

# MightyScape Extension Collection

- [FabLab Chemnitz - Modify existing Path\(s\)](#)
  - [Convert Vertical/Horizontal To Line](#)
- [FabLab Chemnitz - Paths - Join/Order](#)
  - [Longest Continuous Path](#)
  - [Reverse Order Of Subpaths](#)
  - [Move Path Node](#)
  - [Duplicate + Reverse + Join](#)
  - [Circle Tangents \(Replaced by Snap Objects Feature\)](#)
  - [Chain Paths](#)
- [FabLab Chemnitz](#)
  - [Layer Clip](#)
  - [About/Upgrade MightyScape](#)
  - [Table Support](#)
  - [Open Extension Directory](#)

- Reload
- PathOps
- Simple Inkscape (Interface)
- Ray Tracing
  
- Integrated into default Import/Export Menu
  - Encapsulated PostScript - AI compatible (\*.eps)
  - FreeCAD Sketch Export
  - Photoshop PSD (\*.psd)
  - WebP Import
  - Roland CutStudio \*.eps Export
  
- FabLab Chemnitz Boxes.py
- FabLab Chemnitz Shape Generators
- FabLab Chemnitz Boxes/Papercraft

# FabLab Chemnitz - Modify existing Path(s)

# Convert Vertical/Horizontal To Line

This extension converts an SVG path's d attribute the following way: find each V (vertical line) and each H (horizontal line) and replace it by a generic line (L type). A lot of extensions do not work with V and H, but with L commands. So this is just a helper extension for other extensions. It behaves similar to To Absolute (Lächeln)

## Example conversion

from:

```
M 60.403,71.0937 V 89.268022 135.773
```

to:

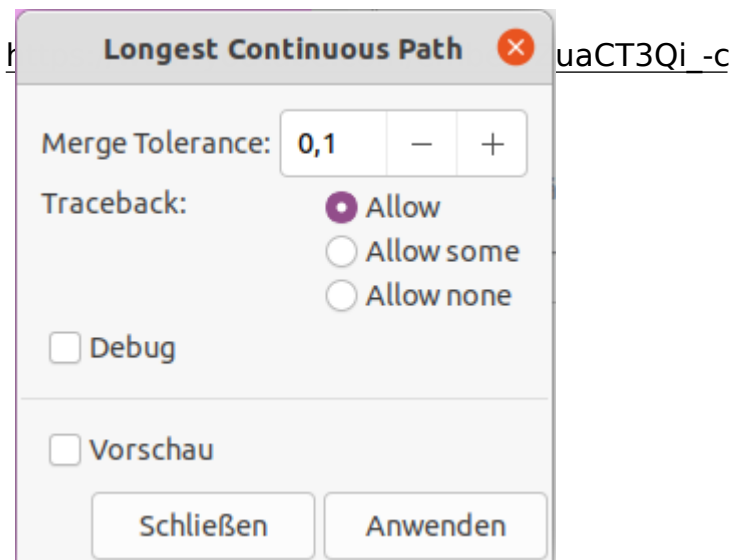
```
M 60.403 71.0937 L 60.403 89.268 L 60.403 135.773
```

# FabLab Chemnitz - Paths - Join/Order

# Longest Continuous Path

Source of documentation: <https://github.com/Daekkyn/inkscapeOptimizePath>

An Inkscape extension that tries to make the longest continuous paths. It converts the paths to a graph, converts the graph to an Eulerian graph and finds an Eulerian cycle. It was written for the Axidraw pen plotter to reduce writing time of graph-like drawings like Voronoi diagrams and meshes.



## Usage

1. Make sure to ungroup everything → Extensions → Reorder → Deep Ungroup
2. Make sure that the paths use only absolute coordinates (see trick below) or use → Extensions → Modify Path → To Absolute
3. Make sure to not have transforms on the paths. You can use the Apply Transform extension to remove them
4. Select all the paths you want to optimize (currently only works with polylines). Conversion to polylines can be done with Approximate Curves by Straight Lines (Flatten Beziers) or Convert To Polylines (as second choice). Bezier curves, which were not converted, are getting **lost** running this extension
5. Open the extension

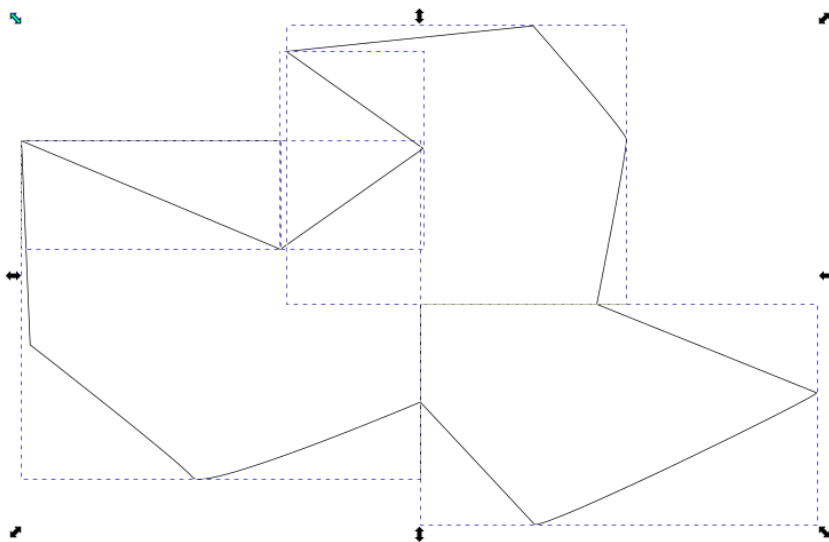
6. Set the merge tolerance (0.1 should work in most cases)
7. Choose the Overwrite rule:
  1. "Allow" means that the result will be a single path which might (probably) will have some overlapping edges.
  2. "Allow none" means that the results will be multiple disconnected paths but there will be no overlapping edges.
  3. "Allow some" is an in-between, overlapping edges are allowed, but only in short numbers. This is probably the best choice in most cases.
8. Apply

### Trick to remove relative coordinates

1. Change the preferences for 'SVG Output > Path Data' to always use absolute coordinates. This will only affect newly created paths, or existing objects for which a rewrite of the path data is triggered.
2. For existing paths, use 'Edit > Select All in All Layers', and nudge the selection with the arrow keys (e.g. one step up and one back down again). This will trigger a rewrite of the path data in 'd' which will follow the changed preferences for optimized path data.

## Another Example

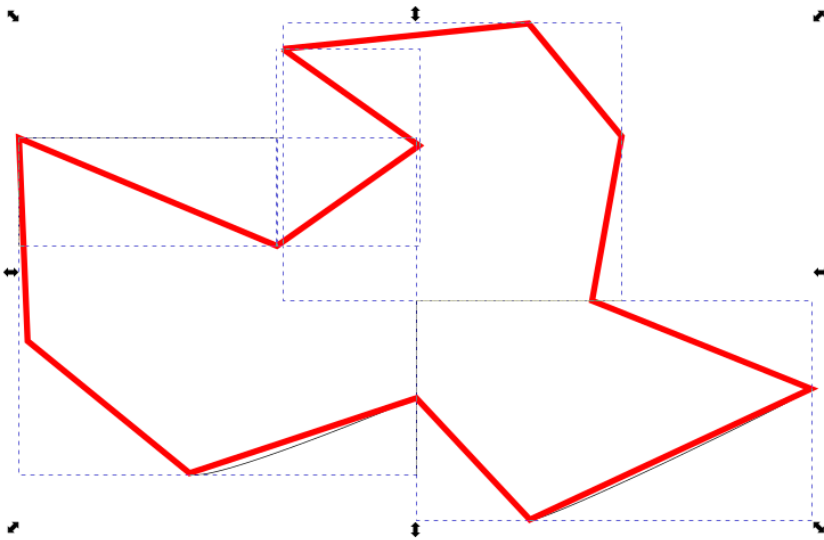
1. **Draw some paths and put the ends together by drag and drop with snap**



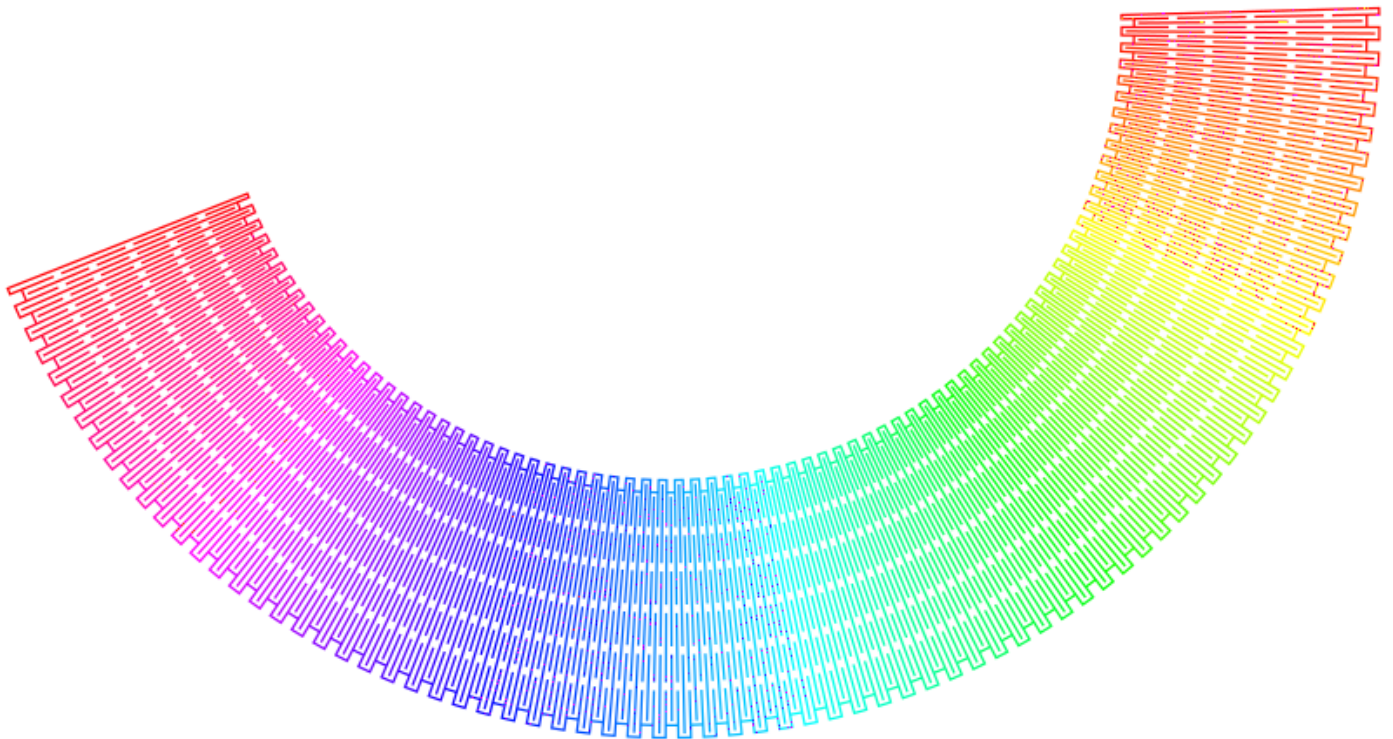
### 2. Get the result

The red contour is an optimized contour. It works similar like [Close Paths](#), [Chain Paths](#) or [Join Paths / Create Tabs And Dimples](#) except that it will not handle bezier curves but only polylines. So it acts like a combination of those plugins in junction with [Approximate](#)

Curves by Straight Lines (Flatten Beziers) or [Convert To Polylines](#).



This extension can also create really cool artwork





# Reverse Order Of Subpaths

Reverse order of subpaths (combined paths) without changing the direction of each subpath.

When drawing paths and combining these the order of the subpaths is from the last to the first drawn. This extension reverse the order of the subpaths and thus the order in which the paths were drawn can be restored.

Use together with the extension to [Number Subpaths](#) to visualize the effect of reversing the subpaths.

See <https://inkscape.org/~EllenWasbo/%E2%98%85reverse-order-of-subpaths> and <https://gitlab.com/EllenWasbo/inkscape-extension-reverse-order-of-subpaths>

# Move Path Node

Extension to change starting / end node of a path and visualize it by dots and numbers. You can also use this extension as a trimmer for open paths. It is additionally highly useful when using the function to map text to path. By moving the node, the mapped path is moving too. It helps for fine-tuning a lot.

Modify Path Start Node

Modify Path Start Node

About

Donate

☐ Handle closed paths only

Move starting node n nodes further

3

–

+

☒ Visualize first two nodes

Font size:

30px

Dot size:

10px

☐ Debug Output

**Notes**

Use "Path > Reverse" to change path direction

Use extension "Chain Paths" to make closed paths out of segments.

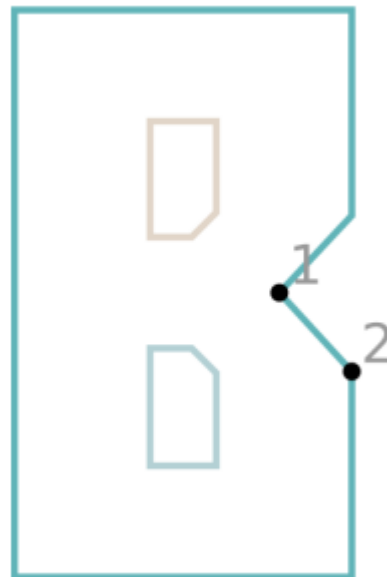
☐ Live preview

Close

Apply

direction

closed paths out of segments.



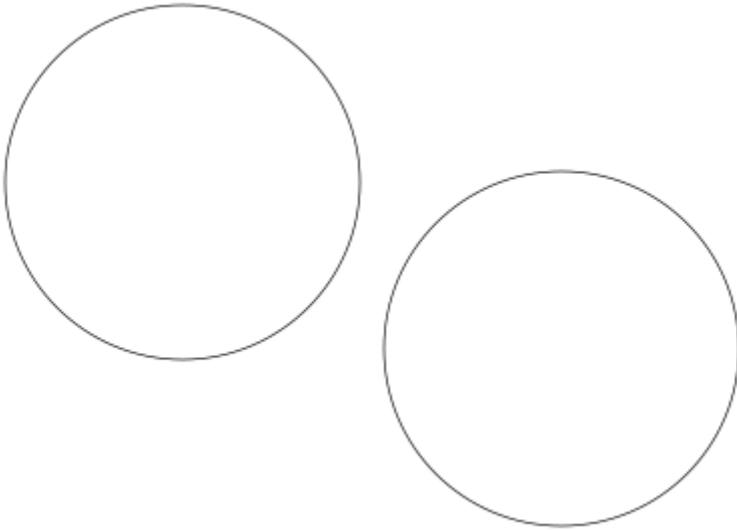
# Duplicate + Reverse + Join

An extension to effectively convert a single line svg font to a stick font by duplicating, reversing and joining each selected path and subpath. Read more on

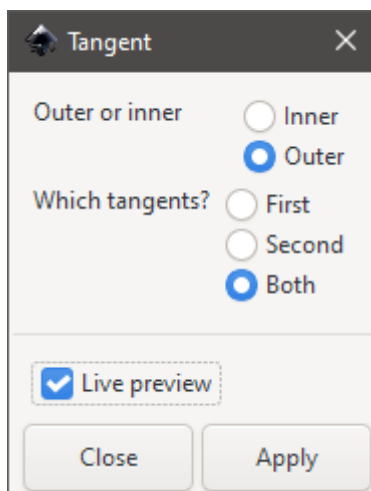
<http://cutlings.wasbo.net/single-line-font-to-stick-font>

# Circle Tangents (Replaced by Snap Objects Feature)

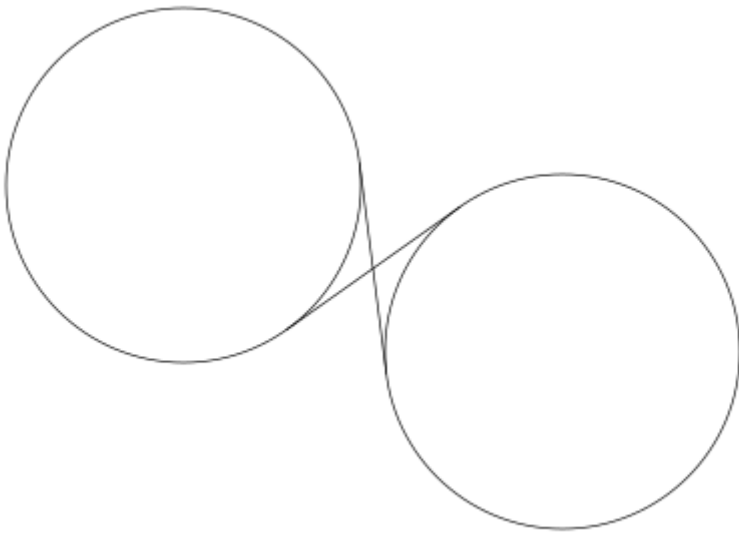
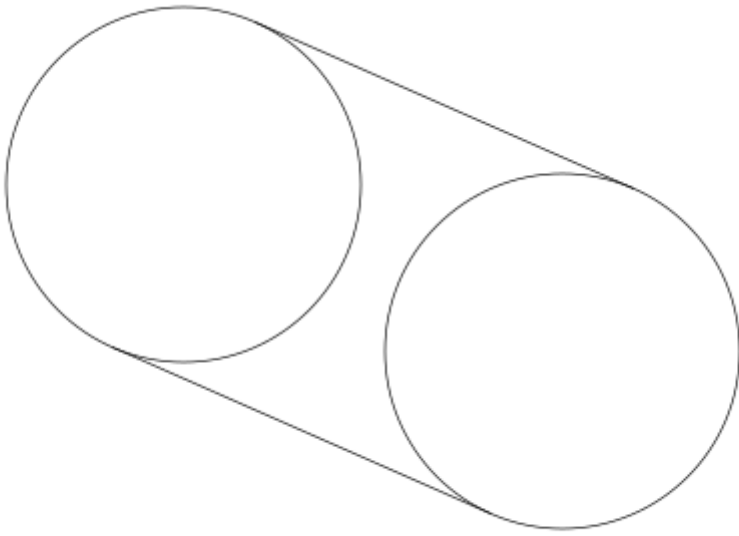
## 1. Draw some circles



## 2. Apply the extension



You can have two different outputs:



If you want to get sort of tangents at the path's point handles instead you can also try to run the extension [Draw Handles](#)

# Chain Paths

Source of documentation:

<https://github.com/fablabnbg/inkscape-chain-paths>

An inkscape extension to combine paths. Like really combining path snippets into longer paths. The stock inkscape path operation "combine" does not do that. It only creates a single path object consisting of multiple distinct segments. Many commercial CAD packages create object contours consisting of separate path snippets using adjacent end points. Such objects cannot be used in path operations like "add", "intersect", "difference", as they are technically a set of objects, rather than a single stroke. This extension forms a longer path from multiple shorter path segments. It is irrelevant if the path segments are separate path objects in Inkscape, or if the path segments belong to the same path object. If two path segments have an end point in common, or if their end points are close together, they are linked together to form a longer path. It is optional whether the linking end points meld into a single common point, or if an optional straight line ('chain link') fills the gap, if any. The maximum distance for end points is a user setting. Usually a fraction of a millimeter works fine.

## Usage

Select multiple pathlike objects. If the status line shows you different object types, then use "Path → Object to Path". This is needed as we operate only on paths only. You can choose the maximum endpoint distance for path ends to be linked, and the combination method: snap the points together, or create a linking path segment.

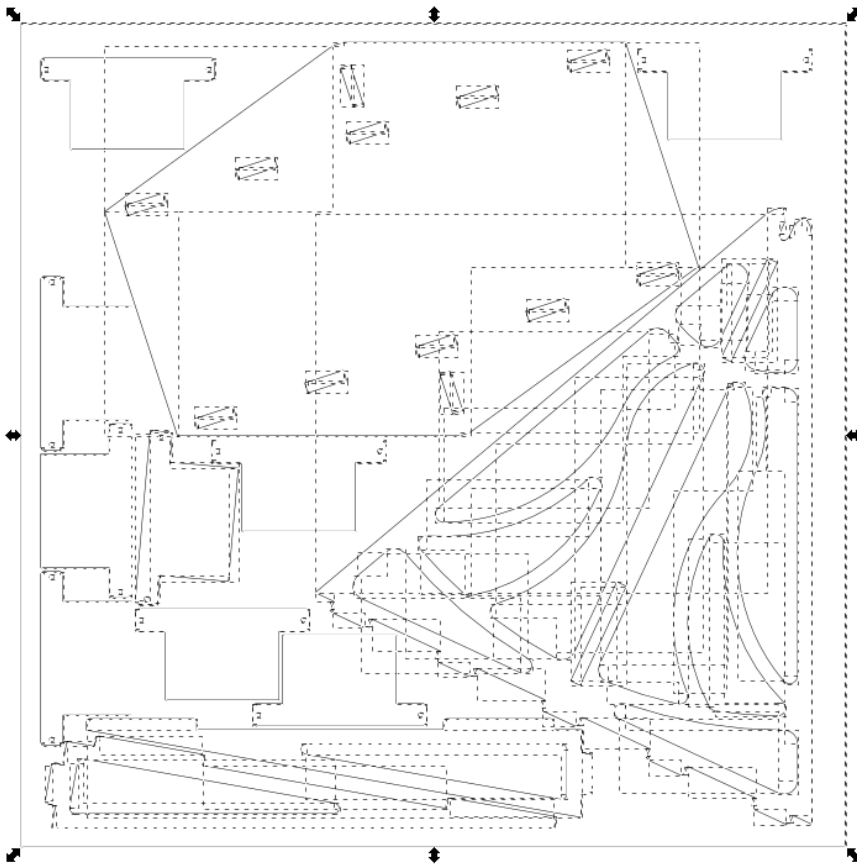
Note, that **paths never fork**. This means, that if there are three or more path ends at the same location, only two are chained together. The others are left unchanged.

## Additional notes

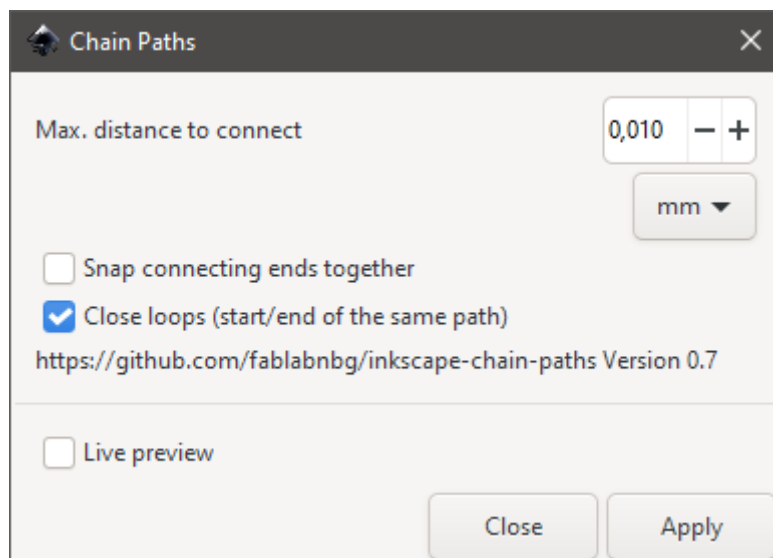
- this plugin can be used to reduce node by chaining (linking) overlapping points
- by selecting "Snap connecting ends together" you are creating closed loops. Otherwise they are open!

- This plugin does not apply for paths in groups. Please ungroup before!

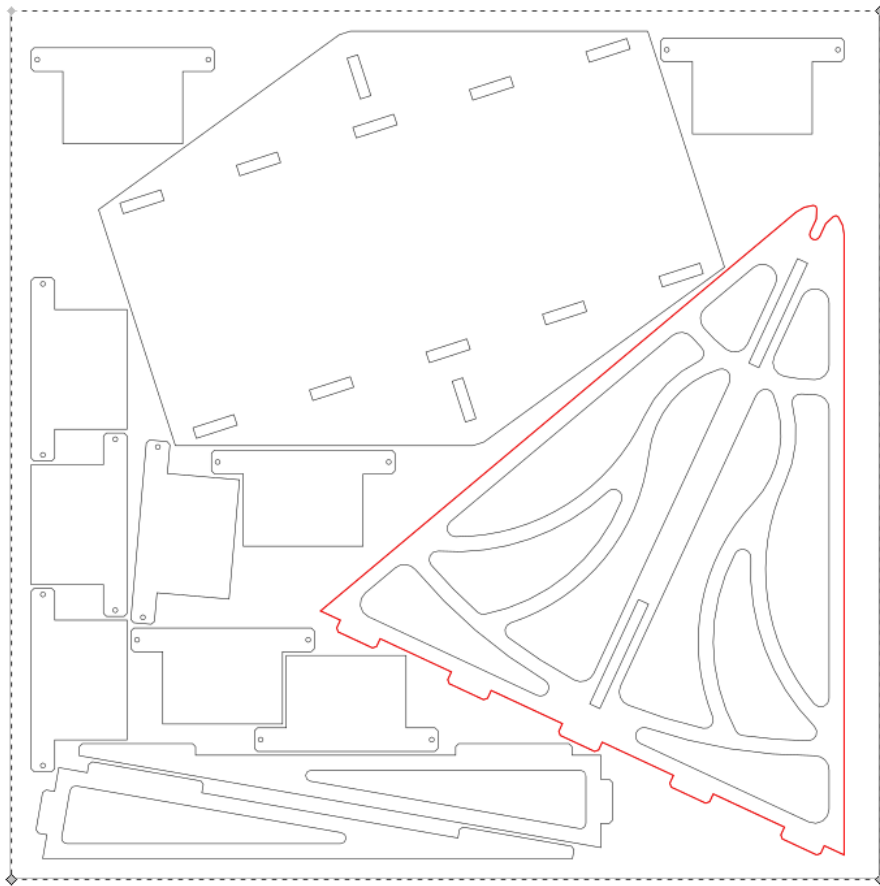
## Step 1 - Having a loose set of lines



## Step 2 - Apply extension



### Step 3 - Get closed (chained) contours



Similar extensions: Line Merging (Combine Paths), Close Paths, Join Paths / Create Tabs  
And Dimples



FabLab Chemnitz

# Layer Clip

- Original idea from <http://www.pernsteiner.org/inkscape/layerclip/>
- Ported to Inkscape 1.0 by <https://gitlab.com/jczapla79/inkscape-extension-layer-clip>
- Fixed and ported to Inkscape 1.2+ by Mario Voigt

## Usage

See an example with screenshots

1. Clip containing layer
  - Select a single object and then run this extension to set that object as the clip path for the layer that contains it.
  - This will mark the layer with "(c\*)" to indicate that the layer contains an object that is being used to clip its containing layer, and with "(C)" to indicate that the layer has a clip path applied to it. (These markers are simply added to the layer name to make it easier to tell when clip paths are in use.)
2. Clip layer above
  - Like "Clip containing layer", but sets the object as the clip path for the layer above the one that contains it
  - The layer containing the object will be marked with "(c+)" (instead of "(c\*)" for "Clip containing layer").
3. Clip layer below
  - Like "Clip containing layer", but on the layer below the layer containing the selected object.
  - Marks the object's layer with "(c-)".
4. Clip parent layer
  - Like "Clip containing layer", but on the parent layer of the layer containing the selected object.
  - Marks the object's layer with "(c\*\*)".
5. Remove layer clip
  - Select an object in a clipped layer and then run this extension to remove the clip path from a layer.
  - The "(C)" marker will also be removed, if one is present.

- The marker on the layer that held the actual clip path will not be removed, but you can remove it yourself by manually editing the layer name.

#### 6. Reset clippath transforms

- If you move an object that is being used as a clip path for a layer, the layer will still be clipped based on the old position. Run this extension to fix the problem.

FabLab Chemnitz

# About/Upgrade MightyScape

# Table Support

- Source: <https://sourceforge.net/projects/inkscape-tables/files/>
- Updated to be compatible with Inkscape 1.2+
- The functions "Duplicate Table" and "Fit Guides" options were removed due to defectness

## Possible actions to perform

- Create
- Add
  - Columns
  - Rows
  - Guide Lines
- Edit
  - Selected cells text
  - Selected columns width
  - Selected rows height
  - Selected table width and height
- Fit
  - Selected table to page
  - Selected table to page width
  - Selected table to page height
- Merge cells
  - Merge selected cells
  - Split selected cells
- Remove
  - Selected columns
  - Selected rows

The number of columns:

2

−

+

The number of rows:

2

−

+

The width of the cell:

100mm

The height of the cell:

20mm

VERY IMPORTANT: You will notice that the newly created table does not contain text. In order to edit any cell's text, just select the cell and then from the Extensions menu select Table -> Edit -> Selected cells text

☐ Live preview

Close

Apply

	1		
	S		
	4		



FabLab Chemnitz

# Open Extension Directory

Just what it says to be! ... Opens the MightyScape main folder (Lächeln)



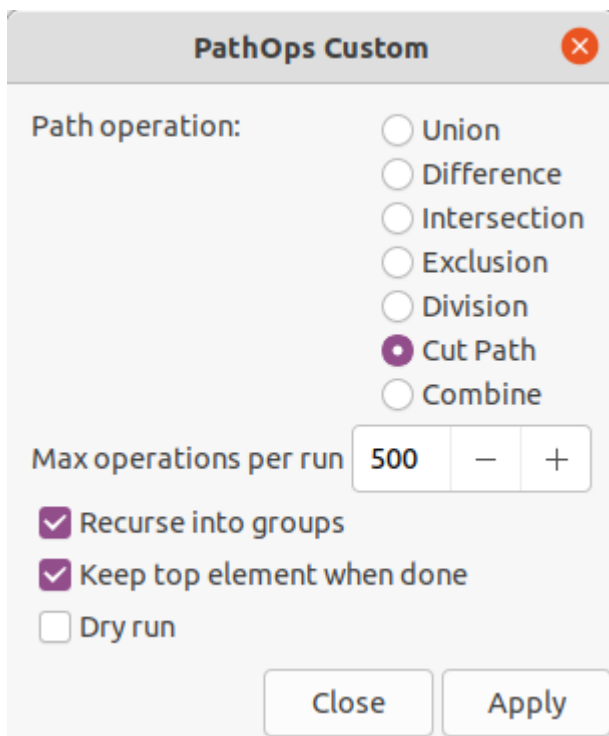
# Reload

An extension which allows to reload the drawing in case of any change without opening the file in a new window. Helpful if the file is opened and changes in background by another instance of Inkscape or any routine which processes and overwrites it in the meantime.

# PathOps

A tool suite to run boolean operations multiple times, like Combine, Cut Path, Division, Exclusion, Intersection, Difference, Union. This extension takes a selection of one or more groups, or of several elements (paths, shapes, text), or a mix of both, and applies a chosen path operation with the top-most valid object in paint order and each other valid object which is part of the selection (directly or as member of a processed group) that is lower in Z-order. Objects which are considered "valid" for path operations are paths, shapes and text. Other graphic elements like clones or bitmap images are silently ignored.

More info about the usage can be found at [https://gitlab.com/moini\\_ink/inx-pathops](https://gitlab.com/moini_ink/inx-pathops)



The image shows a software dialog box titled "PathOps Custom" with a red close button in the top right corner. The dialog contains the following elements:

- Path operation:** A group of radio buttons with the following options: Union, Difference, Intersection, Exclusion, Division, **Cut Path** (which is selected with a purple dot), and Combine.
- Max operations per run:** A numeric input field showing "500", flanked by minus and plus buttons for adjustment.
- Checkboxes:** Three checkboxes are present: "Recurse into groups" (checked with a purple checkmark), "Keep top element when done" (checked with a purple checkmark), and "Dry run" (unchecked).
- Buttons:** Two buttons at the bottom right: "Close" and "Apply".

# Simple Inkscape (Interface)

Source: <https://inkscape.org/~Mr.Raven/%E2%98%85simpleinkscapev01+2>

Original description:

“This extension changes the Inkscape interface into a simpler one.

It hides quite some elements from the menus and toolbars so that a more basic version of the interface remains. This can be useful for educational purposes. It is very easy to use and you can always go back to the default Inkscape user interface.

After installing the extension, you restart Inkscape, open the menu 'Extensions', submenu 'Interface' and then 'Simple Inkscape'. It explains itself.

This extension was made in collaboration with @Kirstie, one of the users of the Inkscape forum and writer of a very good Dutch book about Inkscape. She made the changes to the ui, I made the extension that installs or removes them.

There are two forum topics about this extension.

<https://inkscape.org/forums/extensions/simple-inkscape-what-items-do-we-want/>

<https://inkscape.org/forums/extensions/simple-inkscape-extension-children-education-learning-simplified-user-interface/>

It works in Windows and Linux for Inkscape 1.0 and above.

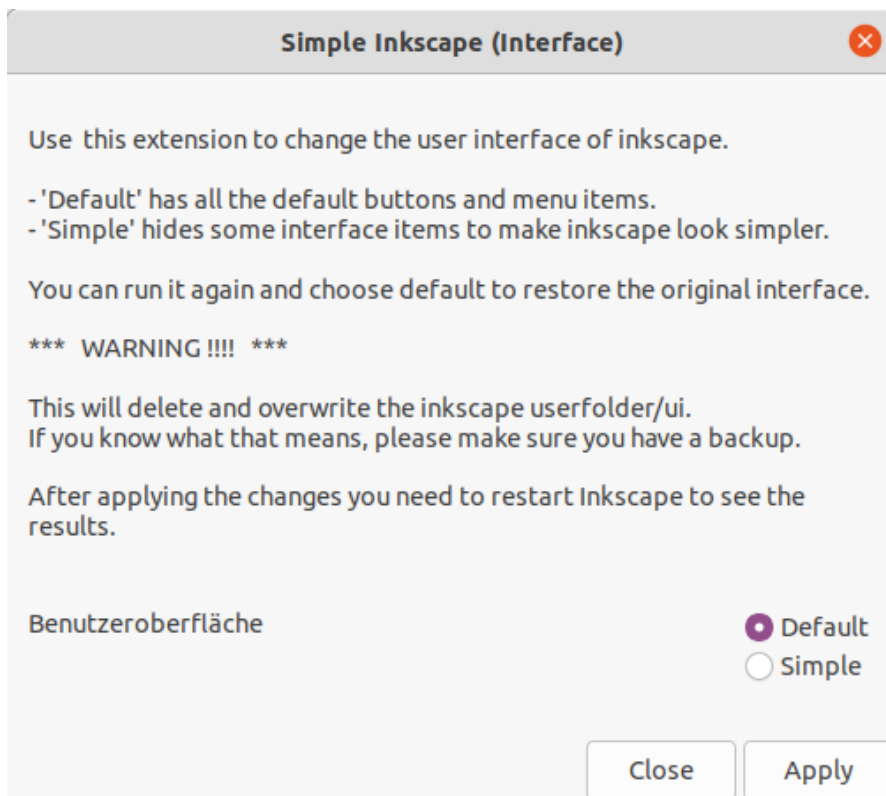
Lower versions of Inkscape will not work.

Mac is currently not supported since I have no Mac to test it on. It will simply give an error message on Mac. If anyone who has a Mac wants to help me test it, I believe I can make it work for Mac as well. Please

contact me if you want to help me with it.

If you have questions - please feel free to ask.

I would also be interested to know if the extension is used and what it is used for so that I can keep that in mind for future versions. So if use it - leave a post if you like.



# Ray Tracing

Sources:

Ray Tracing - Insert Lens Opt... x

Focal length:

100,000

-

+

mm ▾

Diameter:

1,000

-

+

In ▾

Edge thickness:

2,000

-

+

mm ▾

Optical index:

1,5168

-

+

Lens type:

Plano-concave/convex ▾

☐ Vorschau

Schließen

Anwenden

[E2%98%85ray-optics](#)

[kscape-raytracing](#)

Ray Tracing - Set Lens Ma... x

Select material:

Mirror ▾

Optical index:

1,5168

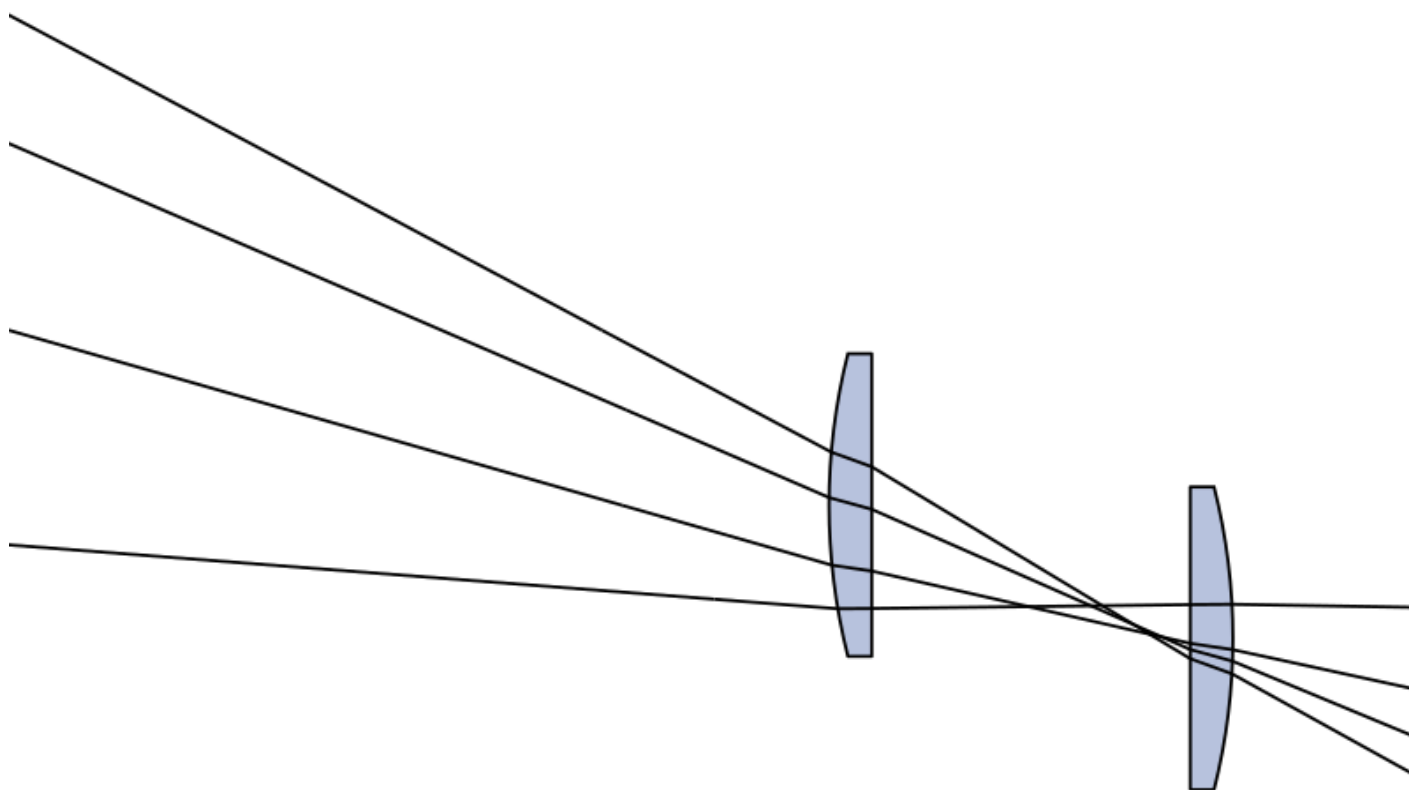
-

+

☐ Vorschau

Schließen

Anwenden



Integrated into default Import/Export Menu

Integrated into default Import/Export Menu

# Encapsulated PostScript - AI compatible (\* .eps)

Source of documentation:

<https://github.com/tzunghaor/inkscape-eps-export>

AI compatible EPS export for Inkscape 0.92 (might work with older versions too). This script converts an Inkscape SVG to Adobe Illustrator 7 compatible EPS. The generated EPS file uses custom Illustrator PS operators, and includes PostScript processes that stands in place of said operators when the file is not opened with Adobe Illustrator. The script exports layers, groups, paths, clones, clipping paths, fill, stroke, gradient fill into a format that Illustrator understands.

**Warning** This script is not extensively tested. Since its dual nature, it is possible that the result looks different in Illustrator than in other programs.

## Known limitations

- Text is not supported: convert them to paths, then ungroup them.  
You can regroup them afterwards, but without ungrouping, this script may think they are invisible.
- Path node types are not retained.
- Circles and elliptical arc segments in paths are converted to bezier curves.
- Layer names lose non 7 bit ASCII characters.
- Radial gradients cannot be elliptical: all radial gradients will be converted to circular. (Although I think it is possible to save elliptical radial gradients, but I did not figure out how to save it in a way that is compatible with different Illustrator versions.)
- Outline gradients are not supported. (Illustrator 7 does not support them.)
- Clones are exported as copies.
- No transparency: everything is exported opaque. (EPS does not support transparency.)
- Filters (including radial blur) are not exported.



- Path effects are not exported, only the result of the effects.

## Features

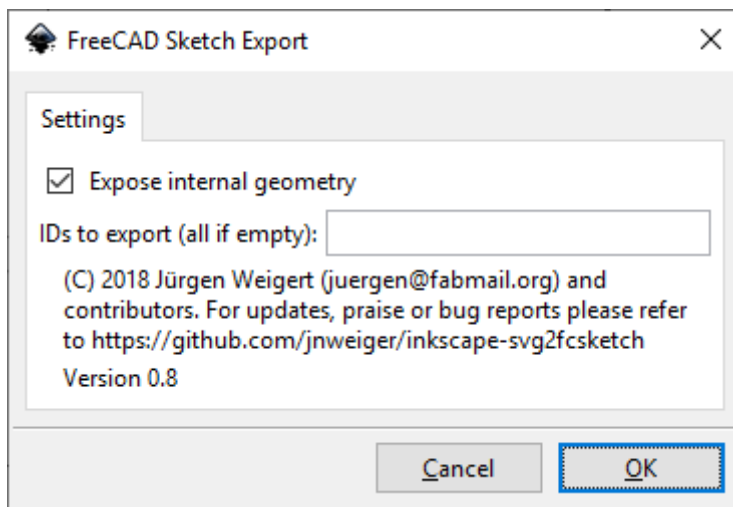
(It's not a bug, it's a feature!)

- Automatically closes all paths that are filled. It makes visual difference with unclosed paths that have fill and stroke.
- Invisible objects are not exported (invisible layers and objects, objects with neither stroke nor fill, stray points)

The generated \*.eps files do not work in Roland CutStudio. For this case please use [Open in Roland CutStudio Plugin](#)

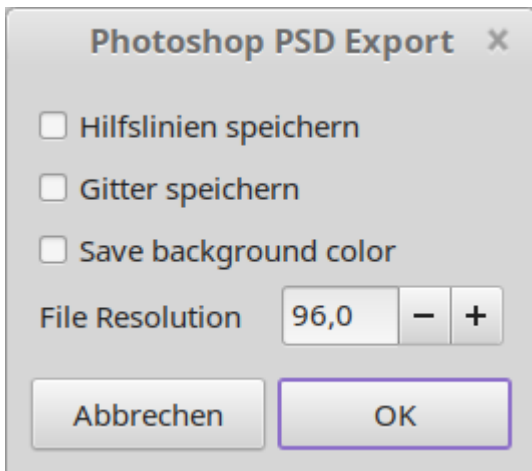
Integrated into default Import/Export Menu

# FreeCAD Sketch Export



Integrated into default Import/Export Menu

## Photoshop PSD (\*.psd)



Requires gimp to be installed. The command `gimp` needs to be in `%PATH% ($PATH)`

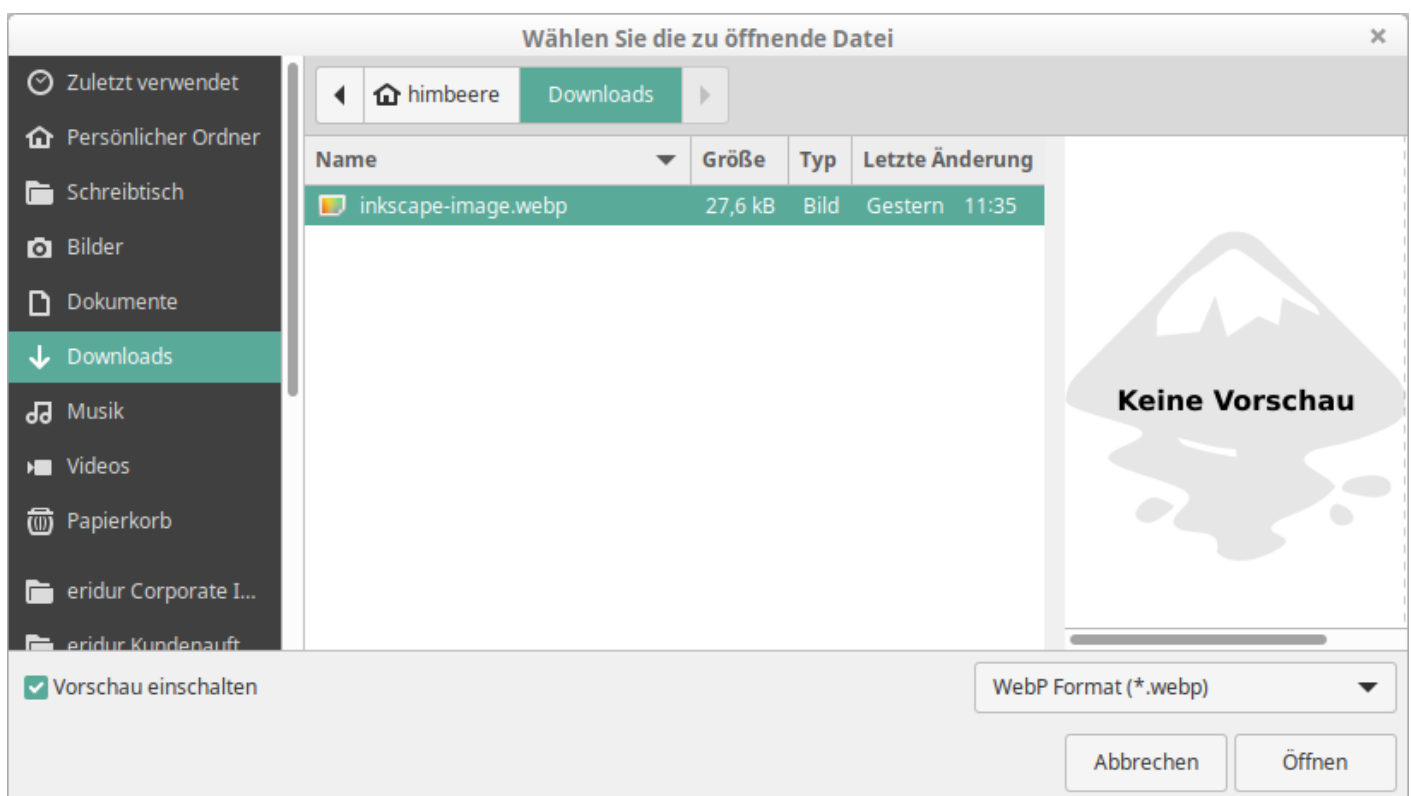
Integrated into default Import/Export Menu

# WebP Import

Import \*.webp image files using ImageMagick conversion. Routine is webp → png → svg → write into Inkscape canvas.

You might need to install ImageMagick. The command "magick" or "convert" needs to be in path variable.

```
sudo dnf install ImageMagick #install on Fedora
```



Integrated into default Import/Export Menu

## Roland CutStudio \*.eps Export

See [Open in Roland CutStudio](#). This extension does the same export but it will not call CutStudio. Instead it will save an importable \*.eps file to the selected target directory by export menu.

# FabLab Chemnitz Boxes.py

Boxes.py is an Open Source box generator written in Python. It features both finished parametrized generators as well as a Python API for writing your own.

Sources of documentation

- <https://www.festi.info/boxes.py/index.html>
- <https://florianfesti.github.io/boxes/html/generators.html>

Source files for boxes.py

- <https://github.com/florianfesti/boxes>



The images are from <https://www.festi.info/boxes.py>

# FabLab Chemnitz Shape Generators

# FabLab Chemnitz Boxes/Papercraft