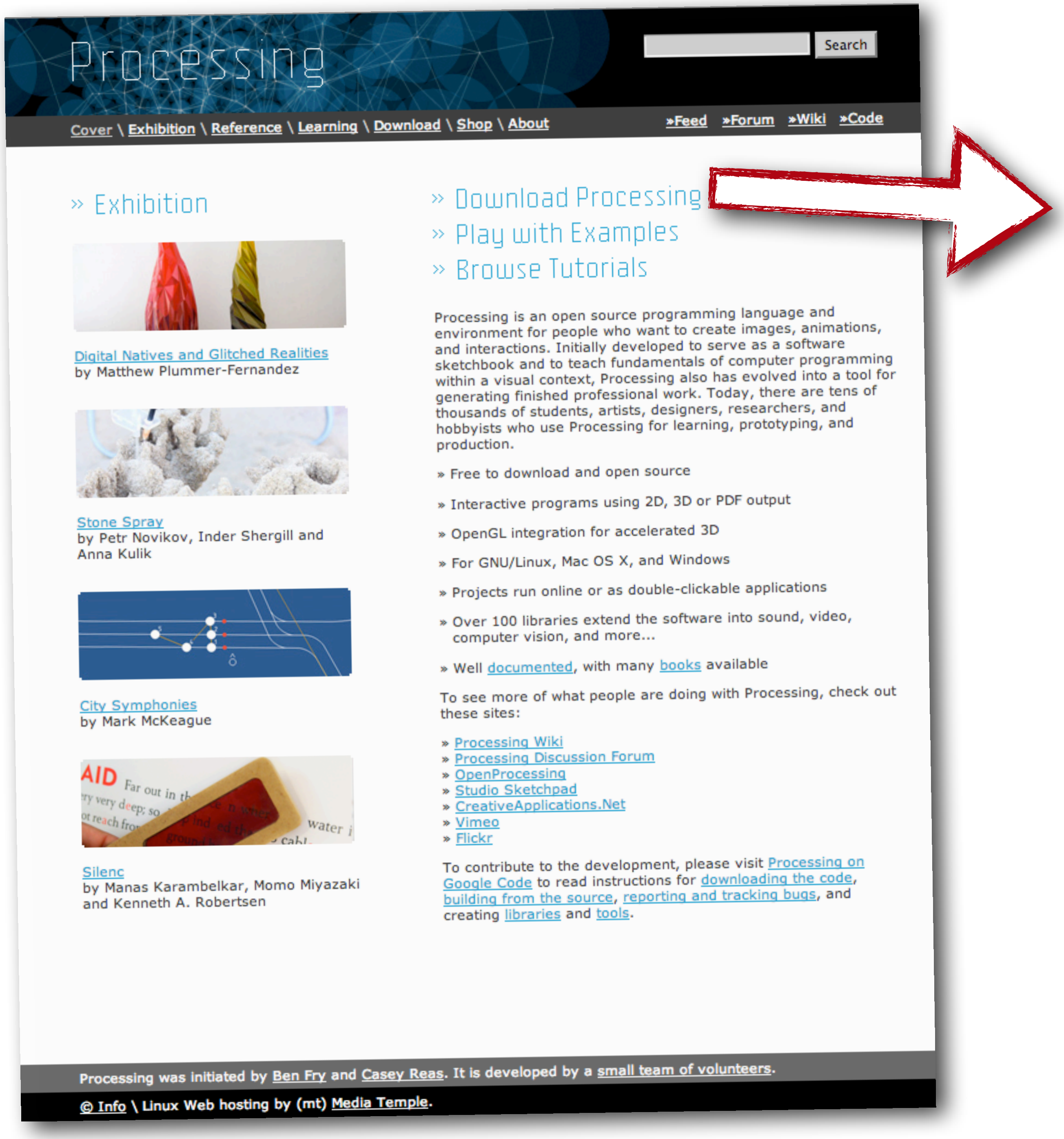
Processing was initiated by Ben Fry and Casey Reas. It is developed by a small team of volunteers.

© Info \ Linux Web hosting by (mt) Media Temple.

http://processing.org/

Intro to Processing

- Download the software v 2.0 beta 6



Processing
Search
Cover \ Exhibition \ Reference \ Learning \ Download \ Shop \ About
»Feed »Forum »Wiki »Code

» Exhibition

Digital Natives and Glitched Realities
by Matthew Plummer-Fernandez

Stone Spray
by Petr Novikov, Inder Shergill and Anna Kulik

City Symphonies
by Mark McKeague

Silenc
by Manas Karambelkar, Momo Miyazaki and Kenneth A. Robertsen

» Download Processing
» Play with Examples
» Browse Tutorials

Processing is an open source programming language and environment for people who want to create images, animations, and interactions. Initially developed to serve as a software sketchbook and to teach fundamentals of computer programming within a visual context, Processing also has evolved into a tool for generating finished professional work. 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 using 2D, 3D or PDF output

» OpenGL integration for accelerated 3D

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

» Projects run online or as double-clickable applications

» Over 100 libraries extend the software into sound, video, computer vision, and more...

» Well documented, with many books available

To see more of what people are doing with Processing, check out these sites:

» Processing Wiki
» Processing Discussion Forum
» OpenProcessing
» Studio Sketchpad
» CreativeApplications.Net
» Vimeo
» Flickr

To contribute to the development, please visit Processing on Google Code to read instructions for downloading the code, building from the source, reporting and tracking bugs, and creating libraries and tools.

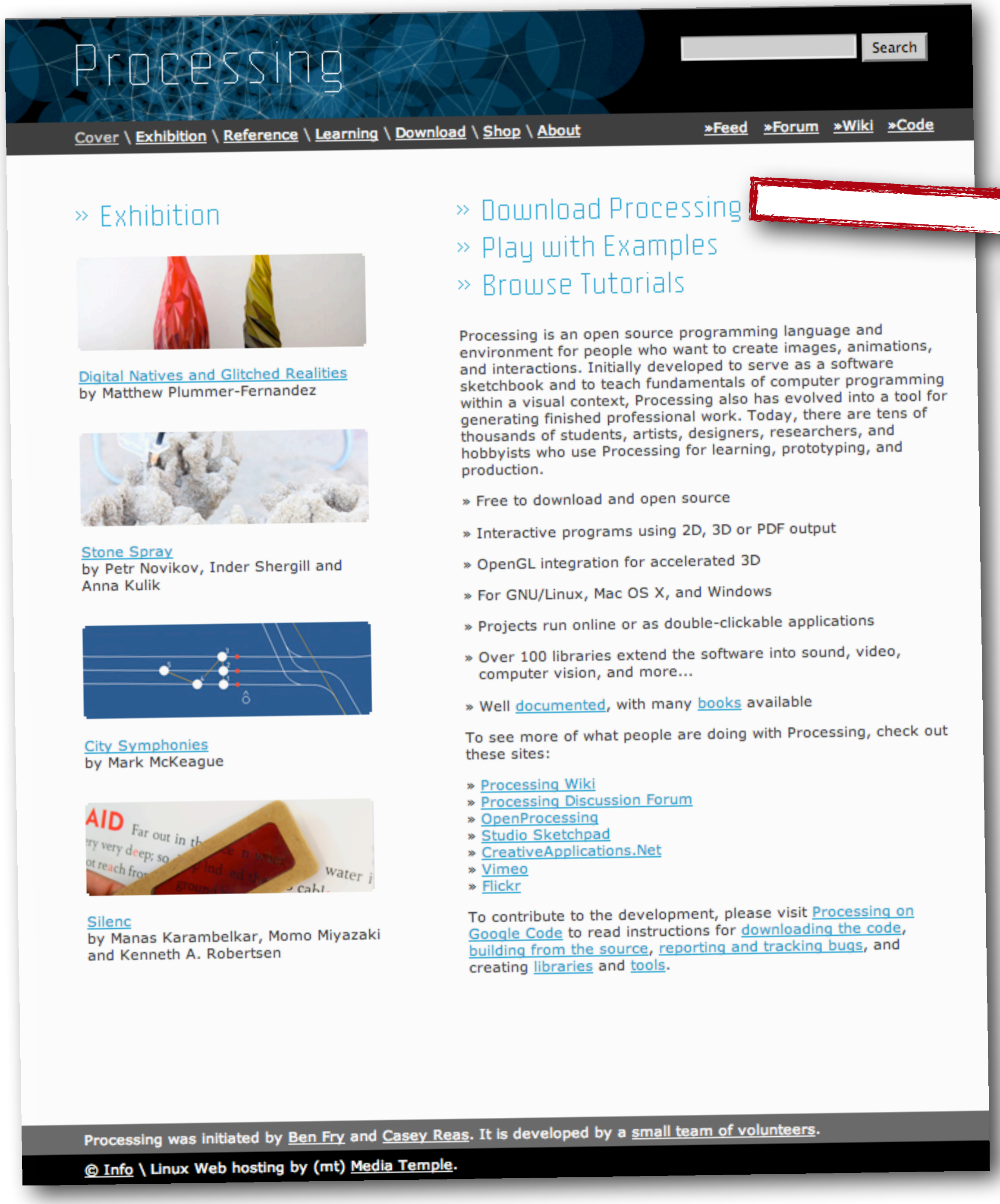
Processing was initiated by Ben Fry and Casey Reas. It is developed by a small team of volunteers.

© Info \ Linux Web hosting by (mt) Media Temple.

http://processing.org/

Intro to Processing

- Download the software v 2.0 beta 6



Processing
Search
Cover \ Exhibition \ Reference \ Learning \ Download \ Shop \ About
»Feed »Forum »Wiki »Code

» Exhibition

Digital Natives and Glitched Realities
by Matthew Plummer-Fernandez

Stone Spray
by Petr Novikov, Inder Shergill and Anna Kulik

City Symphonies
by Mark McKeague

Silenc
by Manas Karambelkar, Momo Miyazaki and Kenneth A. Robertsen

» Download Processing
» Play with Examples
» Browse Tutorials

Processing is an open source programming language and environment for people who want to create images, animations, and interactions. Initially developed to serve as a software sketchbook and to teach fundamentals of computer programming within a visual context, Processing also has evolved into a tool for generating finished professional work. 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 using 2D, 3D or PDF output

» OpenGL integration for accelerated 3D

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

» Projects run online or as double-clickable applications

» Over 100 libraries extend the software into sound, video, computer vision, and more...

» Well documented, with many books available

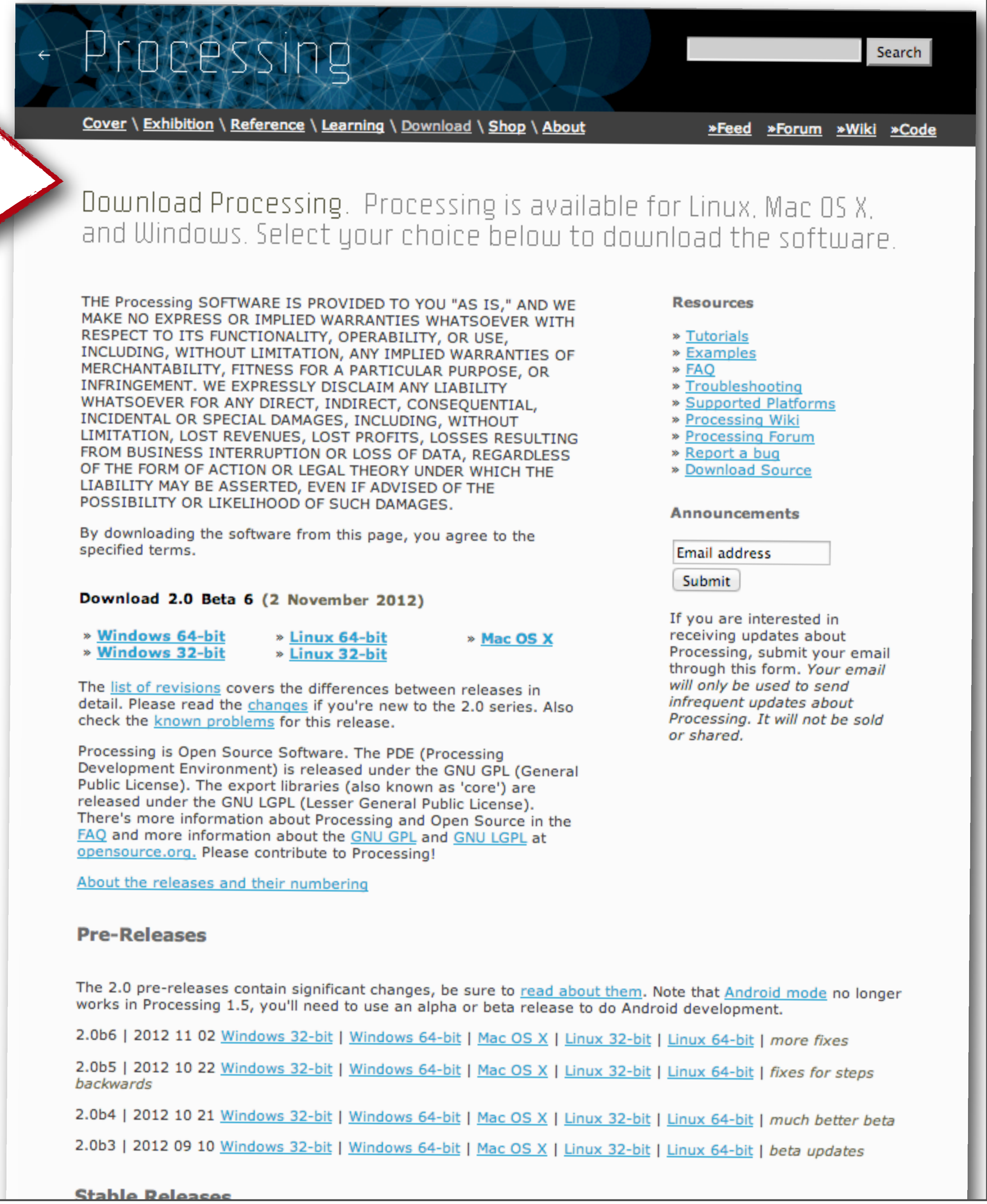
To see more of what people are doing with Processing, check out these sites:

» Processing Wiki
» Processing Discussion Forum
» OpenProcessing
» Studio Sketchpad
» CreativeApplications.Net
» Vimeo
» Flickr

To contribute to the development, please visit Processing on Google Code to read instructions for downloading the code, building from the source, reporting and tracking bugs, and creating libraries and tools.

Processing was initiated by Ben Fry and Casey Reas. It is developed by a small team of volunteers.

© Info \ Linux Web hosting by (mt) Media Temple.



← Processing
Search
Cover \ Exhibition \ Reference \ Learning \ Download \ Shop \ About
»Feed »Forum »Wiki »Code

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

THE Processing SOFTWARE IS PROVIDED TO YOU "AS IS," AND WE MAKE NO EXPRESS OR IMPLIED WARRANTIES WHATSOEVER WITH RESPECT TO ITS FUNCTIONALITY, OPERABILITY, OR USE, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR INFRINGEMENT. WE EXPRESSLY DISCLAIM ANY LIABILITY WHATSOEVER FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, INCIDENTAL OR SPECIAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST REVENUES, LOST PROFITS, LOSSES RESULTING FROM BUSINESS INTERRUPTION OR LOSS OF DATA, REGARDLESS OF THE FORM OF ACTION OR LEGAL THEORY UNDER WHICH THE LIABILITY MAY BE ASSERTED, EVEN IF ADVISED OF THE POSSIBILITY OR LIKELIHOOD OF SUCH DAMAGES.

By downloading the software from this page, you agree to the specified terms.

Download 2.0 Beta 6 (2 November 2012)

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

The list of revisions covers the differences between releases in detail. Please read the changes if you're new to the 2.0 series. Also check the known problems for this release.

Processing is Open Source Software. The PDE (Processing Development Environment) is released under the GNU GPL (General Public License). The export libraries (also known as 'core') are released under the GNU LGPL (Lesser General Public License). There's more information about Processing and Open Source in the FAQ and more information about the GNU GPL and GNU LGPL at opensource.org. Please contribute to Processing!

About the releases and their numbering

Resources

» Tutorials
» Examples
» FAQ
» Troubleshooting
» Supported Platforms
» Processing Wiki
» Processing Forum
» Report a bug
» Download Source

Announcements

Email address

Submit

If you are interested in receiving updates about Processing, submit your email through this form. *Your email will only be used to send infrequent updates about Processing. It will not be sold or shared.*

Pre-Releases

The 2.0 pre-releases contain significant changes, be sure to read about them. Note that Android mode no longer works in Processing 1.5, you'll need to use an alpha or beta release to do Android development.

2.0b6 | 2012 11 02 Windows 32-bit | Windows 64-bit | Mac OS X | Linux 32-bit | Linux 64-bit | *more fixes*

2.0b5 | 2012 10 22 Windows 32-bit | Windows 64-bit | Mac OS X | Linux 32-bit | Linux 64-bit | *fixes for steps backwards*

2.0b4 | 2012 10 21 Windows 32-bit | Windows 64-bit | Mac OS X | Linux 32-bit | Linux 64-bit | *much better beta*

2.0b3 | 2012 09 10 Windows 32-bit | Windows 64-bit | Mac OS X | Linux 32-bit | Linux 64-bit | *beta updates*

Stable Releases

http://processing.org/

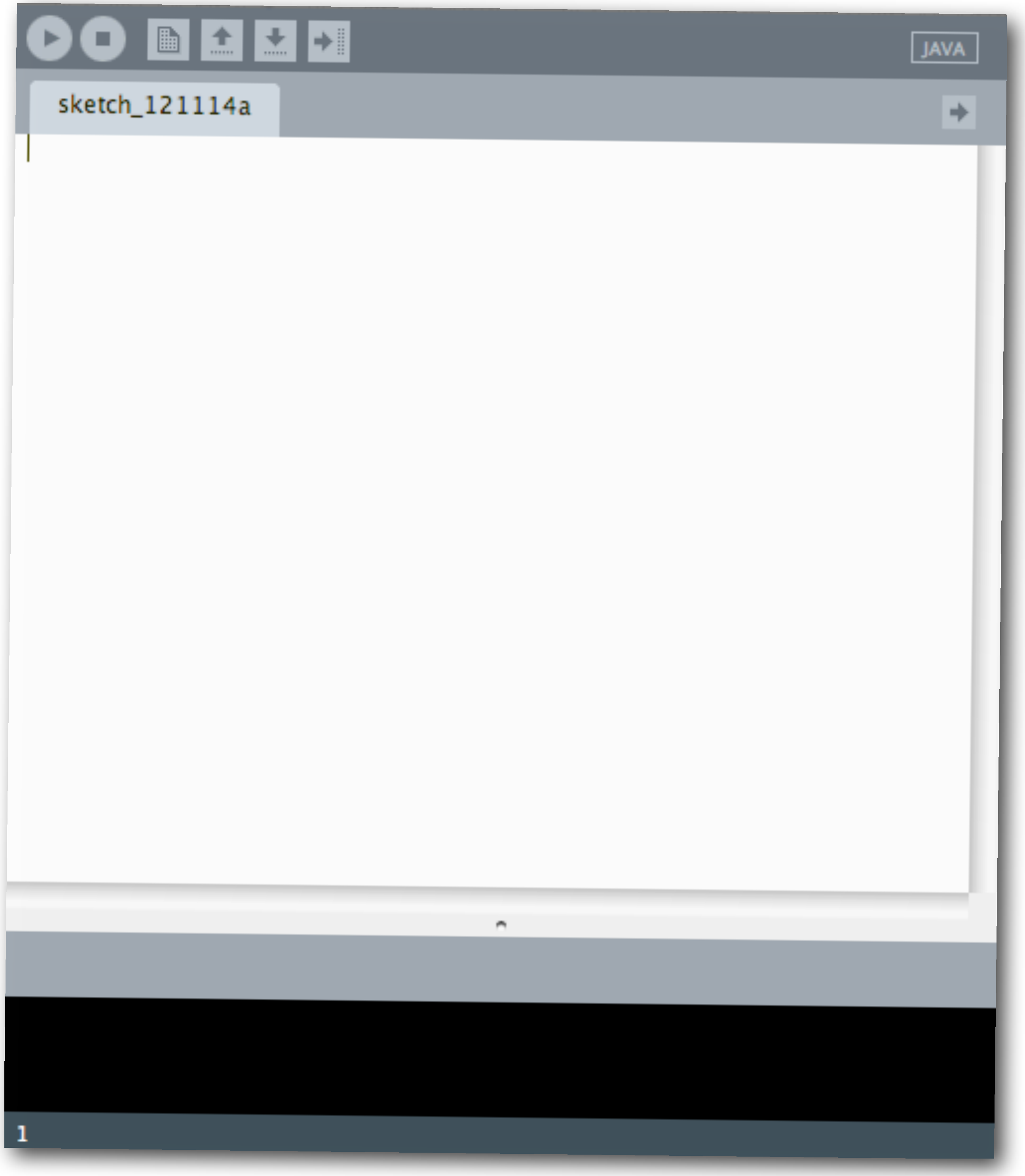
Intro to Processing

- Run the software

http://processing.org/

Intro to Processing

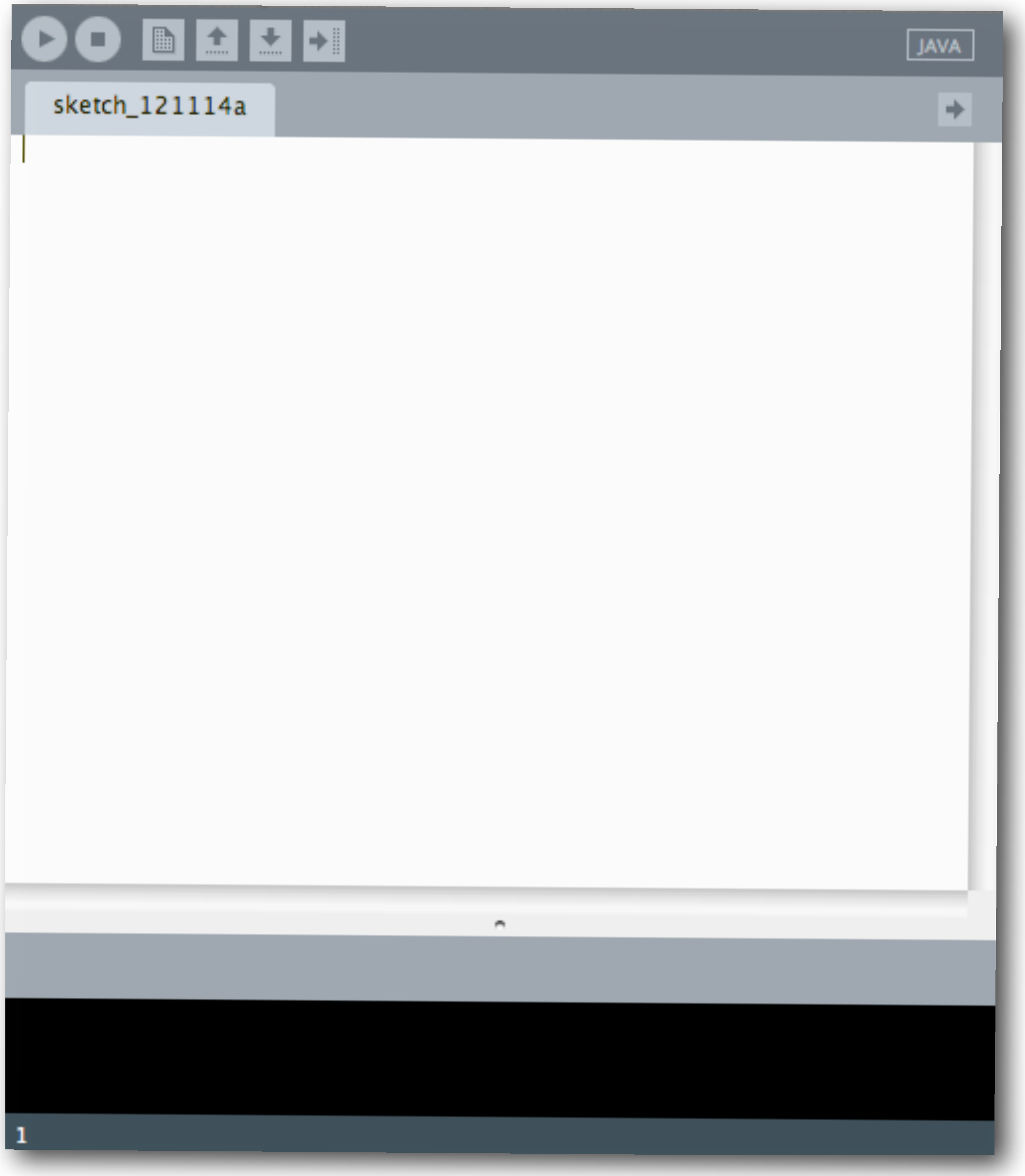
- Run the software



http://processing.org/

Intro to Processing

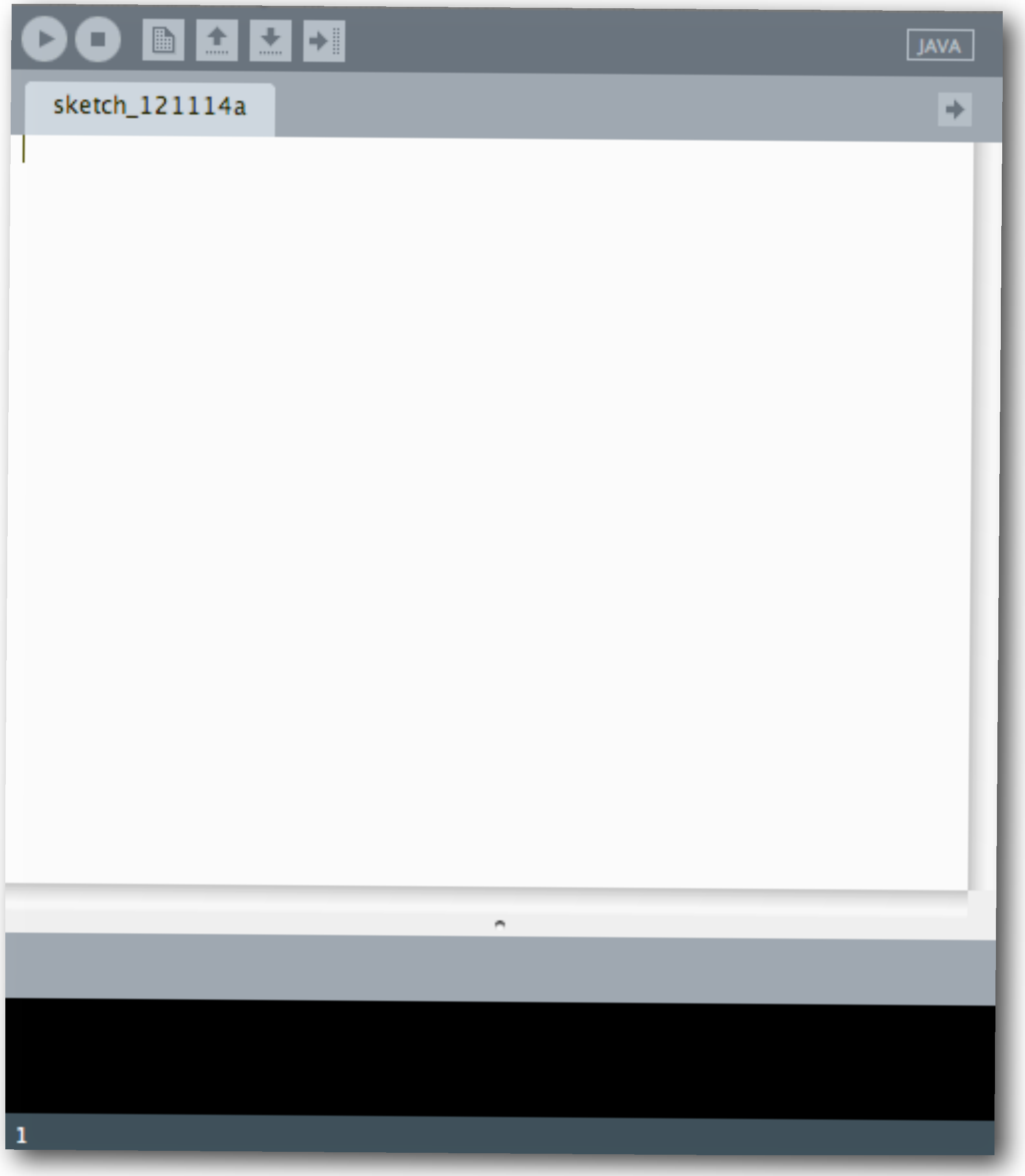
- Run the software



http://processing.org/

Intro to Processing

- Run the software

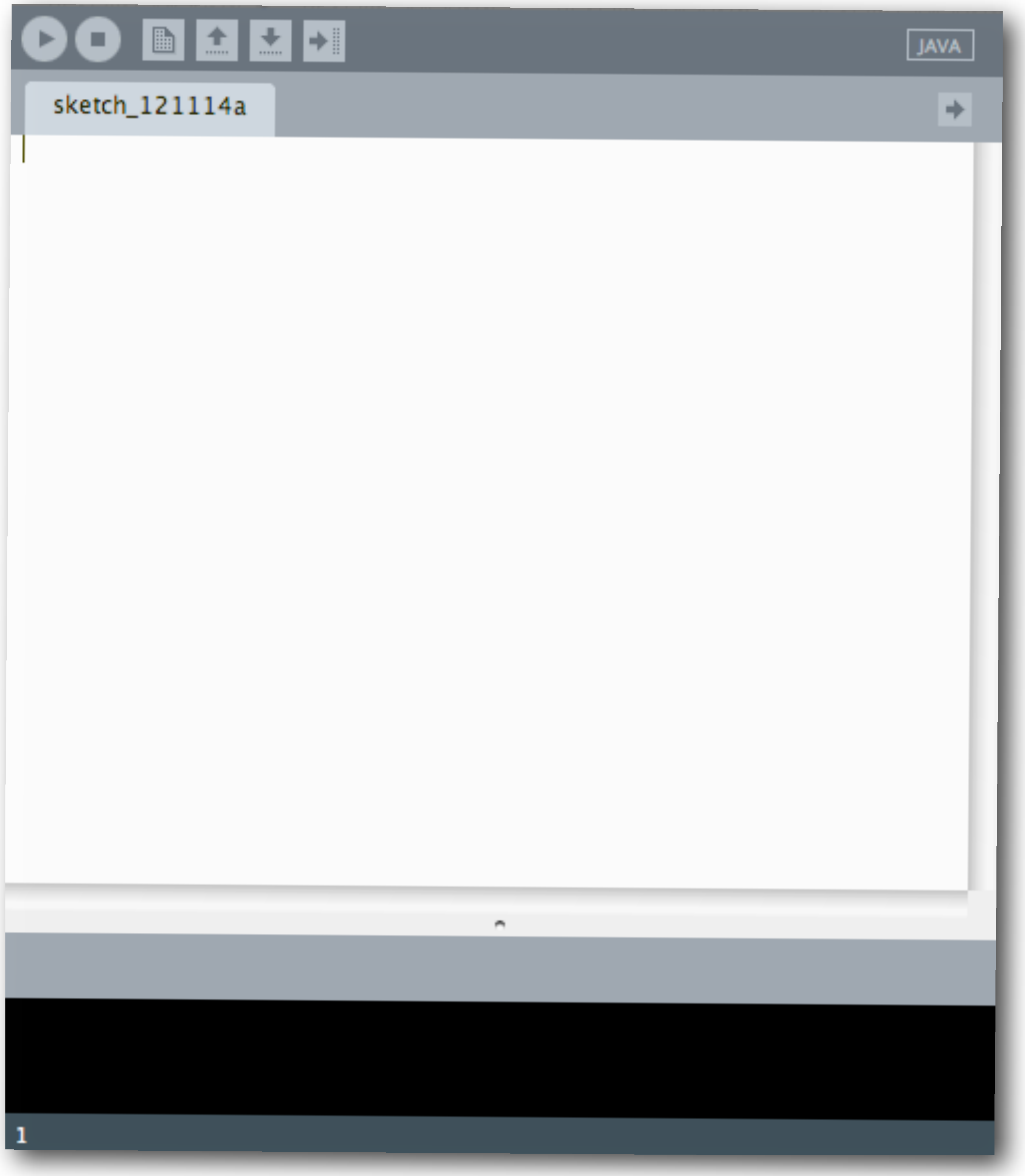


- This is the sketch window

http://processing.org/

Intro to Processing

- Run the software

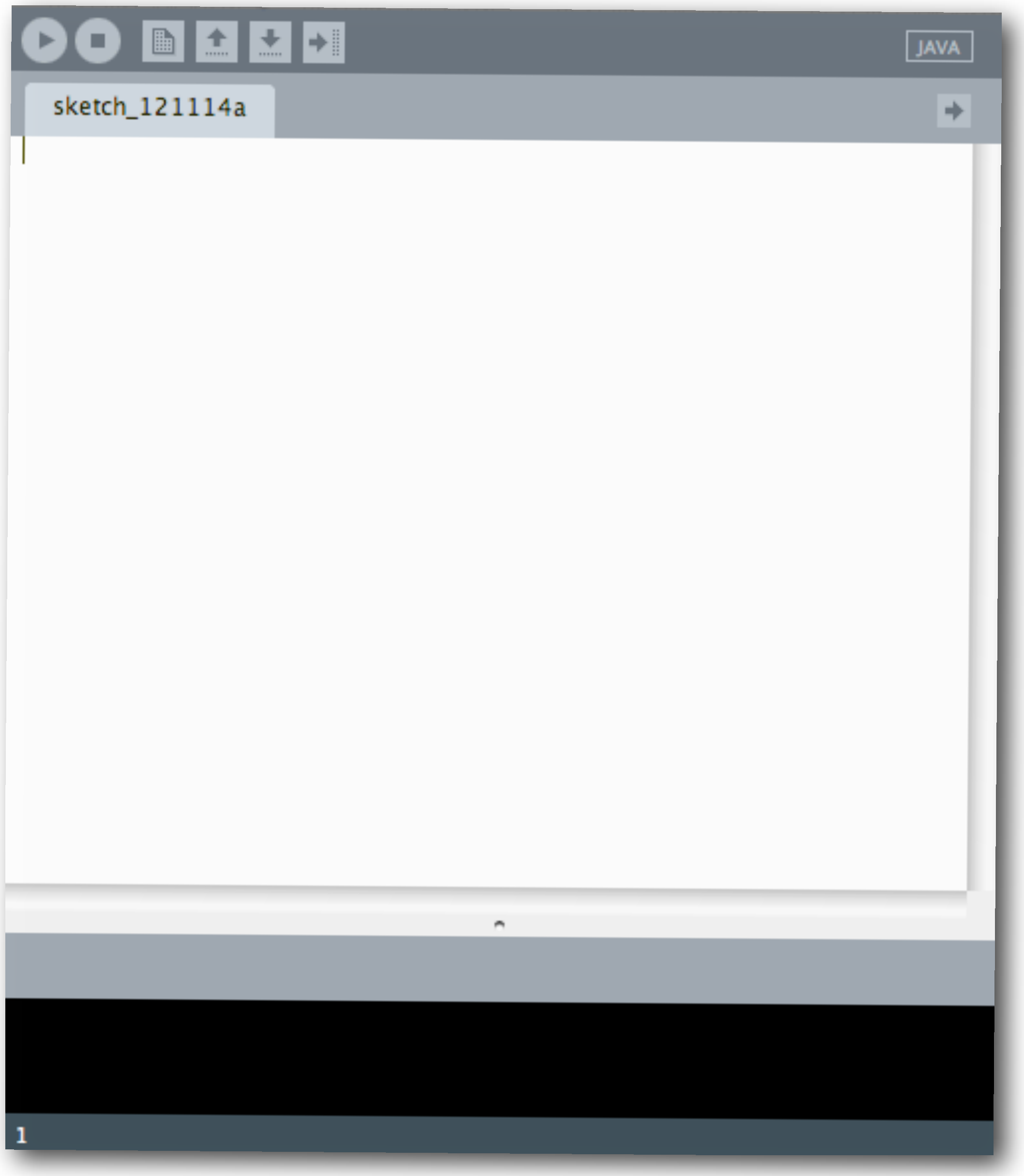


- This is the sketch window
  - It is part of the Processing Development Environment (PDE)

http://processing.org/

Intro to Processing

- Run the software

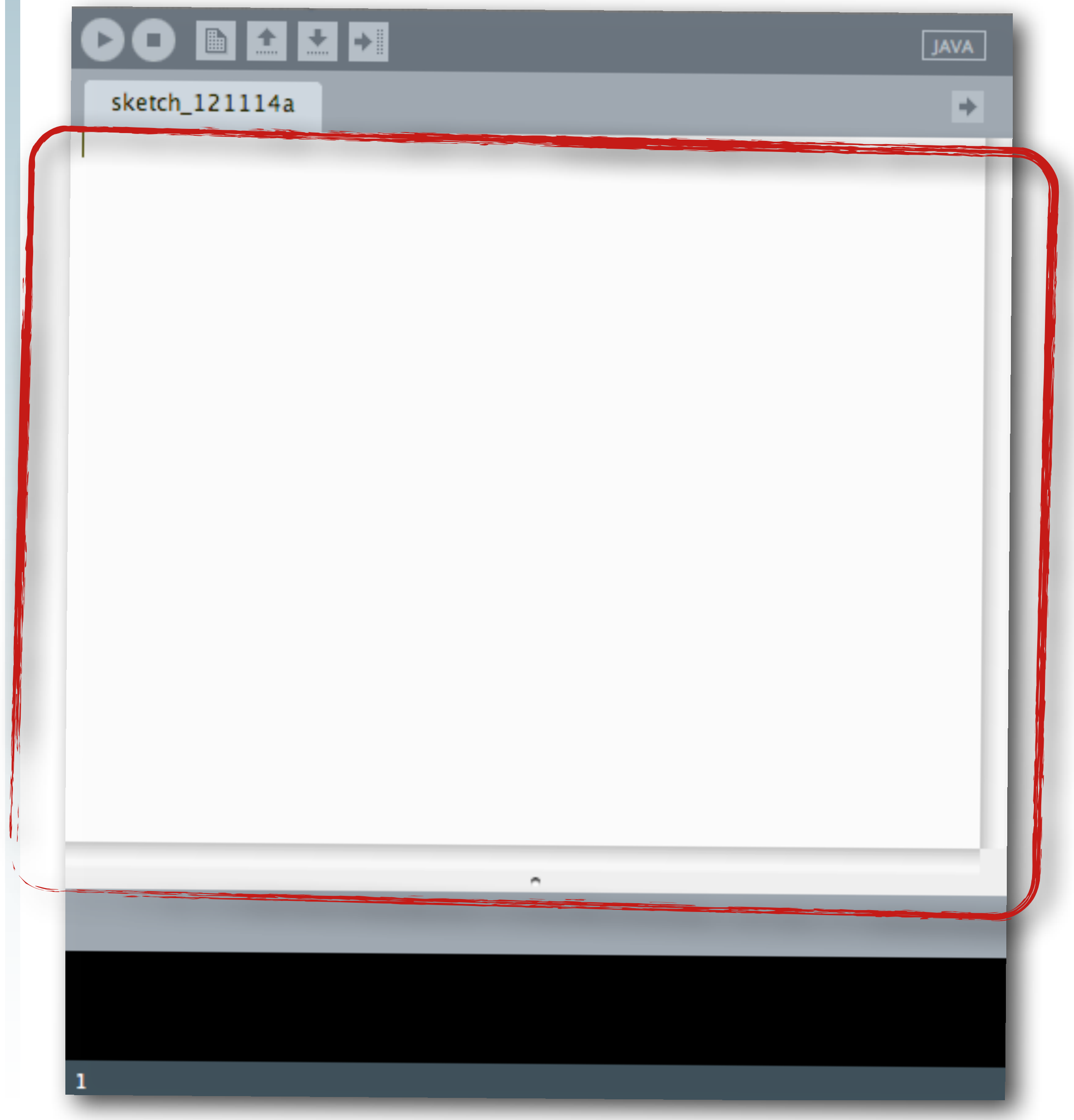


- This is the sketch window
  - It is part of the Processing Development Environment (PDE)
  - This is where you put your program's instructions

http://processing.org/

Intro to Processing

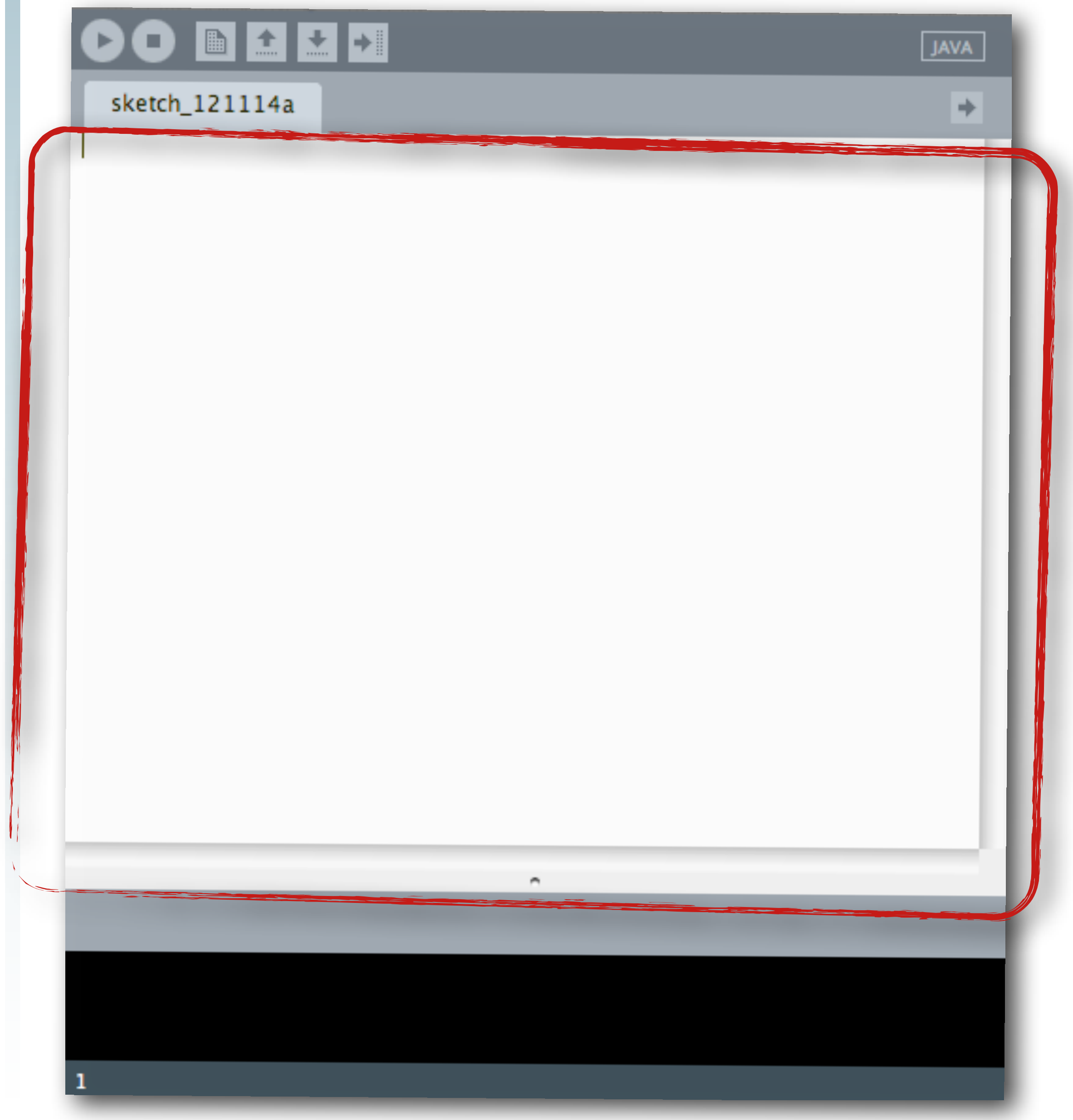
- Run the software



http://processing.org/

Intro to Processing

- Run the software

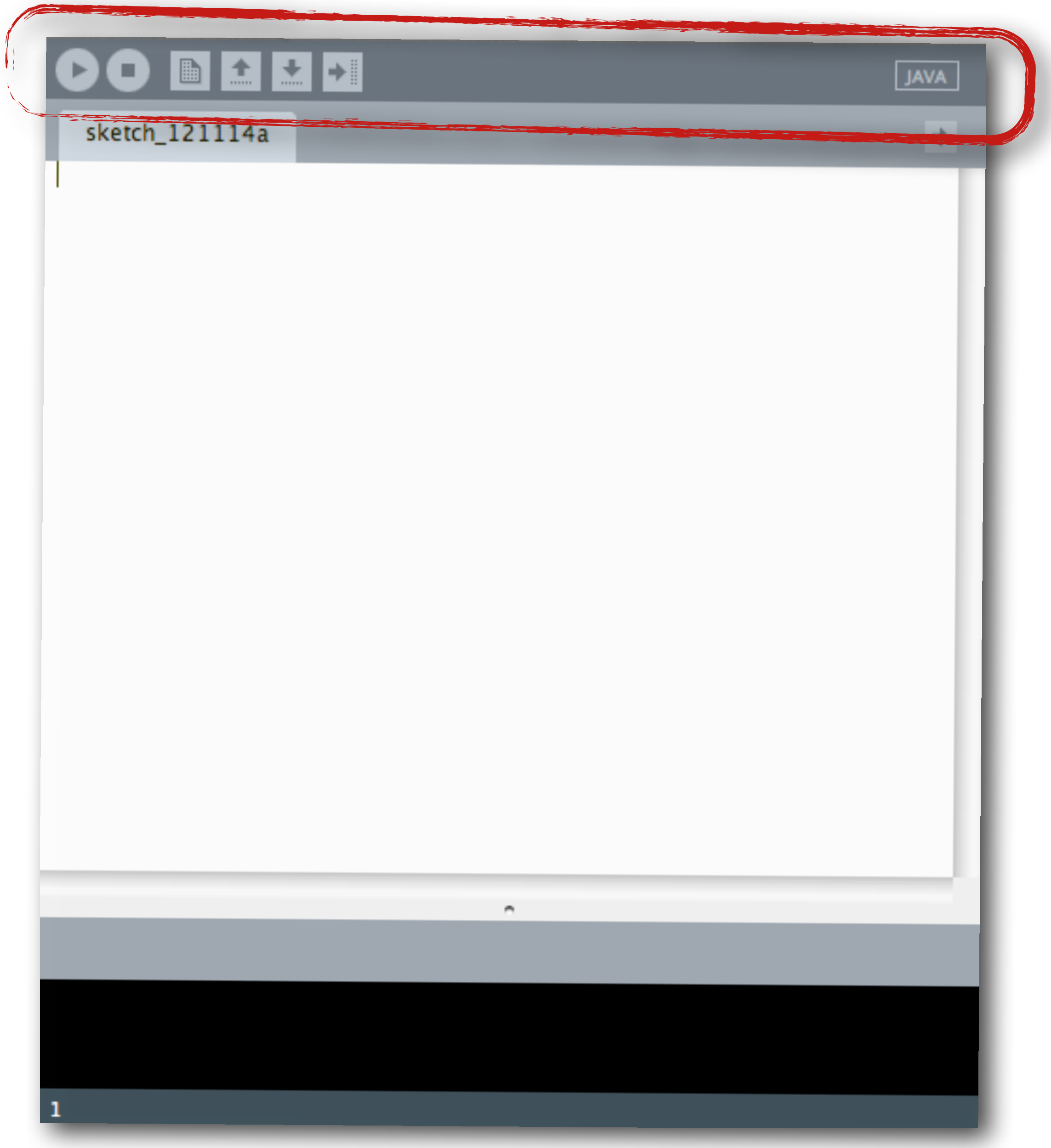


- This is the Text Editor

http://processing.org/

Intro to Processing

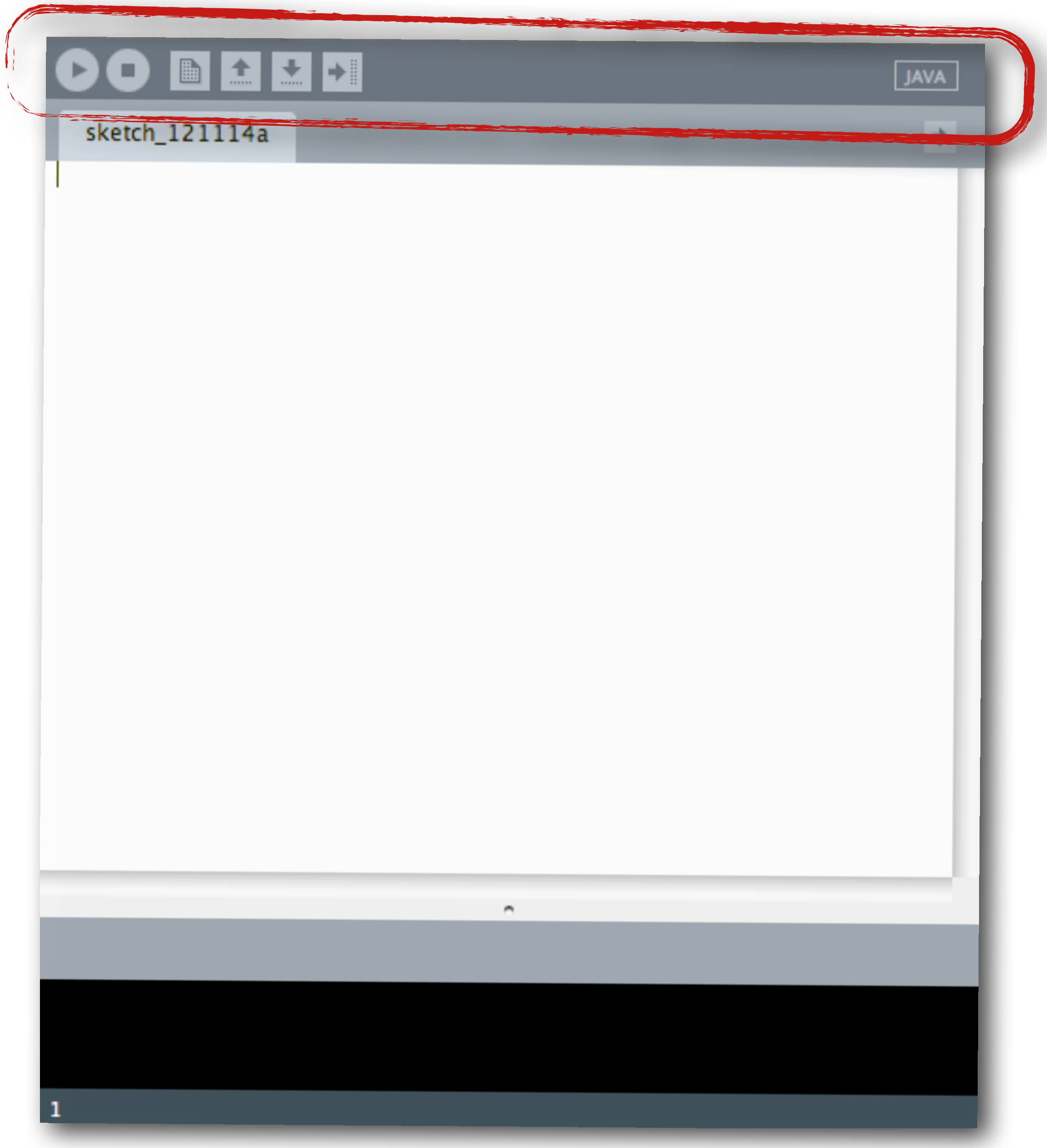
- Run the software



http://processing.org/

Intro to Processing

- Run the software

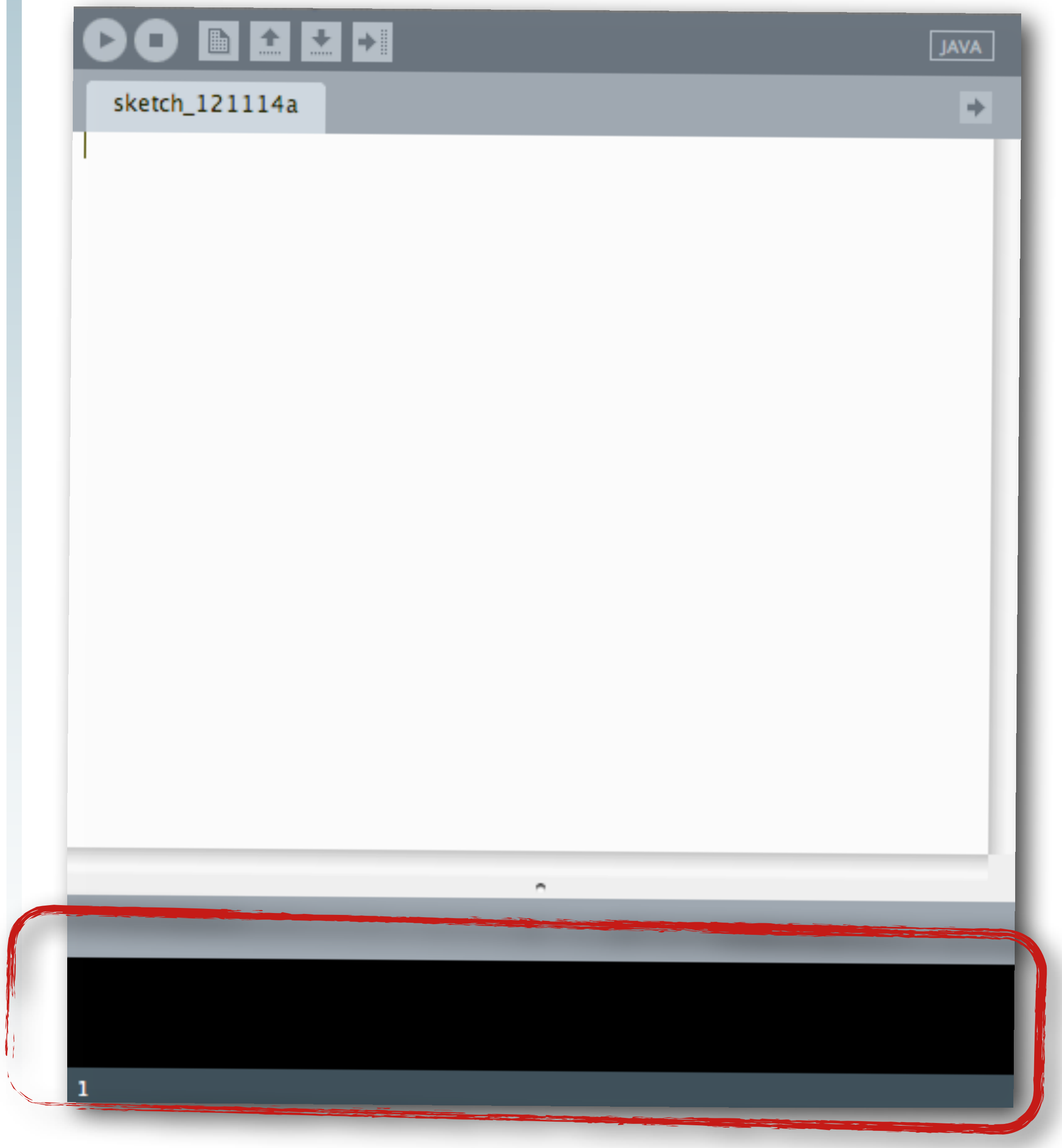


- This is the toolbar

http://processing.org/

Intro to Processing

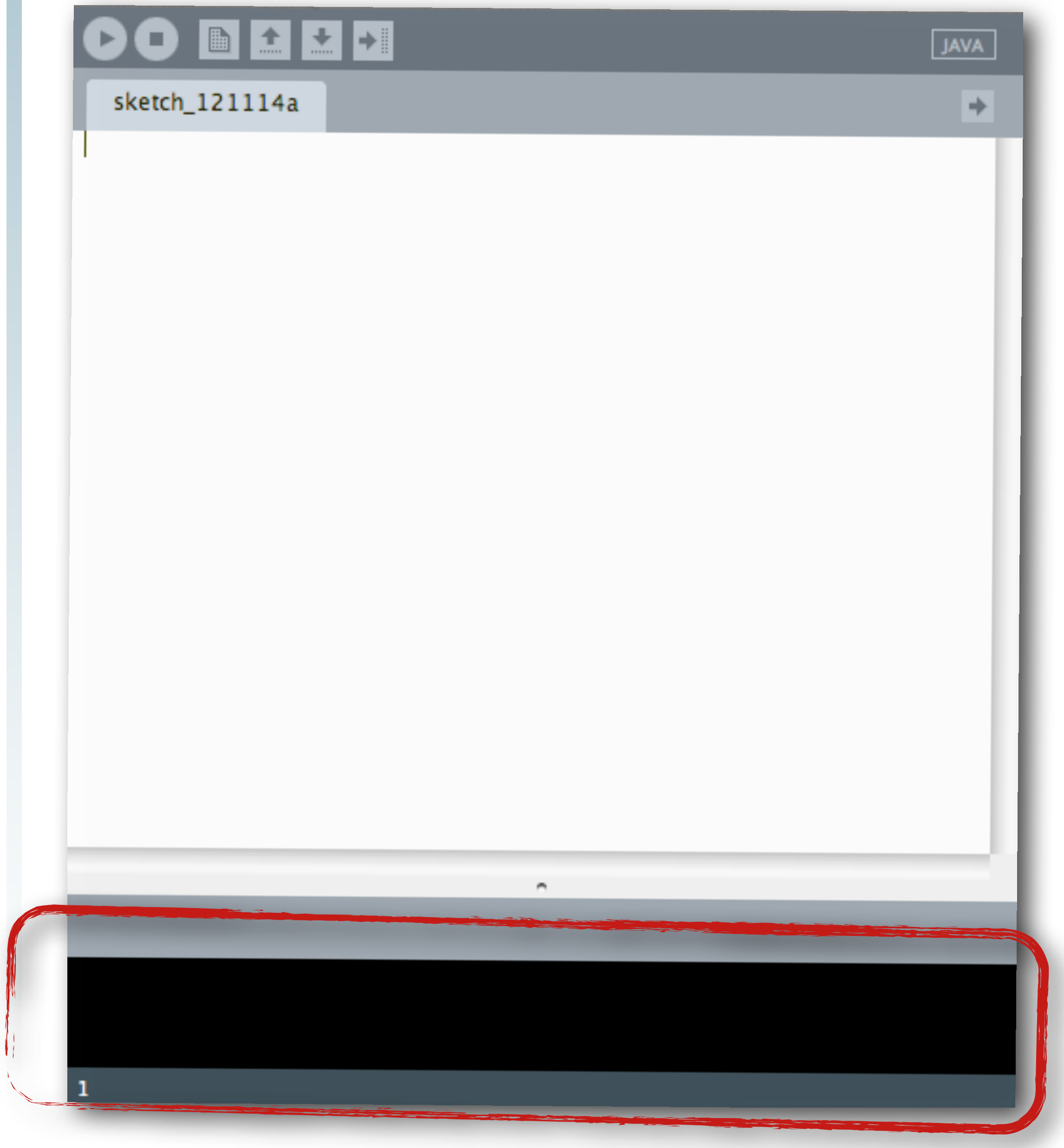
- Run the software



http://processing.org/

Intro to Processing

- Run the software

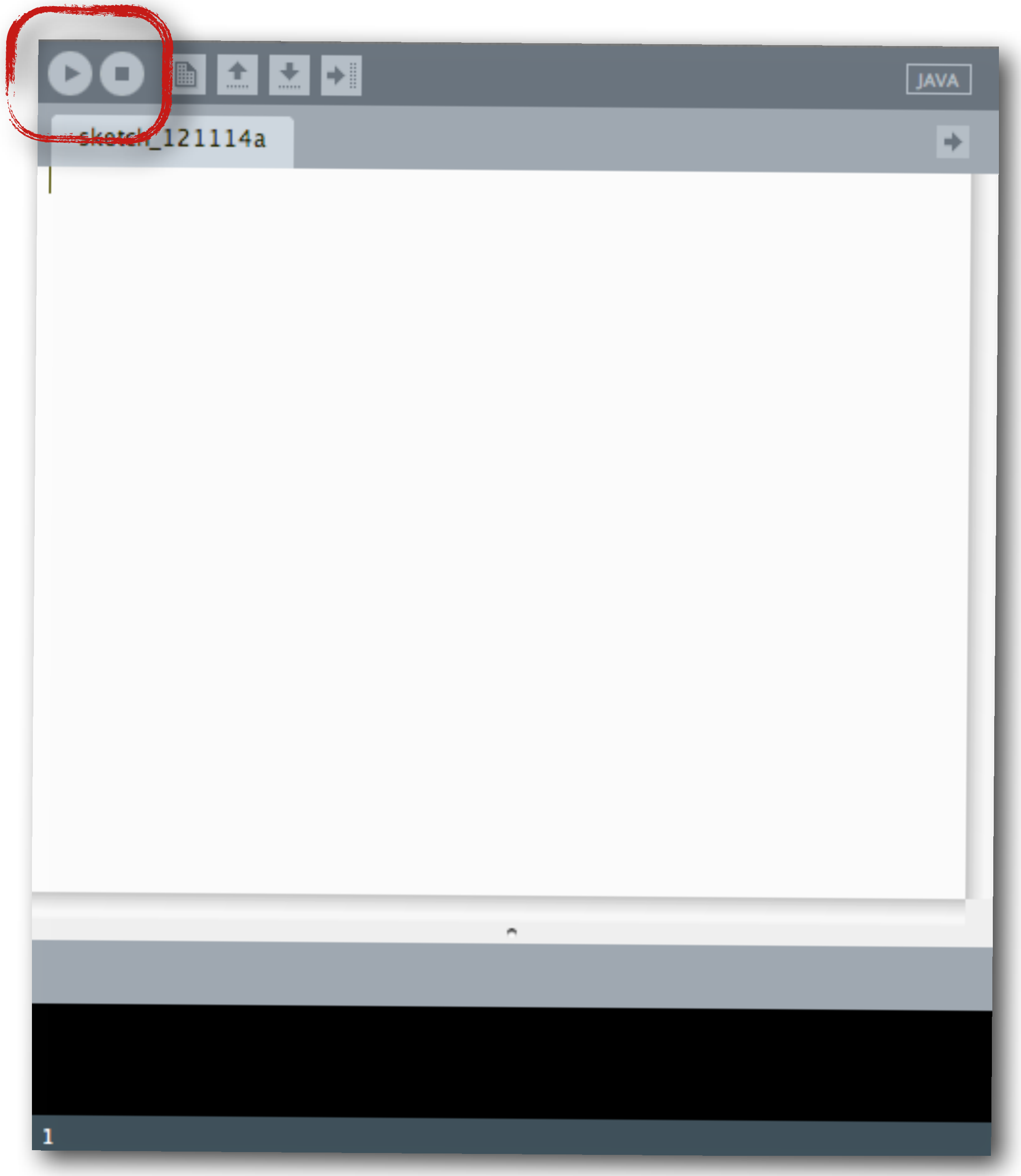


- This is the message area

http://processing.org/

Intro to Processing

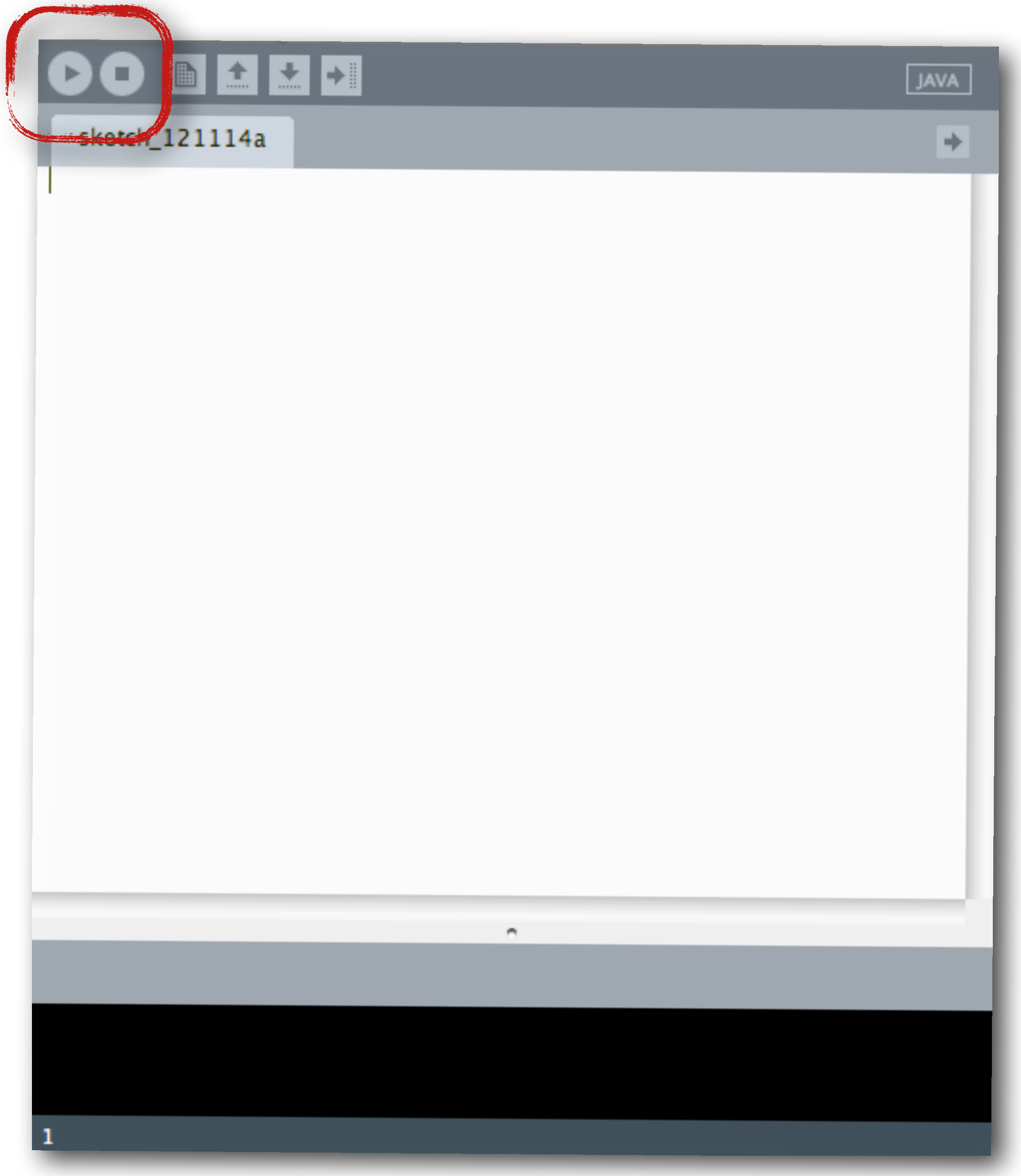
- Run the software



http://processing.org/

Intro to Processing

- Run the software

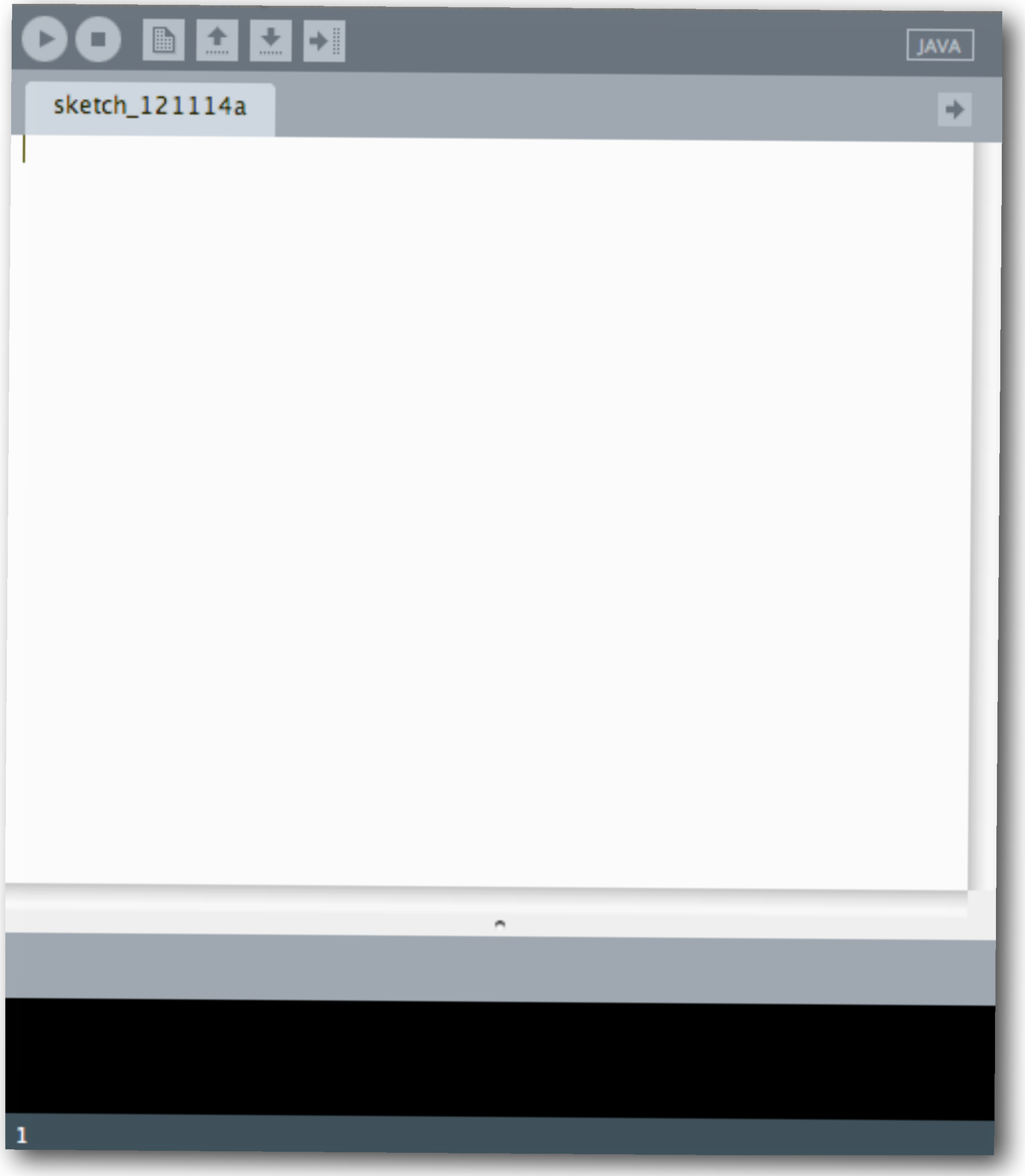


- This is the run and stop buttons

http://processing.org/

Intro to Processing

- Run the software

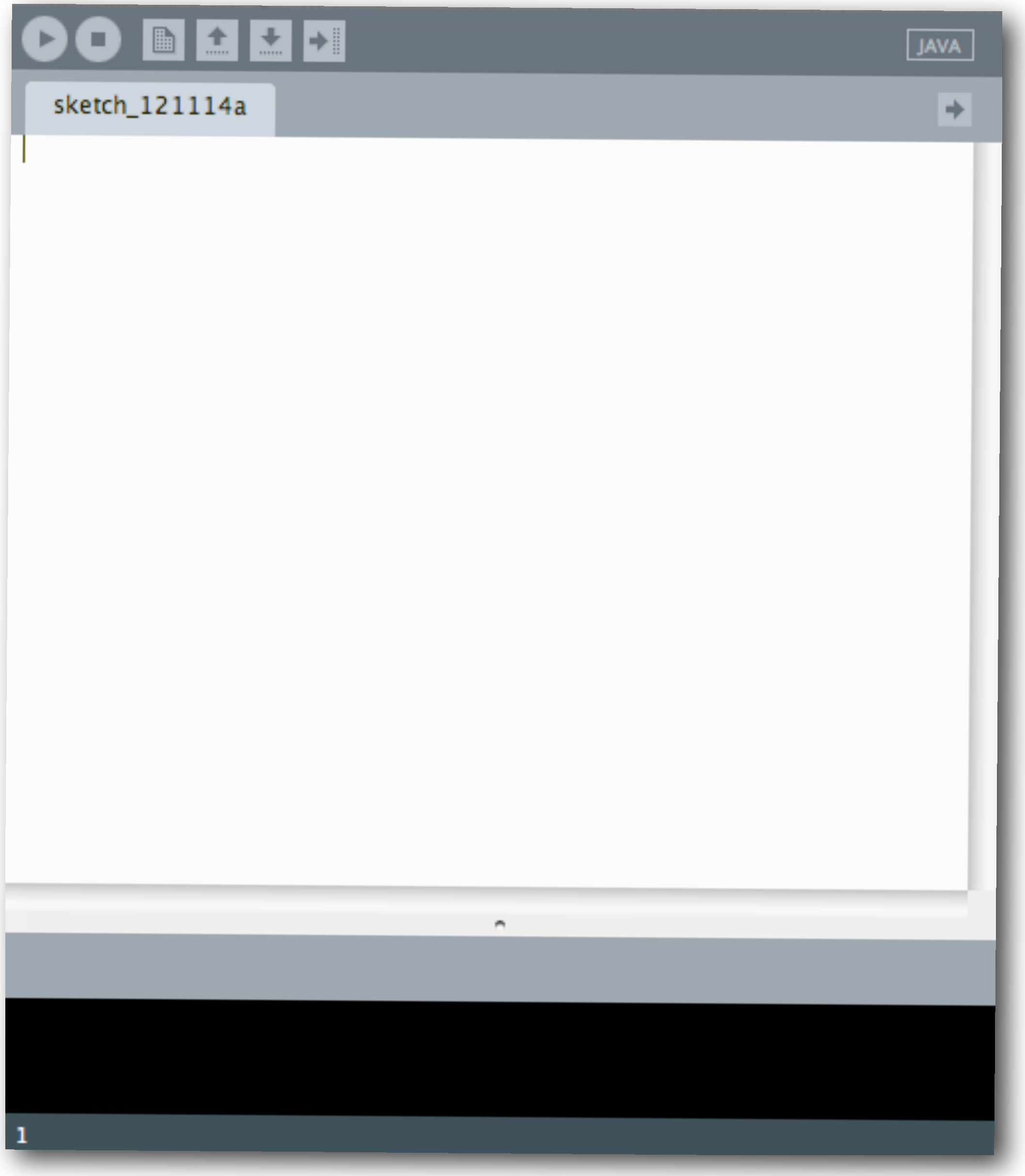


- To run a program
  - type in the program
  - hit run
  - look for the display window

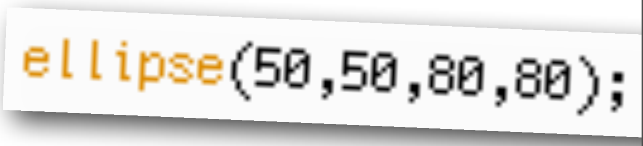
http://processing.org/

Intro to Processing

- Run the software



- To run a program
  - type in the program
  - hit run
  - look for the display window

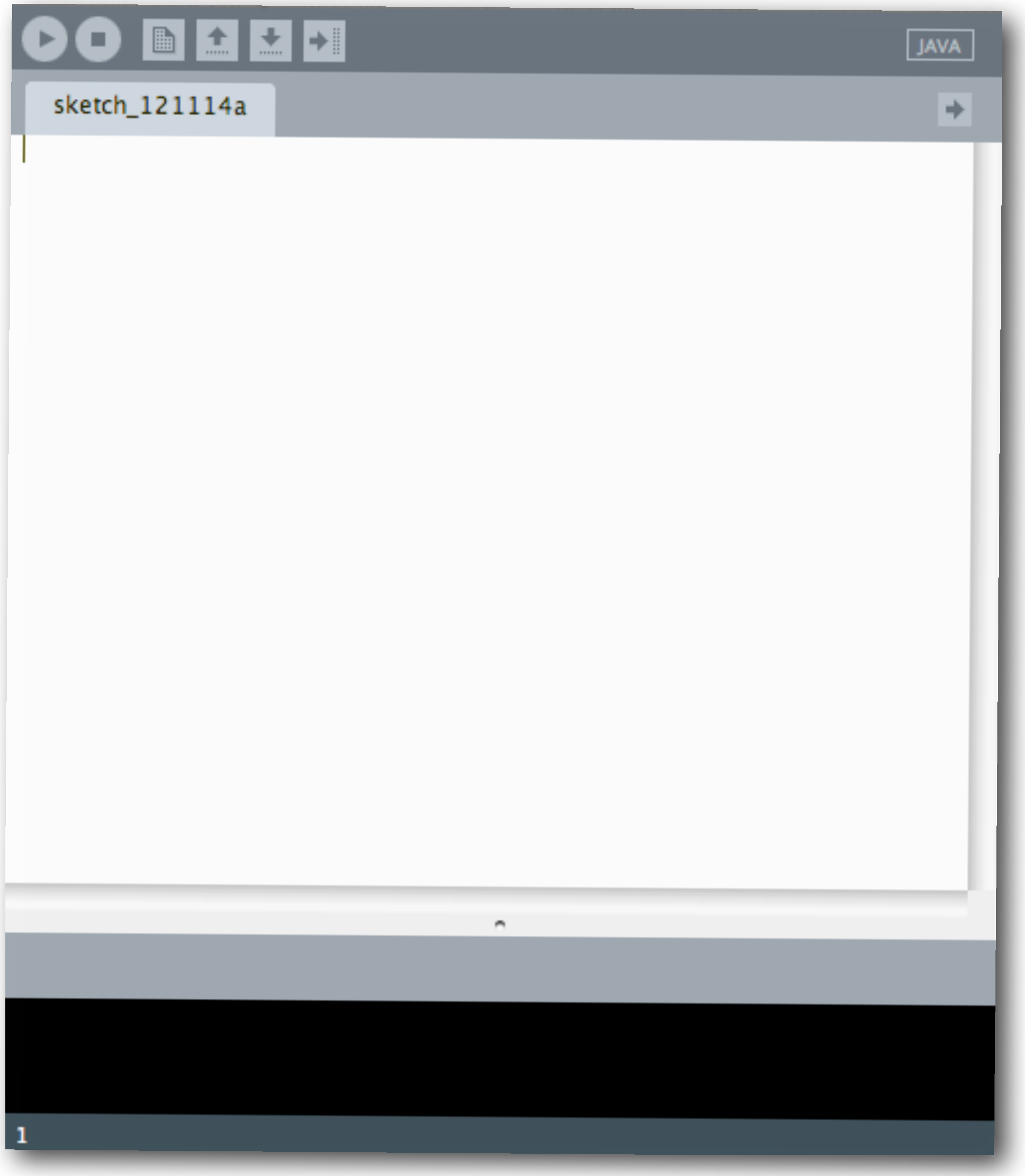


```
ellipse(50,50,80,80);
```

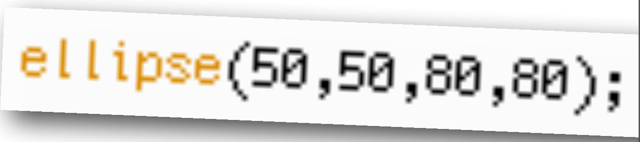
http://processing.org/

Intro to Processing

- Run the software



- To run a program
  - type in the program
  - hit run
  - look for the display window

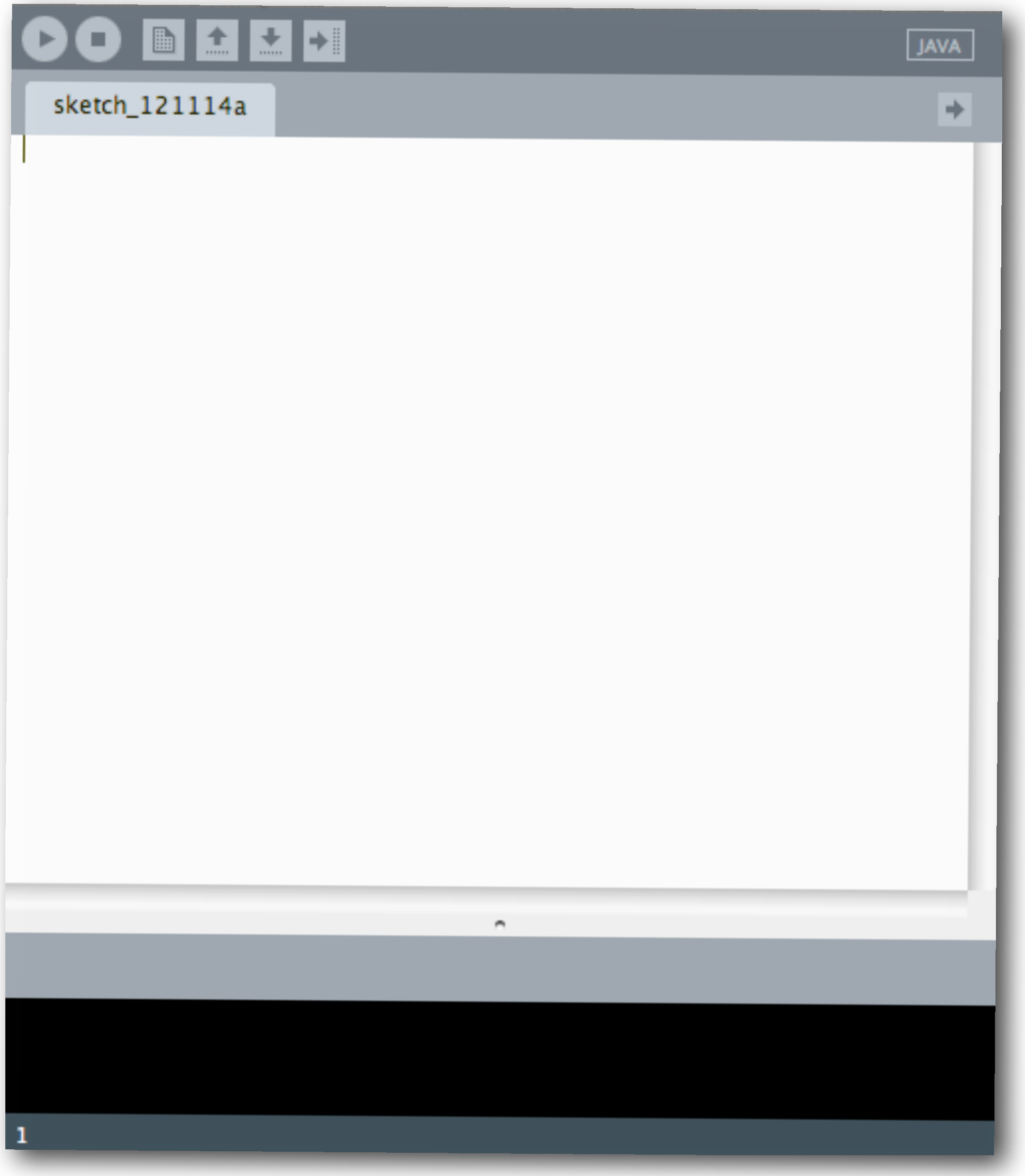


```
ellipse(50,50,80,80);
```

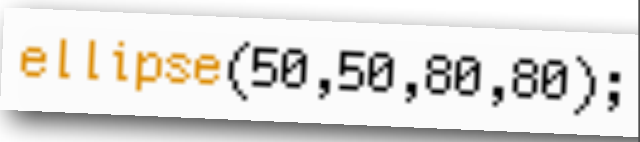
http://processing.org/

Intro to Processing

- Run the software

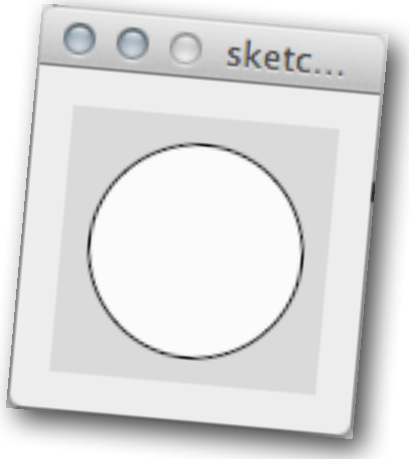


- To run a program
  - type in the program



```
ellipse(50,50,80,80);
```

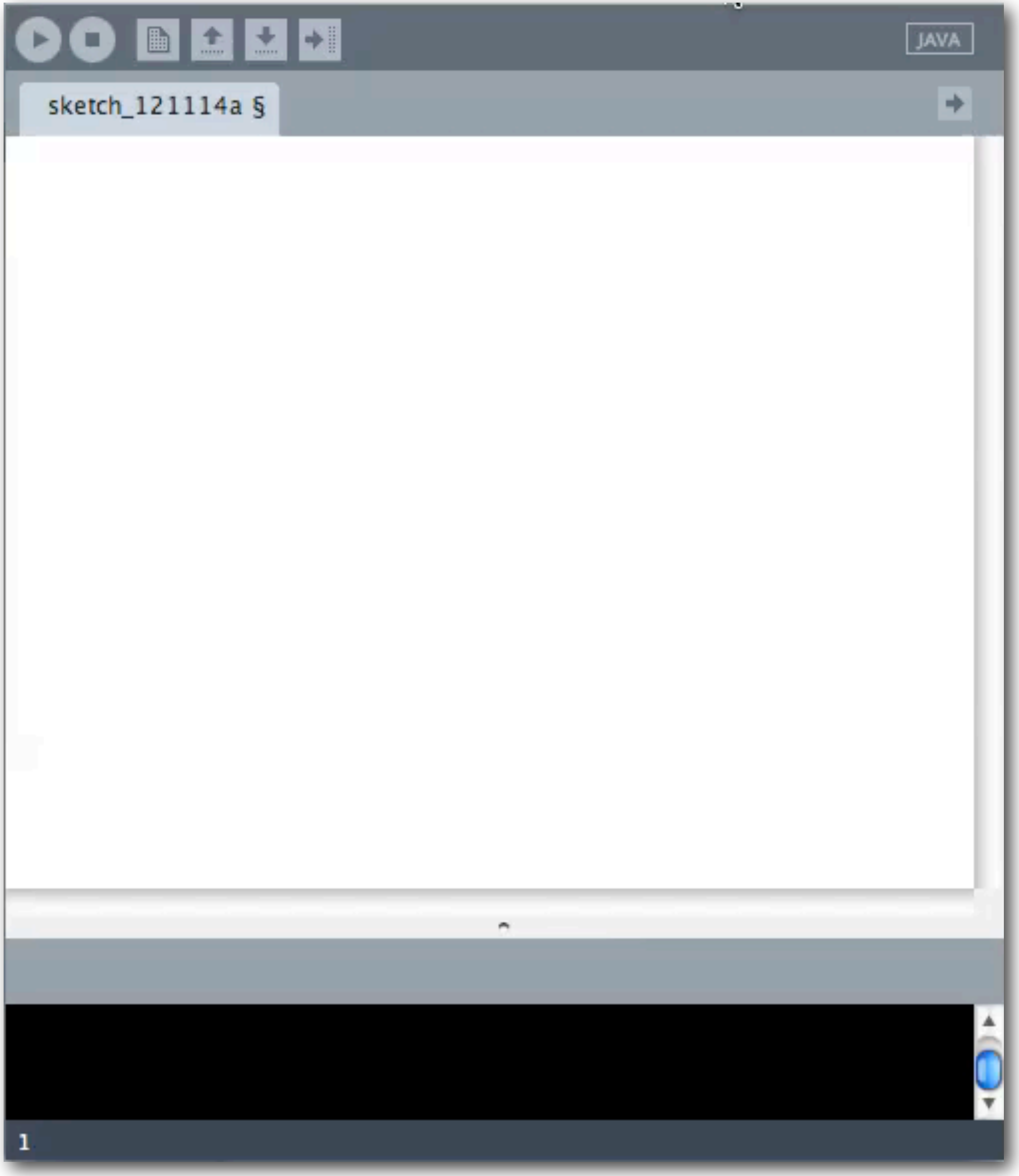
  - hit run
  - look for the display window



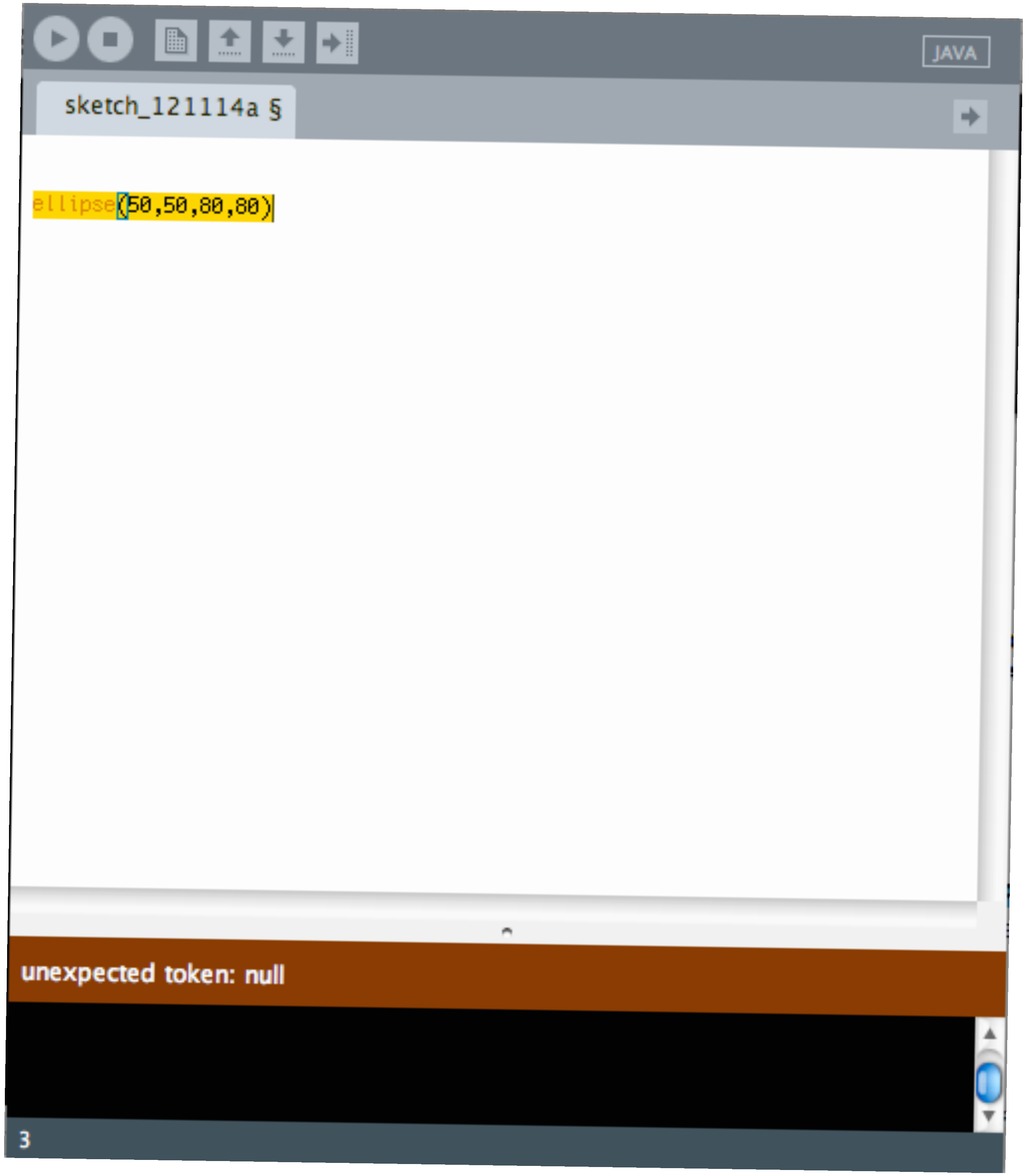
http://processing.org/

Intro to Processing

Intro to Processing



Intro to Processing



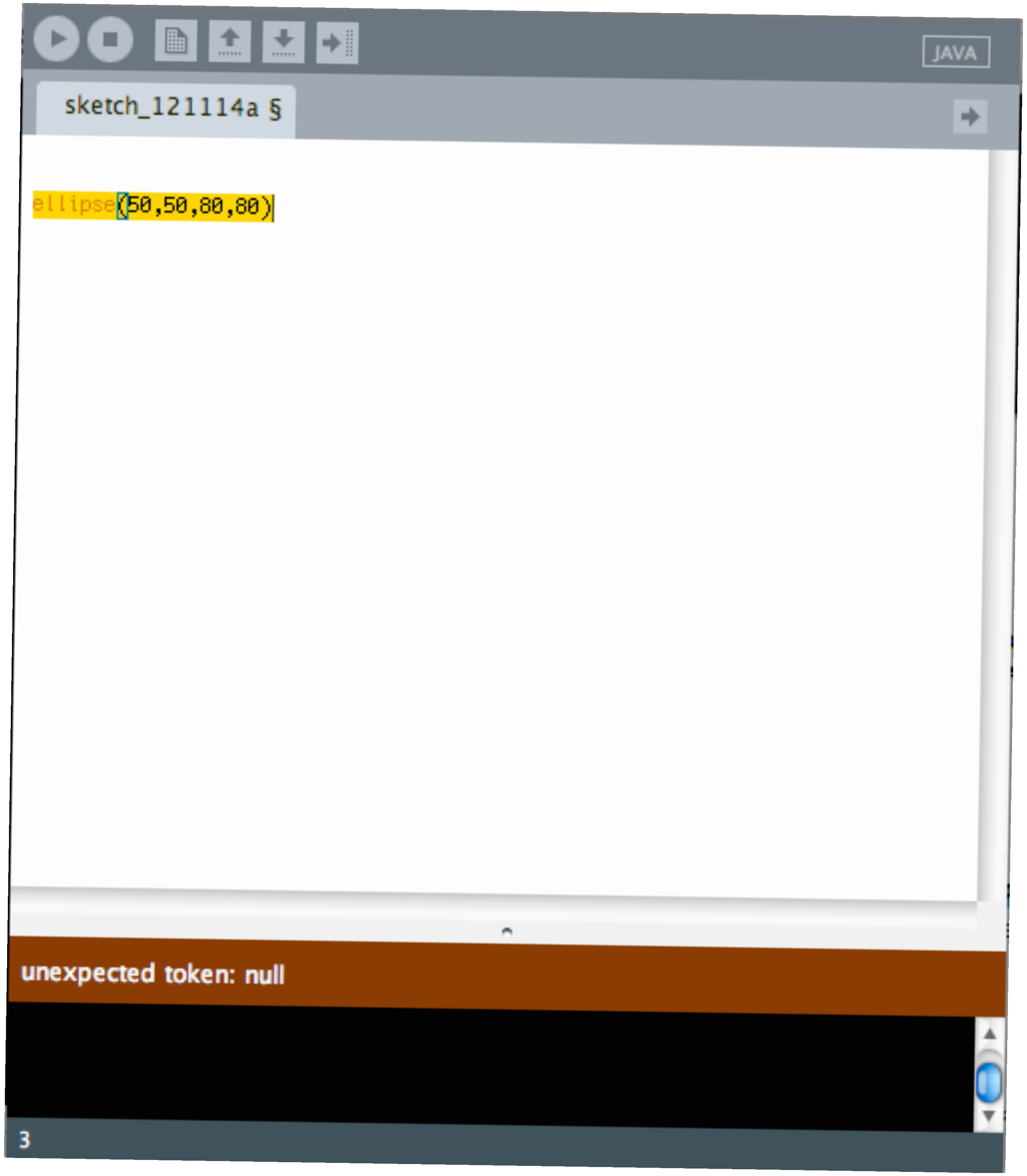
```
ellipse(50,50,80,80)
```

unexpected token: null

3

Intro to Processing

- What if something goes wrong?

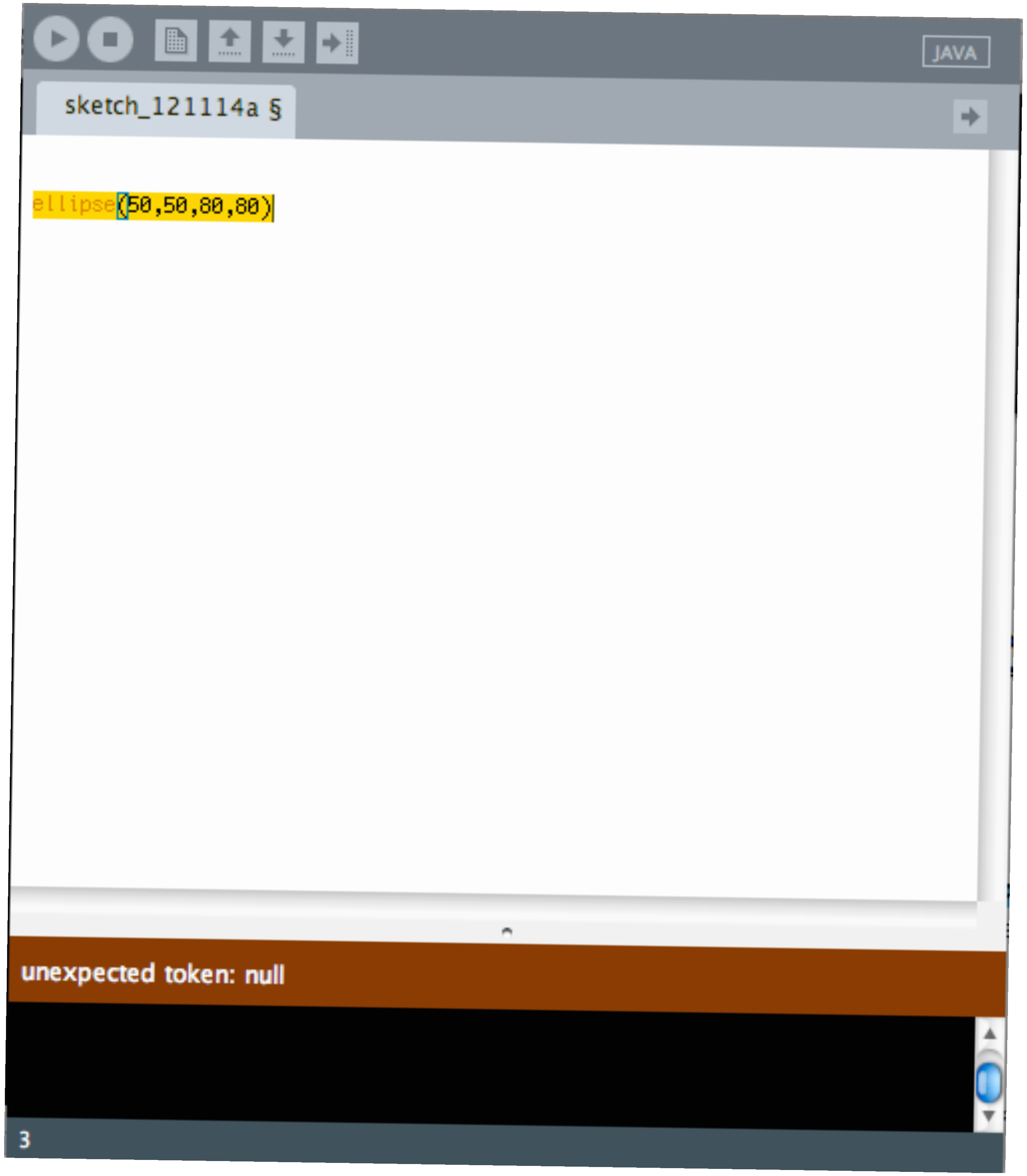


```
ellipse(50,50,80,80)
```

```
unexpected token: null
```

Intro to Processing

- What if something goes wrong?



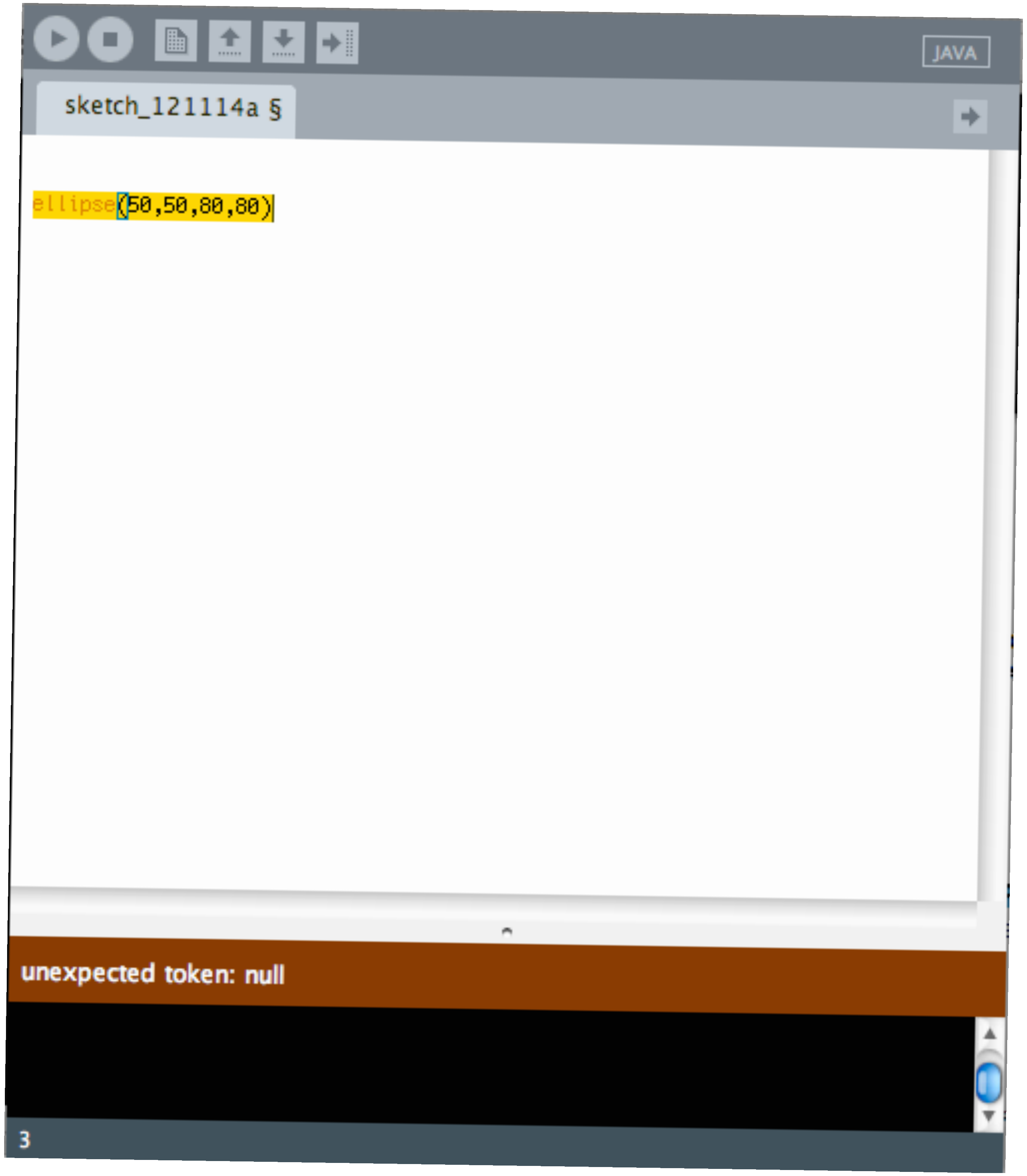
```
ellipse(50,50,80,80)
```

```
unexpected token: null
```

- An error will show up in the message area

Intro to Processing

- What if something goes wrong?



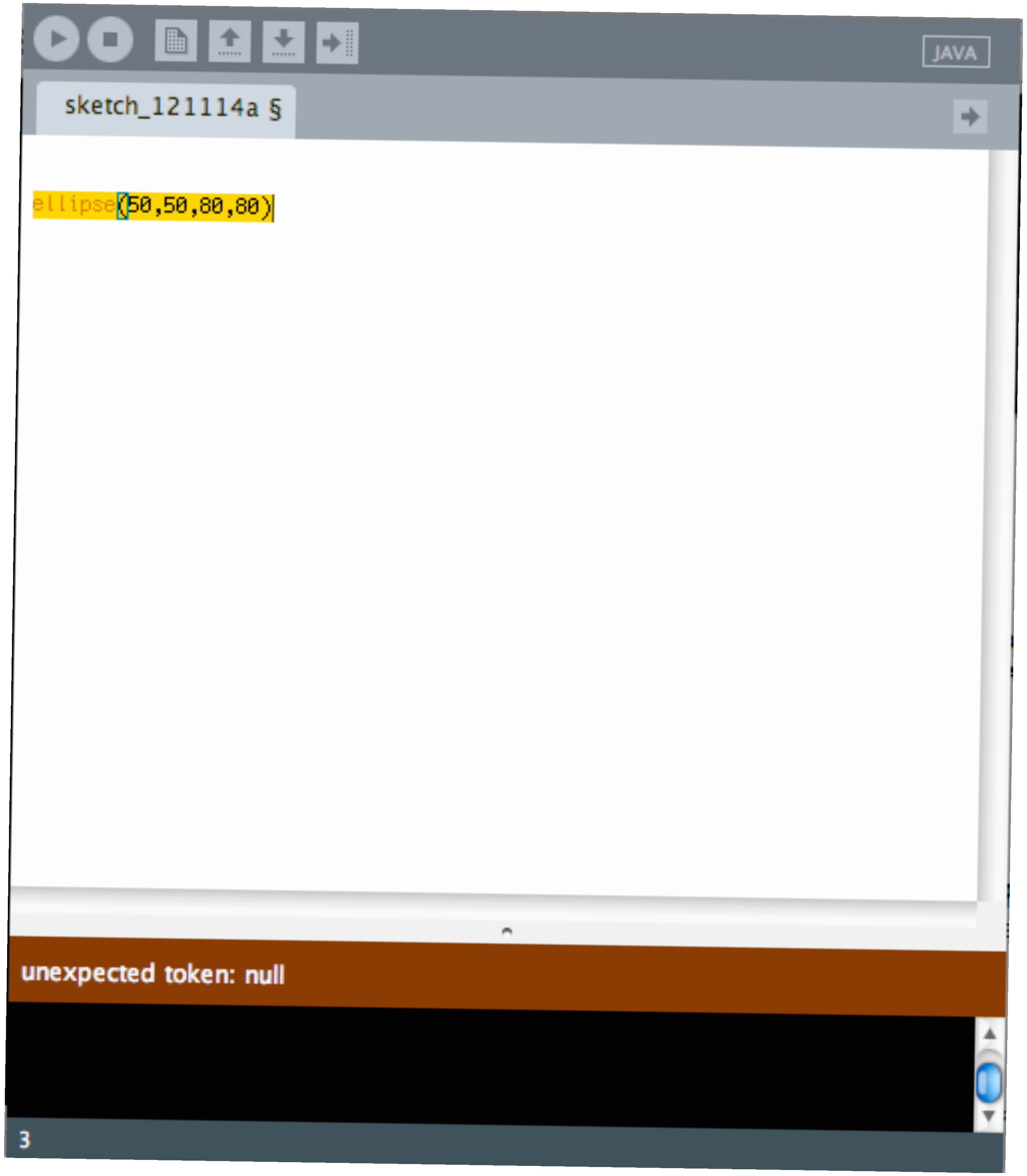
```
ellipse(50,50,80,80)
```

```
unexpected token: null
```

- An error will show up in the message area
- Sometimes it will give you a clue about the problem

Intro to Processing

- What if something goes wrong?



```
ellipse(50,50,80,80)
```

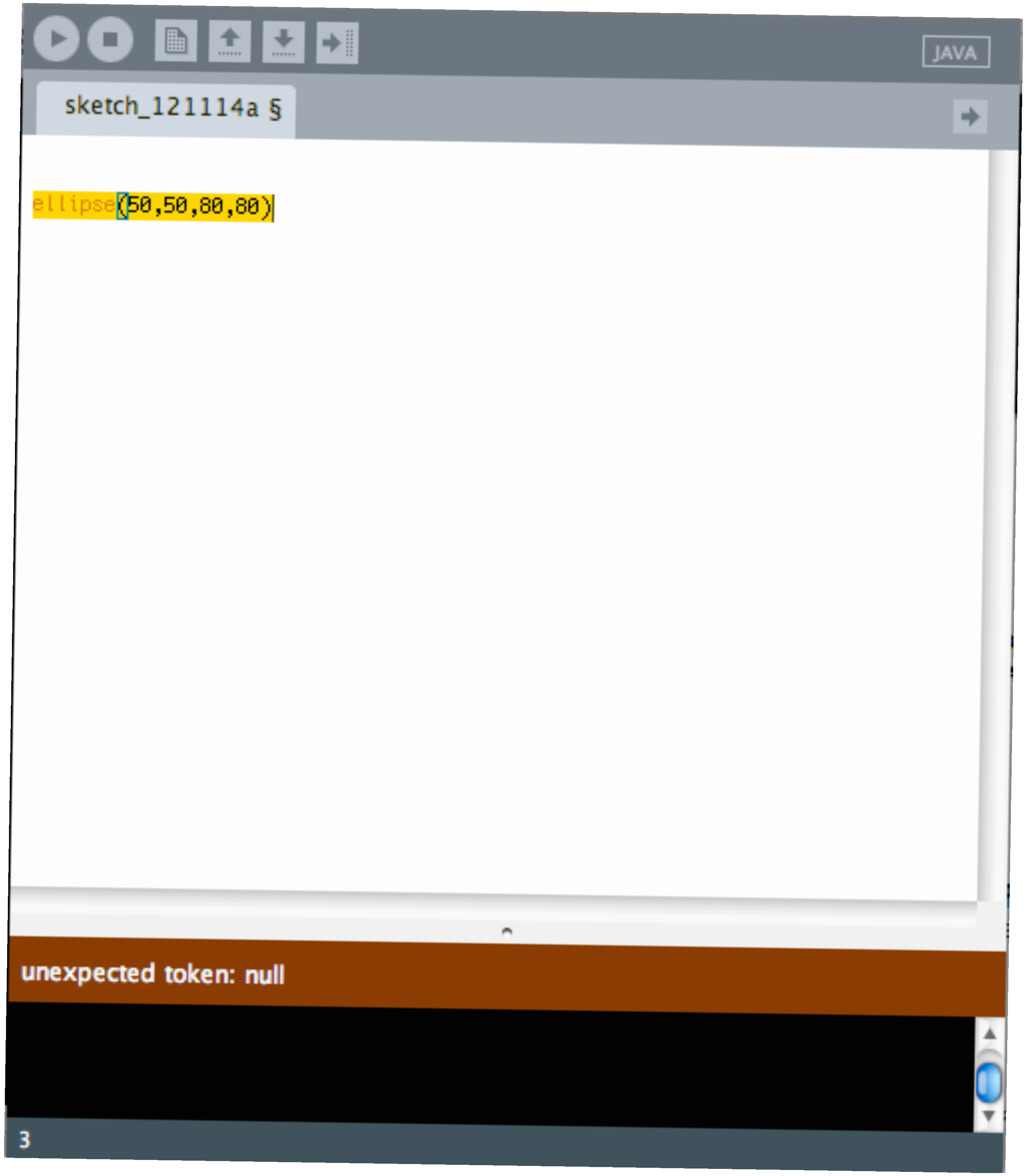
unexpected token: null

3

- An error will show up in the message area
- Sometimes it will give you a clue about the problem
- Make sure you are using parentheses in pairs

Intro to Processing

- What if something goes wrong?



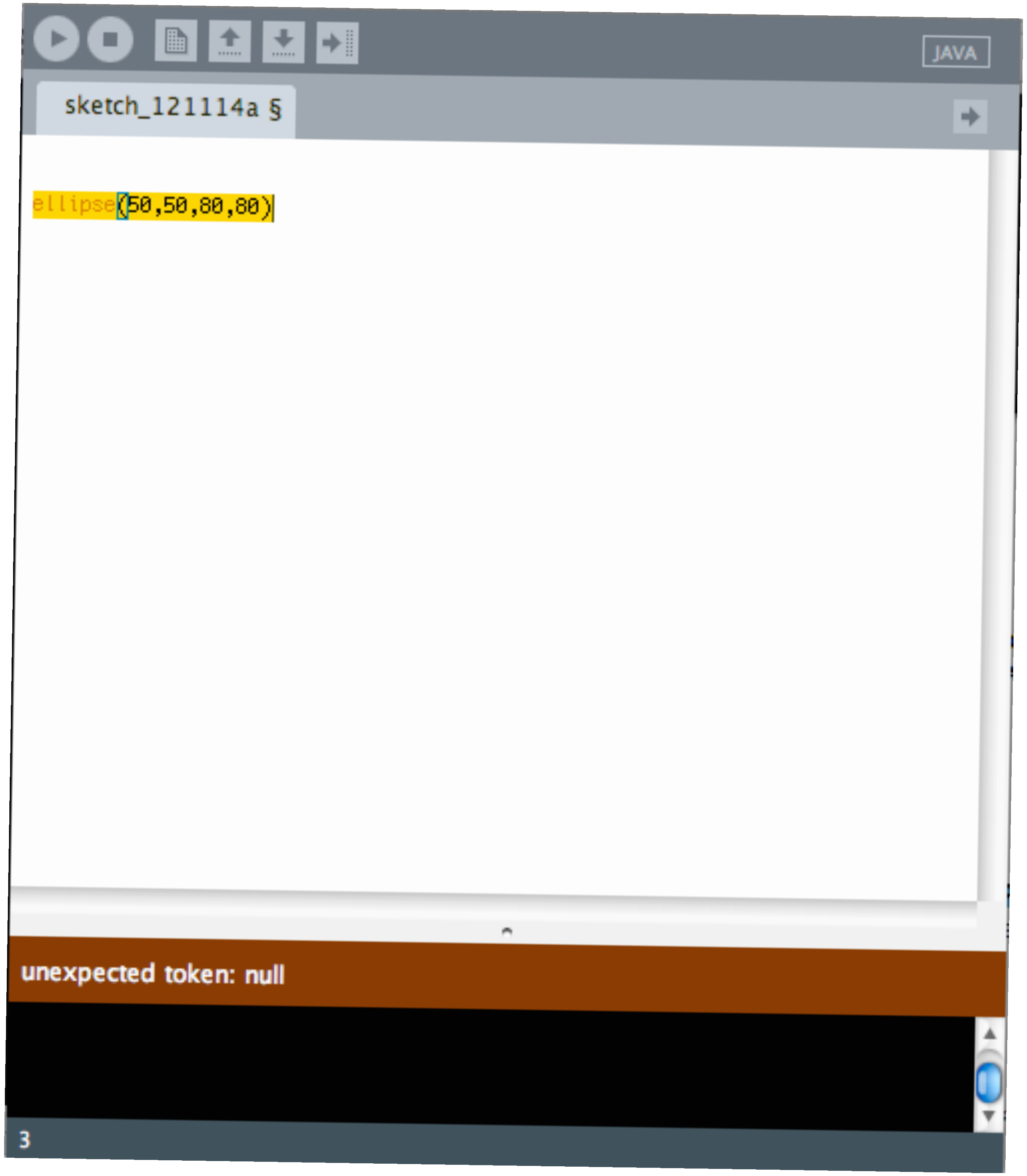
```
ellipse(50,50,80,80)
```

```
unexpected token: null
```

- An error will show up in the message area
- Sometimes it will give you a clue about the problem
- Make sure you are using parentheses in pairs
- Make sure you end a line with a semi-colon

Intro to Processing

- What if something goes wrong?

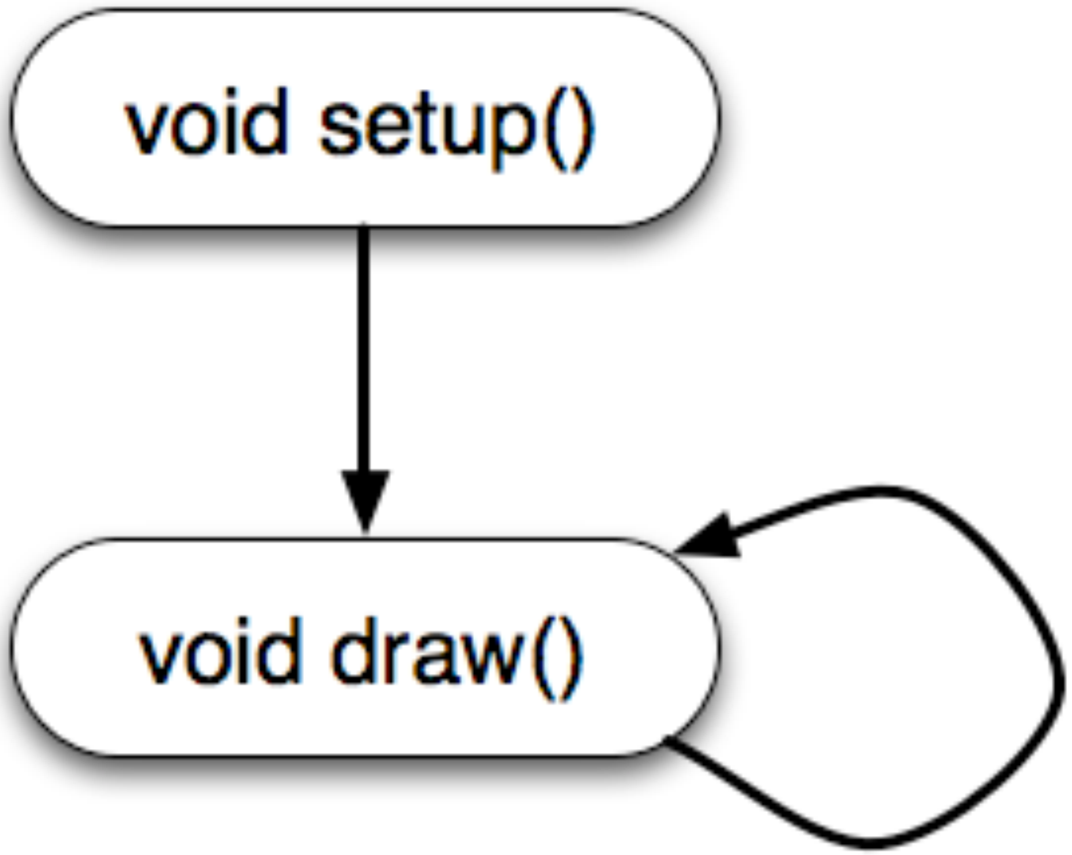


```
ellipse(50,50,80,80)
```

```
unexpected token: null
```

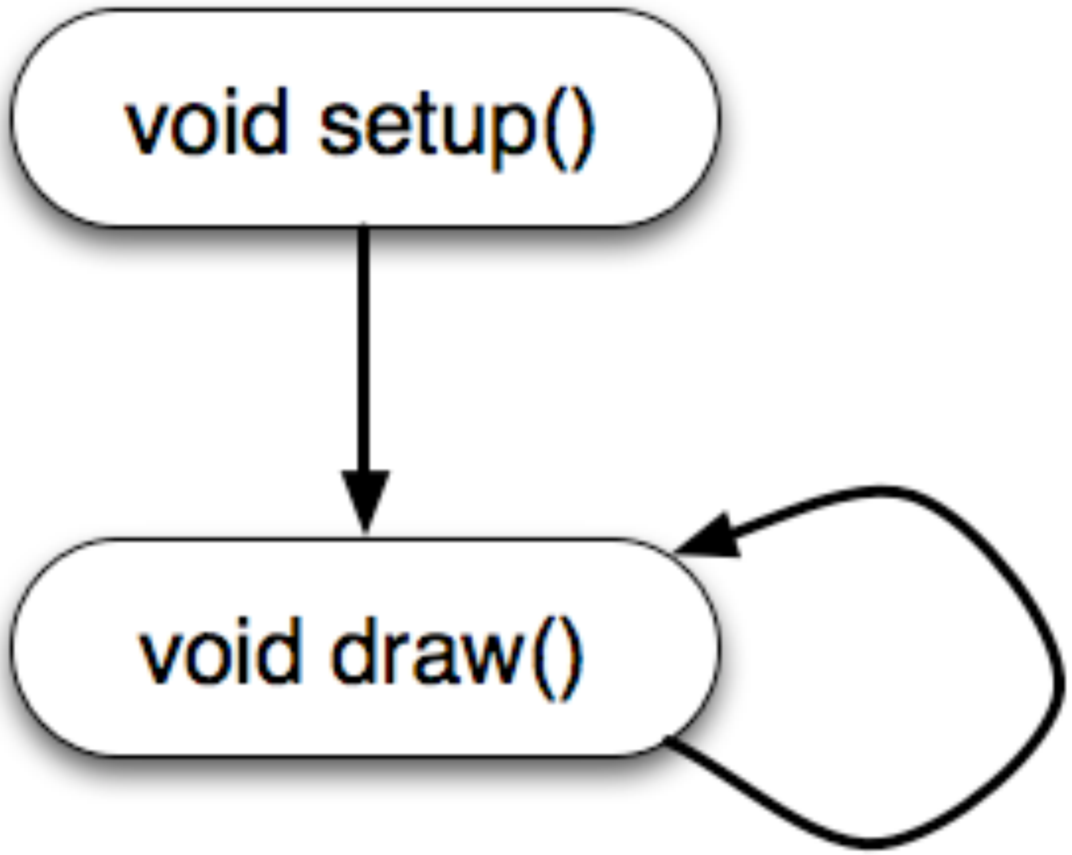
- An error will show up in the message area
- Sometimes it will give you a clue about the problem
- Make sure you are using parentheses in pairs
- Make sure you end a line with a semi-colon
- Make sure you have the right number of parameters for your function

Intro to Processing



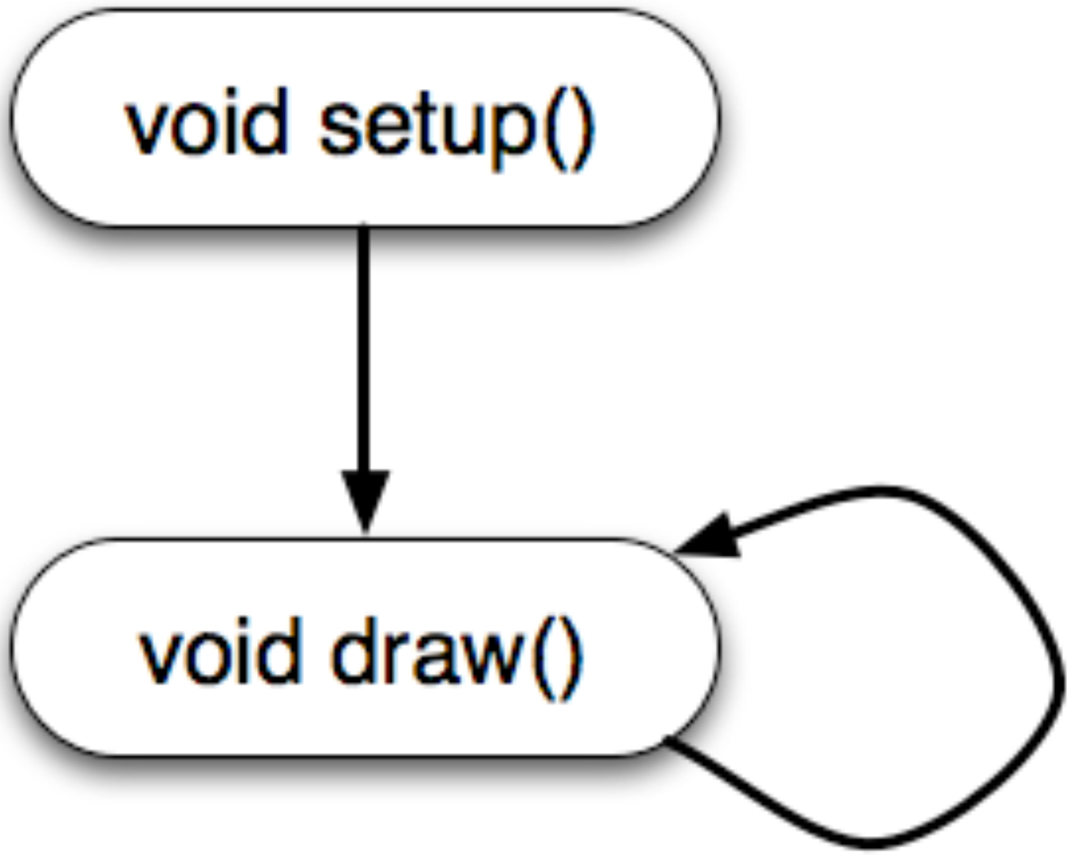
Intro to Processing

- Program flow



Intro to Processing

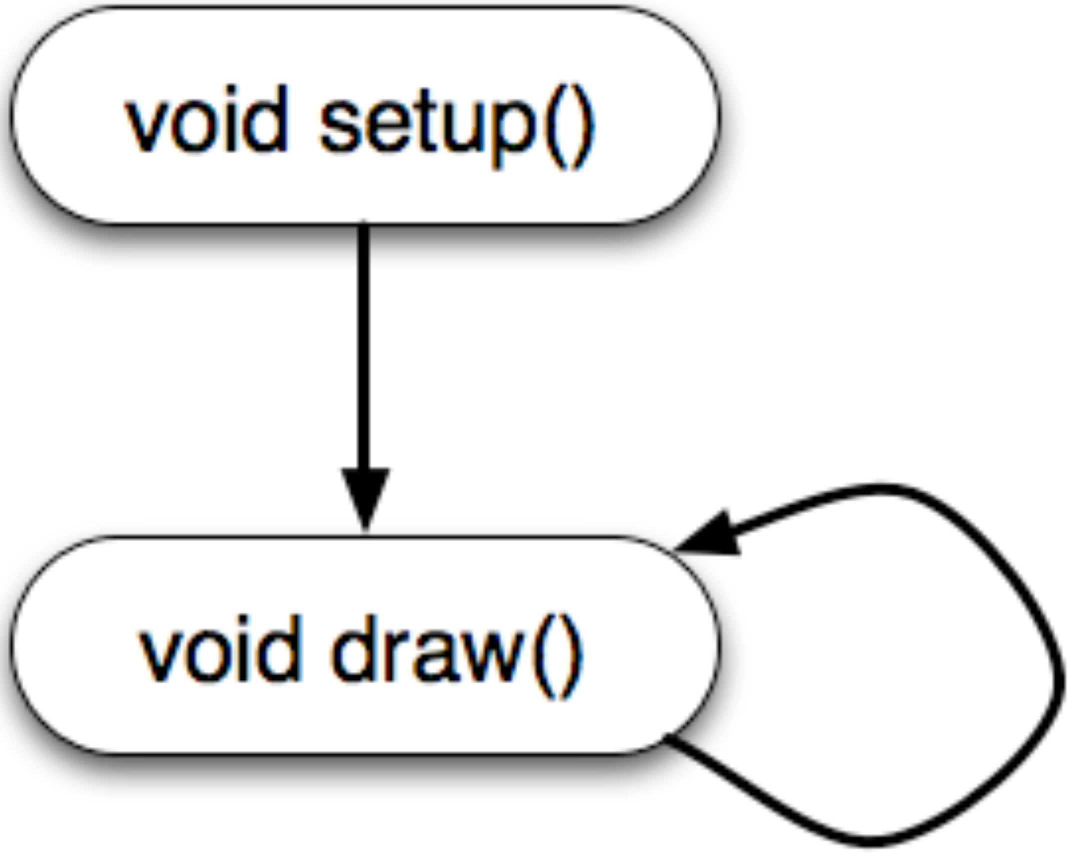
- Program flow



- You can write collections of commands that get run in particular ways by Processing

Intro to Processing

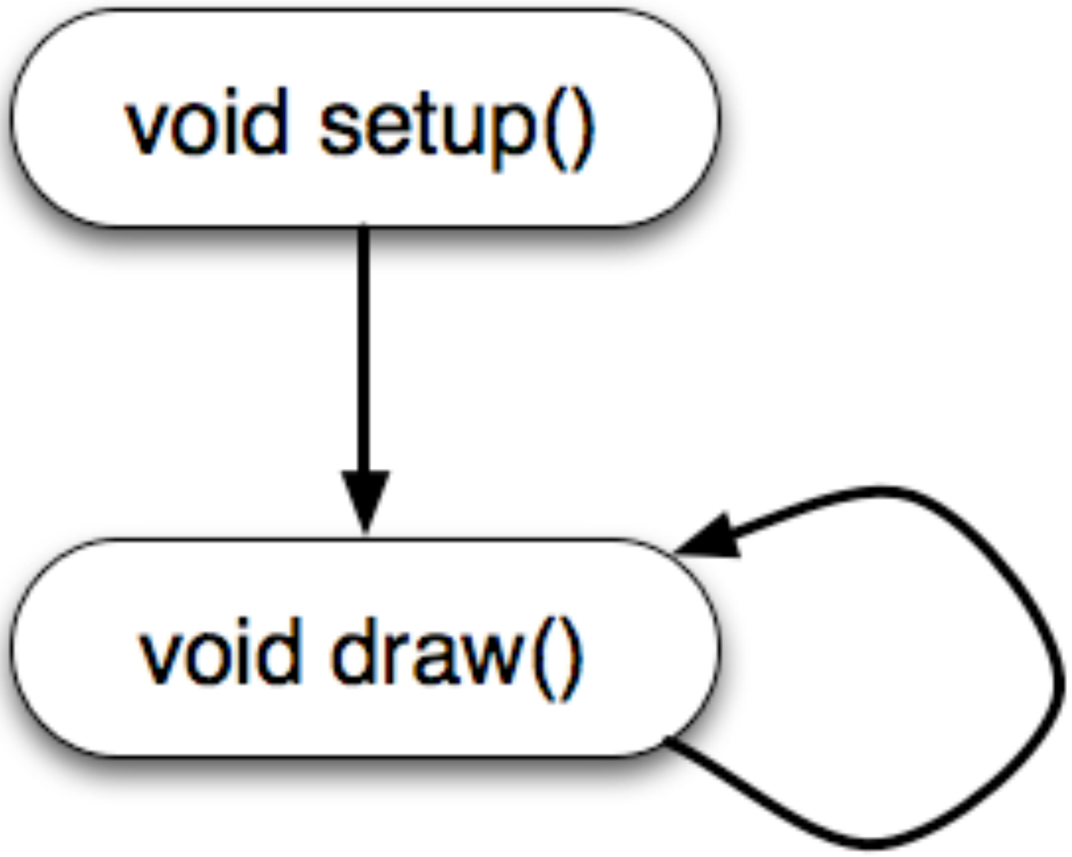
- Program flow



- You can write collections of commands that get run in particular ways by Processing
- the setup function is run once at the beginning

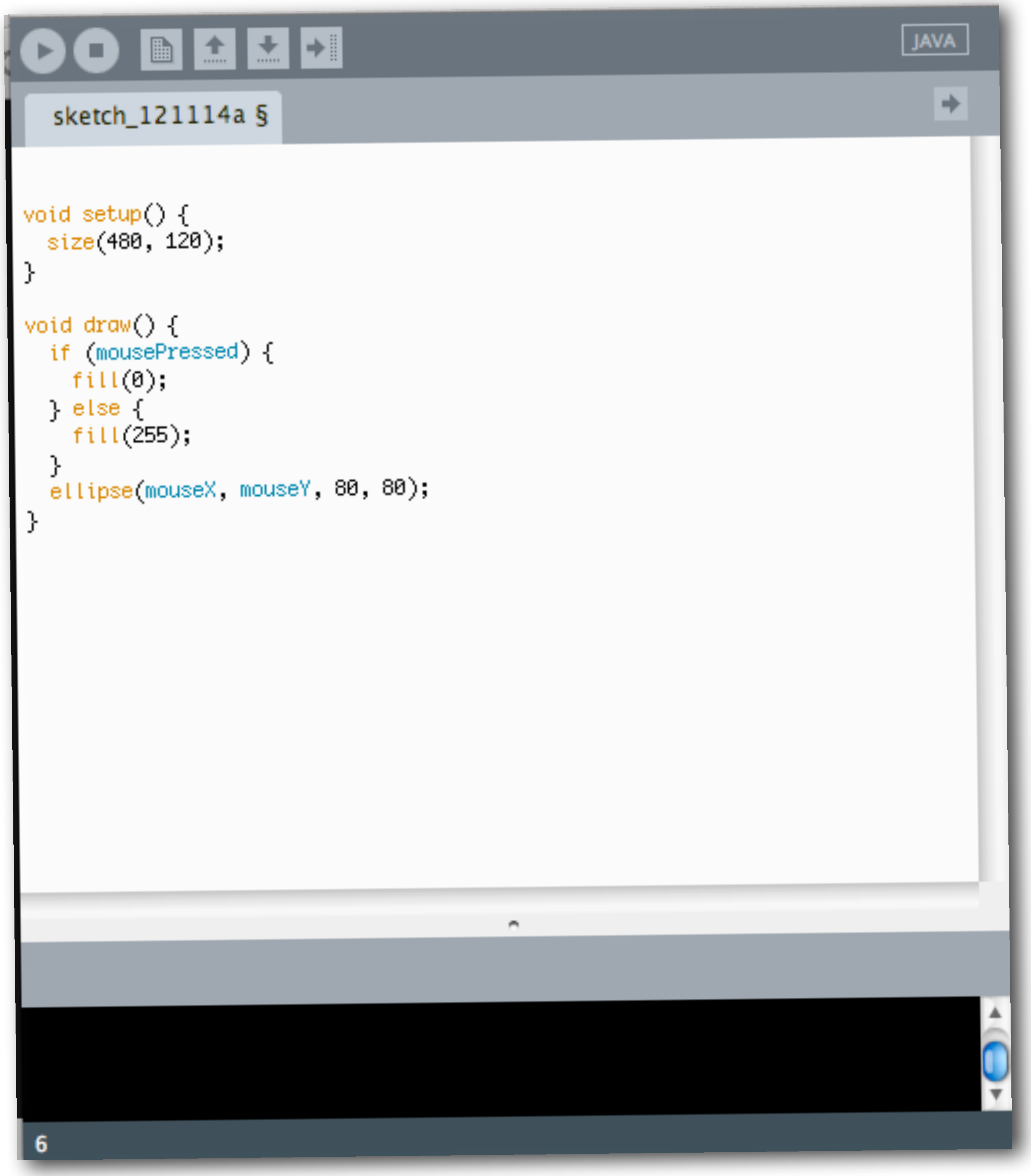
Intro to Processing

- Program flow



- You can write collections of commands that get run in particular ways by Processing
- the setup function is run once at the beginning
- the draw function is run repeatedly until the user hits stop

Intro to Processing

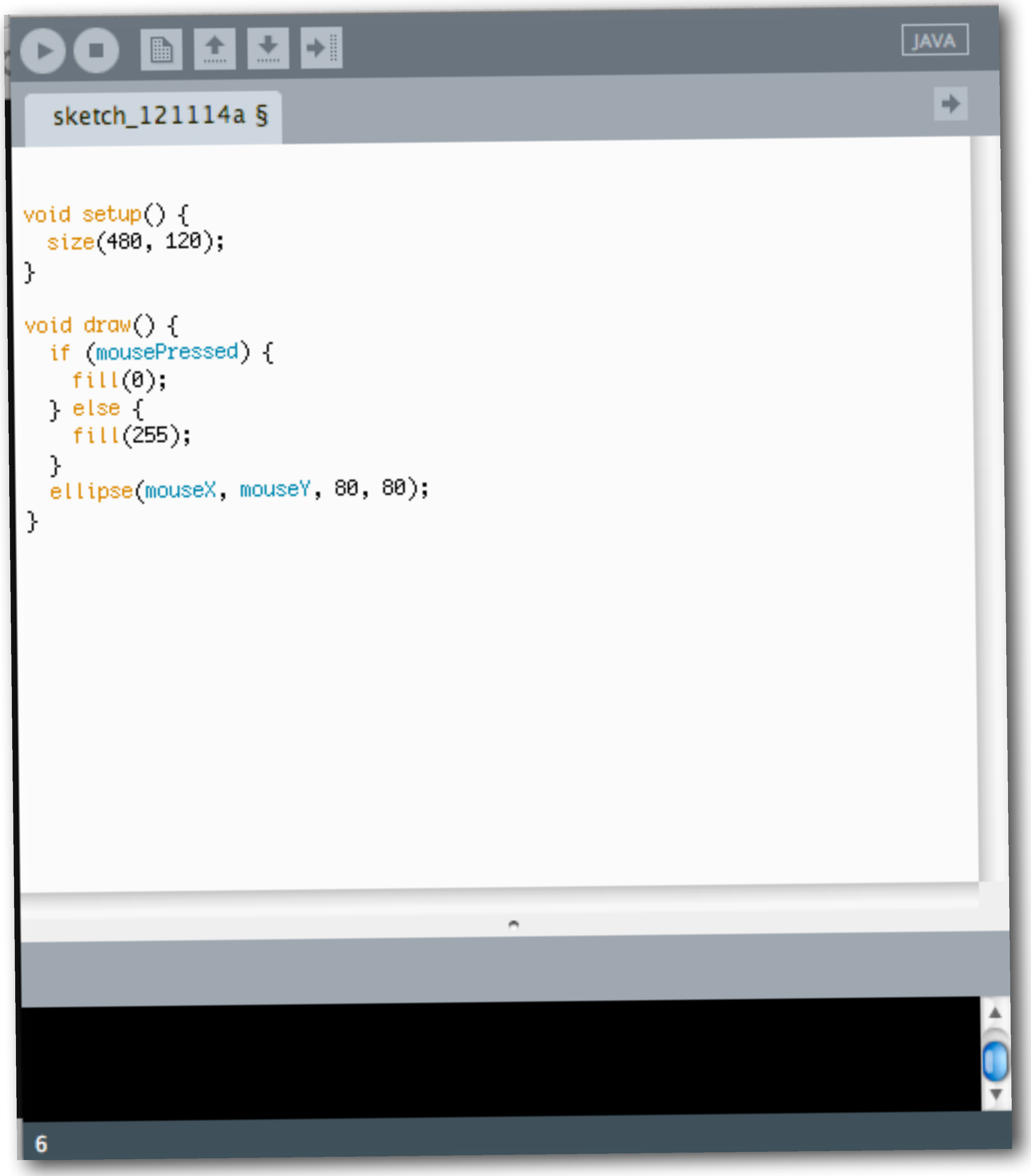


```
void setup() {
  size(480, 120);
}

void draw() {
  if (mousePressed) {
    fill(0);
  } else {
    fill(255);
  }
  ellipse(mouseX, mouseY, 80, 80);
}
```

Intro to Processing

- Program Flow

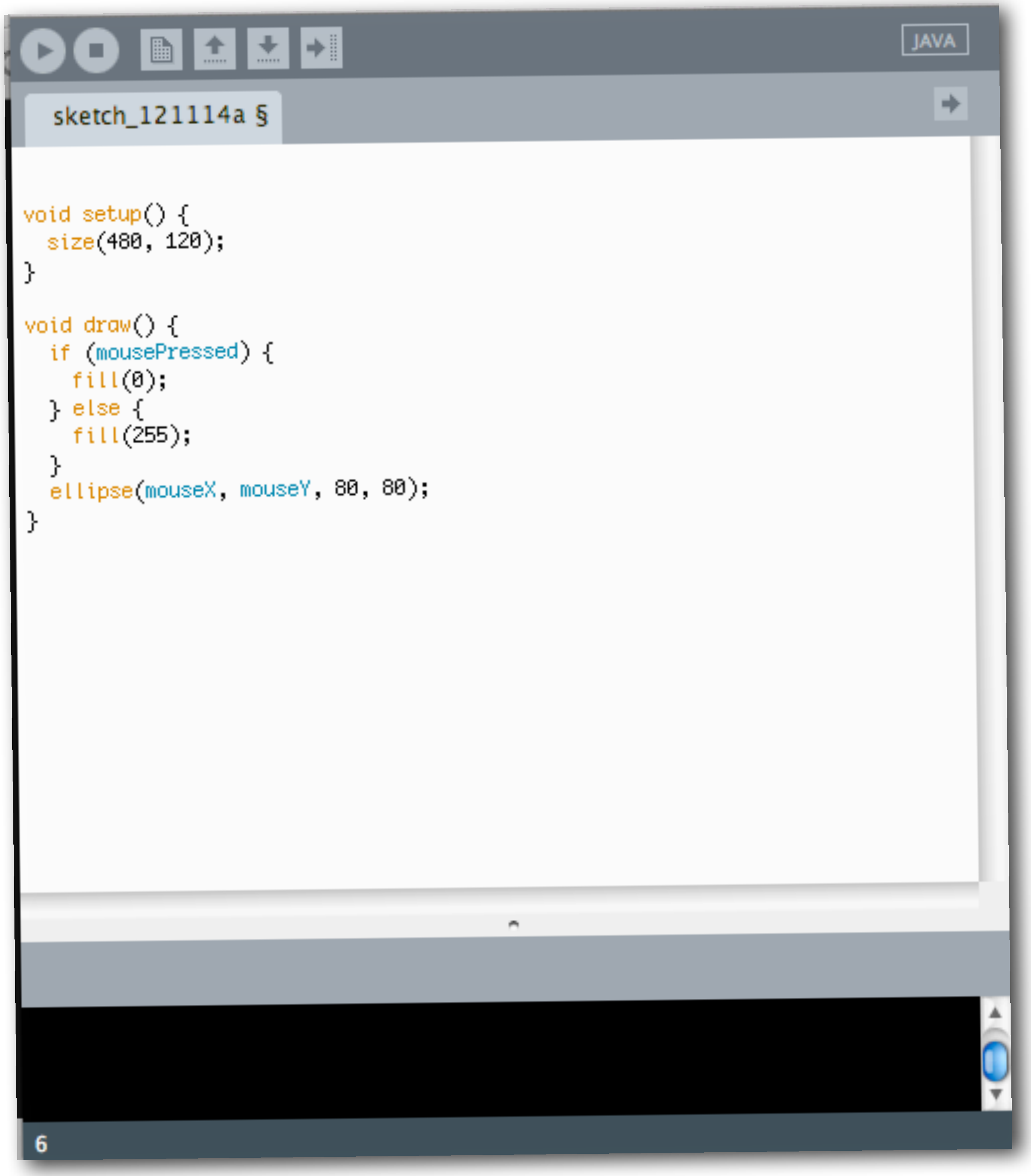


```
void setup() {
  size(480, 120);
}

void draw() {
  if (mousePressed) {
    fill(0);
  } else {
    fill(255);
  }
  ellipse(mouseX, mouseY, 80, 80);
}
```

Intro to Processing

- Program Flow



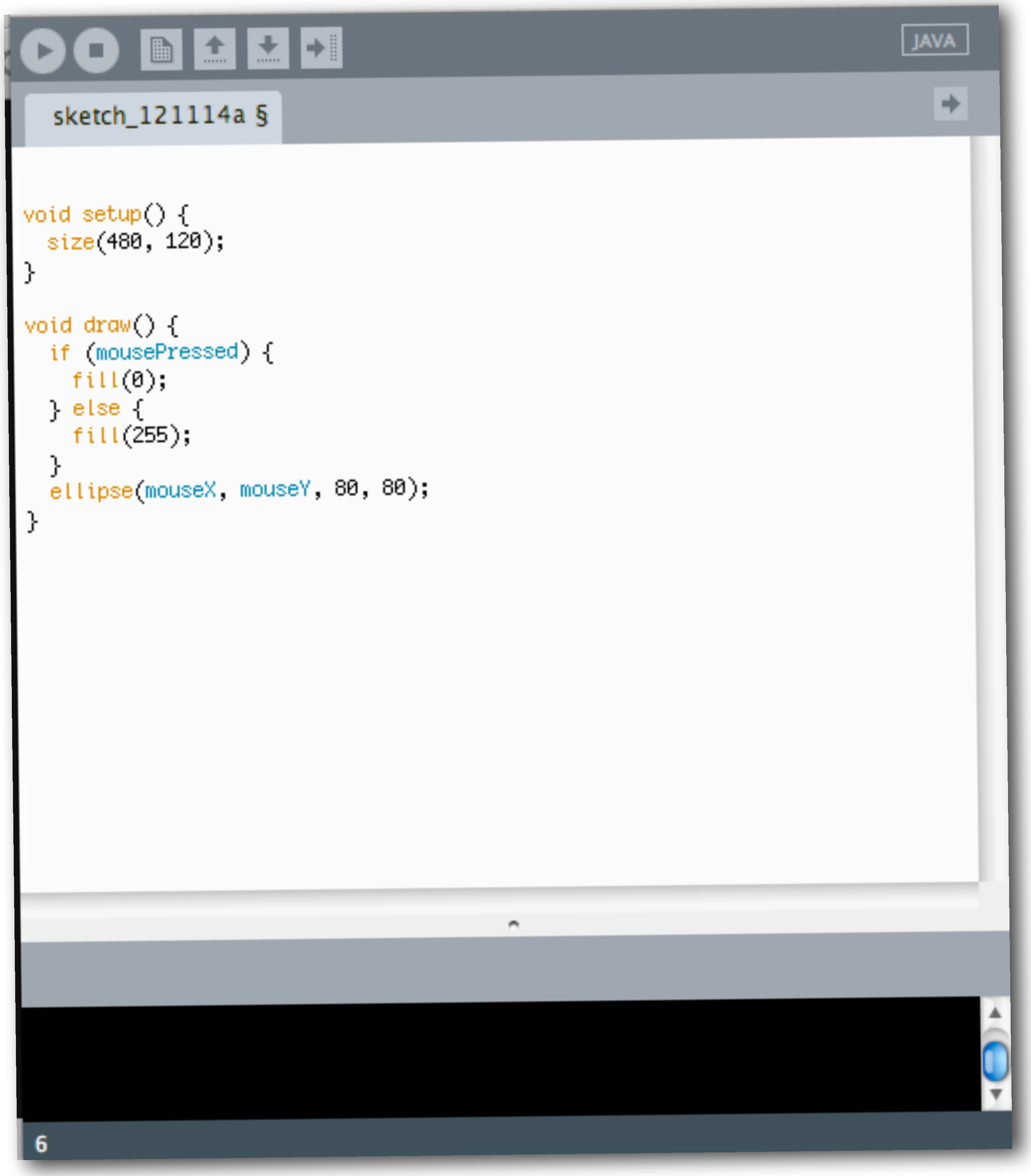
```
void setup() {
  size(480, 120);
}

void draw() {
  if (mousePressed) {
    fill(0);
  } else {
    fill(255);
  }
  ellipse(mouseX, mouseY, 80, 80);
}
```

- functions use curly braces to hold all the commands

Intro to Processing

- Program Flow



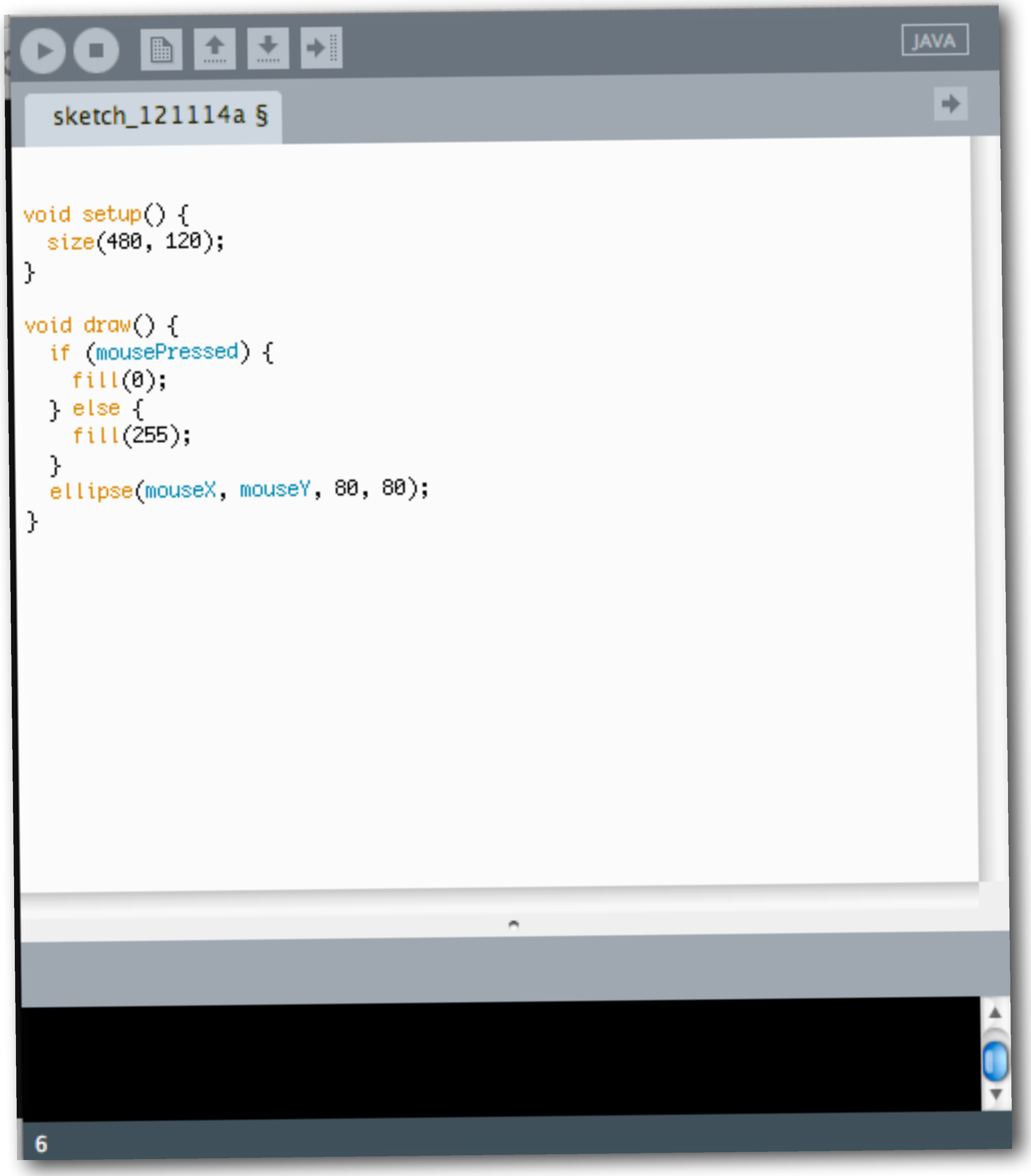
```
void setup() {
  size(480, 120);
}

void draw() {
  if (mousePressed) {
    fill(0);
  } else {
    fill(255);
  }
  ellipse(mouseX, mouseY, 80, 80);
}
```

- functions use curly braces to hold all the commands
- size() changes the display window size

Intro to Processing

- Program Flow



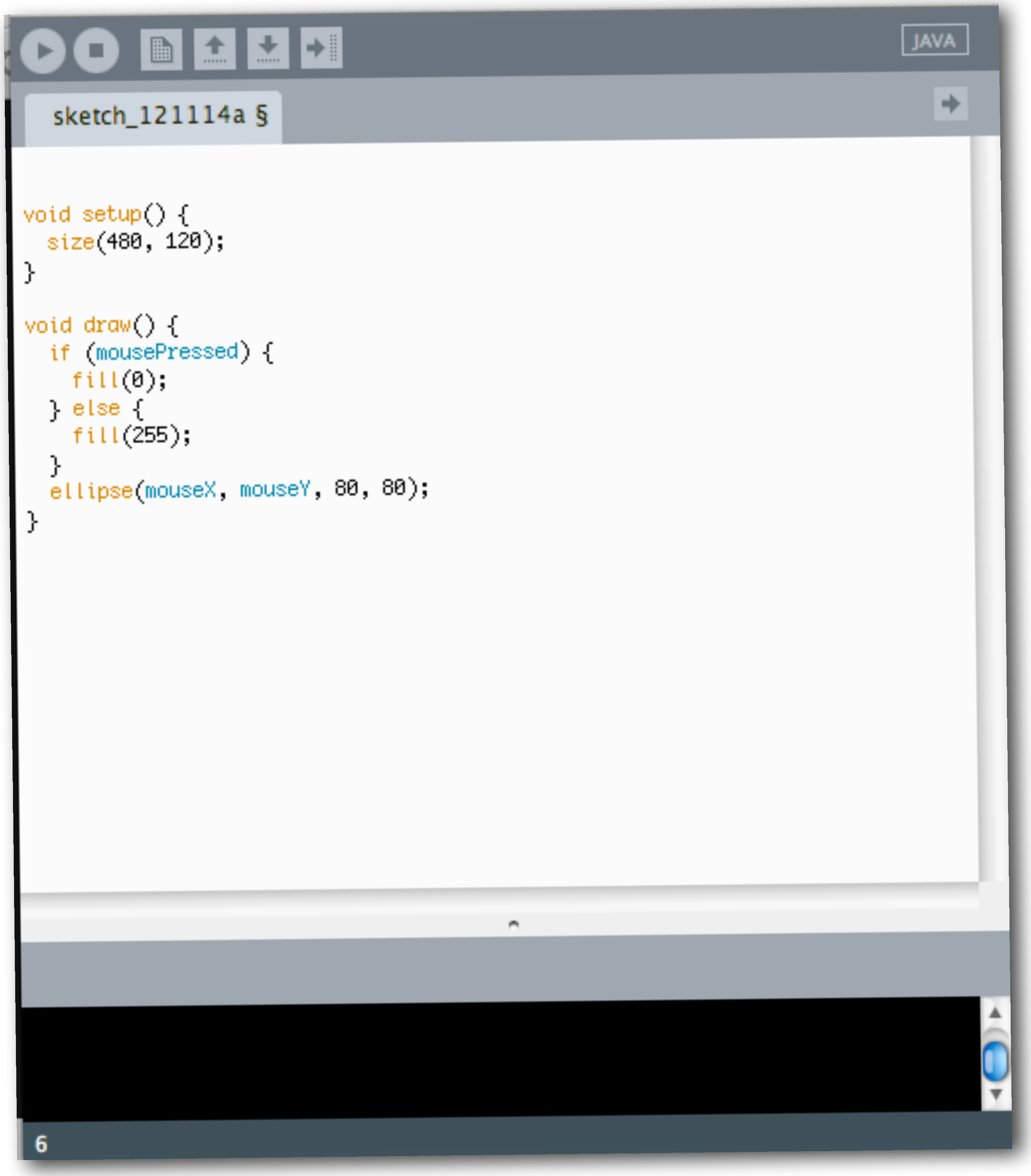
```
void setup() {
  size(480, 120);
}

void draw() {
  if (mousePressed) {
    fill(0);
  } else {
    fill(255);
  }
  ellipse(mouseX, mouseY, 80, 80);
}
```

- functions use curly braces to hold all the commands
- size() changes the display window size
- mousePressed is true if the user is pressing the mouse button

Intro to Processing

- Program Flow



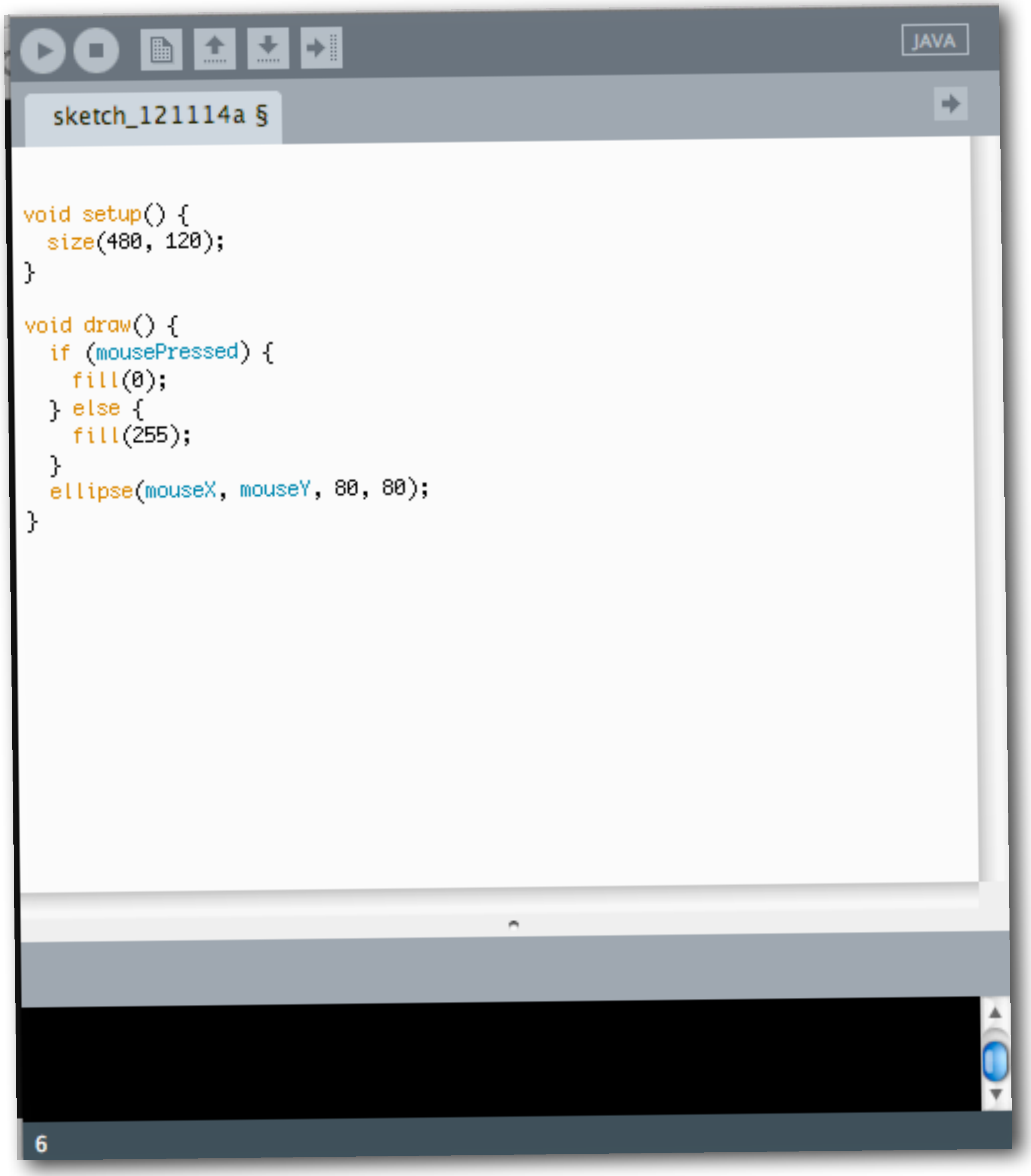
```
void setup() {
  size(480, 120);
}

void draw() {
  if (mousePressed) {
    fill(0);
  } else {
    fill(255);
  }
  ellipse(mouseX, mouseY, 80, 80);
}
```

- functions use curly braces to hold all the commands
- size() changes the display window size
- mousePressed is true if the user is pressing the mouse button
- mouseX and mouseY is the position of the mouse at the current time

Intro to Processing

- Program Flow

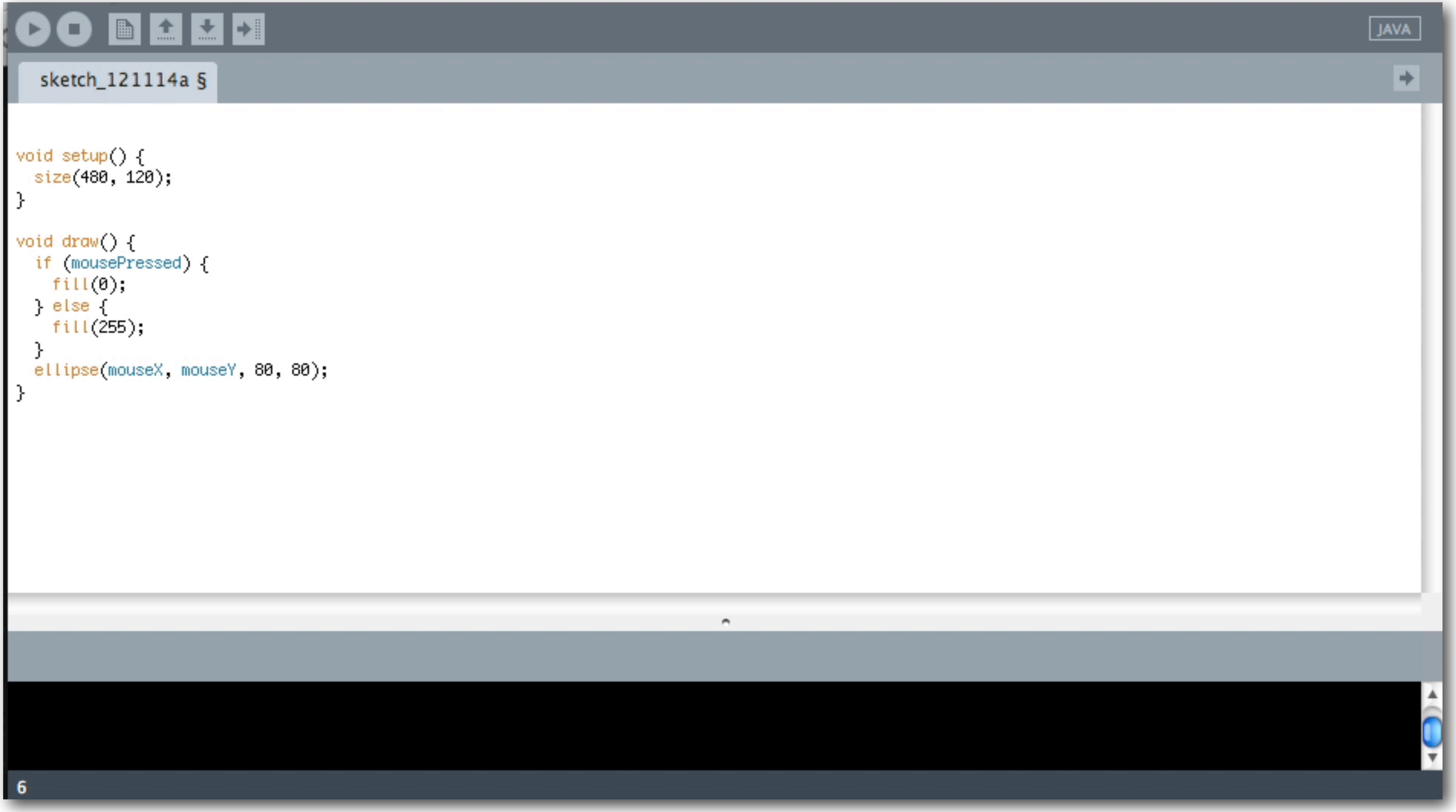


```
void setup() {
  size(480, 120);
}

void draw() {
  if (mousePressed) {
    fill(0);
  } else {
    fill(255);
  }
  ellipse(mouseX, mouseY, 80, 80);
}
```

- functions use curly braces to hold all the commands
- size() changes the display window size
- mousePressed is true if the user is pressing the mouse button
- mouseX and mouseY is the position of the mouse at the current time
- fill() changes the color inside the shape that gets drawn next

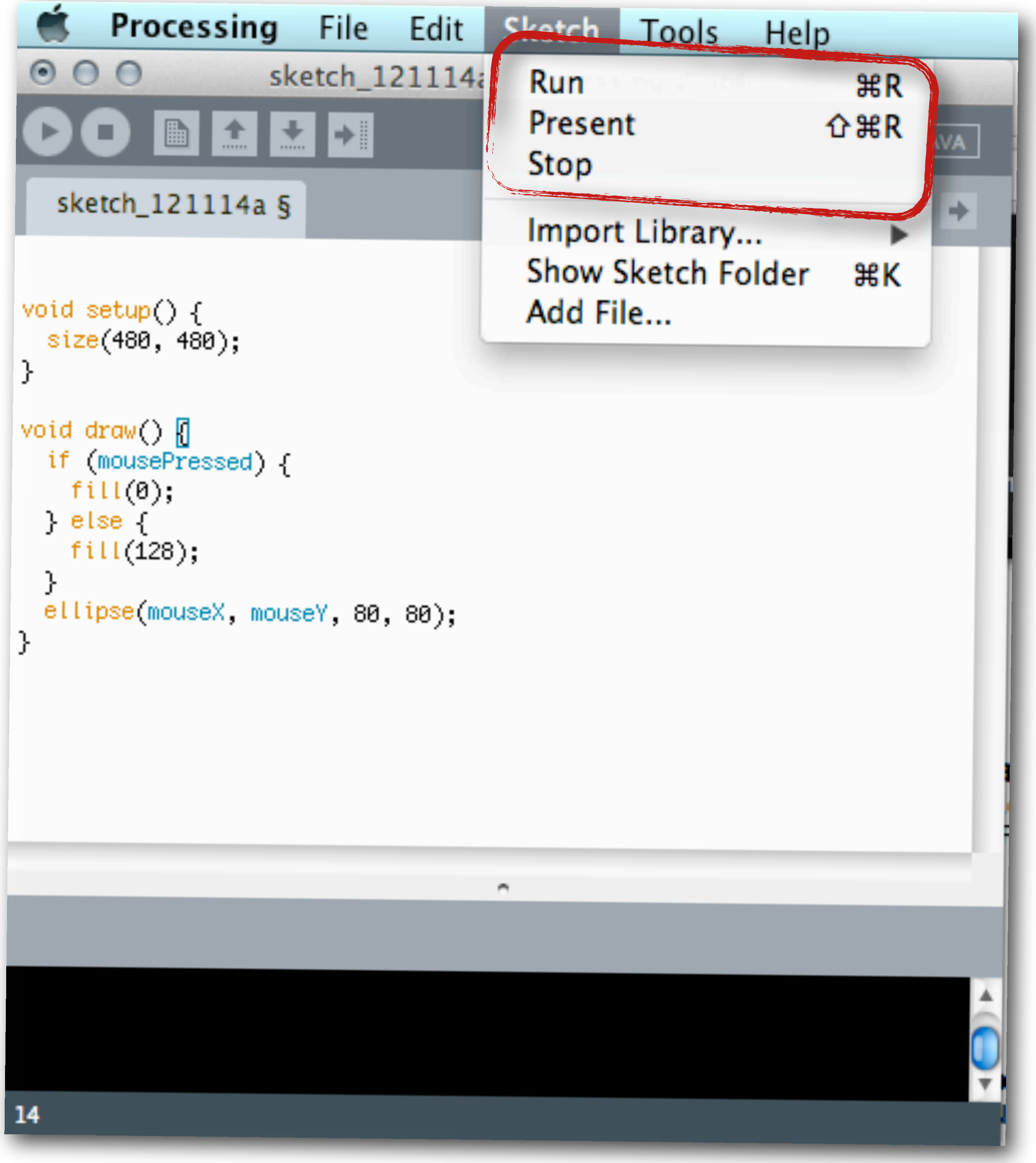
Intro to Processing



```
void setup() {
  size(480, 120);
}

void draw() {
  if (mousePressed) {
    fill(0);
  } else {
    fill(255);
  }
  ellipse(mouseX, mouseY, 80, 80);
}
```

Intro to Processing

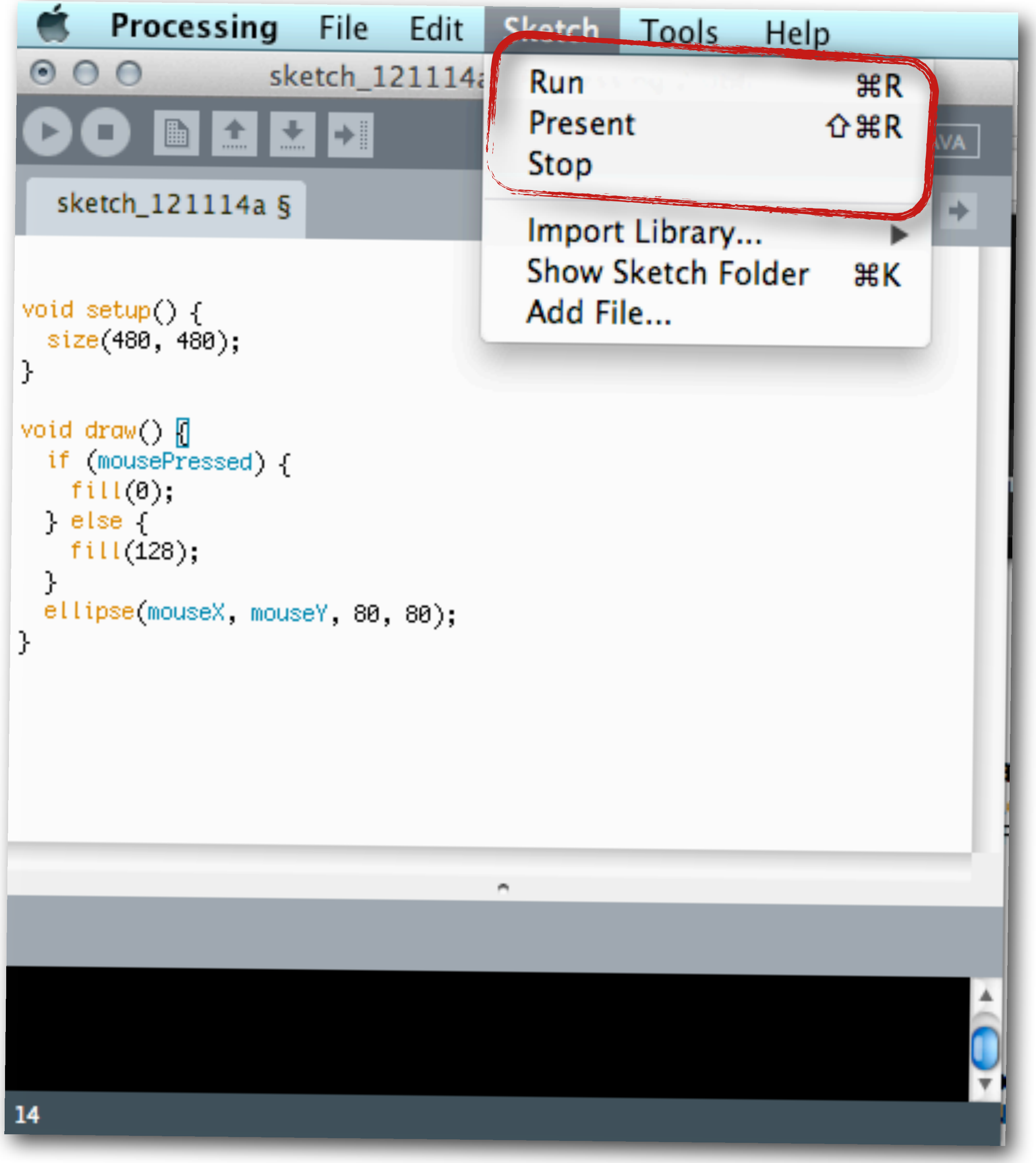


```
void setup() {
  size(480, 480);
}

void draw() {
  if (mousePressed) {
    fill(0);
  } else {
    fill(128);
  }
  ellipse(mouseX, mouseY, 80, 80);
}
```

Intro to Processing

- Options

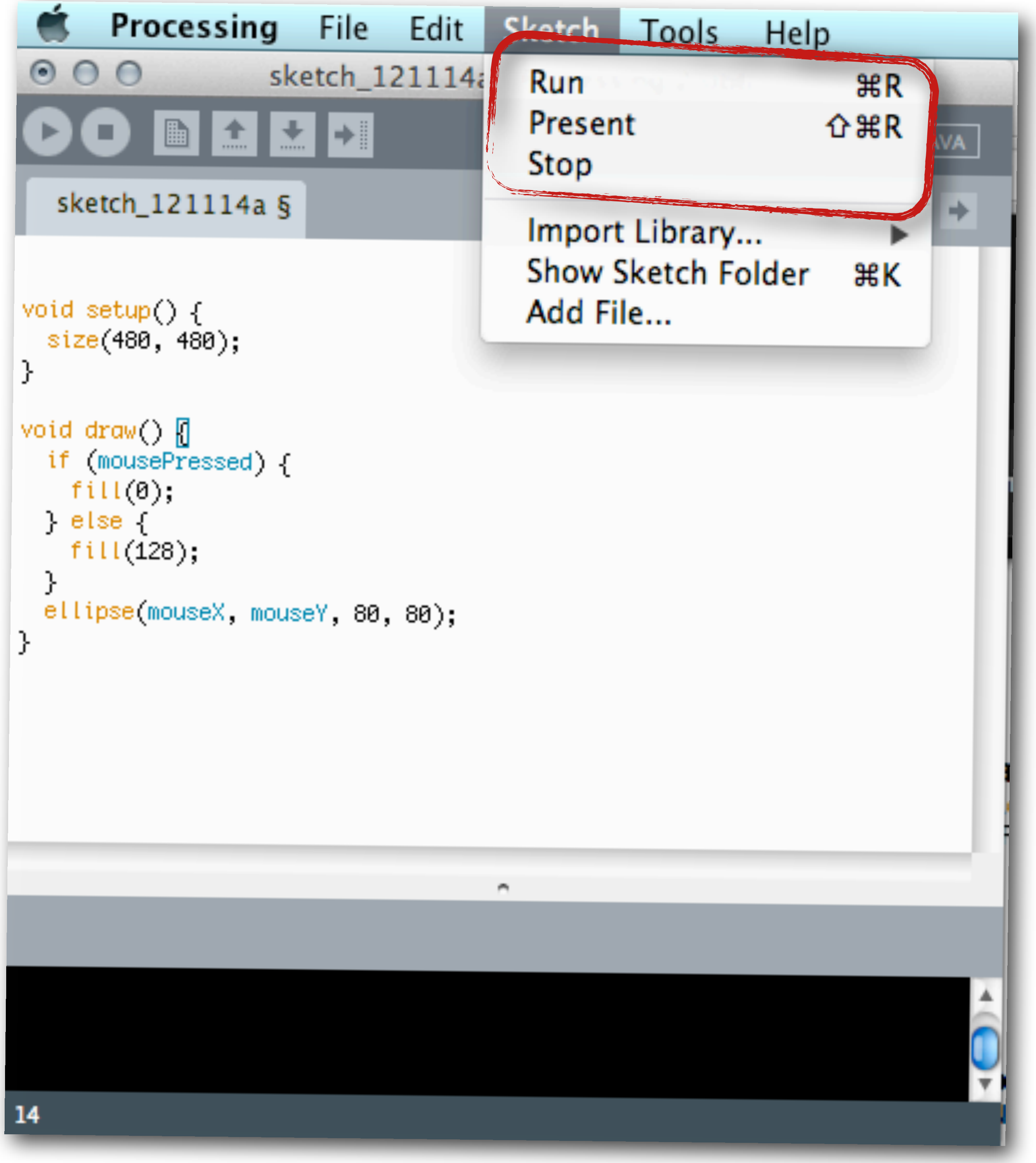


```
void setup() {
  size(480, 480);
}

void draw() {
  if (mousePressed) {
    fill(0);
  } else {
    fill(128);
  }
  ellipse(mouseX, mouseY, 80, 80);
}
```

Intro to Processing

- Options

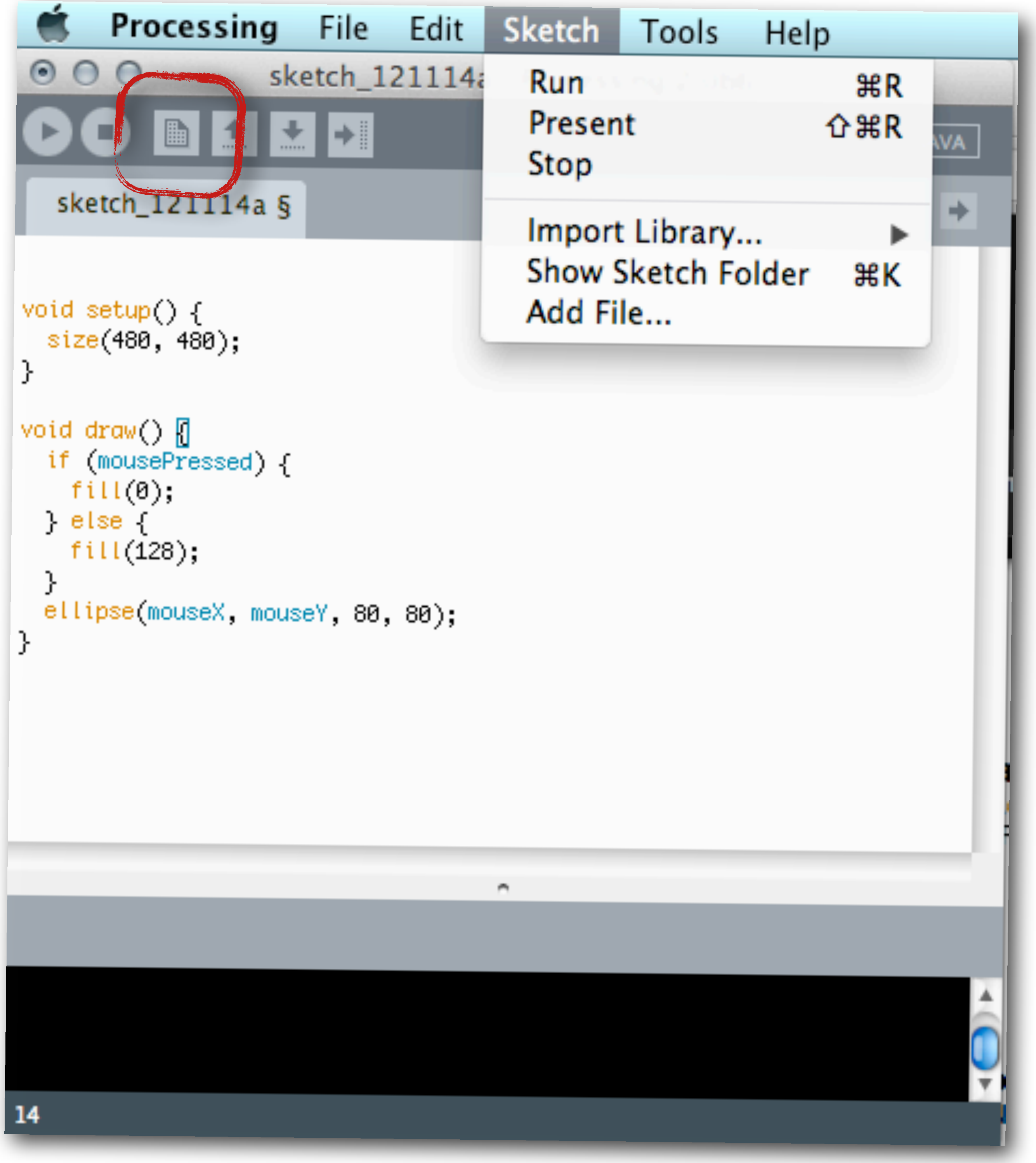


```
void setup() {
  size(480, 480);
}

void draw() {
  if (mousePressed) {
    fill(0);
  } else {
    fill(128);
  }
  ellipse(mouseX, mouseY, 80, 80);
}
```

- Menu bar shows shortcuts instead of requiring the buttons to be used

Intro to Processing

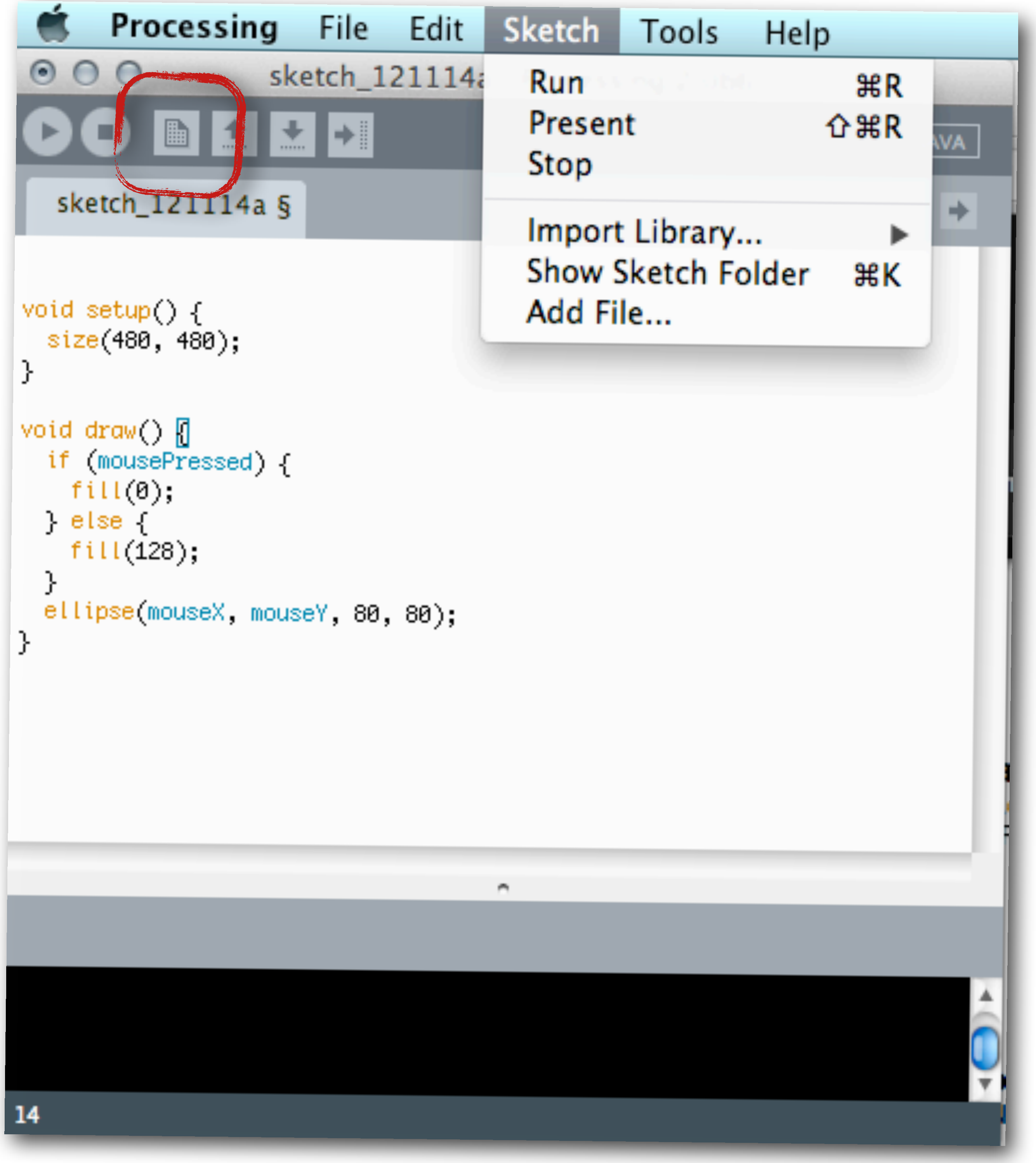


```
void setup() {
  size(480, 480);
}

void draw() {
  if (mousePressed) {
    fill(0);
  } else {
    fill(128);
  }
  ellipse(mouseX, mouseY, 80, 80);
}
```

Intro to Processing

- Options

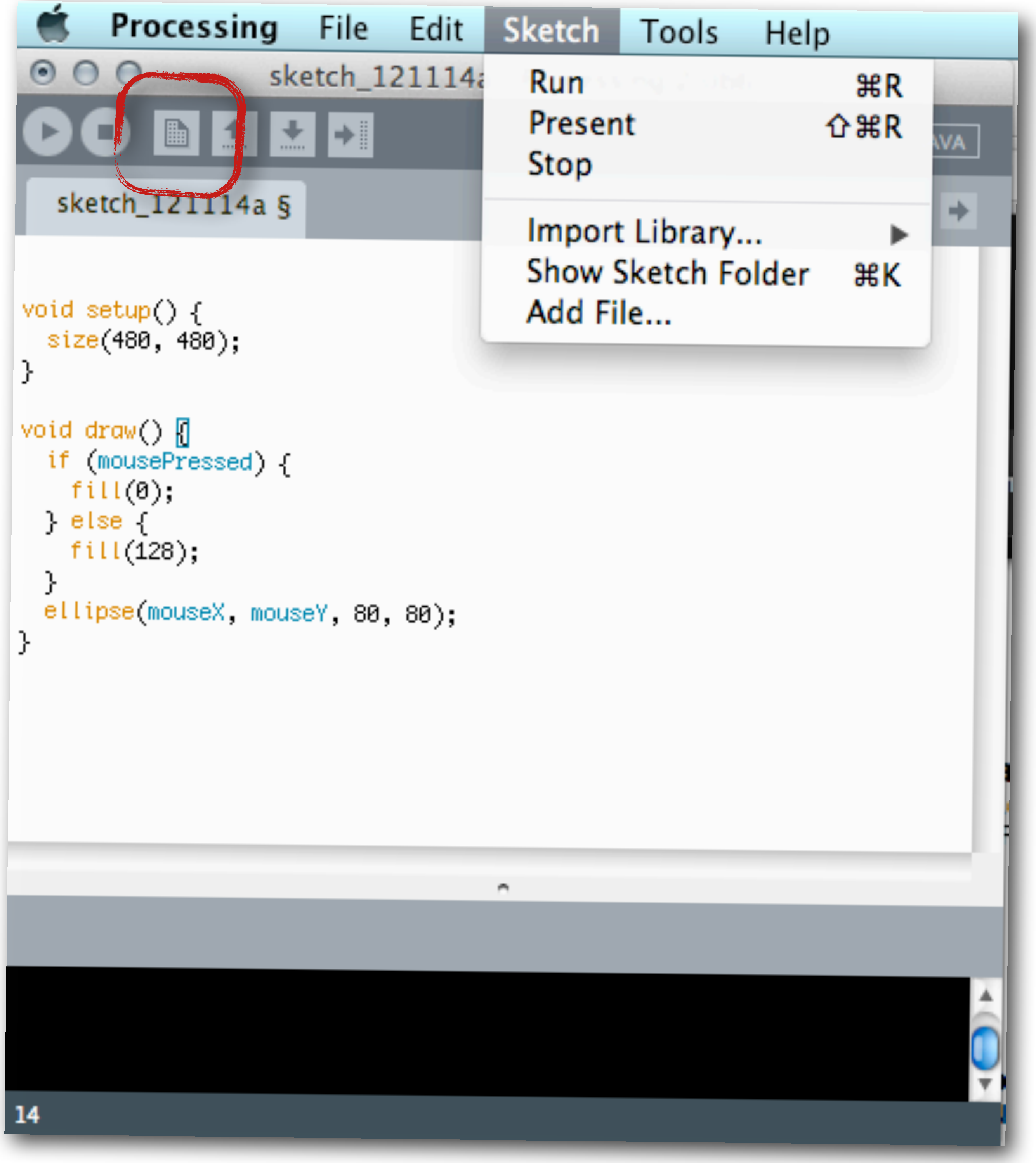


```
void setup() {
  size(480, 480);
}

void draw() {
  if (mousePressed) {
    fill(0);
  } else {
    fill(128);
  }
  ellipse(mouseX, mouseY, 80, 80);
}
```

Intro to Processing

- Options

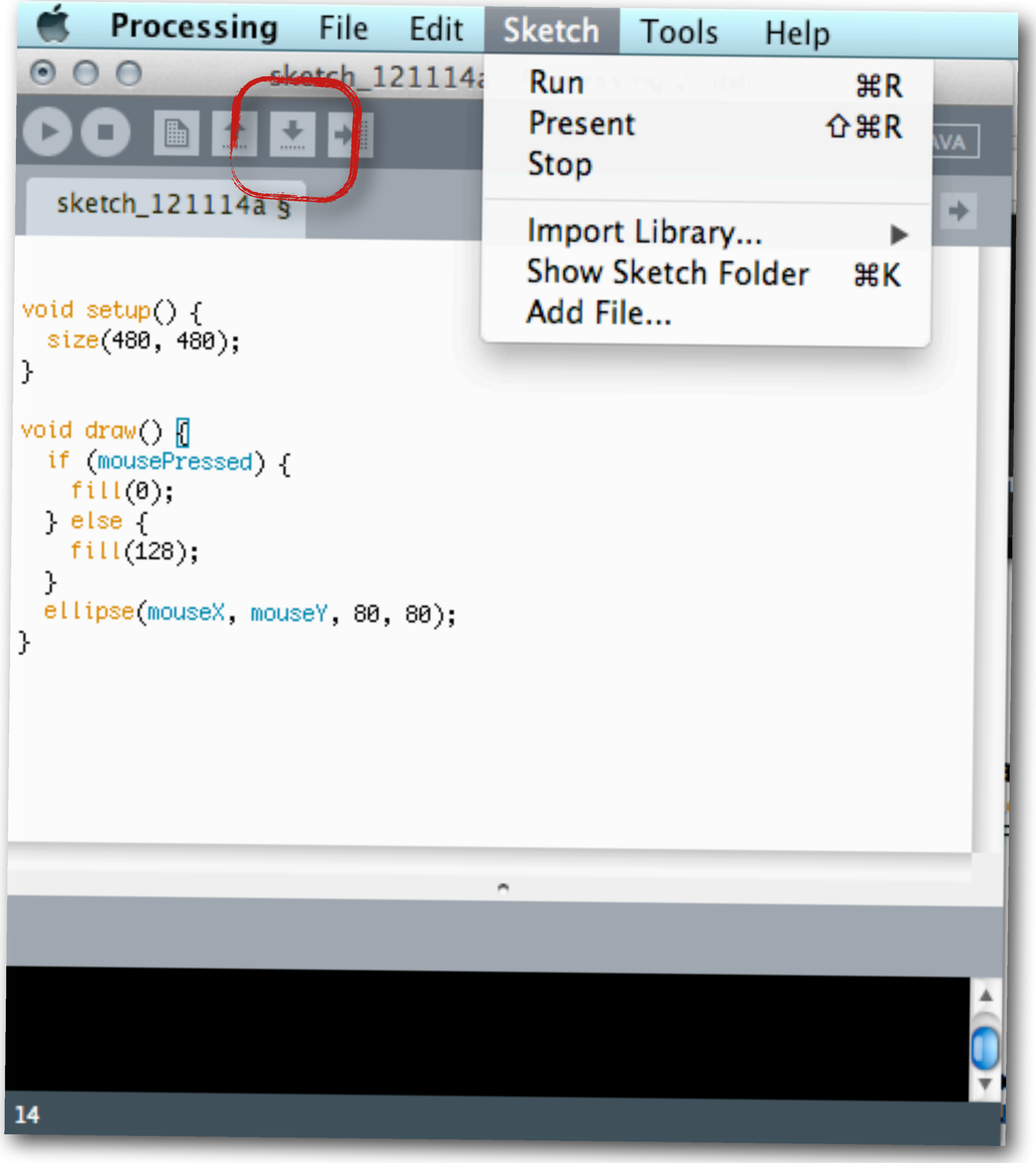


```
void setup() {
  size(480, 480);
}

void draw() {
  if (mousePressed) {
    fill(0);
  } else {
    fill(128);
  }
  ellipse(mouseX, mouseY, 80, 80);
}
```

- Create a new sketch

Intro to Processing

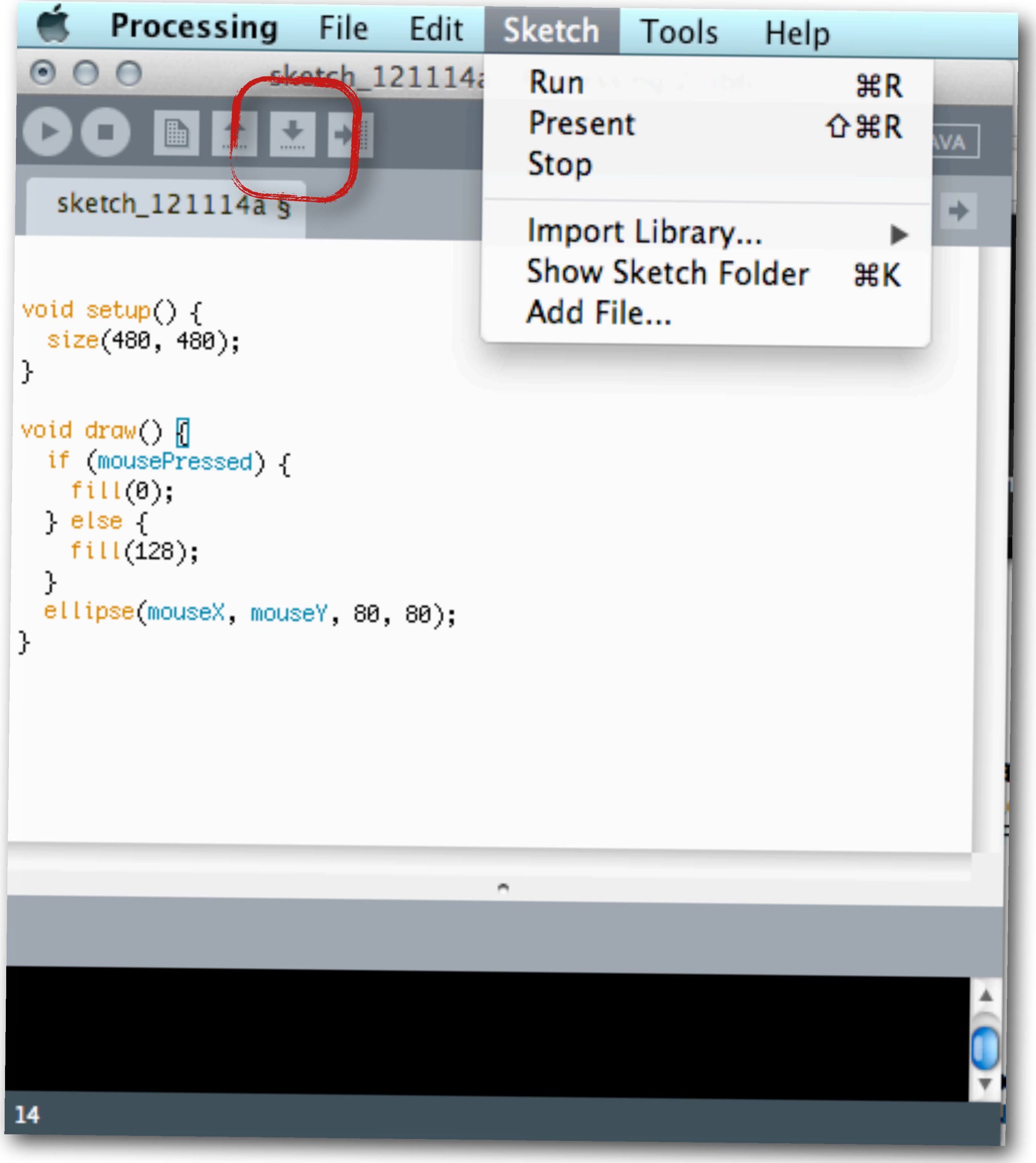


```
void setup() {
  size(480, 480);
}

void draw() {
  if (mousePressed) {
    fill(0);
  } else {
    fill(128);
  }
  ellipse(mouseX, mouseY, 80, 80);
}
```

Intro to Processing

- Options

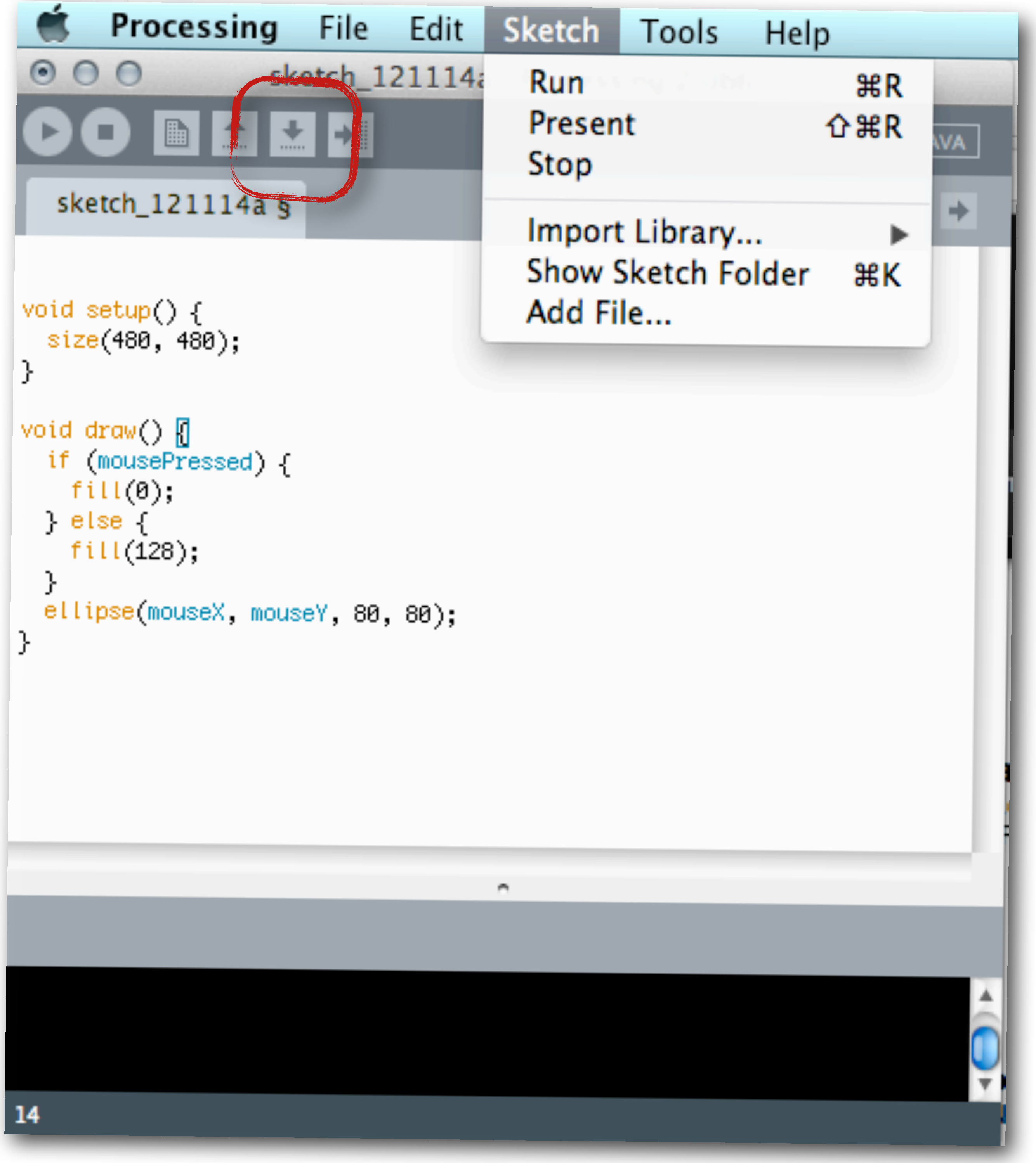


```
void setup() {
  size(480, 480);
}

void draw() {
  if (mousePressed) {
    fill(0);
  } else {
    fill(128);
  }
  ellipse(mouseX, mouseY, 80, 80);
}
```

Intro to Processing

- Options

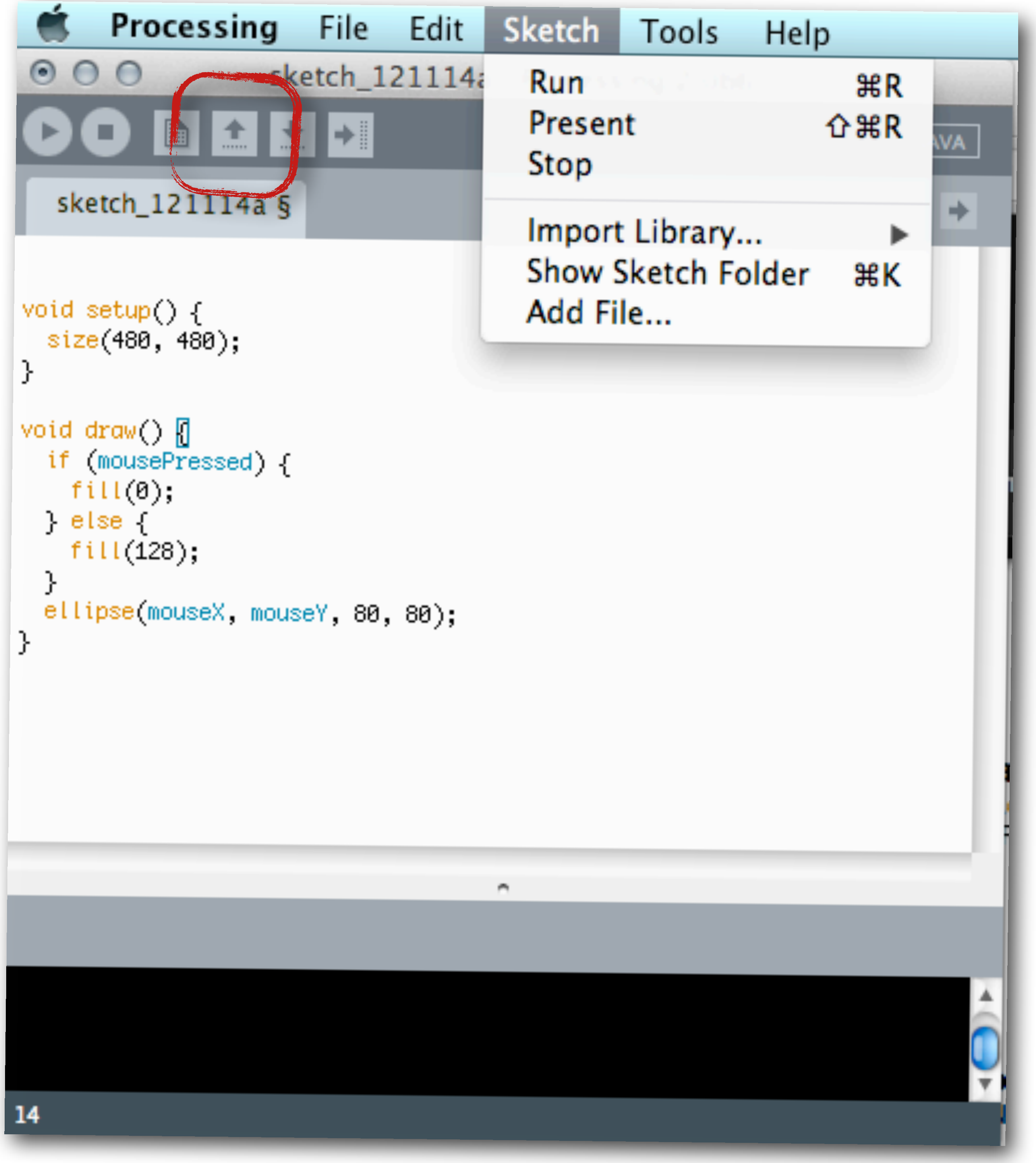


```
void setup() {
  size(480, 480);
}

void draw() {
  if (mousePressed) {
    fill(0);
  } else {
    fill(128);
  }
  ellipse(mouseX, mouseY, 80, 80);
}
```

- Save the current sketch

Intro to Processing

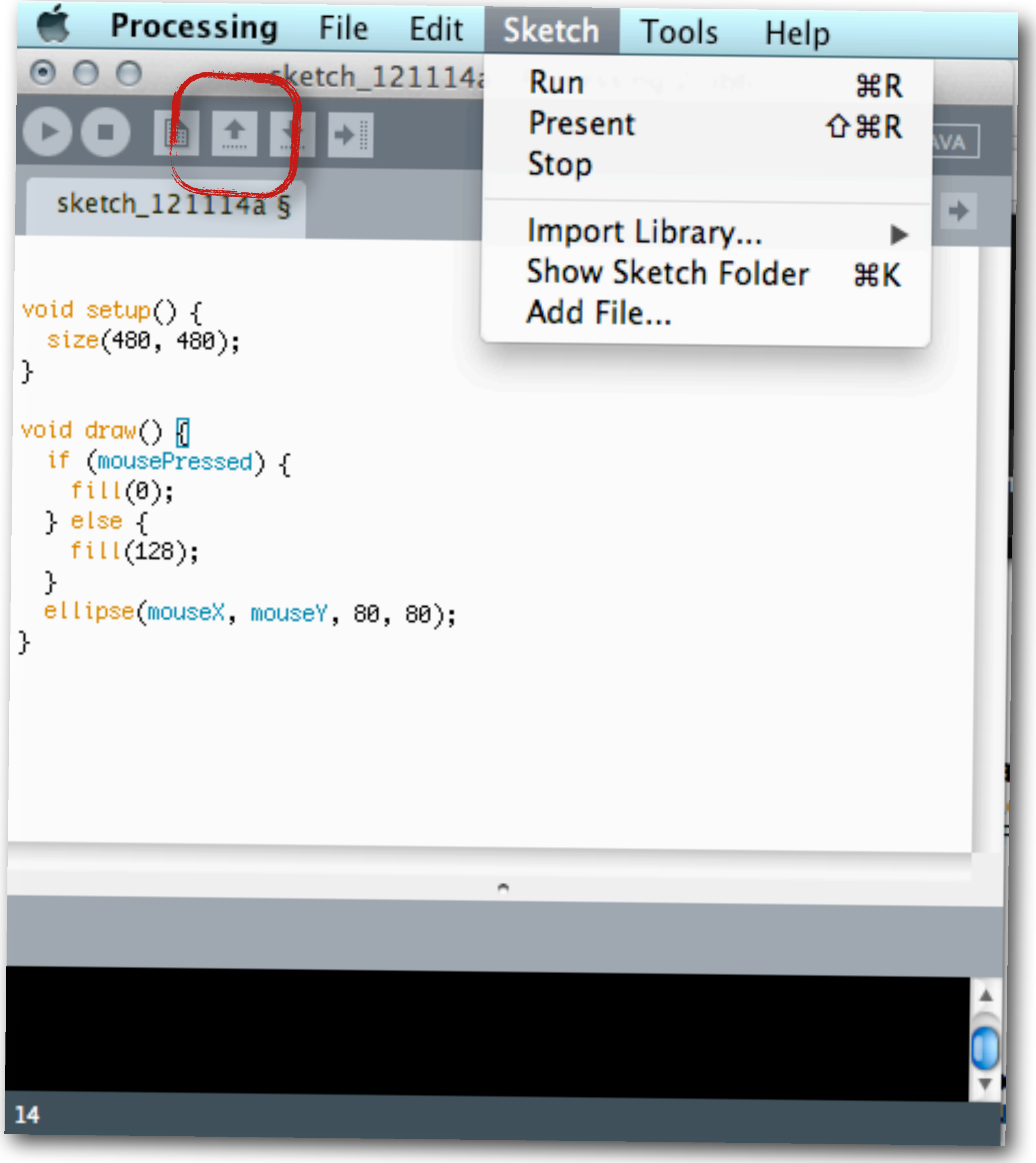


```
void setup() {
  size(480, 480);
}

void draw() {
  if (mousePressed) {
    fill(0);
  } else {
    fill(128);
  }
  ellipse(mouseX, mouseY, 80, 80);
}
```

Intro to Processing

- Options

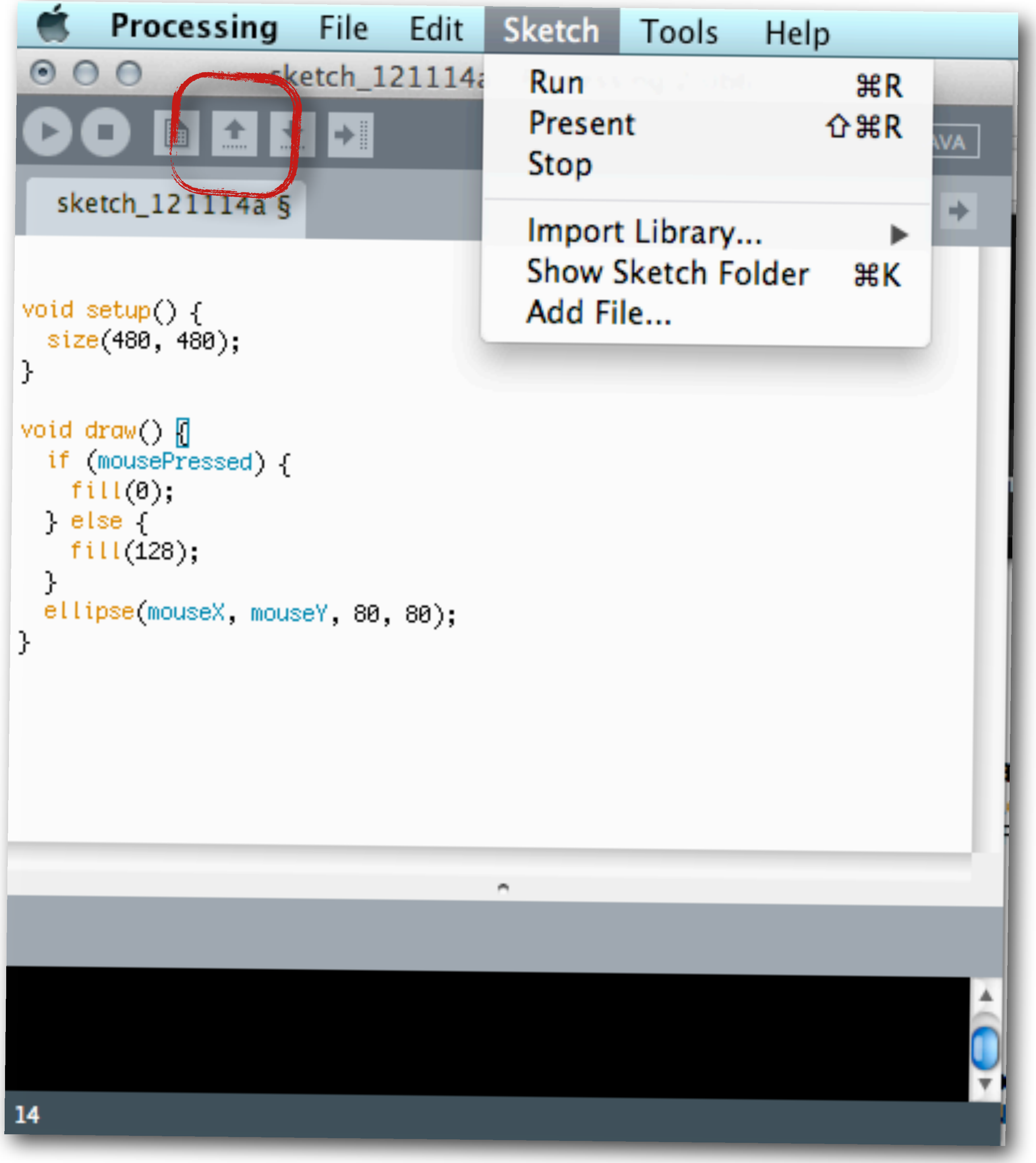


```
void setup() {
  size(480, 480);
}

void draw() {
  if (mousePressed) {
    fill(0);
  } else {
    fill(128);
  }
  ellipse(mouseX, mouseY, 80, 80);
}
```

Intro to Processing

- Options

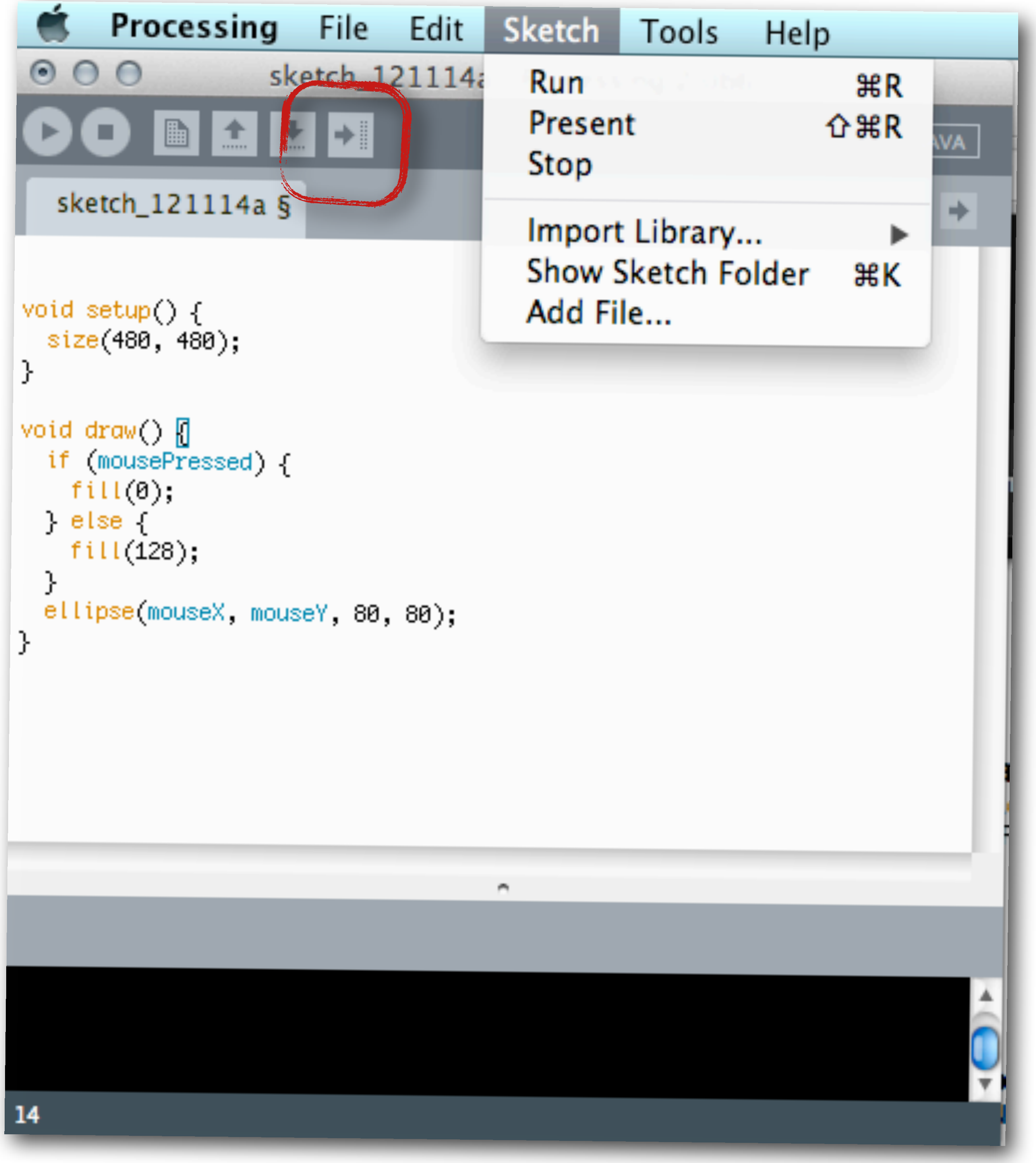


```
void setup() {
  size(480, 480);
}

void draw() {
  if (mousePressed) {
    fill(0);
  } else {
    fill(128);
  }
  ellipse(mouseX, mouseY, 80, 80);
}
```

- Open a previously saved sketch

Intro to Processing

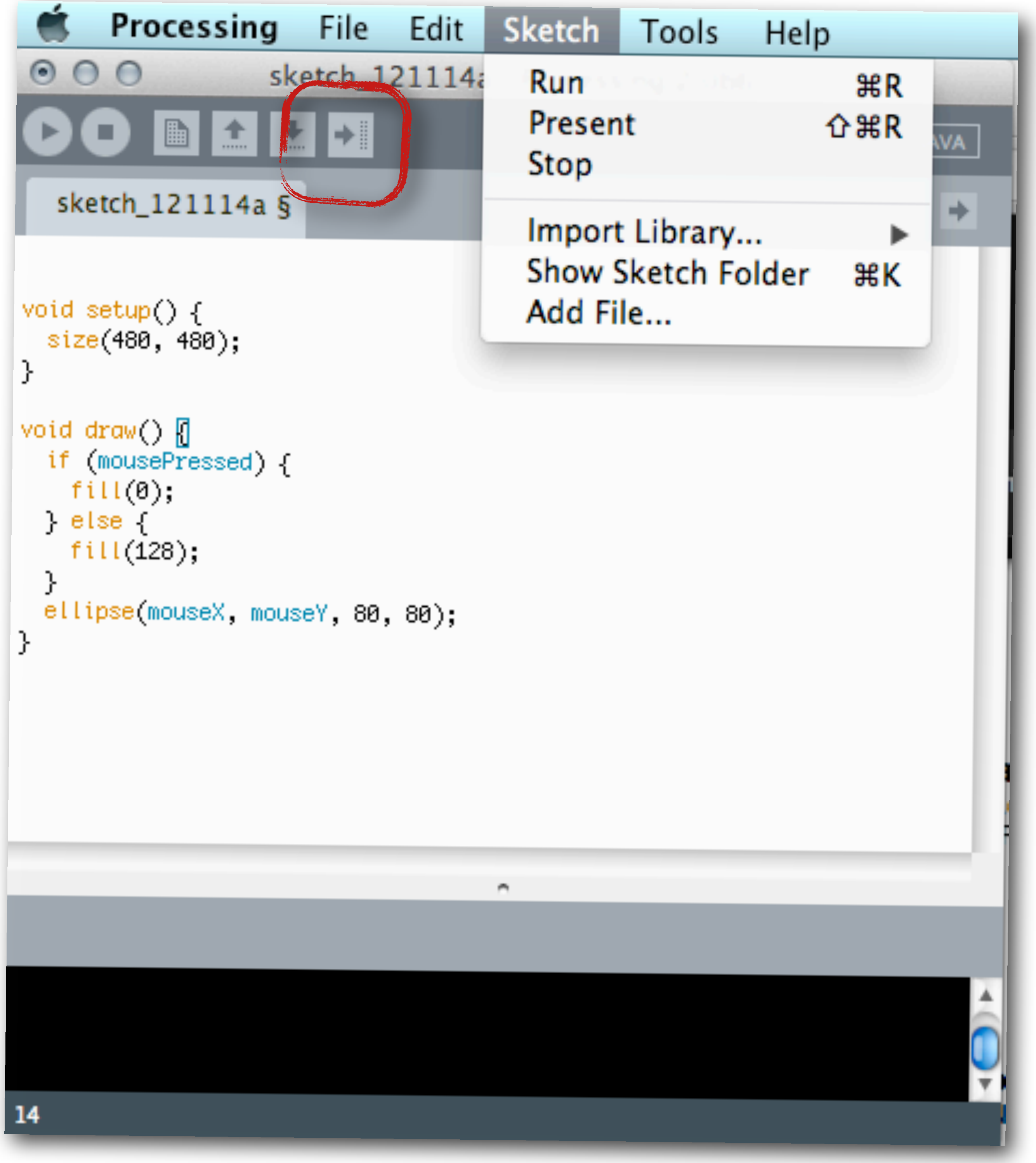


```
void setup() {
  size(480, 480);
}

void draw() {
  if (mousePressed) {
    fill(0);
  } else {
    fill(128);
  }
  ellipse(mouseX, mouseY, 80, 80);
}
```

Intro to Processing

- Options

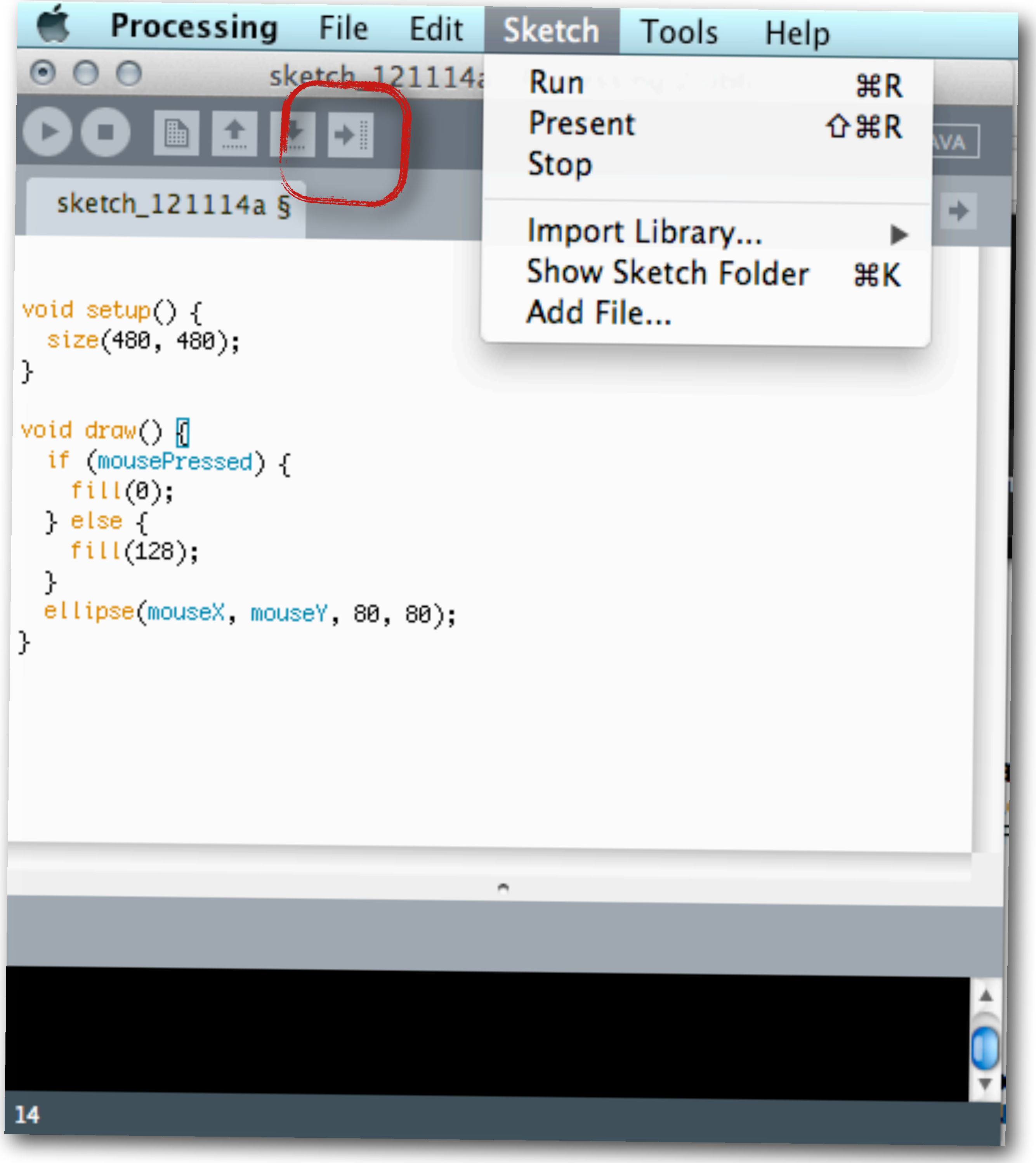


```
void setup() {
  size(480, 480);
}

void draw() {
  if (mousePressed) {
    fill(0);
  } else {
    fill(128);
  }
  ellipse(mouseX, mouseY, 80, 80);
}
```

Intro to Processing

- Options

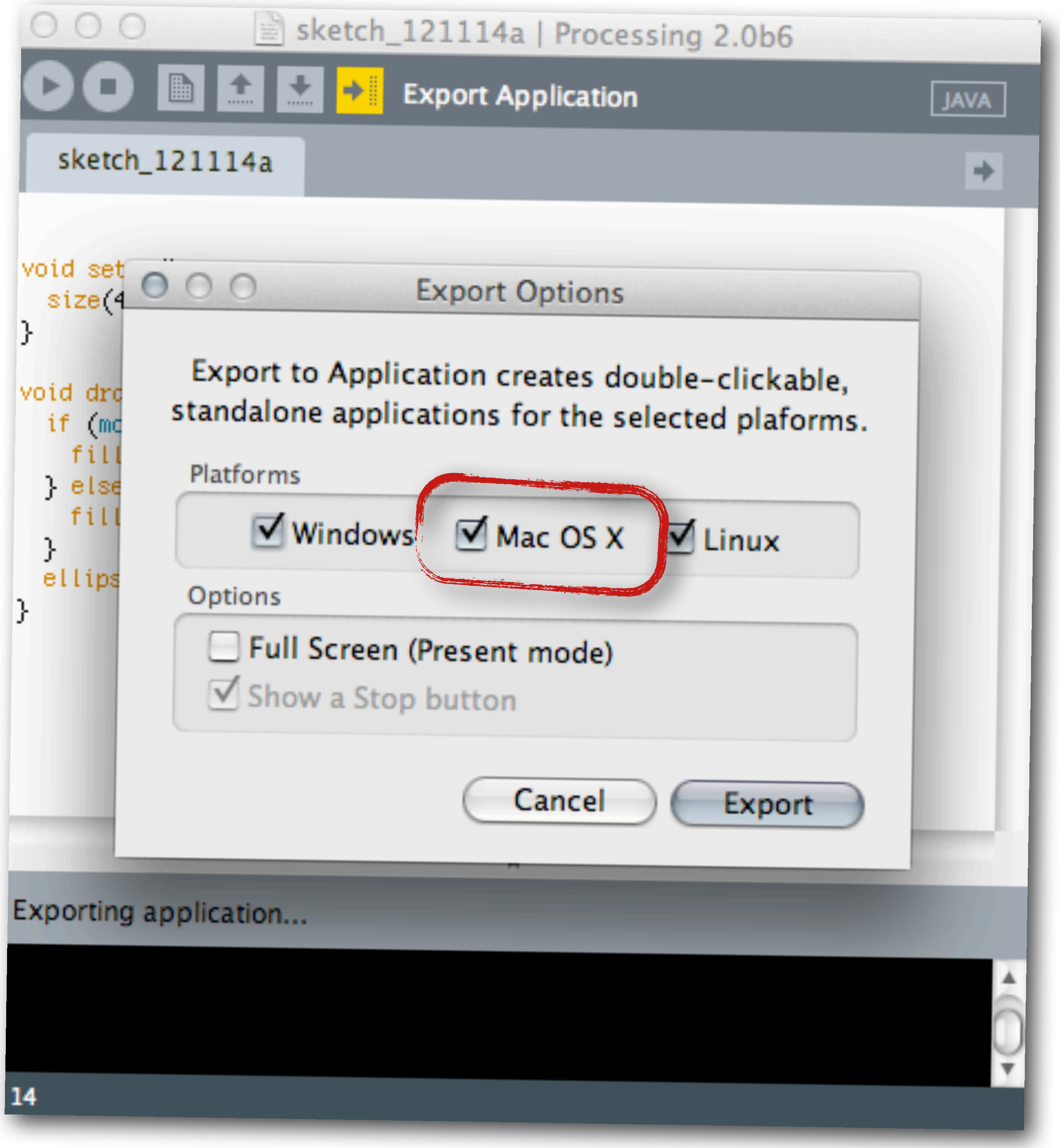


```
void setup() {
  size(480, 480);
}

void draw() {
  if (mousePressed) {
    fill(0);
  } else {
    fill(128);
  }
  ellipse(mouseX, mouseY, 80, 80);
}
```

- Share a sketch as an application

Intro to Processing



```
void set
  size(4
}

void dra
  if (mo
    fill
  } else
    fill
  }
  ellips
}
```

Export Options

Export to Application creates double-clickable, standalone applications for the selected plaforms.

Platforms

☑ Windows ☑ Mac OS X ☑ Linux

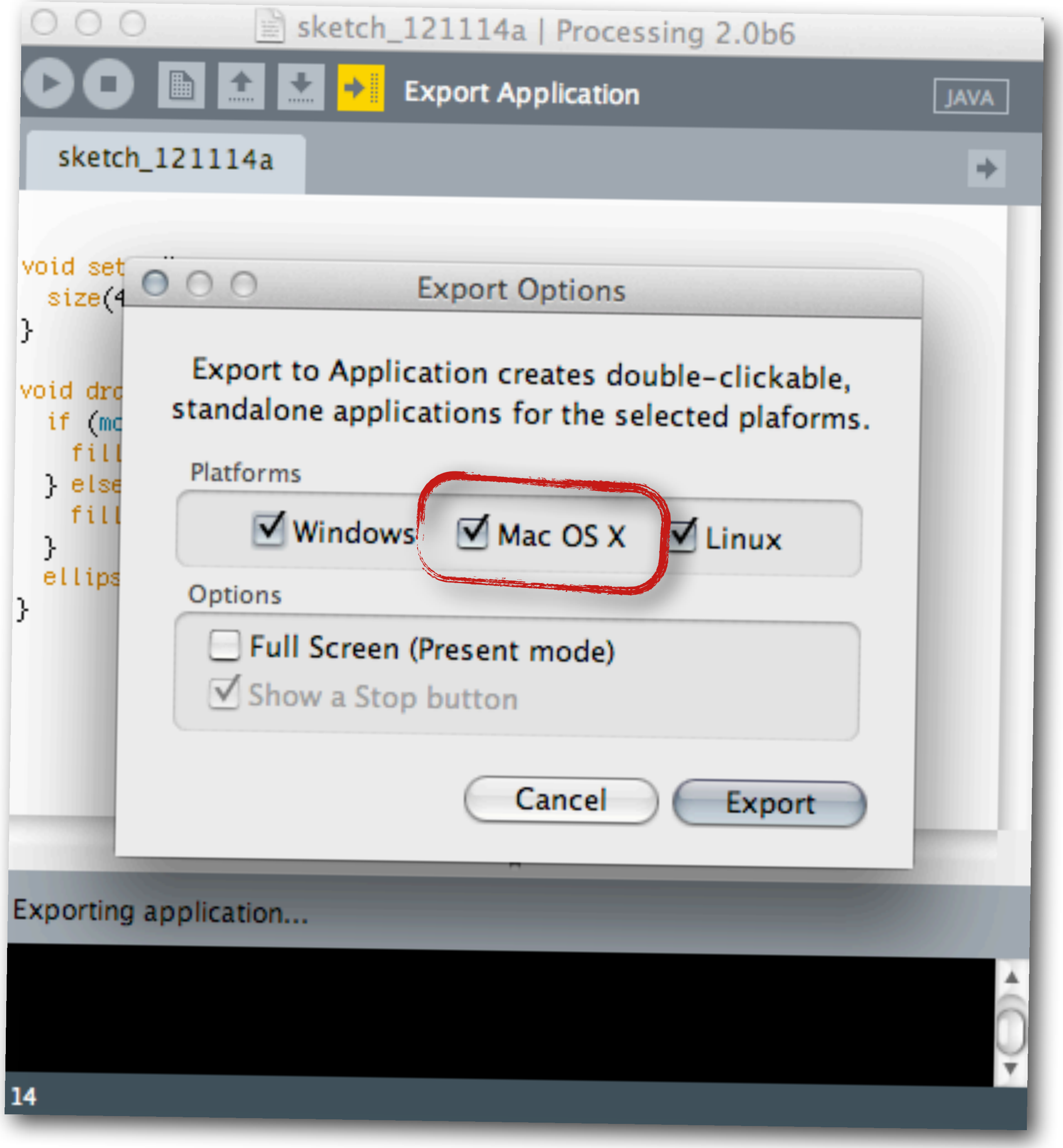
Options

□ Full Screen (Present mode)
☑ Show a Stop button

Exporting application...

Intro to Processing

- Options



```
void set
  size(4
}

void dra
  if (mo
    fill
  } else
    fill
  }
  ellips
}
```

Export to Application creates double-clickable, standalone applications for the selected plaforms.

Platforms

☑ Windows ☑ Mac OS X ☑ Linux

Options

□ Full Screen (Present mode)

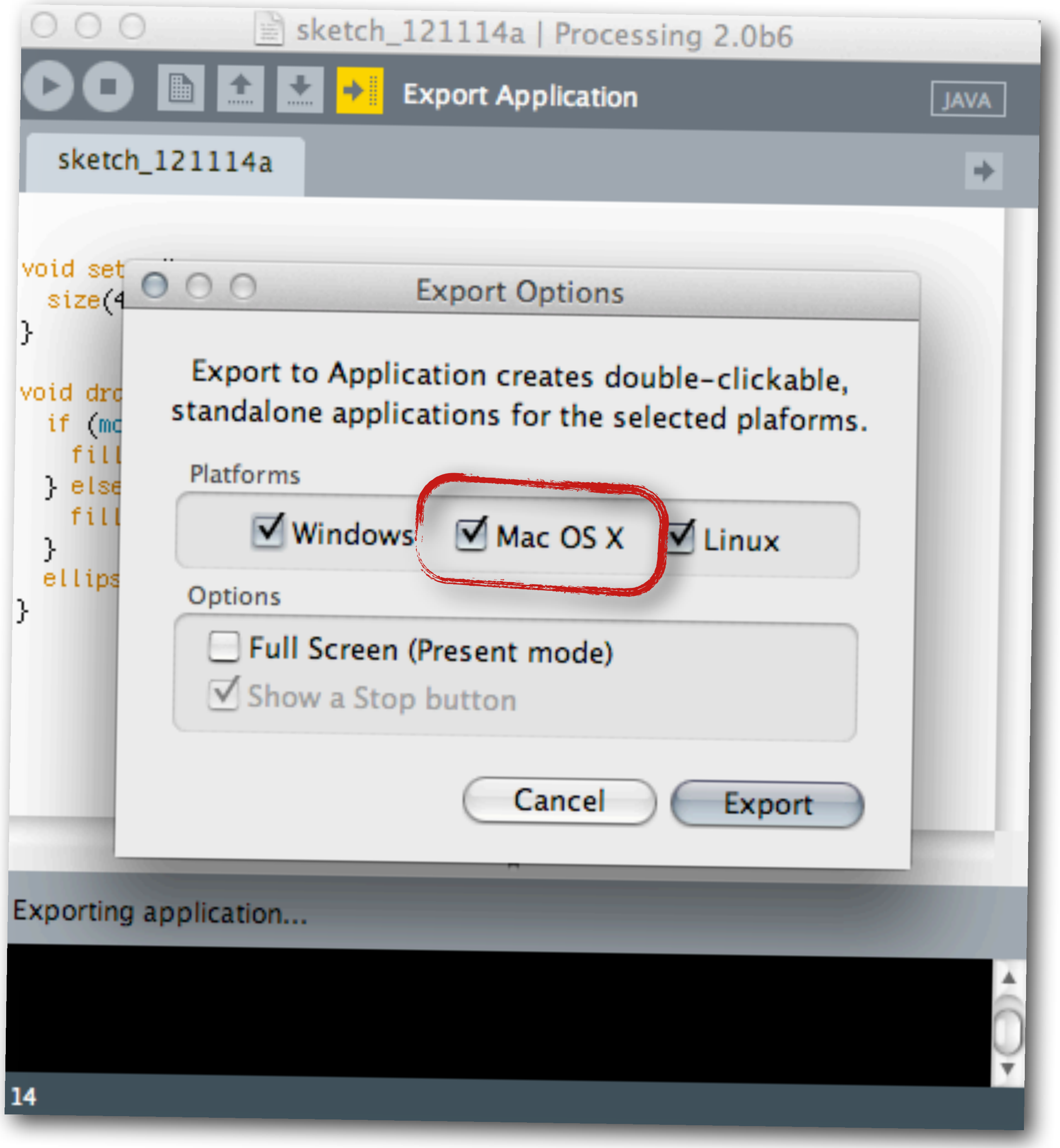
☑ Show a Stop button

Exporting application...

14

Intro to Processing

- Options



sketch_121114a | Processing 2.0b6

sketch_121114a

```
void set
  size(4
}

void dra
  if (mo
    fill
  } else
    fill
  }
  ellips
}
```

Export Options

Export to Application creates double-clickable, standalone applications for the selected plaforms.

Platforms

☑ Windows ☑ Mac OS X ☑ Linux

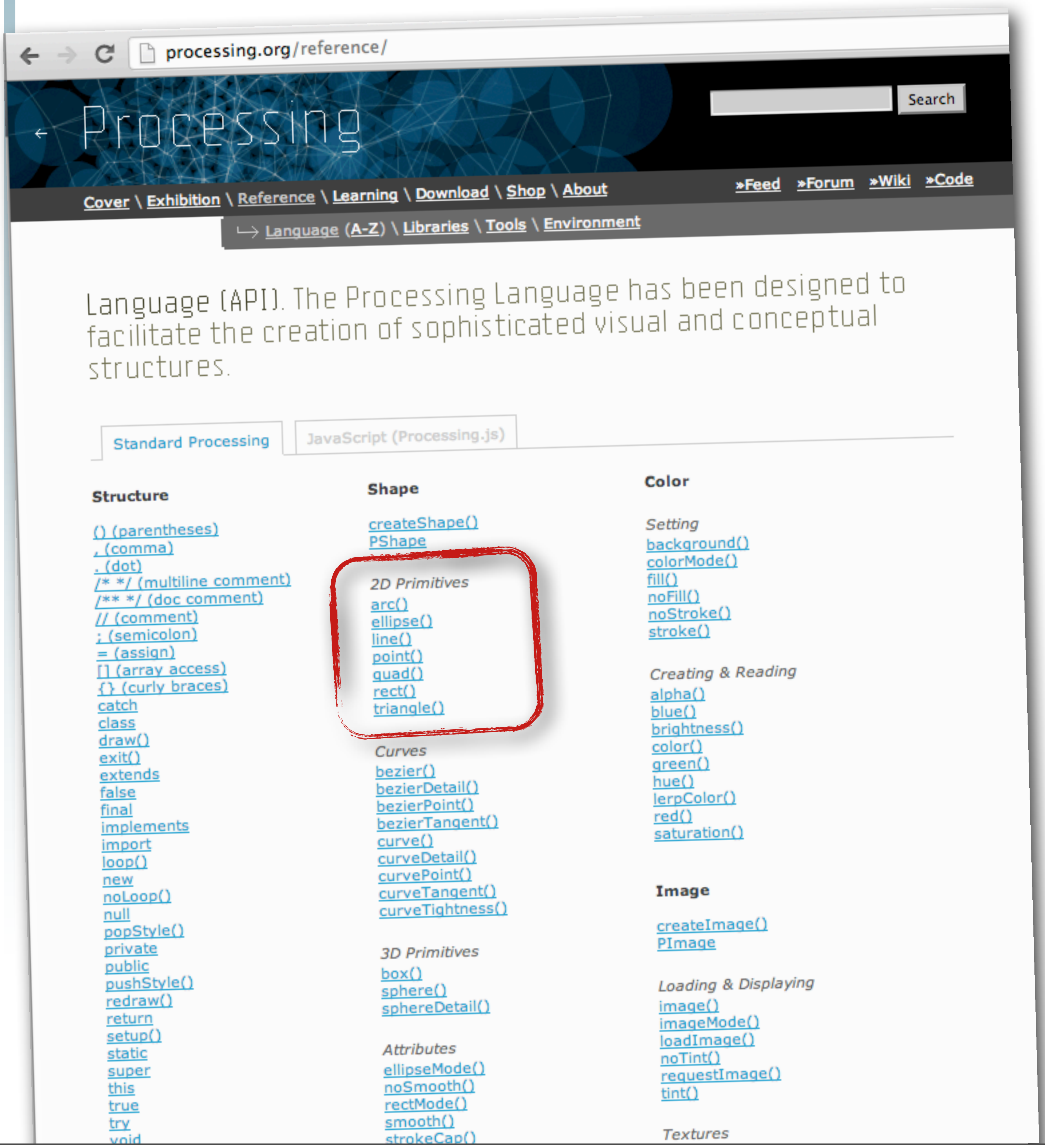
Options

☐ Full Screen (Present mode)
☑ Show a Stop button

Exporting application...

- Creates an application that you can share (or turn in)

Intro to Processing


processing.org/reference/
Processing
Search
Cover \ Exhibition \ Reference \ Learning \ Download \ Shop \ About
»Feed »Forum »Wiki »Code
Language (A-Z) \ Libraries \ Tools \ Environment
Standard Processing
JavaScript (Processing.js)

Language (API). The Processing Language has been designed to facilitate the creation of sophisticated visual and conceptual 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
setup()
static
super
this
true
try
void

Shape

createShape()
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()

Attributes
ellipseMode()
noSmooth()
rectMode()
smooth()
strokeCap()

Color

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

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

Image

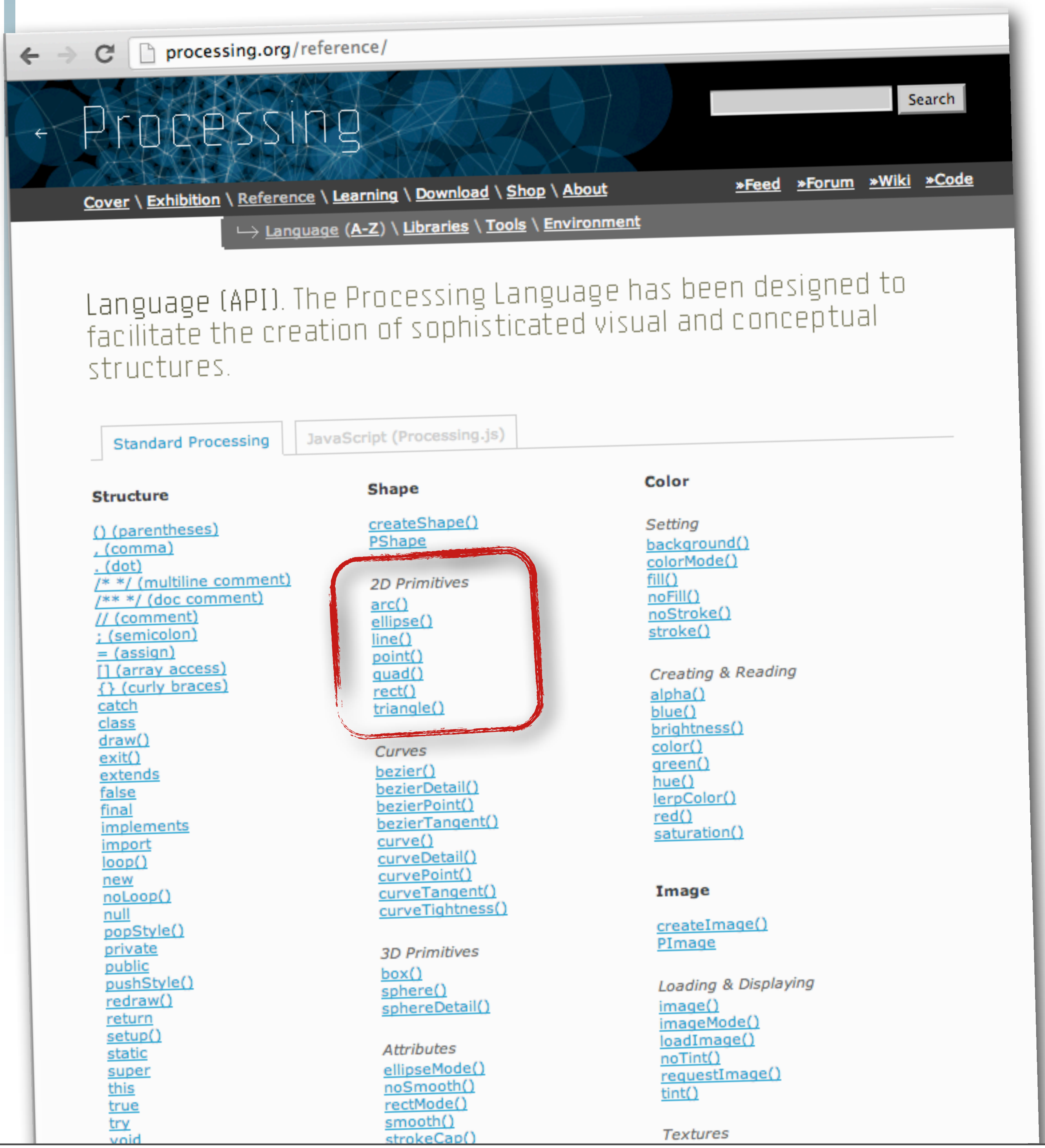
createImage()
PImage

Loading & Displaying
image()
imageMode()
loadImage()
noTint()
requestImage()
tint()

Textures

Intro to Processing

- What else can you draw?



processing.org/reference/
Processing
Search
Cover \ Exhibition \ Reference \ Learning \ Download \ Shop \ About
»Feed »Forum »Wiki »Code
Language (A-Z) \ Libraries \ Tools \ Environment

Language (API). The Processing Language has been designed to facilitate the creation of sophisticated visual and conceptual structures.

Standard Processing
JavaScript (Processing.js)

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
setup()
static
super
this
true
try
void

Shape

createShape()
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()

Attributes
ellipseMode()
noSmooth()
rectMode()
smooth()
strokeCap()

Color

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

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

Image

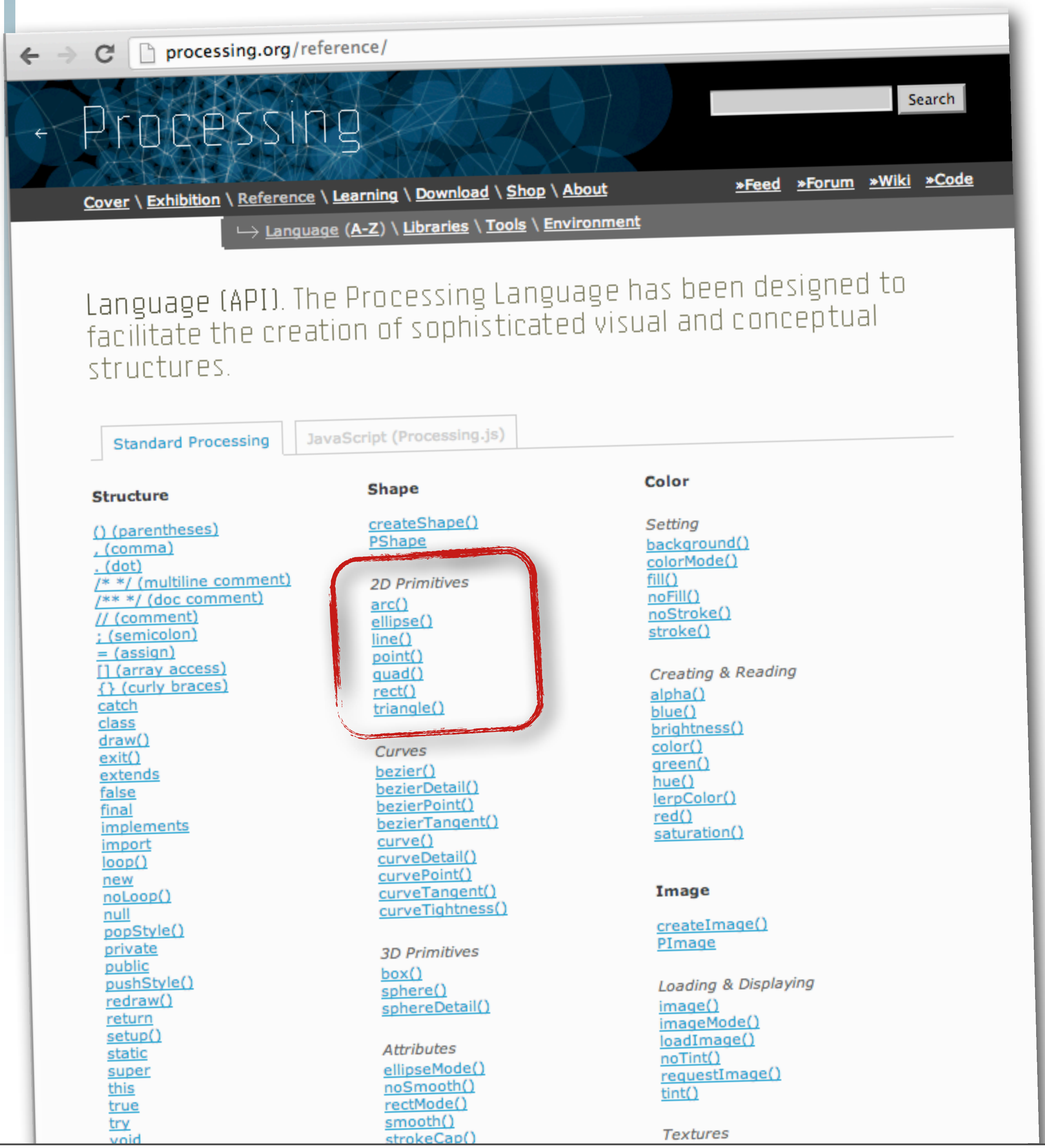
createImage()
PImage

Loading & Displaying
image()
imageMode()
loadImage()
noTint()
requestImage()
tint()

Textures

Intro to Processing

- What else can you draw?


processing.org/reference/
Processing
Search
Cover \ Exhibition \ Reference \ Learning \ Download \ Shop \ About
»Feed »Forum »Wiki »Code
Language (A-Z) \ Libraries \ Tools \ Environment
Standard Processing
JavaScript (Processing.js)

Language (API). The Processing Language has been designed to facilitate the creation of sophisticated visual and conceptual 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
setup()
static
super
this
true
try
void

Shape

createShape()
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()

Attributes

ellipseMode()
noSmooth()
rectMode()
smooth()
strokeCap()

Color

Setting

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

Creating & Reading

alpha()
blue()
brightness()
color()
green()
hue()
lerpColor()
red()
saturation()

Image

createImage()
PImage

Loading & Displaying

image()
imageMode()
loadImage()
noTint()
requestImage()
tint()

Textures

- Click on the link to get an example

LUCI