



Environment (IDE). The Processing Environment includes a text editor, a compiler, and a display window. It enables the creation of software within a carefully designed set of constraints.

[Overview](#)
[Processing Development Environment \(PDE\)](#)
[Preferences](#)
[Sketches and Sketchbook](#)
[Renderers](#)
[Coordinates](#)
[Tabs, Multiple Files, and Classes](#)
[Programming Modes](#)
[Adding Libraries, Tools, and Modes](#)
[Export](#)

↑ Overview

The Processing Development Environment (PDE) makes it easy to write Processing programs. Programs are written in the Text Editor and started by pressing the Run button. In Processing, a computer program is called a *sketch*. Sketches are stored in the *Sketchbook*, which is a folder on your computer. It's easy to open the sketches by clicking on the Open button.

Sketches can draw two- and three-dimensional graphics. The default renderer is for drawing two-dimensional graphics. The P3D renderer makes it possible to draw three-dimensional graphics, which includes controlling the camera, lighting, and materials. The P2D renderer is a fast, but less accurate renderer for drawing two-dimensional graphics. Both the P2D and P3D renderers are accelerated if your computer has an OpenGL compatible graphics card.

The capabilities of Processing are extended with *Libraries* and *Tools*. Libraries make it possible for sketches to do things beyond the *core* Processing code. There are hundreds of libraries contributed by the Processing community that can be added to your sketches to enable new things like playing sounds, doing computer vision, and working with advanced 3D geometry. Tools extend the PDE to help make creating sketches easier by providing interfaces for tasks like selecting colors.

Processing has different *programming modes* to make it possible to deploy sketches on different platforms and program in different ways. The current default programming modes are *Java* and *Experimental*. Other programming modes, such as *JavaScript* and *Android*, are added by selecting "Add Mode..." from the menu in the upper-right corner of the PDE.







↑ Processing Development Environment (PDE)

The Processing Development Environment (PDE) consists of a simple text editor for writing code, a message area, a text console, tabs for managing files, a toolbar with buttons for common actions, and a series of menus. The menu options change from mode to mode. The default Java mode is documented here.



Programs written using Processing are called sketches. These sketches are written in the text editor. It has features for cutting/pasting and for searching/replacing text. The message area gives feedback while saving and exporting and also displays errors. The console displays text output by Processing sketches including complete error messages and text output from sketches with the `print()` and `println()` functions.

The buttons on the toolbar can run and stop programs, create a new sketch, open, save, and export:

-  **Run**
Runs the sketch. In Java mode, it compiles the code and opens a new display window.
-  **Stop**
Terminates a running sketch.
-  **New**
Creates a new sketch (project) in the current window. To create a new sketch in its own window, use File → New.
-  **Open**
Provides a menu with options to open files from anywhere on your computer (Open...), from the Example Menu (Examples...), or one of the programs in the Sketchbook. Opening a sketch from the toolbar will replace the sketch in the current window. To open a sketch in a new window, use File → Open.
-  **Save**
Saves the current sketch to its current location. If you want to give the sketch a different name, select “Save As” from the File menu.
-  **Export**
In Java mode, it exports the current sketch as a Java application and the folder containing the files is opened. (Note: Exporting a sketch will delete the previous contents of the export folder, unless this preference is unchecked in the Preferences.)

Additional commands are found within the five menus: File, Edit, Sketch, Tools, Help. The menus are context sensitive which means only those items relevant to the work currently being carried out are available.

File

- **New (Ctrl+N)**
Creates a new sketch in a new window, named as the current date is the format "sketch_YYMMDDa".
- **Open (Ctrl+O)**
Open a sketch in a new window.
- **Sketchbook**
Open a sketch from the sketchbook folder.
- **Recent**

[Cover](#)

[Download](#)

[Exhibition](#)

[Reference](#)

[Libraries](#)

[Tools](#)

[Environment](#)

[Tutorials](#)

[Examples](#)

[Books](#)

[Overview](#)

[People](#)

[Foundation](#)

[Shop](#)

» [Forum](#)

» [GitHub](#)

» [Issues](#)

» [Wiki](#)

» [FAQ](#)

» [Twitter](#)

» [Facebook](#)

Open a recently closed sketch.

- *Examples*
Open one of the examples included with Processing.
- *Close (Ctrl+W)*
Close the sketch in the frontmost window. If this is the last sketch that's open, you will be prompted whether you would like to quit. To avoid the prompt, use Quit instead of Close when you want to exit the application.
- *Save (Ctrl+S)*
Saves the open sketch in it's current state.
- *Save as... (Shift+Ctrl+S)*
Saves the currently open sketch, with the option of giving it a different name. Does not replace the previous version of the sketch.
- *Export (Ctrl+E)*
In Java mode, exports a Java application as an executable file and opens the folder containing the exported files.
- *Page Setup (Shift+Ctrl+P)*
Define page settings for printing.
- *Print (Ctrl+P)*
Prints the code inside the text editor.
- *Preferences (Ctrl+,)*
Change some of the ways Processing works. (This item is located in the Processing menu on Mac OS X.)
- *Quit (Ctrl+Q)*
Exits the Processing Environment and closes all Processing windows. (This item is located in the Processing menu on Mac OS X.)

Edit

- *Undo (Ctrl+Z)*
Reverses the last command or the last entry typed. Cancel the Undo command by choosing Edit » Redo.
- *Redo (Shift+Ctrl+Z)*
Reverses the action of the last Undo command. This option is only available, if there has already been an Undo action.
- *Cut (Ctrl+X)*
Removes and copies selected text to the clipboard (an off-screen text buffer).
- *Copy (Ctrl+C)*
Copies selected text to the clipboard.
- *Copy as HTML (Shift+Ctrl+C)*
Formats code as HTML, the same way it appears in the Processing environment and copies it to the clipboard so it can be pasted somewhere else.
- *Paste (Ctrl+V)*
Inserts the contents of the clipboard at the location of the cursor, and replaces any selected text.
- *Select All (Ctrl+A)*
Selects all of the text in the file which is currently open in the text editor.
- *Auto Format (Ctrl-T)*
Attempts to format the code into a more human-readable layout. Auto Format was previously called *Beautify*.
- *Comment/Uncomment (Ctrl+//)*
Comments the selected text. If the selected text is already commented, it uncomments it.
- *Increase Indent (Ctrl+])*
Indents the selected text two spaces.

- *Decrease Indent (Ctrl+I)*
If the text is indented, removes two spaces from the indent.
- *Find... (Ctrl+F)*
Finds an occurrence of a text string within the file open in the text editor and gives the option to replace it with a different text.
- *Find Next (Ctrl+G)*
Finds the next occurrence of a text string within the file open in the text editor.
- *Find Previous (Shift+Ctrl+G)*
Finds the previous occurrence of a text string within the file open in the text editor.

Sketch

- *Run (Ctrl+R)*
Runs the code (compiles the code, opens the display window, and runs the sketch inside)
- *Present (Ctrl+Shift+R)*
Runs the code in the center of the screen with a solid-color background. Click the "stop" button in the lower left to exit the presentation.
- *Stop*
If the code is running, stops the execution. Programs written without using the draw() function are stopped automatically after they draw.
- *Import Library*
Adds the necessary import statements to the top of the current sketch. For example, selecting Sketch » Import Library » video adds the statement "import processing.video.*;" to the top of the file. These import statements are necessary for using Libraries.
- *Show Sketch Folder*
Opens the folder for the current sketch.
- *Add File*
Opens a file navigator. Select an image, font, or other media files to add it to the sketch's "data" folder.

Tools

- *Create Font...*
Converts fonts into the Processing font format and adds to the current sketch. Opens a dialog box which give options for setting the font, it's size, if it is anti-aliased, and if all characters should be generated. If the "All Characters" options is selected, non-English characters such as ù and Å are generated, but the font file is larger in size. The amount of memory required for the font is also determined by the size selected. Processing fonts are textures, so larger fonts require more image data.
- *Color Selector*
Interface for selecting colors.
- *Archive Sketch*
Archives a copy of the current sketch in .zip format. The archive is placed in the same folder as the sketch.
- *Movie Maker*
Creates a QuickTime movie from a sequence of images. Options include setting the size, frame rate, and compression, as well as an audio file.

Help

- *About Processing*
Opens a concise information panel about the software. (This item is located in the Processing menu on Mac OS X.)
- *Environment*
Opens the reference for the Processing Development Environment (this page) in the default web browser.

- *Reference*
Opens the reference in the default web browser. Includes reference for the language, programming environment, libraries, and a language comparison.
- *Find in Reference (Ctrl+Shift+F)*
Select a word in your sketch and select "Find in Reference" to open its reference HTML page.
- *Getting Started*
Opens a tutorial on getting started with Processing.
- *Troubleshooting*
Opens the troubleshooting information in the default web browser.
- *Frequently Asked Questions*
Answers to some basic question about the Processing project.
- *Visit Processing.org*
Opens default web browser to the Processing.org homepage.

↑ Preferences

The Processing Development Environment (PDE) is highly configurable. The most common preferences can be modified in the Preferences window, located in the File menu on Windows and Linux and in the Processing menu on Mac Os X. The full list of preferences are stored in the "preferences.txt" file. This file can be opened and edited directly only when Processing is not running. You can find the location of this file on your computer by reading the bottom-left corner of the Preferences window.

- *Sketchbook location*
Any folder can be used as the Sketchbook. Input a new location or select "Browse" to set the folder you want to use.
- *Editor font size*
Sets the font size of the code in the text editor. Restart Processing after making this change.
- *Use smooth text in editor window*
By default, the text in the editor is aliased. When checked, the editor switches to an anti-aliased (smoothed) font. Restart Processing after making this change.
- *Increase maximum available memory*
Allocates more RAM to Processing sketches when they run. Sketches that use media files (images, audio, etc.) sometimes require more RAM. Increase the amount of RAM if a sketch is throwing Out of Memory Errors.
- *Delete previous folder on export*
When checked (default behavior), Processing deletes the complete export folder before re-creating it and adding the new media.
- *Check for updates on startup*
When checked (default behavior), you'll be informed of new Processing software releases as they become available through a small dialog box that opens as Processing starts.

↑ Sketches and Sketchbook

All Processing projects are called sketches. Each sketch has its own folder. The main file for each sketch has the same name as the folder and is found inside. For example, if the sketch is named "Sketch_123", the folder for the sketch will be called "Sketch_123" and the main file will be called "Sketch_123.pde". The PDE file extension is an acronym for the Processing Development Environment.

Processing sketches can be stored anywhere on your computer, but by default they are stored in the sketchbook, which will be in different places on your computer or network depending if you use PC, Mac, or Linux and how the preferences are set. To locate this folder, select the "Preferences"

option from the File menu (or from the "Processing" menu on the Mac) and look for the "Sketchbook location".

A sketch folder sometimes contains other folders for media files and other code. When a font or image is added to a sketch by selecting "Add File..." from the Sketch menu, a "data" folder is created. Files may also be added to your Processing sketch by dragging them into the text editor. Image and sound files dragged into the application window will automatically be added to the current sketch's "data" folder. All images, fonts, sounds, and other data files loaded in the sketch must be in this folder.

↑ Renderers

Processing has three built-in renderers. The default renderer is for drawing two-dimensional shapes. *P2D* is a faster, but less accurate renderer for drawing two-dimensional shapes. *P3D* is for three-dimensional geometry, it can also control the camera, lighting, and materials. The P2D and P3D renderers are accelerated if your computer has an OpenGL compatible graphics card.

The renderer used for each sketch is specified through the `size()` function. If a renderer is not explicitly defined in `size()`, it uses the default renderer. For example, the following program:

```
void setup() {
  size(200, 200);
}

void draw() {
  background(204);
  line(width/2, height/2, mouseX, mouseY);
}
```

To change the renderer, add a third parameter to `size()`. For example:

```
void setup() {
  size(200, 200, P2D);
}

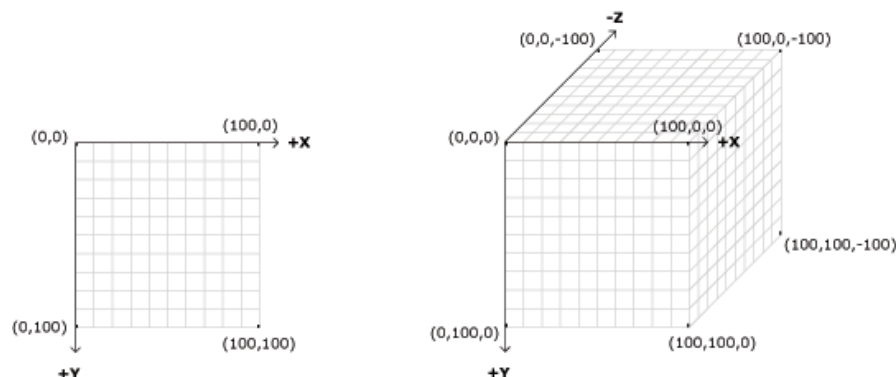
void draw() {
  background(204);
  line(width/2, height/2, mouseX, mouseY);
}
```

A large effort has been made to make the Processing renderers behave similarly across the different renderers, but there are currently some inconsistencies that are explained in the reference.

For more information, see the [size\(\)](#) reference entry.

↑ Coordinates

Processing uses a Cartesian coordinate system with the origin in the upper-left corner. If your sketch is 320 pixels wide and 240 pixels high, coordinate (0, 0) is the upper-left pixel and coordinate (320, 240) is in the lower-right. The last visible pixel in the lower-right corner of the screen is at position (319, 239) because pixels are drawn to the right and below the coordinate.



Using the three-dimension coordinate system of P3D, the z-coordinate is zero at the surface of the image, with negative z-values moving back in space. When drawing in 3D, the *camera* is positioned in the center of the screen.

↑ Tabs, Multiple Files, and Classes

It can be inconvenient to write a long program within a single file. When Processing sketches grow to hundreds or thousands of lines, breaking them into modular units helps manage the different parts. Processing manages files with the Sketchbook and each sketch can have multiple files that are managed with tabs.

The arrow button to the right of the tabs in the Processing Development Environment is used to manage these files. Click this button to reveal options to create a new tab, rename the current tab, and delete the current tab. If a project has more than one tab, they can also be hidden and revealed. Hiding a tab temporarily removes that code from the sketch (it will not be compiled with the sketch when you press Run).

Tabs are intended for more advanced users, and for this reason, the menu that controls the tabs is intentionally made less prominent.

Advanced

When a program with multiple tabs is run, the code is grouped together and the classes in other tabs become inner classes. Because they're inner classes, they cannot have static variables. Simply place the "static" variable outside the class itself to do the same thing (it need not be explicitly named "static" once you list it in this manner). If you don't want code to be an inner class, you can also create a tab with a .java suffix, which means it will be interpreted as straight java code. It is also not possible to use static classes in separate tabs. If you do this, however, you'll need to pass the PApplet object to that object in that tab in order to get PApplet functions like `line()`, `loadStrings()` or `saveFrame()` to work.

↑ Programming Modes

Processing has different *programming modes* to make it possible to deploy sketches on different platforms and program in different ways. The current default programming modes are *Java* and *Experimental*. Other programming modes, such as *JavaScript* and *Android*, are added by selecting "Add Mode..." from the menu in the upper-right corner of the PDE.

Java Mode

This mode makes it possible to write short programs to draw to the screen, but also enables complex Java programs as well. It's can be used simply by beginners, but it scales to professional Java software development. Sketches written in this mode can be exported as Java Applications to run on Linux, Mac OS X, and Windows operating systems.

Advanced

Java files with the extension .java can be included with a Java mode sketch. They may be created directly in the PDE or copied into the sketch folder through the "Add File..." item in the Sketch menu or dragged into the text editor. It's possible to write any Java code in files with the .java extension. In fact, complete Java code can be written from inside the Processing Environment by subclassing PApplet like this:

```
public class MyDemo extends PApplet {
```

This is for advanced developers only and is not really recommended. Using this technique means that any additional tabs will no longer be inner classes, meaning you'll have to do extra work to make them communicate properly with the host PApplet. It is not necessary

to use this technique just to get features of the Java language. Advanced developers can also program with Processing in another Java Editor if higher-level code editing and tools are needed. We've documented how to use [Processing in Eclipse](#), but core.jar can be used as a part of any Java project.

Experimental Mode

This is a prototype of a potential future version of the Processing Development Environment. It includes features to check for errors in the code while it's written, to follow variables as they change, to debug a program with break points, and more.

JavaScript Mode

Sketches written in this mode can be exported to run inside web browsers using HTML5 and WebGL. This mode is documented on the [JavaScript](#) page of the Processing Wiki. To add this mode, click on the mode button in the upper-right corner of the PDE and select "Add Mode..."

Android Mode

Sketches written in this mode can be exported to run on Android phones and tablets. This mode is documented on the [Processing for Android](#) page of the Processing Wiki. To add this mode, click on the mode button in the upper-right corner of the PDE and select "Add Mode..."

↑ Adding Libraries, Tools, and Modes

Processing 2.0 includes a set of new features to make it easier to install, update, and remove Libraries, Tools, and Modes.

Add contributed libraries by selecting "Add Library..." from the "Import Library..." submenu within the Sketch menu. Not all available libraries have been converted to show up in "Add Library...". If a library isn't there, it will need to be installed manually. Follow the [How to Install a Contributed Library](#) instructions on the Processing Wiki for more information.

Add contributed tools by selecting "Add Tool..." from the Tools menu to select a Tool to download.

Add contributed modes by selecting "Add Mode..." from the Mode menu in the upper-right corner of the PDE.

↑ Export

The [Export information and Tips](#) page on the Processing Wiki covers the details of exporting Applications from Java mode.

Exporting from JavaScript mode is discussed on the [JavaScript](#) page of the Processing Wiki.

Exporting from Android mode is discussed on the [Android](#) page of the Processing Wiki.

Processing was initiated by [Ben Fry](#) and [Casey Reas](#). It is developed by a [small team of volunteers](#).

[© Info](#)