A vector drawing program for Unix

Sketch Tutor

User's Guide

Berhnard Herzog, Toussaint Frédéric, Yves Ceccone, André Pascual

Translation : Michel ARMAND

Index

Index

Sketch User's Guide

Introduction

What is Sketch ?

Sketch is an interactive, object oriented drawing program. This means that the drawing is composed of objects like rectangles, lines or pieces of text. Sketch allows you to manipulate the objects by moving them around, resizing them or changing their color, etc.

This type of program is also called a *vector drawing software* because objects are stored as mathematics coordinates.
There are other similar softwares : Xfig, Tgif, Sodipodi, Gyve are open source ones therefore free. The most popular are commercial : Corel Draw or Adobe Illustrator.

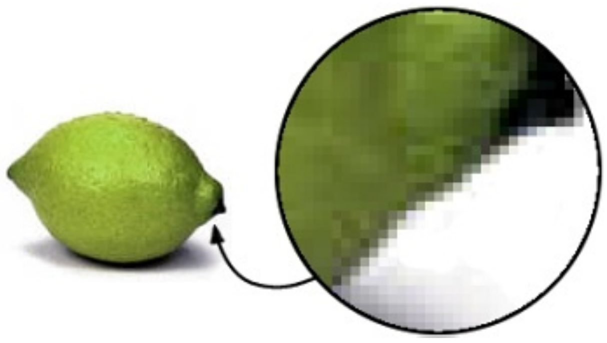
Sketch is written in an interpreted language called Python, which makes it unique

Difference between Vectorial and Bitmap

Two types of software can be used to draw : vector drawing programs such as Sketch and bitmap softwares like The Gimp or Adobe Photoshop (r).
Although twice are able to help you draw graphics, the way they work, the way you understand them and finally their use are totally different.

Bitmap drawing :

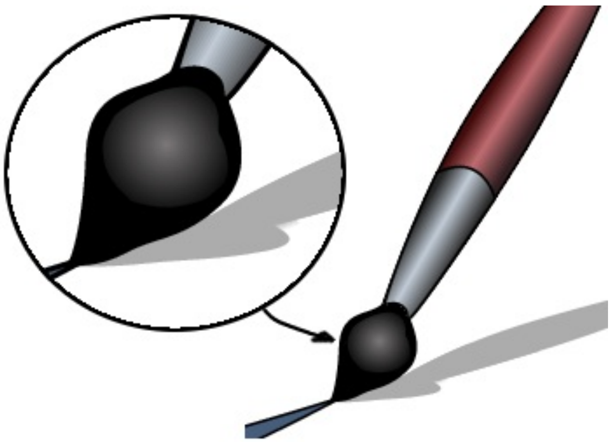
Bitmap drawing means you work with pixels. Pixel is a point which can have color and transparency values. Images used on the internet or digital pictures consist of a cloud of pixels. Let's magnify a bitmap image :



We can see that the zoom factor (600%) makes pixels visible. Bitmap drawing software can select these pixels and manipulate them. This kind of image suits retouching, mounting, special effects or working with a drawing tablet allowing the use of tools like pencil, airbrush, ... You can easily understand that the "weight" of an image is proportional to the number of pixels it is made of.

Vectors based drawing :

Contrary to bitmap (with pixels), a vector drawing consists in objects that are mathematics curves (Bezier curves) or lines and segments which have a color and a size. Their primary advantage comes from the fact that there is no visible pixel even if you magnify a lot.



When a zoom is done, curves that compose the drawing are automatically re−calculate so as to be scaled. That allows you to work on an element or a group of elements without changing the quality. High resolution works often requires this kind of image. In that case, the "weight" of the image is independent of the size.

Requirements

Fortunately, Sketch just requires 6 Mo RAM at the maximum (it depends on the number of objects on screen). Therefore, the use of layers optimizes the use of memory. In the processus list, Sketch appears under the name of Python. Pentium 200 with 64 Mo RAM is sufficient. Based on floating toolboxes, Sketch requires a large screen (17").
Although it's impossible to recommend a Linux distribution, Sketch needs the following libraries:

- Python 1.52
- Python imaging 1.0
- tcl/tk 8.0

This libraries are available in .rpm, .deb (Debian and Corel) or sources if you want to adapt it on special platforms (Sparc or alpha).Please visit the official site : http://sketch.sourceforge.net for further information.

Installation

Installation or libraries checking

Sketch needs these libraries:

− Tcl/tk
Generally included in the major distributions, Sketch uses tcl/tk 8.0 but Tcl 7.6 et Tk 4.2 should also

work.
Find it at http://www.scriptics.com/software/download.html

− Python
Python is a interpreted programming language, available at http://www.python.org in .rpm format or directly in source code.
This package is almost always included in all distribution.

Sketch works with Python 1.51.1; previous versions are not supported.
If you compile Python on your system, please check that you allow the compilation of the Tkinter module (see Installation of Python). Tkinter is the TK interface.

−Python Imaging Library (PIL)
The Python Imaging Library is a library that allows manipulation of different picture formats. .rpm package is available at http://www.pythonware.com . If you decide to compile it, please refer to the installation instruction included with the sources. Version 1.0 works perfectly.

−Python XML (optional)
This library is required if you want to import SVG files. If you don't know what is XML an SVG jump to next step. Sources are available at this address http://www.python.org/topics/xml/download.html

Installation from the .rpm package (redhat package manager)

It's the simpliest method. Provided all the quoted libraries previously are correctly installed, you just have to run :
rpm −i /the_folder_containing_the_rpm/sketch−0.6.x−1.i386.rpm
Sometimes, one or several libraries are correctly installed but don't appear in the libraries database.
When the package manager check the libraries, it could block the installation if a library doesn't appear.
If you are sure the library is present (you have compiled it) you can add −force in command line so as to "force" the installation.

Installation from the sources

Extract the archive, open an xterm from the directory and then:

- configuration : ./setup.py configure
- compilation : ./setup.py build
- installation : ./setup.py install

Running sketch in a xterm should work if all has been correctly installed (libraries) and compiled (source).
Please note that if the libraries are not compiled on your system, you MUST install the development packages of the libraries so as to compile Sketch.

By default, Sketch is not compiled to take charge of local languages; if you want Sketch to speak french or another available language put the following option in the command line:
./setup.py configure −−with−nls

Licences

Here is what says **Bernhard Herzog,** the author of Sketch, concerning the licence under whom Sketch has been placed:

- Sketch itself is published under the terms of the GNU Library General Public License (LGPL). Some of the code distributed with Sketch is not directly part of Sketch and is distributed under a

Python−style license.
- I've put it under the LGPL, because many parts of Sketch can be use as a library (and are used by Sketch that way), and the LGPL is less restrictive than the normal GNU General Public License when it come to distributing an LGPL'ed library with software covered by a different license.

Configuration

Fonts

Introduction

Fonts are a somewhat complex matter in Linux and X. Sketch's main output format for printing is PostScript. Because of that, Sketch identifies fonts with their PostScript names and must know how to get the appropriate metrics (these tell Sketch how to place the individual letters for instance) and it must know X's name for the font (to be able to show the text on the screen appropriately).

The information Sketch needs to map the PostScript names to metric−filenames (.afm−files) and X font names is stored in .sfd−files.

sfd−Files

An sfd−file maps font names to the filenames of the afm−files and to various font attributes like weight and width and to two strings that are used to build the font name for X.

For example, the line concerning the Courier font in Resources/Fontmetrics/std.sfd is:

```
Courier, ... ,−adobe−Courier−medium−r−normal,iso8859−1,pcrr8a
```

This tells Sketch that the metrics file is pcrr8a.afm and that the first part of the font name is −adobe−Courier−medium−r−normal and its last part is iso8859−1. The parts missing from the font name describe the size and transformation and are automatically filled in by Sketch. Sketch assumes that fonts can be arbitrarily scaled and transformed. (The XFree servers often used on Linux have this capability; other servers on other platforms may not)

If the font is not installed in the server (either the X−server or the font server), or if it is installed under a different name, Sketch will not be able to display the text.

On start−up, Sketch reads all sfd−files it finds in the directories listed in its font−path.

The font−path

Sketch maintains a list of directories where it searches for font−related files like sfd−files and metrics. Note that this is a sketch−specific font−path and has nothing to do with X's font−path.

By default (as of version 0.6.2) the font−path consists of the directories Resources/Fontmetrics (relative to where Sketch is installed), /usr/X11R6/lib/X11/fonts/Type1 and /usr/share/ghostscript/fonts.

If you need to add directories or otherwise modify the font−path, you can do so in Sketch's start−up file, ~/.sketch/userhooks.py, which is executed by Sketch if it exists. E.g. to add a directory to the font−path you could put this code into userhooks.py:

```
from Sketch import config

config.font_path.append('/usr/local/share/fonts/bitstream')
```

The next step explains you how to add fonts to Sketch.

Adding fonts

If you want to install new "Type1" fonts for Sketch you first have to install them for X and create the **sfd file.**

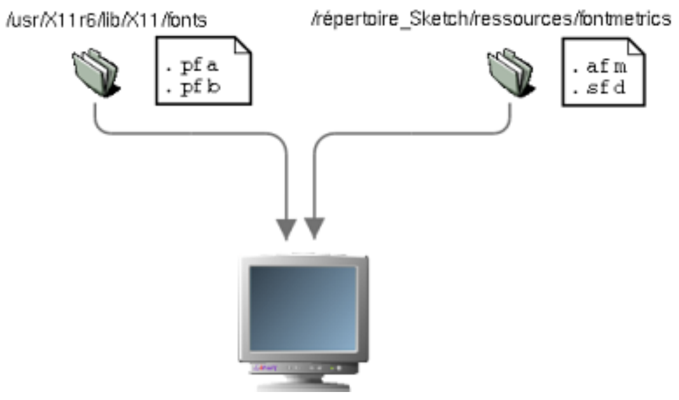
Man pages are available if you need help to install fonts for X:

- mkfontdir (how to create the file containing font.scale and font.alias)
- Xset
- Xserver (general information)
- XF86Config (X settings)

The configuration of fonts is easy because Sketch is distributed with **mkfontdb.py** enabling the simultaneous creation of the **.sfd** file and the **fonts.scale** for X. The author thinks recommend this method so as to be sure the files fit. If they do not, Sketch can display fonts.

The easiest way to use **mkfontdb.py** is the following:

- Create a directory in usr/local/share/fonts and place **.afm and .pfa or .pfb files in** (you can place this directory in /usr/X11R6/lib/X11/fonts, but usr/local/share/fonts is commonly used).
- Create a symbolic link of **mkfontdb.py** from the /tool directory (installation of Sketch) to /usr/local/bin
- Run **mkfontdb.py** with the −s option which writes the **std.sfd** file and with the −x option xhixh writes the **fonts.scale** file.
- Add the fonts directory to the configuration file of X (*/etc/XF86Config*)
- Add the **.afm** and **.sfd** files in */Sketch_directory/ressources/FontMetrics*
- Edit the */etc/X11/XF86Config* file and add the path to the new directory.



You may have to rename fonts.scale into fonts.dir to make it recognized by X. Sketch will detect the new fonts if you copy the **sfd** and **.afm files in** */ressources/fontmetrics* or if you create the **userhooks.py** file. Restart X and Sketch, here are the new fonts ! Note that if the files are compressed (afm.tgz), you must decompress them.

If you want to save your work in postscript format or print it, you will notice that the font will be replaced by a general use font. The font is correctly installed but Ghostscript, the Postscript interpreter for Linux, converts files into data understandable by the monitor or the printer. If Ghostscript doesn't "knows" the font, it will replace it.

There are 2 solutions:

- You convert the text into curves with "curves−>convert into curve"
- You add your fonts to the Ghostscript fonts database

The first one is the easiest. Moreover you will be sure that the result will be the same if you export your work, if someone else edit it or print it.

Note :
ttf2tp1 is a tool very useful to quickly convert TrueType fonts into type1 (.pfa and afm files).
It is available at : http://www.netspace.net.au/~mheath/ttf2pt1/

Options....

File−>Option opens the general options box of Sketch.

Here are the available settings:

Undo levels :

Provided your system is equipped with a large quantity of RAM (64 Mo), you can set it to infinity.

Duplicate :

The 2 settings here are specific to the shift between elements.
In the following example vertical shift is set to 0 and horizontal is set to 50

If you use the transformation tool you will obtain the same results buts elements will not be independent, so you won't be able to modify them individually. Therefore, duplicate is more precise.

Unity by default :

Concerning its grid and guides, Sketch offers several measurement units : dots, inches, cm and mm.
This parameters are active only when you click OK in the dialog box.

Tools and commands

Menus

File Menu

New :
Create a new file. Only one document can be opened in Sketch.

Open :
Opens a document. The type is automatically recognized. Supported formats are: Xfig, ai (adobe illustrator), CMX (Corel), SVG (the new internet standard in vector−based drawing) and WMF.

Save, Save As...
Save the document. If the file was already saved with the Sketch format (.sk), it will be saved and Sketch creates backup file. If it's a new document or in a different format, Sketch asks you a name. Sketch can write files in the following formats: SK, AI (illustrator) and SVG (uncomplete).

Save as Postscript :
Save the current document in Postscript format with the eps specification. This format is very widespread in the Computer Assisted Production world.

Print :
Print the current document.

Import a file :
Insert a document in the opened file. A dotted frame is located near the cursor to help you place the drawing.

Options :
Configuration of the options...See configuration

About Sketch :
About Sketch....

List of last opened :
Very useful, Sketch stores a list of the last opened documents you worked on.

Quit :
Quit Sketch, if the document has been modified without having been saved, a dialog box appears and asks you if you want to save it or not.

Edit Menu

Undo
Undo the last action.

Redo
Redo what have been canceled.

Delete the Undo History
Sketch stores the actions you do. You can limit the number in the options dialog box.
You can empty the history with this button.

Copy...
Used to copy the selected object or group of objects and put it in the clipborad of Sketch. It doesn't work over layers.

Cut...
Cut the selected object or group of objects and put it in the clipboard.

Paste...
Paste the content of the clipboard. The object is representaed by a dotted frame attached to the cursor. Click where you want to place it.

Delete
Delete the selected object or group of objects.

Select all
Select all the objects. Objects placed on a hidden layer are ignored. Only the visible elements are taken into account.

Duplicate
Creates a clone of the selected object or group of objects. The clone apperas automatically near the original.

Create
Here are the submenus available with this control:

◊ Draw a rectangle
◊ Draw an ellipse
◊ Draw a poly−line
◊ Insert text
◊ Insert image....
◊ LCD Text
◊ Creates text with the LCD aspect (you can specify the size in the dilog box)
◊ Regular polygon

Opens a dialog box allowing the creation of polygon. You can enter the number of vertices and the radius. **Selection Mode**
Activate the Selection Mode. See selections.

Edit Mode
Toggle to Edit Mode. See Objects Edition.

Effects Menu

Flip horizontally

Flip the current object(s) horizontally

Flip vertically
Flip the current object(s) vertically

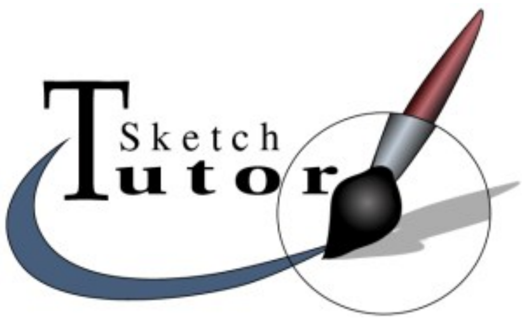
Remove transformation

Some objects have an intrinsic geometry, like text objects, where the geometry is defined by the font and the font size, and bitmap images or EPS files. Any transformations (rotations, reflections, ...) later applied are stored in the object. **Remove Transformation** reverts these objects to their natural size and orientation.

Transformations...
This command opens the Transformations floating toolbox.
Transformations only work with 2 objects : the first and the last element. You must select this 2 elements (a circle and a rectangle for example) and the transformation tool will automatically generate x intermediate elements. This tool is very useful to create complex color gradations.
More details at Floating toolboxes and Special affects.

Cancel Transformation
Cancel the transformation.

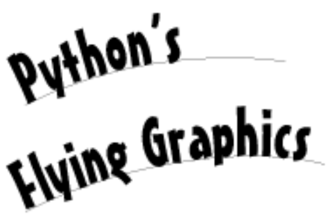
Create a mask
This is useful to create intersection between objects.
For example, you want to create a text filled with a texture.
Import the image designed to be the background. Cut it according to the desired text (select the 2 objects and apply "Create a Mask". Please note that this command will work only with curves, not with groups... first ungroup them). The hierarchy of objects on the layer is not neutral: it's the object at the top of the stack that cuts.
The following image shows a group of object (the head of the paintbrush), the mask is a circle. It is possible to modify the size of the circle : it will modify the size of all the objects grouped with the circle (here the head of the paintbrush) and create a zoom effect (note that the paintbrush behind is a copy made before the creation of the mask).

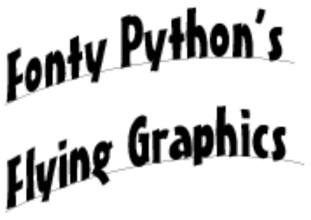


You can find an example with the detailed method in the special effects rubric.

Create a text on path
This function allows you to create a text that "follows" a path.
Create a text, a curve and select them. At that time, the function is available and once applied, the text will follow the curve. Then you can modify the text and the curve as you want.
If you right click on the selected object, the menu will offer you 2 options: rotated letters or skewed

letters. With this menu you can also select the text or the letters to modify them. Here are examples of the result with the 2 options:





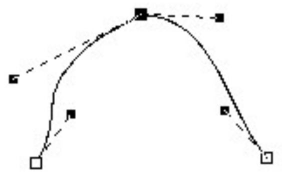
Text on a path can have either rotated letters... or skewed letters.

You can find an example with the detailed method in the special effects rubric.

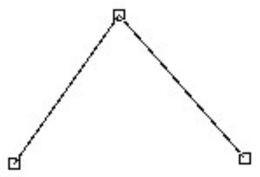
Curves menu

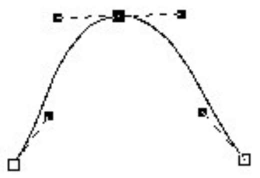
Most of the commands (except "Convert into curves") are only available in Edit Mode and by the Curves floating toolbox. Curves are often called Bezier curves (Bezier was the french mathematician who invented them).

Angle
Allow to modify only one handful on a knot in Bezier Mode.

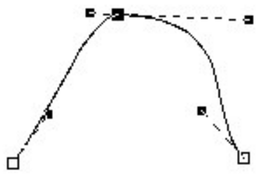


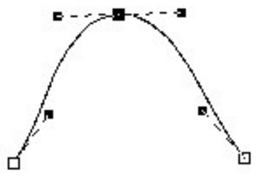
Smooth
Transform a straight angle by smoothing it (Bezier). Select the knot(s) then apply.





Symmetry
Make handfuls of Bezier curves symmetric. You have identical segments.





Curve−−>Line

Transform a curves (Bezier knots) into a line (straight angles).

Line−−>Curve
The opposite of the previous command.

Delete knots
Choose a knot and delete it.

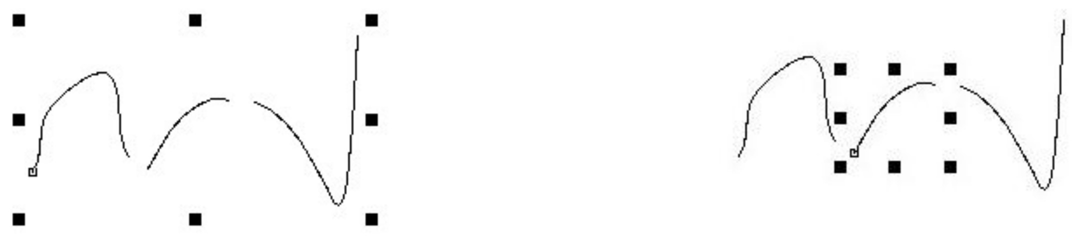
Add a knot
First click where you want the knot to be placed (a dot appears) click this command... you have added a knot.

Join knots
You can merge 2 ends by selecting knots (click on the first one then old <shift> and click on the second) and join them. Essential if you want to close a curve.

Cut the curve
Select the point and cut the curve at this point.

Associate Bézier
You can merge 2 Beziers curves with this command.
First go out of the Edit Mode (spacebar), select the 2 curves and launch this command. You obtain 1 object.

Separate Bézier
First verify that the curve comes from a association of different segments. If it's not the case, use "Cut the curve". You will obtain as many individual objects as segments.



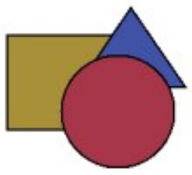
Convert into curves
This command allows you to transform a text into curves.
You must do this if you want to transform it into a mask or if you want to use it with transformations.

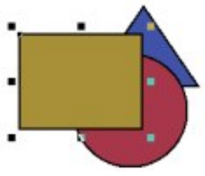
Dispose menu

Align..
Open the Alignments toolbox.

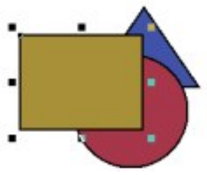
Move to top
Move the current object to the top of its layer.

−−>

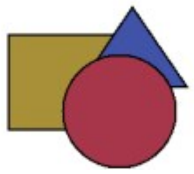




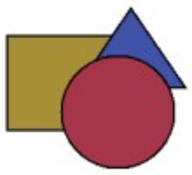
Move to bottom
Move the current object to the bottom of its layer.



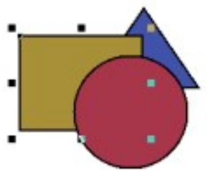
-->



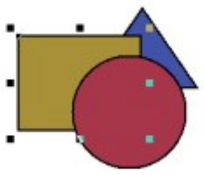
Move One Up...
Swap the current object and the next higher one.



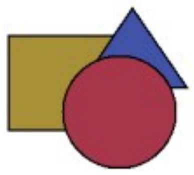
-->



Move One Down...
Swap the current object and the next lower one.



-->



Horizontal limit
Align objects on their horizontal limit (each one above the other).

Vertical limit
Align objects on their vertical limit (each one aside the other).

Group
Group objects into an element.

Ungroup
The opposite of the previous command

Snap to objects
If snapping to objects is active, dragged objects snap to the nearest special point of any other visible object. Special points are the nodes of a bézier curve or the corners of a rectangle.

Snap to grid
When this option is activated, the objects magnetize themselves on the grid. Very useful to place them with precision. By default the grid of Sketch is not visible (see layers section).

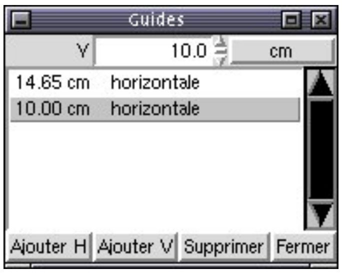
Grid
Open the Grid floating toolbox.

Magnetic guides
Guides are lines aimed at "guiding" objects on the canvas. They are similar to rules.
To add rules, click on a rule around the canvas and drag the mouse on the canvas : the guide automatically appears. You can also choose one of the following commands.
As grid, guides can attract objects. Very useful if you want objects to be lined up on the same axis.

Add an horizontal guide
Add a guide!...

Add a vertical guide
Add a guide...

Guides...
Open the "Guides floating toolbox" displaying the current guides on the canvas.



You can click on one of the existing guides, specify its position (in cm here) on the canvas. Press Enter to validate. You can also add or delete guides.

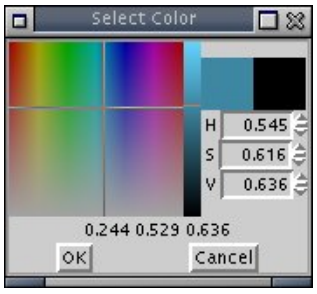
Make up
This command open the floating make up toolbox of the document.

Styles menu

No filling up
The object or the group f object cannot have a filling up attribute.

Filling up
Open the floating filling up toolbox.

Filling up color



This command opens the "floating colors toolbox" and allows you to choose a color and apply it to the selected element.

No line
This command delete the line around the object. By default, this line is black when the object is created.

Line
Open the floating line toolbox.

Name of style...
Create a style.
A style consists in parameters applied to an object (lines type, filling up type...) you can save. You can simply apply with saved style to a lot of different objects. You must give a name to a style. You can also deactivate some options of styles.

Styles...
Open the floating styles toolbox.

Update the style
All the parameters you will apply from a style to the others objects are linked. You can modify the parameters of an object, the others will be modified too simply with this command.

Font
Open the floating fonts toolbox.

Styles menu

Settings of the canvas:

100%
Zoom set to normal size (100%)

Zoom
Predefined values for zoom (25%, 200%, etc.)

Zoom forward
The value increases of the double of the value indicated at the bottom of the Sketch window.

Zoom backward
The value decreases of the double of the value indicated at the bottom of the Sketch window.

Zoom (option)
When activated, the mouse will be a magnifier. Click to zoom.

Adapt to window
Adapt all the objects to the window size.

Adapt the selection to the window.
Automatically adapt the selected object to the window size.

Adapt the page to the window
Adjust the page to the window size.

Restore the previous view
Return to the view before a zoom.

Redraw
This command is used to re−calculate the display of the objects on the canvas. Use it when some moves let parts of an object on the canvas..

Underline
Objects are displayed as lines. Useful when objects hide others.

Display the page
Hide or show the page on the canvas. The page is set with the Sketch's options. This option is activated by default.

Open the palette
Open the palette at the bottom of the Sketch's window. Double−click on a color to validate it in the "Filling up palette".

Scripts menu

Sketch offers the possibility to create you own scripts.
This scripts appear in the "Scripts menu".
See the scripts section for more details.

Windows menu

How to hide and show the floating toolboxes of Sketch?

Hide
Hide all the opened floating toolboxes.

Show
Show all the opened floating toolboxes

Layers...
Open the floating layers toolbox.

Align...
Open the floating alignments toolbox.

Grid..
Open the floating grid toolbox.

Line...
Open the floating line parameters toolbox.

Filling up...
Open the floating filling up parameters toolbox.

Fonts...
Open the floating Fonts toolboxes.

Style...
Open the floating styles toolbox.

Make up
Open the floating make up parameters toolbox.

Transformations...
Open the floating transformations toolbox.

Curves
Open the floating curves toolbox.

Main Toolbar

Operations on files

New file

Create a new file. Sketch can open only one document at once time.

Open a file

This button opens a dialog box : choose the file to open.
Supported file formats are: Sketch (.sk), Adobe Illustrator (.ai), Corel (.cmx), SVG (new standard for the web) and WMF.
Sketch automatically recognize the type.

Save as

Save the current document. If the opened document was already saved in Sketch format, Sketch creates a backup file and save the file.
If the document was not previously saved, you must give it a name.
Sketch can save in Sketch (.sk) an Illustrator (.ai) formats..

Save As PostScript

Save the current drawing into a PostScript file. The file conforms to the EPS specification (I hope).

Edit commands

Edit Mode

Toggle Sketch into Edit Mode.
This mode gives access to nodes points allowing you to modify a curve.
See the Modes section for more details.

Selection Mode

Toggle Sketch into Selection Mode.
This mode allows you to select, resize, place, rotate objects.
See the modes section for more details.

Undo

Cancel the last operations. You can specify the number of cancels in the configuration options.

Redo

Sketch allows you to undo every operation and maintains a virtually unlimited undo history. If you really want to limit the undo history, you can do so from the preferences dialog (**File/Options...**).

Delete

Delete the current object (only in Selection Mode).

Duplicate

Create a duplicate of the current object just above the current object and select it. The duplicate is slightly offset. You can set the duplication offset in the preference dialog.
See also configuration options.

Effects

Vertical flip

Flip vertically an object or a group of objects.

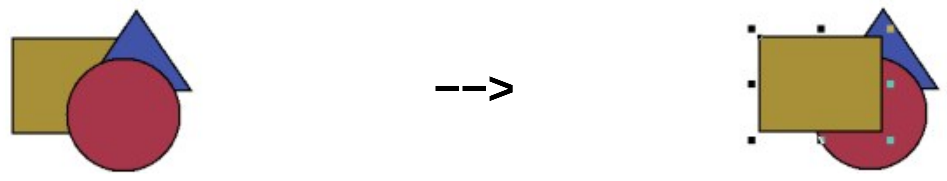
Horizontal flip

Flip horizontally.

Objects disposition

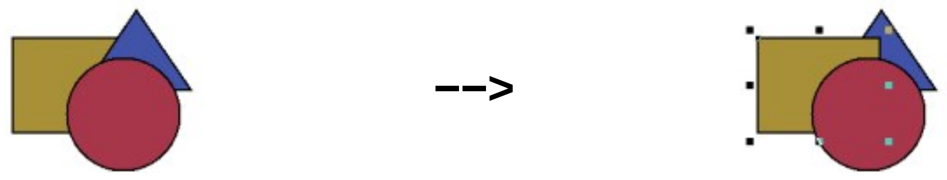
Move To Top

Move the current object to the top of its layer.



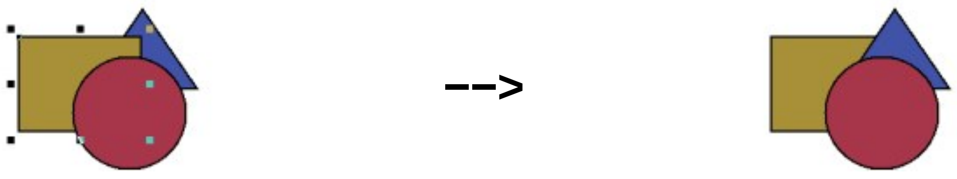
Move One Up

Swap the current object and the next higher one.



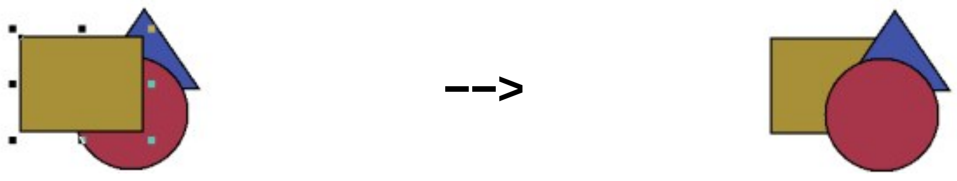
Move One Down

Swap the current object and the next lower one.



Move To Bottom

Move the current object to the bottom of its layer.



Grouping Objects

Group
Replace the currently selected objects with a group containing these objects and select it.

Ungroup
Replace the currently selected group with the objects it contains and select them.

Viewing Commands

Zoom
After invoking the zoom command you can either

- Zoom into a region. Click and drag **<Button1>** to indicate a rectangular area in the main window. This area will be magnified such that it just fits into the window.
- Double the magnification Just click (and don't drag) **<Button1>** on a point in the main window. This point will be centered and the magnification factor doubled.

The maximum zoom factor is 1600%.

Utilities

Snap To Grid
Turns the grid on (or off, if it's on). If gridding is on, objects will snap to grid points when they are edited.
This command doesn't affect the visibility of the grid. The visibility is controlled via the layer dialog. The grid is a special layer which is usually called "Grid".
See Layers section.

Objects creation

Draw a rectangle
Create a Rectangle by pressing the left mouse button and dragging the mouse. One corner of the newly created rectangle is where the button was pressed, the other where it was released. Holding **<Ctrl>** while dragging creates a square, holding **<Shift>** creates a rectangle or square centered on the starting point.

In Edit Mode, you can drag the corners of a rectangle to create rounded corners.

Draw an ellipse
Create an Ellipse by pressing the left mouse button and dragging the mouse. Holding **<Ctrl>** while dragging creates a circle, holding **<Shift>** creates a ellipse or circle centered on the starting point.
In Edit Mode, you can drag the handle of an ellipse to create arcs.

Draw a curve
Create a Bézier curve. This requires at least two click−drag−release cycles. The first cycle defines the start point of the curve and its tangent. The next cycles define the rest of the points and tangents in the same fashion. Click **<Button2>** or press **Space** to finish.

You can edit curves in Edit Mode and modify points with the curves menu or the Floating curves toolbox.

Draw a line

Create a polygon. The first click−drag−release cycle defines the first line segment. The next cycles define the rest of the segments. Click **<Button2>** or press **Space** to finish. The object created is actually a Bezier curve consisting only of straight lines.
Go in Edit Mode if you want to modify it.

Insert text

Create a text object by clicking where you want the text to be and type. If you click and drag you create rotated text.
You can modify the font in the floating fonts toolbox, you can transform the text into curves so as to modify it in Edit Mode, You can also apply special effects.

Insert an image

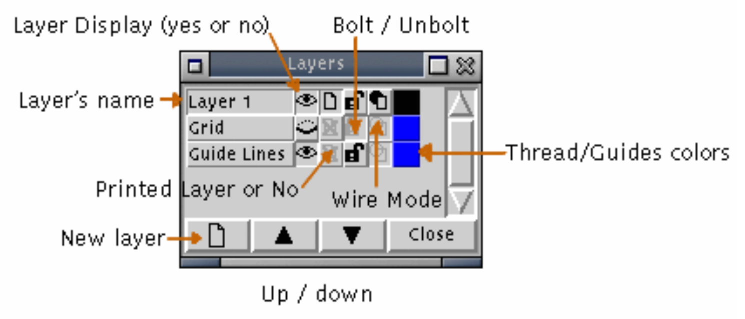
This command opens a dialog box to let you specify the image file. After selecting the file, place the image on the page by clicking at the desired position. A dashed rectangle indicates the size and position.

Sketch can load any image the Python Imaging Library (PIL) can read. EPS files are not read by the PIL, they are imported and printed unmodified. Sketch uses Ghostscript to render a preview image, so an EPS file looks like an ordinary bitmap image on the screen.
Supported formats are : Joint Photographers Expert Group (.jpeg, .jpg), GIF (.gif), Portable Bitmap (.pbm), portable Greymap (.ppm), portable pixmap (.ppm), Targged Image File Format (.tif, .tiff), windows, OS2 bitmap (.bmp), PCX (.pcx), Portable Network Graphic (.png).

Floating toolboxes

Layers

As bitmap softwares (The Gimp) You can work on different layers.
Here are the possibilities offered by this toolbox :



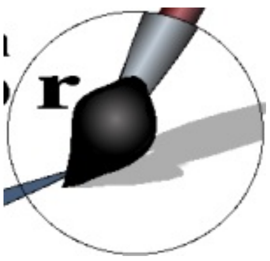
New layer :
Click on this button to add a layer; a dialog box appear and ask you a name for the layer. You can modify this name by double−clicking on it.
The layer you are currently working on is the one with a sunken button (layer 1 in the image above).
You can move them with the arrows.

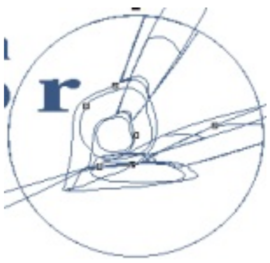
Eye :
The eye indicates to you if the layer is visible or not. It very useful when you work on complex

composition.
This button is also used to show/mask the grids and guides of the corresponding layer.

Bolt :
This function is used when you want to protect a layer. In this case, Edit and Selection Mode are unavailable.

Wire Mode :
The Wire Mode shows only the outline of the objects.
This can speed up the display without masking a layer.

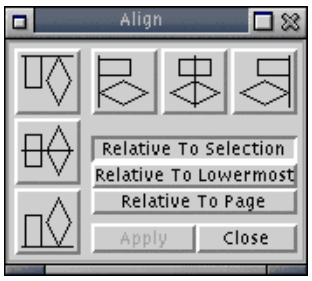




Thread/Guides colors :
The color shown in this box is the color used to draw threads and guides on the current layer.

Printed layer
You can specify if you want to print or no a layer. This works with Postscript export.

Align



If you want to align objects, you first have to select them.

On top horizontal alignment :
Objects aligned on their upper limit.

Centered horizontal alignment :
Objects aligned horizontally on their center.

On bottom horizontal alignment :
Objects aligned on their lower limit.

Vertical left alignment :
Objects aligned on their left limit.

Centered vertical alignment :

Objects aligned vertically on their center limit.

Vertical right alignment :

Objects aligned vertically on their right limit.

Relative to selection :

If activated, the alignment will be done relatively to the global selection (the frame around the objects)

Relative to lowermost

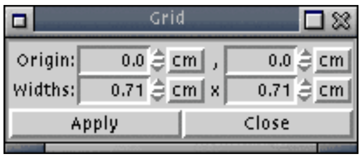
The lower object will be the reference.

Relative to page

The borders of the pages will be the reference.

Grid

This toolbox is used to configure the grid.
If you want to show/hide it, you must use the Floating layers toolbox.

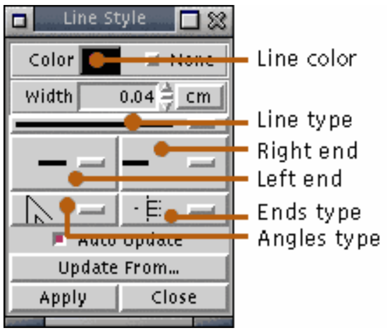

Grid

Origin: 0.0 cm , 0.0 cm
Widths: 0.71 cm x 0.71 cm

Apply Close

The origin parameter is the point of reference for the origin of the grid. By default its value is 0.
The spacing parameter modify the spaces between the grid's points.
The first column is for the X coordinates and the second for the Y coordinates.
By default, the grid is not magnetic, You can activate it with the Dispose Menu.

Line



Line color :

Click on this to change the color of the line. If "none" is activated, the line is transparent.

Width :

You can specify a value in cm, mm, inches, or points. To modify the unit, maintain the button hold down, a dialog box will appear.

Line type :

(continuous, dotted...)

Left / Right end :
Arrows, none...

Angles type :
rounded, flat...

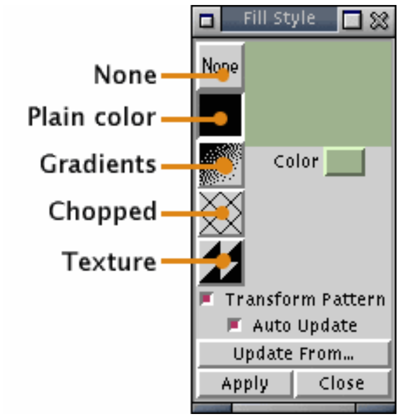
Ends type :
Flat, rounded, square...

Auto−update :
If activated, changes will be applied without having to click on "Apply".

Update from... :
This function, very useful, enables you to recover the style of an object so as to apply it on another one.
Just create a style, apply it to an object, create a second object and use this function to apply the same style to this object.

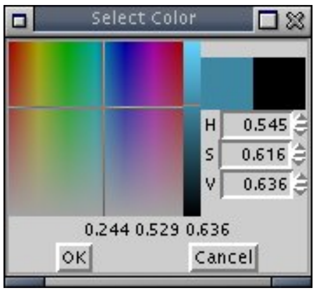
Filling up

All the parameters here doesn't apply any changes to lines.



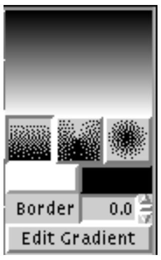
None :
The object will be considered as a transparent object (only lines or frames will be visible).

Plain color :
Click to change the filling up color.



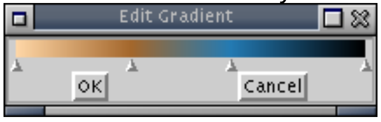
You can specify the corresponding value of the color you want.
If you doesn't want any lines around the object, use the floating line toolbox with none as parameter for "color line".

Gradients :



You can apply different sorts of gradation:
▦ Linear
▦ Conic
✺ Circular

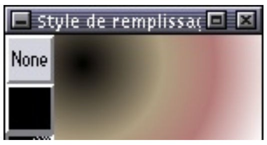
Choose the colors (black and white zones above)
"**Edit Gradient**" allows you to create complex gradations :

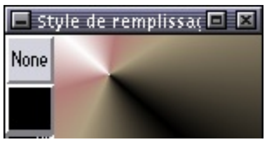


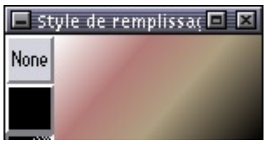
Create as many points as you want to mix colors and create your own gradation.
Right clicking on it opens the properties of the point.

The "border" button allows to create a border with the end colors of the gradation.

Sketch offers also the possibility to modify the orientation of the gradation.
In the case of a linear gradation, you can click and drag to orient the gradation.
If its a circular or conic gradation, you can chose the starting point.
Here are some examples :



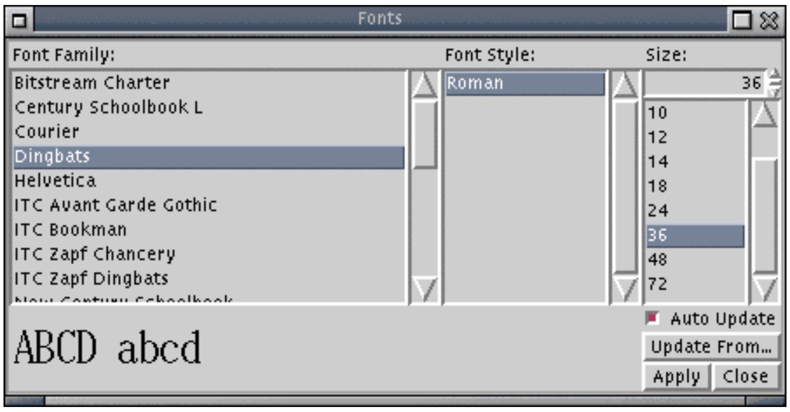




Chopped filling up :
You can the colors, the orientation (click, hold and drag) of the lines and the space between them.

Texture filling up :
You can fill an object with an image.
The "Adapt" option resize the image to fill up all the object.
The "Capture" button, allows you to "copy" an image (in your document) and use it to fill up an object.

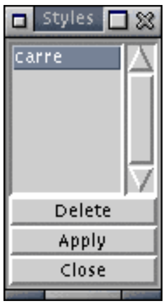
Fonts



Here you can choose a font and apply it.
If a text has been converted into curves, you can modify it; otherwise, you can do everything with it.
Note that Sketch doesn't use the same fonts as X server; if you want to add fonts to Sketch see the Fonts configuration section.

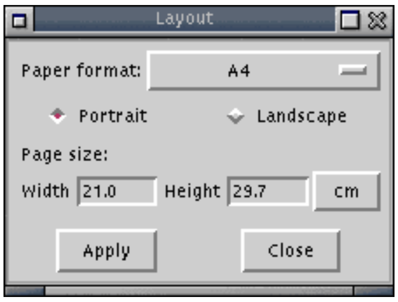
Styles

This toolbox helps you to manage styles created with "styles menu"−>Name of the style...



A style consists in parameters, applied to an object (line type, filling up type...) and that you can save.
You can apply it to others objects.
All the style created and saved appear in this dialog box so that you can select and apply them easily.

Make up

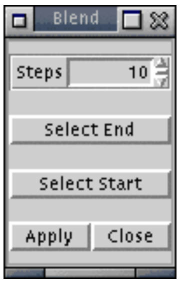


Paper format: A4
Portrait [✓] Landscape
Page size:
Width 21.0 Height 29.7 cm

Chose the page format (A3, A4 letterUS,...).
You can modify the orientation of the page, the width (largeur) and weight (hauteur) and even the unit.

Blend

More details can be found in the Special effects section.



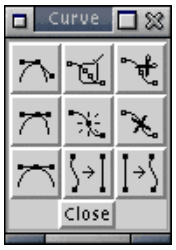
Steps 10

You can apply the blend effect only with 2 objects.
You can specify the number of steps.
When it's finished, you still have the possibility to modify the "First Element" and the "Last Element" by selected them.

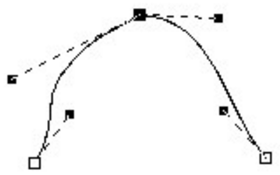
Curves

This toolbox is available only in Edit Mode.

Here are essentials functions to create/manage Bezier curves.

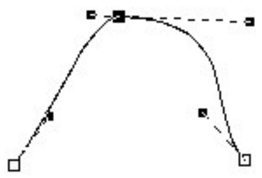


Asymmetric angle



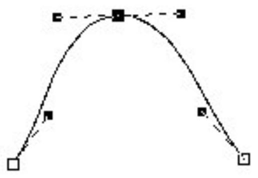
Allows to modify only one handful of a curve.

Semi−symmetric angle



Allows to modify only one handful but keep the symmetry.

Symmetric angle



Perfect symmetry on the size and orientation of a knot.

Join knots

You can join/merge 2 ends by selecting the concerned knots and applying this command. Essential if you want to close a curve.

Cut the curve

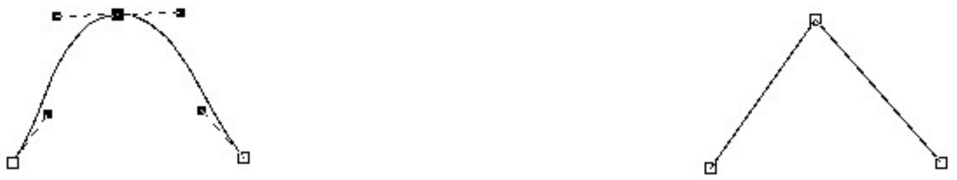
Select the point where you want to cut the curve and click !

Add a knot

First click where you want to place, a point appear, then use this command to create the knot.

Delete knots
Choose a knot and delete it.

Curve−−>Line
Allows you to convert a curve with Bezier knots into a line with straight angles.



Line−−>Curve
The opposite of the previous command.

Modes

2 modes are available in Sketch : **Edit** and **selection**.

The **selection mode** allows to move, resize, orientate an object. It is activated by default or by the button in the main toolbar. When you click on an object, little black squares appear around it indicating you are in the select mode and that the object is selected. If you want to learn how to manipulate objects, you can read the How to select and manipulate objects section.

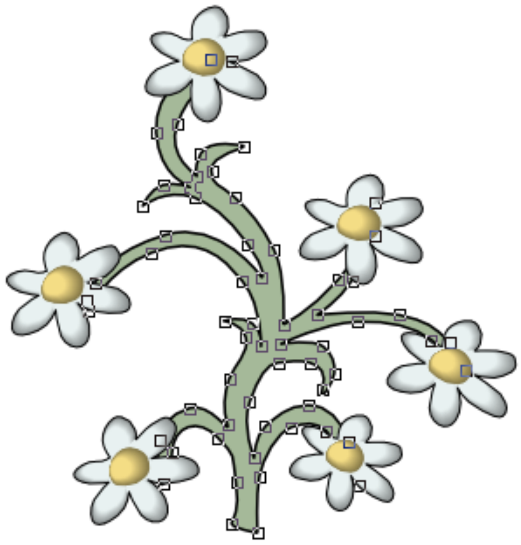
The edit mode allows you to modify the outline of an object. Note that the object must be selected. Then click on the button in the main toolbar. Every knot of the outline appear; you can click on them and modify them. If you want to learn how to modify tan outline, read the Objects edition section.

However it is advisable to read the following general principles:

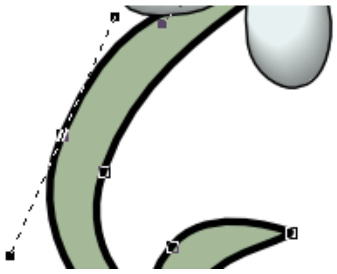
◊ You cannot activate the two modes at the same time. That means you cannot resize or orientate an object when you are in edit mode.
◊ You cannot edit a group of objects. You must ungroup them by "*dispose−>Ungroup*" and then modify them individually.
◊ If you want to merge objects so as to edit them as a single outline, first associate them by "*Curves−>associate Bezier*".
◊ You cannot edit text while it hasn't be converted into curves by "*Curves−>convert into curves*".
◊ You can quickly change the active mode with the space bar.

Objects Edition

The edit mode allows to modify the outline of an object.
You first have to select an object by the selection mode and then press on the space bar or the button of the main window.
When edit mode is activated, the outline, constituted of knots represented by little squares, is visible.



Move knots or modify them by their handfuls that appear when you click on it.
This handfuls only exist for Bezier curves and won't appear for angles.

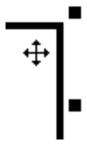


Sketch offers lots of possibilities to modify Bezier curves. Additional tools and information are available with the Curves menu and the Curves floating window. You will notice that a knot is represented by a black square when selected and a transparent one when not.
When Edit mode is activated, you cannot modify the scale and the pitch of the object (first activate the select mode).

Press on the space bar to quit this mode.

How to select and manipulate objects ?

Move, resize or orientate objects in Sketch is rather simple. When you click on a visible part of an object, little black squares (handfuls) automatically appear around the object. That means you are in **selection mode**.



The multiple arrows cursor indicates we can move the object. This function is available when you move upon a visible pat of the object. Click and drag the object wherever you want on the work area.

If you move the mouse on a handful, the cursor becomes a cross. If you click and hold down the button, you will resize the object by dragging the cursor. The object will be resized from the side you selected the black square. If you click on a handful in an angle, the object will be resize horizontally end vertically. If you want to modify the size without changing the height/width ratio, hold down Ctrl while resizing. Respect the following sequence :
Click −> Ctrl −> Move the cursor. Otherwise it won't work.

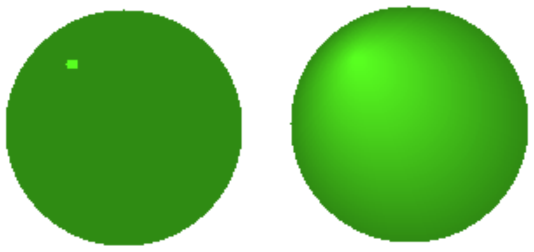
If you select an object by double−clicking, you will alternate between select/move move and rotate/incline mode. Your are in this mode when handfuls becomes arrows. At this moment, if you select an handful in an angle (the cursor change) you will rotate the object (note that the rotation angle is displayed in degrees at the bottom of the Sketch window). If you hold down the Ctrl button, the rotation will follow the magnetic grid which increase the precision.
If you select an handful on a side of the object, it will be inclined.

Press the space bar to quit this mode.

Special effects

Create a sphere

The method consists in using the "**blend**" function also called "**form gradation**" in a famous commercial software.
Create a circle without an outline filled with a solid color then another one smaller without an outline too and filled with a bright color (white is better). Place the little circle where you want the light impact point to be...
Select the 2 elements and launch "Blend" write 100 steps in the dialog box and click "Apply".
his is the impressive result :

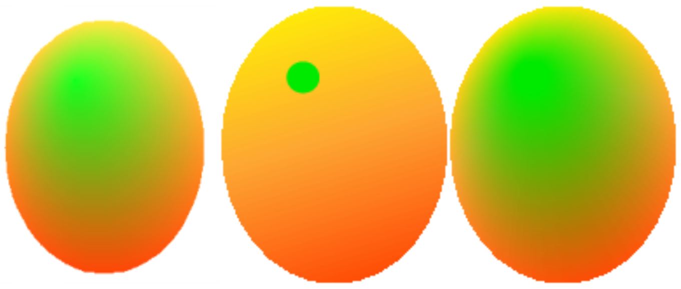


You can change the number of steps, use a gradation into the circles, modify the forms, or select one of the elements so as to apply on it a blend with a third.
Note that after a blend effect, the 2 elements are grouped (In fact it's 100 elements here that are grouped), and it is interesting to ungroup them so as to apply effects on them easily, or use de "deselect" function "Last/First Element" of the "Blend" dialog box.

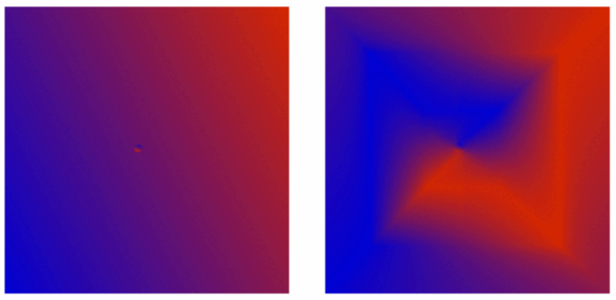
Next step : complex gradations...

Complex gradations

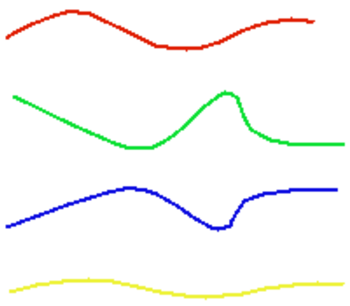
The different types of gradations here are impossible to realize with the classical tool "**gradient**", they use the "**Blend**" function.



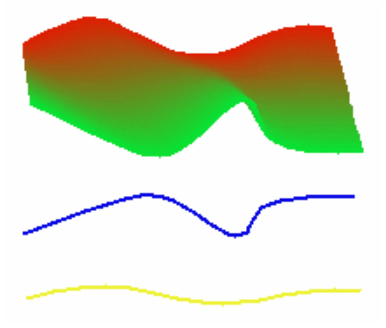
This gradation was obtained with the following method: create a gradation yellow/orange/red in an ellipse, place a little green circle on it, apply "**Blend**" with 100 steps.



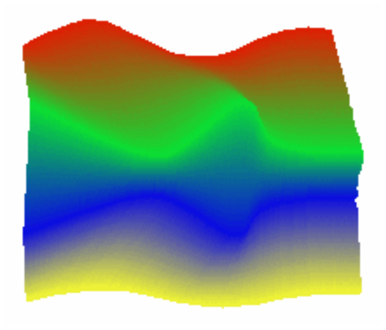
In this case, fill a square with a red/blue gradient diagonnaly oriented (blue in the left lower corner)
Place on this square a tiny circle filled with the same gradient but oriented at 180° (blue in the right upper corner).
Use "**Blend**" with 300 steps and see the result!



Thi effect needs curves with different colors.



We first apply "**Blend**" on the red and green curve

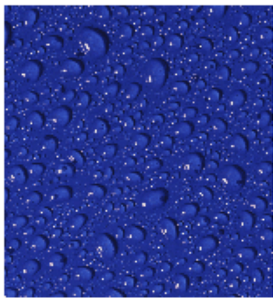


Then on all the curves. To select a curve that already belongs to a blend effect, use deselect function "Last/First Element" of the "Blend" dialog box.
Use your imagination with clors, curves, steps...

If you want to use this kind of gradation in an ordinary form, use "create a mask", see the Effect section of the "Menu rubric" and the next page.

A frame with a letter form

This effect is based on the "Effect" −> "**Create a mask**" command.



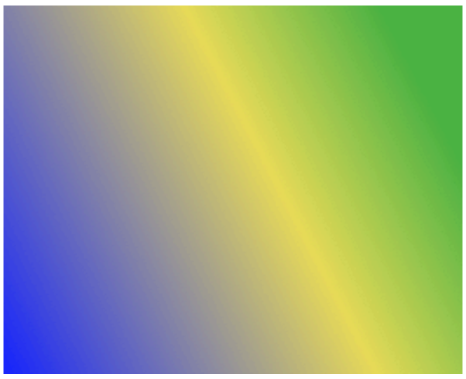
First insert a bitmap image as a background.

Insert a text (be careful not to write letters bigger than the background).

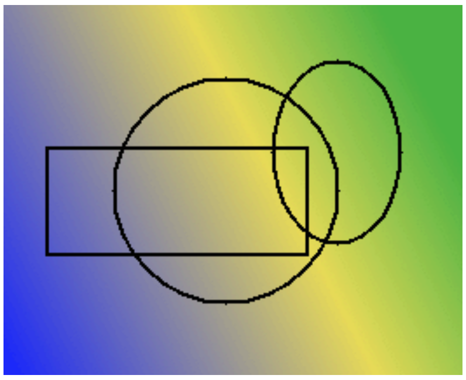
Select the 2 elements, launch "Effects" –> "**Create a mask**". Everything outside the letter disappears. Do not try to change the color of the letter especially if it bigger than the background (you may it for a special composition...)
You can for example let an outline to the letter (see the Effect section of the "Menu" rubric"). But it will work only if you convert the letter into curves before creating the mask ("Curves" –> "Convert into curves"). See also the Curves section of the "Menu" rubric) then ungroup the result so as to create the outline.

The "Transparent mask" effect

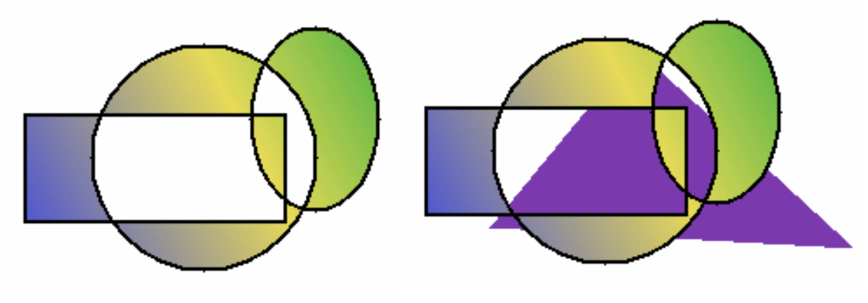
This effect is the combination of the "**Create a mask**" command and "**Associate Bezier**" command :



First, create a background.



Then draw 3 forms that overlap on the background.
Select each form and launch "Curves" –> "Convert into curves"



Now select all the forms (without the background) and launch "Curves" –> Associate Bezier".
At last, select all the elements and execute "**Create a mask**".

All the areas that have lost the background image are now transparent : you can verify it by lacing a colored element under it.

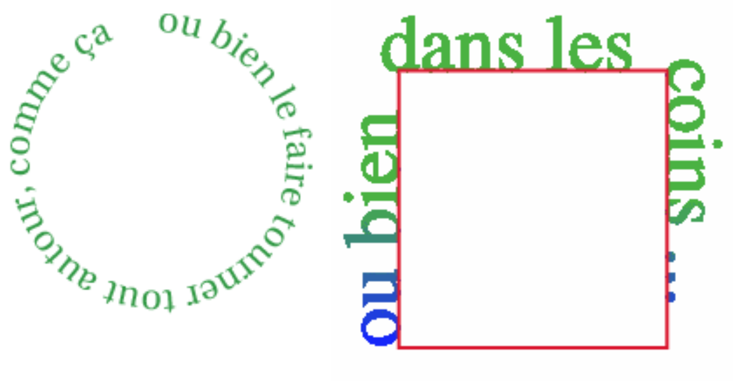
Text on path

The only thing you have to do is launching the following command : "Effects" –> "Create path text"

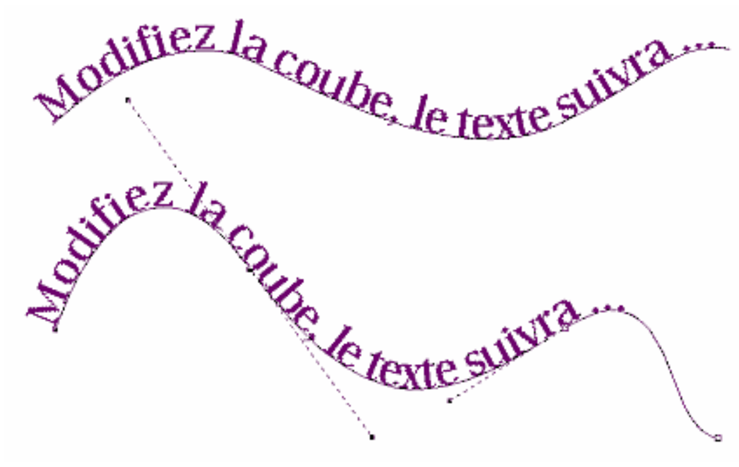
Here are some examples :

avec Sketch, on peut faire surfer du texte

Text on a path can have either rotated letters... or skewed letters. (change it with a right click on the path).



After its creation, you still have the possibility to modify, resize...
A right click enables you to select only the text or the curve so as to modify it.
In Edit Mode, you can modify the Curve with all the "Bezier tools":


Modifiez la coube, le texte suivra ...
Modifiez la coube, le texte suivra ...

In Selection Mode select the Path Text, with a right click do "Select the path", toggle in Edit Mode" (spacebar) and modify the curve.

Concepts

Predefined objects

Basic tools are :

Rectangle

Create a Rectangle by pressing the left mouse button and dragging the mouse. One corner of the newly created rectangle is where the button was pressed, the other where it was released. Holding **<Ctrl>** while dragging creates a square, holding **<Shift>** creates a rectangle or square centered on the starting point.
Both **<Ctrl>** and **<Shift>** while dragging creates a square centered on the starting point.
In Edit Mode, you can drag the corners of a rectangle to create rounded corners.
The objects created with this command must be converted into curves ("Curves" −> "Convert into Curves")

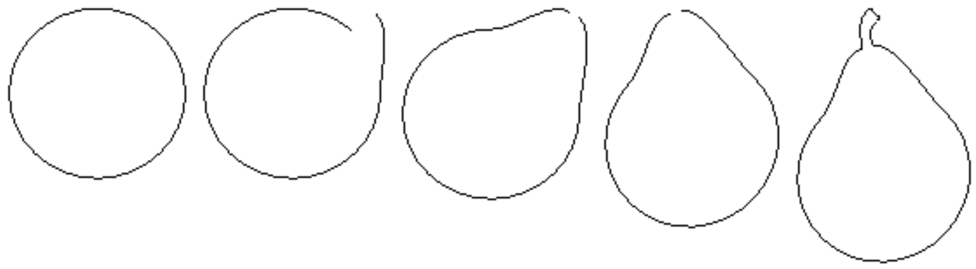
Ellipse

Create an Ellipse by pressing the left mouse button and dragging the mouse. Holding **<Ctrl>** while dragging creates a circle, holding **<Shift>** creates a ellipse or circle centered on the starting point.
In Edit Mode, you can drag the handle of an ellipse to create arcs.
Don't forget to convert into curves.

Curves (introduction...See next page for details)

Create a Bézier curve. This requires at least two click−drag−release cycles. The first cycle defines the start point of the curve and its tangent.
See also Objects Edition from the "Tools and commands" sections to learn how to manipulate curves and transform them.

Example:

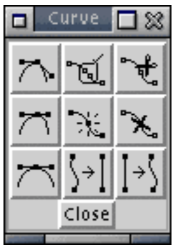


Free forms

Curves

Create a Bézier curve. This requires at least two click−drag−release cycles. The first cycle defines the start point of the curve and its tangent. The next cycles define the rest of the points and tangents in the same fashion. Click **<Button2>** or press **Space** to finish.

To modify an outline with this tool, you must be in Edit Mode (icon)



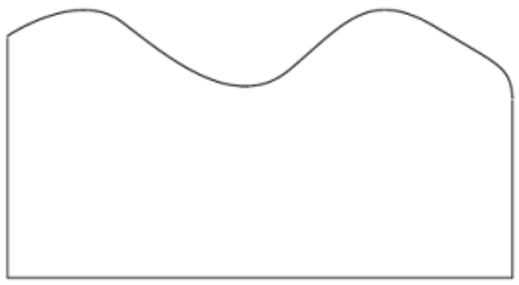
then use this dialog box.

First place a point, then a second holding down the button of the mouse you can drag it and create a curve with its tangent; the second tangent will be created with the third click.
This tool can be difficult when you begin with it but with a little practice you should be able to obtain good results.
See floating toolboxes section and toolbar for more details on Bézier curves.
See also Objects Edition from the "Tools and commands" sections to learn how to manipulate curves and transform them.

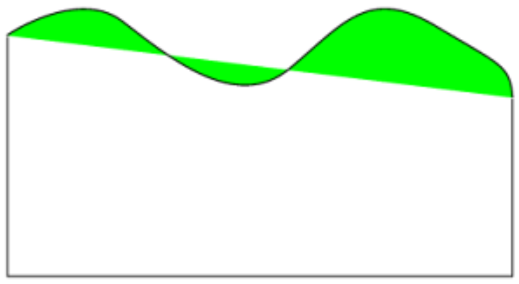
Polygon

Create a polygon. The first click−drag−release cycle defines the first line segment. The next cycles define the rest of the segments. Click **<Button2>** or press **Space** to finish. The object created is actually a Bézier curve consisting only of straight lines.
Holding **<Ctrl>** allows you to move around the centered point with 15°

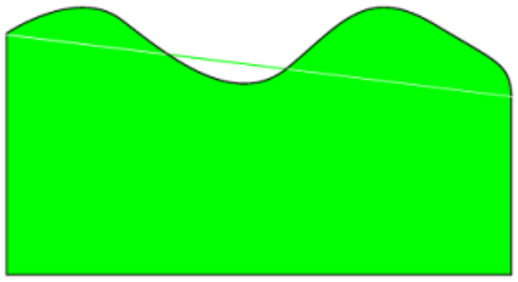
Close an outline
When an outline is complex, closing it can be better if you want to fill it:



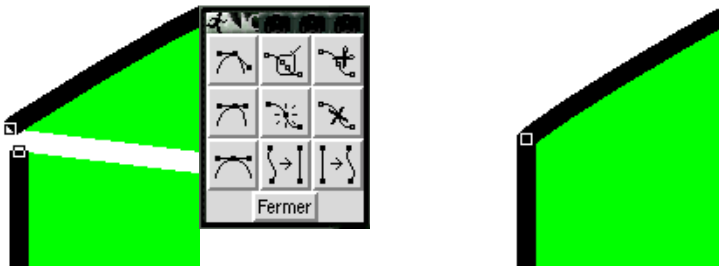
This is an object created this 2 tools: Lines and curves.
It is created with 2 parts.
This is what happen if you want to fill it with a color:



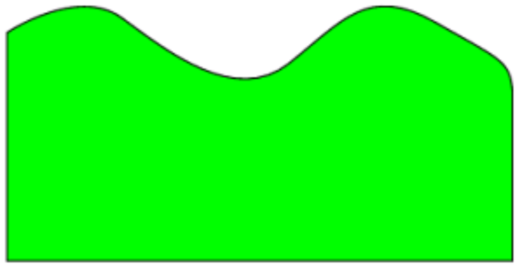
There is color outside the outline because it is not closed.
(Note that the problem remains if you try at the bottom of it)
One solution consists in use "Curves" −> "Associate Bezier". Here is the result:



Although it is possible to bring parts together so as to eliminate the little white and green lines, the result is not perfect.
The best way to proceed consists in associating the curves and join the knots:

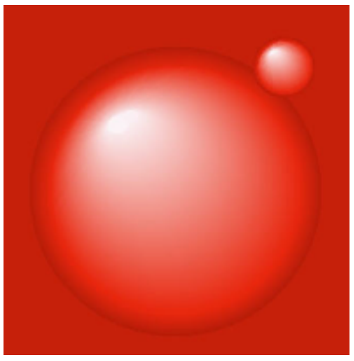


In Edit Mode, select the 2 knots (select the first one, hold **<shift>** and click on the second). In the floating toolbox named "Curves", select "join" (the central upper button.

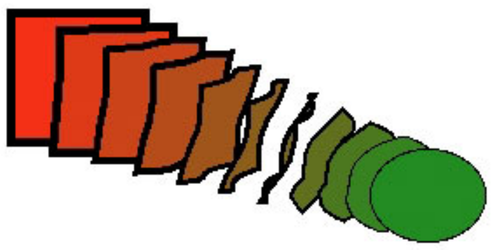


Here is our perfect and fillable form...

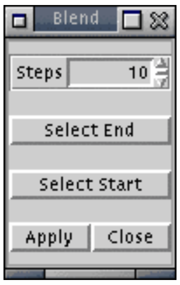
Transform to create



This effect can be obtained with the blend tool ("Effects" −> "Blend") also called "form gradation" in several commercial softwares.



Principle : The red square is transform into a green circle (its form and color are both transformed).



You must select 2 objects, then in the "Blend" dialog box, choose the number of steps that will be used to do the "mix". The more the number is high, the softer will be the effect. For example, setting it to , will show a "sphere" effect simply with a white circle above a colored one.

For the example above, the 3 spheres, we need to use "Blend" 3 times (select 2 circles, Transform, select 2 others, Transform...).
It is possible to modify it. You can change parts of it by selecting "First Element" or "Last Element" in the dialog box or by ungroup elements and redo the blend effect not only with the first and last element.

With this effect, you can create colors gradations that are impossible to realized with the "Gradation" tool..
See Complex gradations section of the "Special Effect" rubric.

How to transform a text into an object ?

T The text tool

Click on the icon and then in the work area to place the text at the cursor position.
Holding the mouse button allows you to pitch the text.
A right click makes the "fonts dialog box appear".
See the fonts rubric of the "floating toolboxes" section for more details.

Transforming a text into an object

This method consists into converting a group of letters into Bezier curves.
This will allow you to modify them as you want.
This transformation (also called "vectorization" in other softwares) has another big advantage: people who don't have the used fonts installed on their system, can even see the text (very interesting for multi−platforms works).
The disadvantage comes from the fact that a vectorized text is definitely fixed, you cannot correct it.

Example of a vectorization then a transformation.

Type the text, select it, click on "Curves" −> "Convert into curves" then if you want to edit it, click on "Dispose" −> "Ungroup"

You can only vectorize fonts Sketch recognize the .pfb file. You can lace it in fonts directory of Sketch : /usr/lib/sketch−0.6.5/Ressources/Fontmetrics (if you use .rpms) or modify the /usr/lib/sketch−0.6.5/Sketch/Base/config.py file in witch you must add the path to the .pfb files in the "font_path" section.
See fonts and adding fonts sections for more details.

How to transform and manage objects ?

Objects edition end transformation
2 work modes are available to modify objects :

− The Selection Mode : use

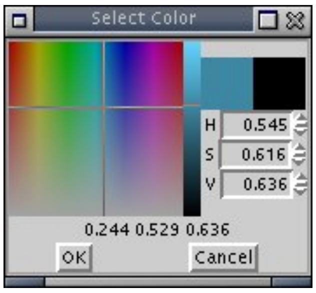
− and the Edit Mode : use
The space bar toggle from one to another mode. It's easier than the icons !

Se the Modes rubric of the "Tools and commands" section.

The **Selection Mode** enables placing, resizing, orientation of objects.
More details : How to select and manipulate objects rubric of the "Tools and commands" section.

The **Edition Mode** allows you to modify the outline and manipulate Bezier curves.
More details : Objects Edition rubric of the "Tools and commands" section.

Background and outline of an object
− In Sketch, an object is defined by a background and an outline. We can assign to them a color (or none), the transparency. There are characteristics that are specific to elements: thickness for the outline, the type, the end, a gradation for the background.



− To assign a color, a gradation, a design or a texture to the background of an object, you must select it, the you can even choose one of the colors at the bottom of Sketch or use the "Filling up" or "Filling up colors" floating toolboxes that appear with a right click (you can select "no filling up" too).
See Filling up section for more details on how to use this floating toolboxes

− You can modify the outline of an object with a right click (the "Line" toolbox appear) : you can specify value for the different options (thickness, color...).
See line section for more details on how to use this floating toolbox.

Grouped Objets

This icon is used to group 2 or more objects (use <shift> to add elements to your selection)

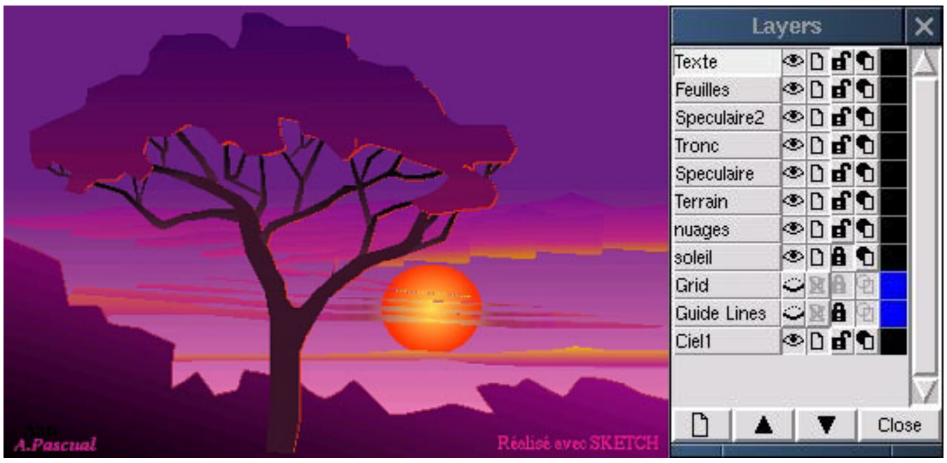
− When 2 elements (or +) are grouped, modifications are applied to both starting objects.
− If you modify the filling color of the background or of the outline, both will be changed.
If you want to modify them separately, you first have to ungroup them.

Dispose objects
Each object can be placed under or above another one.
More details can found in the Dispose rubric of the "Menu and toolbars" section.

You can also use layers for complex drawings:

Use of layers
Layers are transparent and are use to place objects above others.
Modifications made on a layer don't modify the others.
The layer toolbox enables you to modify the position, the hiding, the write protection of a layer.
You can also control the grid, the guides and offers a wired mode very useful for complex compositions.

In the following example, each object is placed on a layer.



See details of the layer toolbox in the "floating toolboxes" section.

The "Insert Image" tool
This tool opens a toolbox enabling bitmap image importation. Supported formats are : ps, eps, jpeg (jpg), gif, tif, bmp, pcx, ppm, pgm, pbm.
When the image has been selected and imported, its outline appears dotted. Place the image and click "ok" to validate.
You can modify the size, the position, ...in Selection Mode like a basic object.

Scripts

Sketch lets you write scripts to automate tasks and add new functionality. The programming language for these scripts is Python, an interpreted, object oriented language.

This chapter explains how to write such scripts and how they end up in the **Scripts** menu, so that you can invoke them. I'll assume that you already know how to program in Python. If you don't know that yet, have a look at Python's web−page for online documentation.

Some parts of this feature are still experimental and may change substantially in newer releases, so watch the NEWS file and the sample scripts in case this documentation is outdated.

Sketch itself is implemented almost completely in Python, so your scripts have access to all areas of the application, including internal data structures. This makes user scripts very powerful, but it also means that they can mess around with Sketch's internals with the result that Sketch might not be able to undo the changes or even save the document. But don't worry, evading the traps is not that difficult (I think) and this chapter tries to explain to avoid them.

Introduction

Safe and Advanced Scripts
As mentioned above user scripts in Sketch can cause data loss. To make things easier for you in this regard, Sketch supports two kinds of user scripts, *safe scripts* and *advanced scripts*.
Safe scripts take care of undo handling for you and they restrict what your script can actually do to keep you from harm. The disadvantage of this approach is that you can't do everything you might want to do with a safe script.
The advanced script has no restrictions but it has to take care of undo handling itself, which isn't difficult but has to be done with care.
The Script Function
Regardless of whether a script is a safe script or an advanced script it is just a Python function that accepts a context object as parameter. The context is an instance with three attributes:

document
The document for which the script was called.
The document contains all objects that make up the drawing and it manages the selection.

application
The application object.
The application object has methods for setting and retrieving the application's clipboard and to pop−up message boxes and file dialogs.

main_window
The top−level window containing the canvas widget that shows the current drawing. It has methods for loading and saving documents, among others.

A simple 'hello world' type script might look like this:

```
 def hello_world(context):
     context.application.MessageBox(title = "My Script",
                                     message = "Hello World!")
```

Registering Scripts

The **Script** menu just shows all the scripts in the script registry, so to have a script appear in that menu, you have to put it into the registry.
The registry is managed in a subpackage Scripting of the toplevel package Sketch. The function needed here is AddFunction and is called like this:

AddFunction(**name, title, function)**

name
is a name for that script. This name should be unique, as the scripts are identified in the registry by this name. It doesn't have to be the same as the function name.

title
The text of the menu entry.

function
the function that implements the script, e.g. hello_world above.

There are a few optional keyword arguments, one of which is type. The value of script_type must be either SafeScript (the default) or AdvancedScript. Both values are
defined in the Scripting subpackage. The registry functions are covered in more detail in the script registry section.

As an example, registering the hello world script might look like this:

```
import Sketch.Scripting
Sketch.Scripting.AddFunction("hello_world", "Hello World", hello_world)
```

First we import the package Sketch.Scripting and then we call the AddFunction function.
For more information about the organization of Sketch's code have a look at the API overview.

The Startup Script

The only thing left to do is to get some user supplied code to run on startup, so it can define and register scripts. To do that just create a file userhooks.py in the directory
~/.sketch/. If such a file exists it is automatically executed when Sketch starts. More precisely, the directory ~/.sketch/ is inserted at the front of Python's modules
search path and a module userhooks.py is imported.
The only thing left to do is to get some user supplied code to run on startup, so it can define and register scripts. To do that just create a file userhooks.py in the directory ~/.sketch/. If such a file exists it is automatically executed when Sketch starts. More precisely, the directory ~/.sketch/ is

inserted at the front of Python's modules search path and a module userhooks.py is imported.

You can put arbitrary code in there not just scripts, but one way to implement and register our hello_world script would be a userhooks.py file like this:

```
# example for ~/.sketch/userhooks.py

def hello_world(context):
    context.application.MessageBox(title = "My Script",
                                   message = "Hello World!")

# register script
import Sketch.Scripting
Sketch.Scripting.AddFunction("hello_world", "Hello World", hello_world)
```

The Example script

Sketch comes with a few example scripts in the Scripts/ directory. This directory works as a Python package, so to import a script, say abut_horizontal.py just execute import Script.abut_horizontal in userhooks.py.

Each of the modules in Script/ contains one script and registers this script when it is imported.

The script Registry

The script registry contains all user scripts and its contents are used to create the **Script** menu. The script registry is implemented in the (sub−)package Sketch.Scripting. Because the Scripting package is not automatically imported by the Sketch package, you have to import Sketch.Scripting explicitly before accessing the registry functions.

Functions in the registry package:

AddFunction(*name*, *title*, *function*[, *kwargs*])

Add the function *function* to the registry under the name *name*. *name* is used internally to identify the script and should be unique.

The parameter *title* defines the text for the menu−entry.

The function accepts these optional keyword arguments:

script_type = *type*
The type of the script. It can be either SafeScript or AdvancedScript. Both values are defined in the Scripting package. The default is SafeScript.
menu = *menu_spec*
Defines the sub−menu the script should appear in. The *menu_spec* can be either a string or a tuple of strings. Each string is the title of a sub−menu.
args = *argtuple*
Defines additional positional parameters passed to the script function after the context parameter. *argtuple* must be a tuple containing these arguments. It defaults to the empty tuple.

Example: Registering the abut_horizontal script in the sub−menu "Arrange"

```
Sketch.Scripting.AddFunction('abut_horizontal', 'Abut Horizontal',
                    abut_horizontal, menu = 'Arrange')
```

Sketch's API

If you want to write scripts for Sketch you have to know how to access the objects in the document and how to manipulate them. This section gives a brief introduction to the structure of Sketch's modules and objects from script author's point of view.

A more detailed description of Sketch's internals can be found in the Developer's Guide.

Module Structure

Sketch's code is organized in a hierarchy of packages. The top−level package that contains all of Sketch's core components is Sketch. It directly contains references to most of Sketch's graphics classes and functions.

Conventions

Sketch's objects have methods that are meant for public use and methods for internal purposes. Naturally, a script is expected to only use the public interface. Since Python has no builtin distinction between public and private/protected methods we have to rely on conventions.

Sketch uses a naming convention for this. Methods with capitalized names, e.g. "SetRadius", are public methods and Methods with lowercase name, e.g. "set_radius", are protected.

Undo Handling

As pointed out above, advanced scripts have to deal with undo information. This isn't difficult, but you have to know which methods return undo information and what you have to do with it.

You only have to deal with undo info if you modify an object that is part of a document. These objects include instances of classes derived from GraphicsObject (class hierarchy) and the objects that define the properties of a graphics object, like patterns.

All methods that modify such an object return an undo info object. What this object looks like is irrelevant here, all you need to know for now is that you have to pass it immediately on to the document method AddUndo.

A typical example:

```
context.document.AddUndo(object.Translate(offset))
```

The Translate method translates an object by offset, a point object that stands for a 2D−vector.

There are some exceptions to the rule that methods that modify the document return undo info.

Firstly, the document methods that modify the selection, that is, that modify which objects are selected not the selected objects themselves, don't return undo info because changing the selection is not considered changing the document.

Secondly, public document methods that modify the selected objects themselves already take care of undo info themselves. The reason for this is that they are called directly as a result of a menu command, button or key−press event.

For more information about undo information in Sketch, have a look at the corresponding section in the Developer's Guide.

Further Information

The example scripts in the Script directory contain extensive comments on what's going on.

The Developer's Guide covers Sketch's internals in more detail. Although it's incomplete it already contains a lot of information about the structure of Sketch's sources, the class hierarchy and some of the base classes, coordinate systems, plugins and more.

And, of course: Use The Source, Luke!

Common Tasks

Unless otherwise noted, all functions and classes mentioned here are directly accessible from the Sketch package. E.g. the CreateRGBColor function is accessible either with:

```
import Sketch

blue = Sketch.CreateRGBColor(0, 0, 1)
```

or

```
from Sketch import CreateRGBColor

blue = CreateRGBColor(0, 0, 1)
```

Creating Objects

Creating new objects for a drawing, usually takes three steps:

- Create the object. This is described in this section.
- Set graphics properties like fill color or line width. This is described below.
- Insert the object into the document

Primitives

Rectangles

This function creates a rectangle object described by *trafo*. *trafo* is an affine transformation (represented by a a transformation object that transforms the unit square into the desired rectangle.

Transformation objects are created by the functions Trafo, Scale, Rotation and Translation.

In the most common case where you want to create a rectangle with a specific width and height at a specific position (x, y) (the position of the lower left corner of the rectangle) and edges parallel to the edges of the page, you could create the rectangle like this:

```
Rectangle(Trafo(width, 0, 0, height, x, y))
```

or like this

```
Rectangle(Translation(x, y)(Scale(width, height)))
```

Lines and Curves

All line and curve−objects in Sketch are instances of the PolyBezier class. Each PolyBezier object consists of one or more paths with each path consisting of one or more line or bézier segments.

To create a PolyBezier object you have to first create the paths and then the PolyBezier instance. A path is created by starting with an empty path returned by the CreatePath function and then appending line− and curve−segments to construct the path. How paths are constructed is covered in detail in the corresponding section in the developer's guide.

Once you have constructed your paths, you create the PolyBezier instance with:

PolyBezier(*path_tuple*)

The argument *path_tuple* must be a tuple of paths. If you have just one path path, you can use the expression (path,).

For an example, see the sample script create_star.py.

Text

Sketch's text capabilities are still very simple. A text object consists of just one line of text in one font and size, hence the class is called SimpleText:

SimpleText(*trafo*, *text*)

Create a simple text object with the text string *text*. The position and orientation is given by *trafo*, a a transformation object.

For an example, see the sample scripts create_text.py.

Compound Objects

Groups

Creating a normal group is very simple. First, you have to construct a list of the objects to be contained in the group. Then you just call the constructor:

Group(*children*)

Create a group object with the objects in the list *children* as children. The children must not be contained in a document.

If you want to create a group of objects that are already contained in the document, the easiest way is to make sure those objects are selected and then to call the document method GroupSelected. Like all document methods this method takes care of undo itself.

Inserting the Object into the Document

You can insert an object into the document with this document method:

Insert(*object*)

Insert the object *object* into the document. The object becomes the topmost object in the current layer. The selection is set to just this object.

Manipulating Objects

Once you have created a graphics object or you have a reference to an object in the document, perhaps from the selection, you can manipulate them in various ways. All methods that modify an object in place return undo information which you have to pass to the document's AddUndo method if your script is an advanced script.

Transformations

Objects have two methods that allow you move and transform them.

Translate(*offset*)

Move the object by *offset*. *offset* must be a point object. For an example, see the sample scripts abut_horizontal.py and abut_vertical.py.

Transform(*trafo*)

Apply the affine transformation *trafo* to the object.

Changing Object Properties

Properties define how an object looks, i.e. they define the fill and line styles and fonts.

The fill is defined by a pattern object. There are several kinds of pattern types:

EmptyPattern

The EmptyPattern is a predefined object that means the object is not filled.

SolidPattern(*color*)

Return a solid pattern object for color *color*. The solid pattern fills the object uniformly.

color has to be a color object which can be created with the following function:

CreateRGBColor(*red*, *green*, *blue*)

which returns a color object for the color given by the RGB components *red*, *green*, *blue* which are floats in the range 0 − 1.

There are also some predefined color objects for some standard colors. They are attributes of the StandardColors object, e.g. StandardColors.blue is blue.

There are more pattern like gradients which I haven't documented yet.

To set the properties of an object, call this method

SetProperties(*keyword_arguments*)

Set the properties described by the keyword arguments and return undo information. The accepted keyword arguments are

fill_pattern

The fill pattern. It can be of any pattern type.

fill_transform

A boolean (i.e. 0 or 1) that defines whether the pattern is transformed as well if the object is transformed.

line_pattern

The pattern for the outline. Must be either EmptyPattern or a SolidPattern. If it's EmptyPattern, no outline is drawn.

line_width

The line width as a float in pt.

line_cap

The cap style. One of CapButt, CapRound or CapProjecting. These constants are defined in Sketch.const

line_join

The join style. One of JoinMiter, JoinRound or JoinBevel. These constants are defined in Sketch.const

line_dashes

The dash−pattern as a tuple of floats. The items of the tuple define the dashes and gaps terms of the line−width. The number of items in the tuple should be even.

line_arrow1
line_arrow2

The arrow heads. Arrow heads are objects you can create with the function Arrow.

font

The font for a text object. The value must be a font object which you can get with the function GetFont(*name*) where *name* is the PostScript−name of the font.

font_size

The size of the font in pt.

Managing the Selection

In Sketch 0.6.x, the selection is managed by the document object. You can use these document methods to query and change the selection:

HasSelection()

Return true if the selection is not empty.

CountSelected()

Return the number of selected objects.

SelectedObjects()

Return a list containing the currently selected objects.

CurrentObject()

Return the currently selected object if exactly one object is selected, return None otherwise.

SelectNone()

Make the selection empty.

SelectObject(*object*[, *mode*])

If *mode* is SelectSet, make *object* the selected object, otherwise, if *mode* is SelectAdd, add *object* to the selection. *mode* defaults to SelectSet.

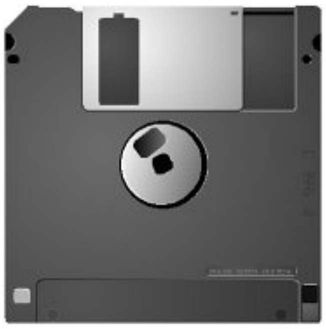
object may be a single object or a list of objects.

DeselectObject(*object*)

Remove *object* from the selection.

Tutorials

How to create a Floppy Disk ?

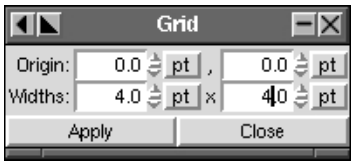


Presentation

The purpose of this tutorial is to show you how to create a floppy disk.
All the basic tools of Sketch are used.
Let's go !

By default, Sketch creates a A4 page.
With "F5", open the layers toolbox and verify that the grid is active (the eye in front of "Grid" is open)

Modify the grid step :

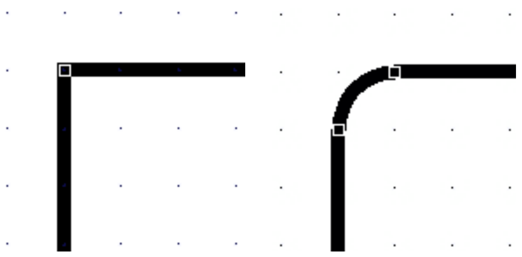


Grid

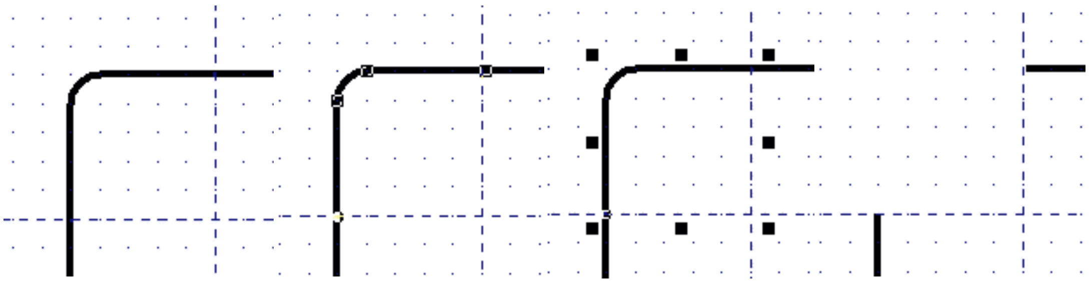
Origin: 0.0 pt , 0.0 pt
Widths: 4.0 pt x 4.0 pt

The body

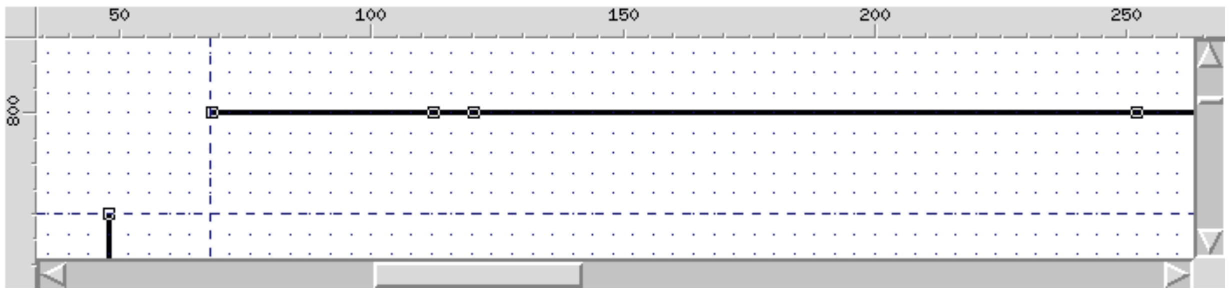
Magnetize the grid, then use the rectangular tool to create a square (hold <ctrl>) using the whole page : start from a point of the grid and finish to another one.
Zoom on the left upper corner; with the edit tool, click on the point at the angle and drag it to the center of the square so as to give the 4 corners a rounded aspect in the same time:



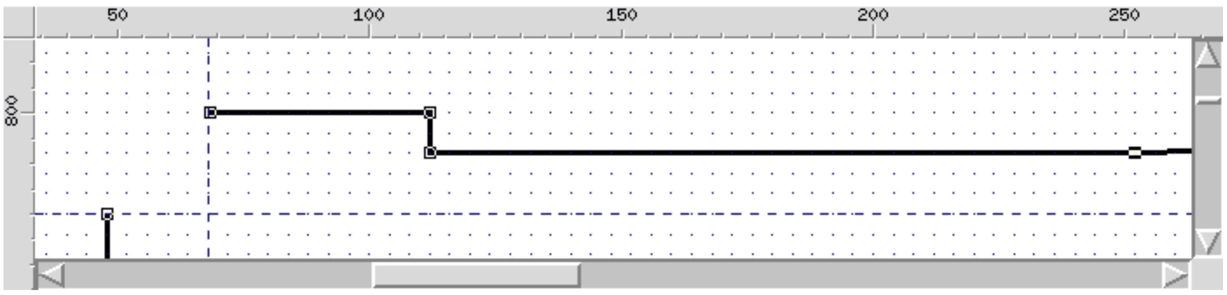
Always in the same angle, place marks at 5 units from the borders, horizontally and vertically :



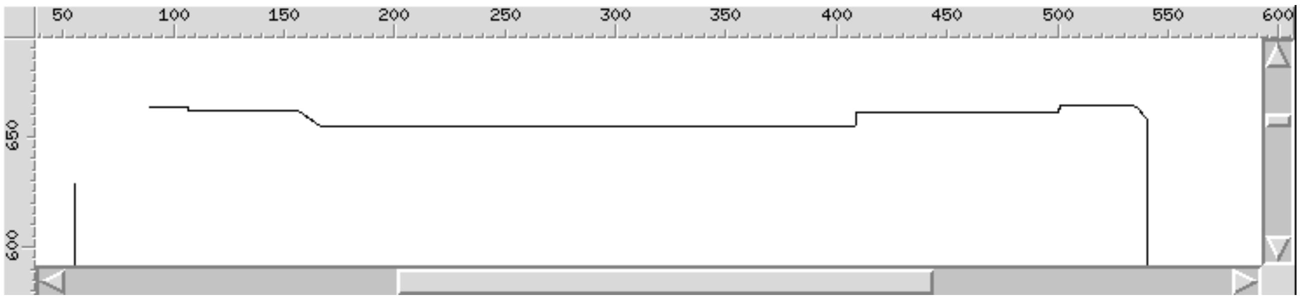
Lock the "Guide" calque (use the padlock).
Convert the object into Bézier curves (*"Curves" –> "Convert into curves"*).
With the Edit tool, click on one of the intersection between mark / outline, a dot will appear.
Click in the Curves toolbox on the "Cut the curve" button.
Do the same thing for the second intersection.
Then *"Curves" –> "Separate Bézier"*.
Select the rounded part and delete it.
You can close the outline but it's not necessary in that case.



Use "Insert a knot" (in the middle of the Curves toolbox) to add 3 knots to the outline.

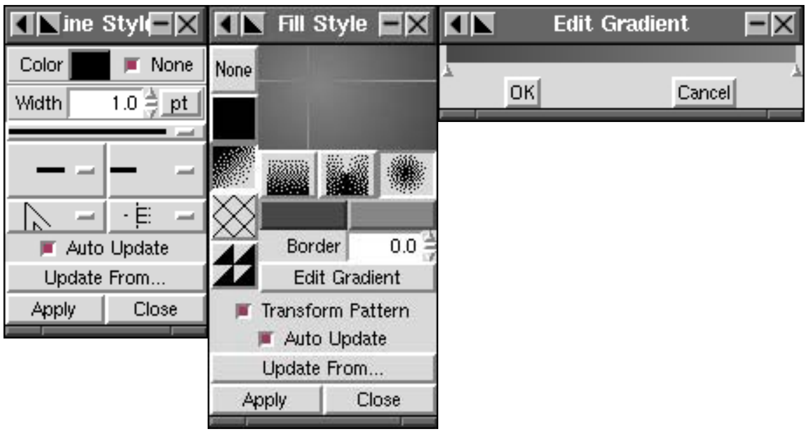


With the Edit tool, moves the points so as to obtain such a result :

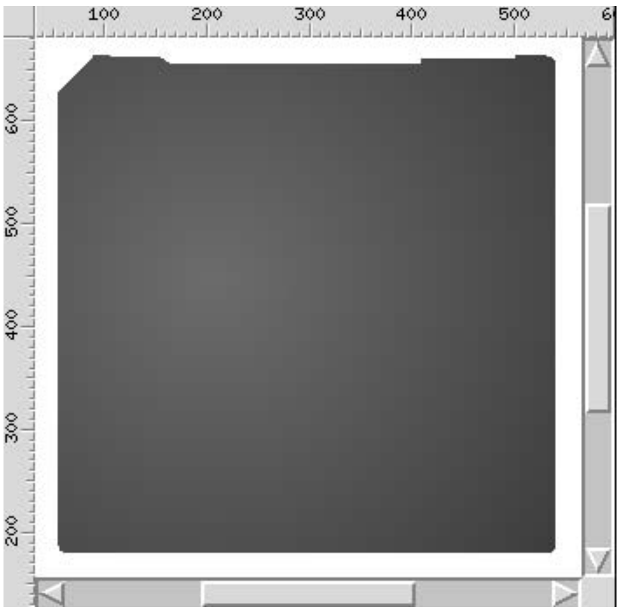


... The outline is finished...
Open the "Filling up" and "Line" toolboxes.

The outline of the floppy disk is selected; click on none in the Line toolbox

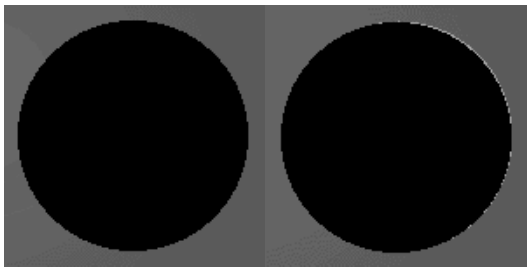


In the Filling up toolbox, select a circular gradient, then "Edit gradient".
Create you own gradient, then apply.
(you must select the object before ;−))

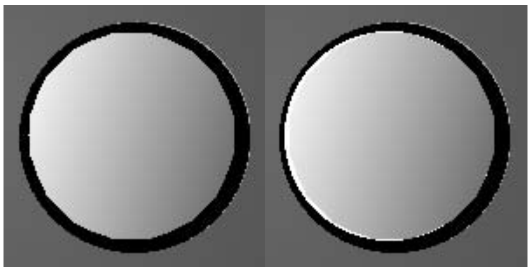


Lock the "Body" layer and create another one (the llower right button of the layers toolbox) for the center part.

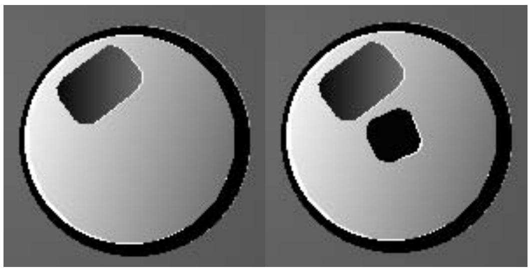
The axis



Draw a circle in the middle of the body (use the "Alignment" toolbox) and fill it with black
Duplicate the circle, fill the duplicated circle with a light grey.
Distort it a little bit so that you can see the black circle from the right side.
Then use the "Move One Down" tool to place the grey circle under the black circle : you have added a reflection.



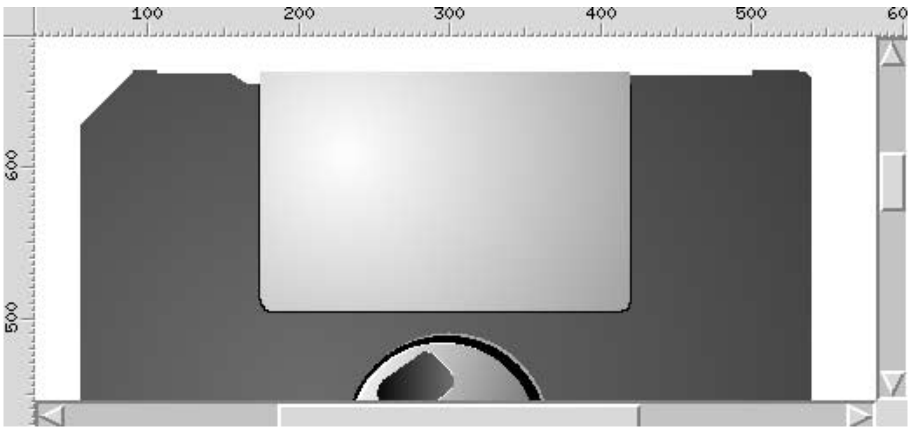
Duplicate the black circle a second time and reduce its size (don't forget to hold <ctrl>).
Fill it with a grey gradient.
Duplicate it, fill the copy with white, place it under the grey circle so as to see it on the left.



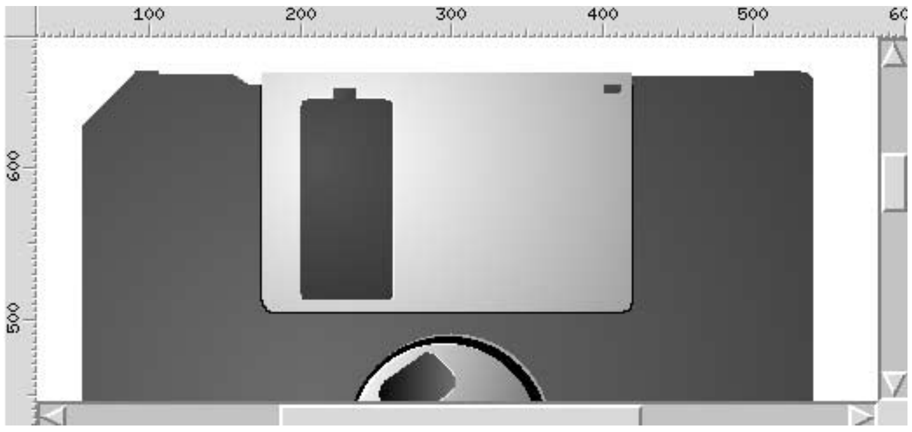
Then draw 2 rectangles with rounded corners (use the "body" method).
Fill the smallest with black and the other with a dark grey gradient.
Then ad the reflections (see above).

The sliding part

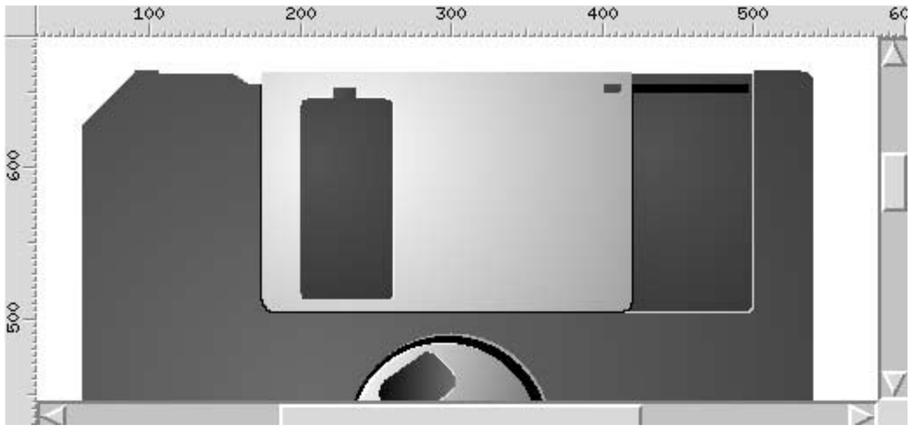
Create a new layer if you want...



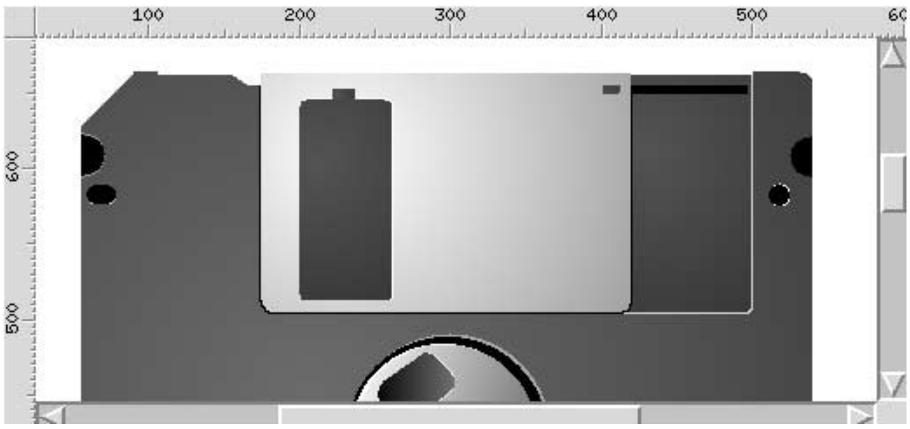
The sliding part consists in a rectangle with rounded corners only at the bottom :
Cut the body's curve to obtain it.
The metallic aspect, choose a circular grey gradient but place the starting point of the gradient on the left part of the rectangle.



Create the last cutting out parts and use the same color that the body to fill them.
Then add some reflections with white/grey filled duplicated parts.

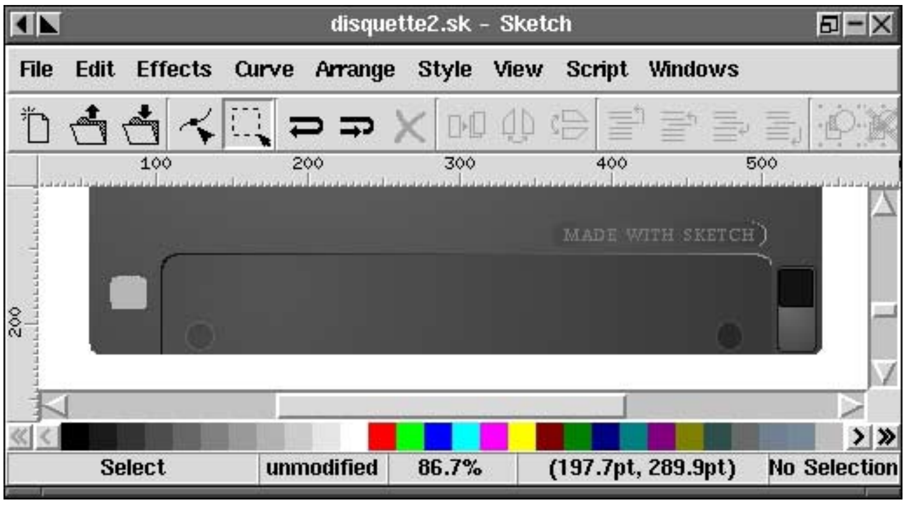


Duplicate the sliding part so as to create the hollow part and its light effect.
Then add the black slot.



Then add the last elements with circles, distorted circles and rounded rectangles.
A semicircle is obtainded by drawing a circle and, in Edit Mode, by modifying the only one point that appear.

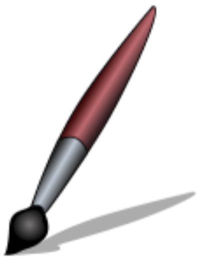
Bottom of the floppy disk and signature



Same method...
For the text, click where you want to place the text with the text tool, choose the font, the size...
The form around the text is a rounded rectangle.
The vertical text : *"Edit" −> "Create" −> "LCD Text"* , set you parameters and apply.
Double−click on it and use the arrows at the angles to rotate it.

Here is the result

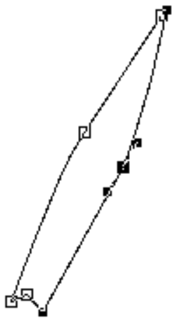
How to draw paintbrush ?



This tutorial will explain you how to create the SKetch Tutor Paintbrush.

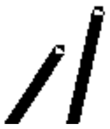
The body

We first draw a rough outline with the Bezier curve tool , note that to close the outline, you must place the last knot over the first one.

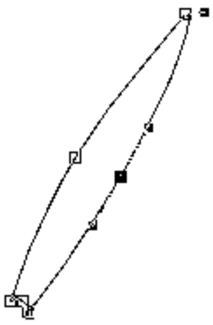


Then we will close the outline : first select th "Point" tool in the main toolbar , then click on the last point of the outline, then on the first one with the <shift> keyboard touch sunken.

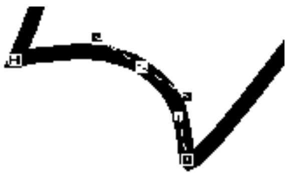
You can click on the zoom tool and drag to cretae a rectangular zoom area :



Then open the Floating curves toolbox (*"Windows"* –> *"Curves"*) and slect the "Join knots" command .

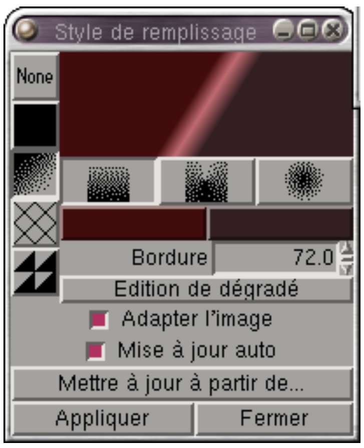


We need to sharpen the outline so as to obtain a rounded body.
Use some tools of the Floating curves toolbox so as to obtain the following aspect :

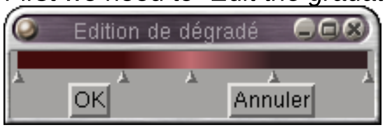


Filling up of the body

We want to create a color gradation s as to recreate a volume effect.
We can either use the blend effect or gradations.
Let use the second method.
Open the Floating filling up toolbox (*"Windows"* –> *"Filling up"*) and select the mode.
Try to obtain a similar aspect :

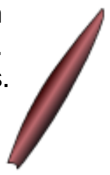


First we need to *"Edit the gradation"* .



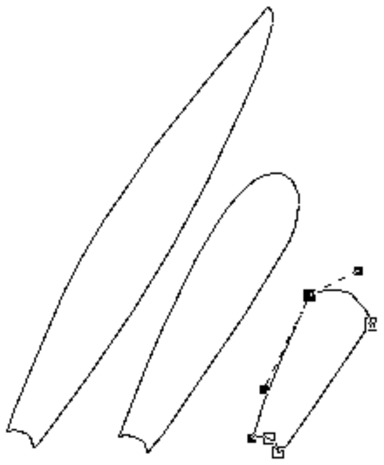
Click on the arrows at the ends to add points. Right click on a point and choose "Choice of the point color".

To orientate the gradation, click in the main window and drag to draw a line. The gradation will follow this line.
Find the right orientation for the paintbrush is not so easy, try several times.



The metallic part

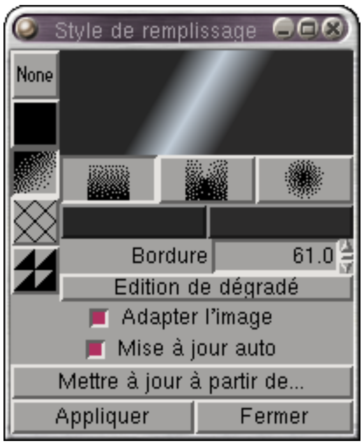
We duplicate and modify the body so as to keep the form.
Duplicate : *"Edit" –> "Duplicate", the duplicated object appear.*
In the filling up toolbox choose none for this object.
To edit the outline we must toggle into Edit Mode (spacebar).
The aim is to take the 2 lateral points down and to delete the point at the end.



We use an asymmetrical modification to the higher points to obtain this rounded form.
(refer to the third drawing that shows the edited points)

First select the appropriate button and then the point.
See the Floating curves toolbox for more details.

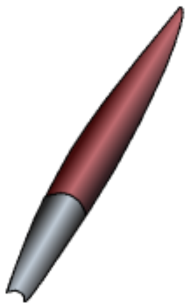
Then fill up the object with a gray gradation. You can reuse the previous gradation used for the body with the command called *"update from"* but don't forget to modify the colors.



Style de remplissage
None
Bordure 61.0
Edition de dégradé
Adapter l'image
Mise à jour auto
Mettre à jour à partir de...
Appliquer
Fermer

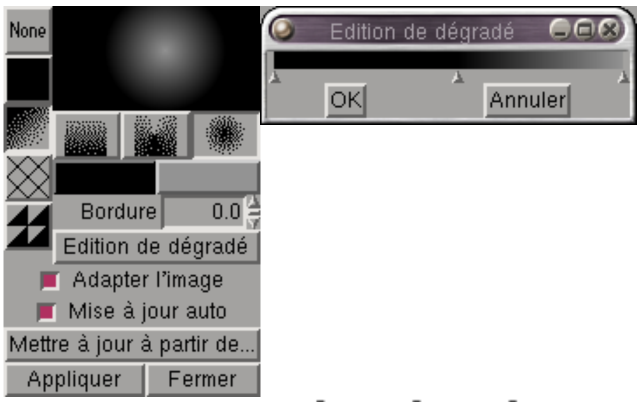
The head

At this moment you should have a similar drawing :

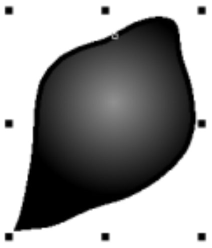


We start with an outline made with the Bézier curves tool , don't forget to close the outline by placing the last point over the first one and by joining them with the join command :

To create the volume effect, will use a radial gradation from the center of the object:
See parameters used :



>>>>>>

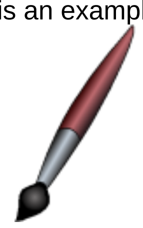


Note that the "Head" created must be placed with the Position commands.

Dropped shadow

Here is an example of what you should

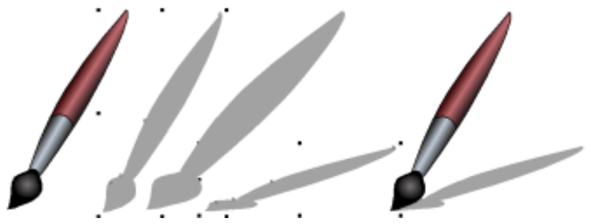
have:



We need to select all the objects we created (body, metallic part and head) and group them (*"Dispose Menu" –> "Group").*
Then we duplicate the group (*"Edit Menu" –> "Duplicate"*).

Select the duplicated group and open the floating filling up toolbox.
Select a gray and apply it.
For the outline, open the Floating lines toolbox and indicate you don't want any outline (don't care if the option is still activated, just click apply it will work !).
Now we can distort the shadow.

Enlarge it a bit from a side (see How to select and manipulate objects ?).
Resize it to simulate that the shadow is plane and place as shown above :



You have finished.

The next step shows you how to transform your image into the bitmap format, keeping the quality of course.

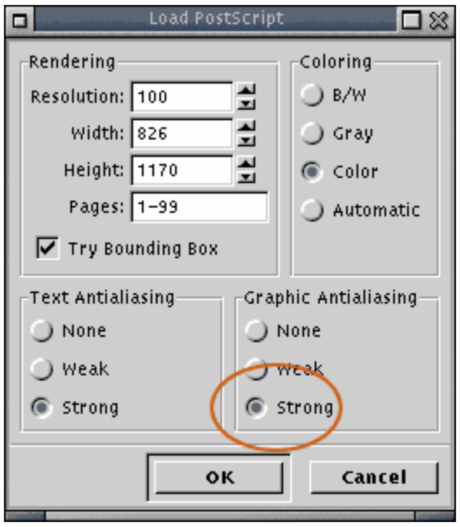
Import with The Gimp

You can save your work under the **.sk** format but only Sketch will be able to open it.
If you want to use it on a web page, you have to convert it into bitmap format.

Sketch can save it into Postscript format : *"File" –> "Save as Postscript"*

You recover your Postscript file with The Gimp and save it into a bitmap format (jpeg, png,...note that a license must be purchased to use gif)

In The Gimp : *"File" –> "Open"* your **.ps** file.
The Gimp shows you several parameters for the importation:



Load PostScript

Rendering
Resolution: 100
Width: 826
Height: 1170
Pages: 1-99
[✓ Try Bounding Box]

Coloring
B/W
Gray
[✓ Color]
Automatic

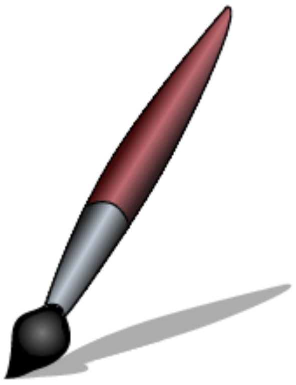
Text Antialiasing
None
Weak
[✓ Strong]

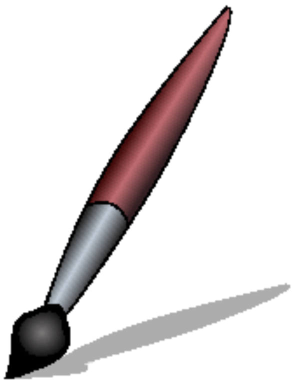
Graphic Antialiasing
None
Weak
[✓ Strong]

OK Cancel

If you want a perfect and smooth outline, choose *"High"* for the *"Graphical smoothing"* parameter.

Below are two images with a different level for this parameter :





Now you can save your work in jpeg or png format :
Right click and *"File"* −> *"Save as"*.
Choose the correct extension and its finished !!!