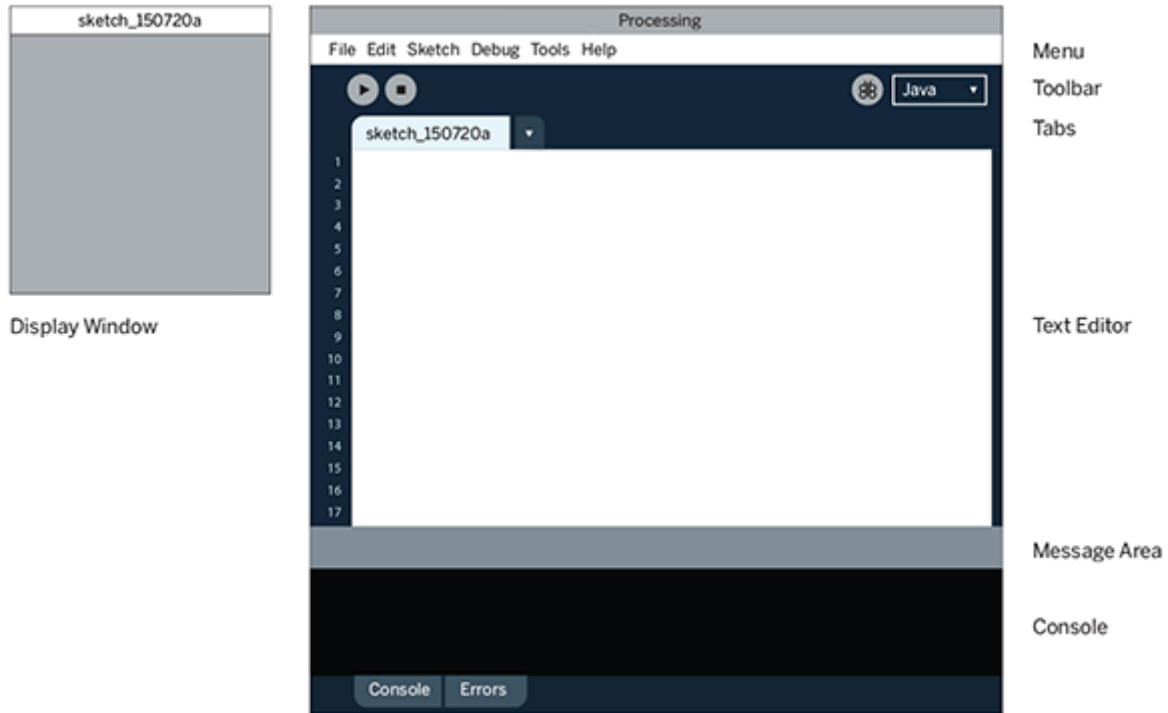


Getting Started: Welcome to Processing!

Processing is a simple programming environment that was created to make it easier to develop visually oriented applications.

1- Your First Program



Double-click processing shortcut to start. You're now running the Processing Development Environment (or PDE). The large area is the Text Editor, and there's a row of buttons across the top; this is the toolbar. Below the editor is the Message Area, and below that is the Console. The Message Area is used for one line messages, and the Console is used for more technical details. In the editor, type the following:

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

Click the Run button (the triangle button in the Toolbar).

1- Describe your result.

2- Change the parameters, click the Run button, and describe yours results.

2- Following the mouse

The setup() block runs once, and the draw() block runs repeatedly. Delete the text from the last example, and try this:

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

```
void draw() {  
  background(255,0,0);  
  if (mousePressed) {
```

```

fill(0);
} else {
fill(255);
}
ellipse(mouseX, mouseY, 80, 80);
}

```

1- Describe your result.

2- Change the parameters for the background(), size(), fill() and ellipse() functions, click the Run button, and describe yours results.

3- Bounce

Variables are used for storing values: int for integer values, float for decimal values. Try this sketch.

```

int rad = 60; // Width of the shape
float xpos, ypos; // Starting position of shape
float xspeed = 2.8; // Speed of the shape
float yspeed = 2.2; // Speed of the shape
int xdirection = 1; // Left or Right
int ydirection = 1; // Top to Bottom

void setup()
{
size(640, 360);
noStroke();
frameRate(30);
ellipseMode(RADIUS);
// Set the starting position of the shape
xpos = width/2;
ypos = height/2;
}

void draw()
{
background(102);
// Update the position of the shape
xpos = xpos + ( xspeed * xdirection );
ypos = ypos + ( yspeed * ydirection );
// Test to see if the shape exceeds the boundaries
of the screen
// If it does, reverse its direction by multiplying
by -1
if (xpos > width-rad || xpos < rad) {
xdirection *= -1;
}
if (ypos > height-rad || ypos < rad) {
ydirection *= -1;
}
// Draw the shape
ellipse(xpos, ypos, rad, rad);
}

```

1- Describe the behavior of this sketch.

2- In this example, change the values of variables to affect the composition.

3- This sketch draw an ellipse. Find out how you can draw a rectangle, or an image instead of this ellipse (you can use the examples of Processing). Modify the sktech to draw a rectangle. Then try to write a sketch with a bouncing Tux.

