Ambigram generator (fully customizable)

This is an OpenSCAD script to generate two-sided 3D texts that merge, each text is visible from a different point of view.

This is a PDF version of the README.md file you can find on original GitHub repository.

This script has been published and can be found and used directly on MakerWorld's page.

Summary

- A guide through the tabs of parameters
  - Texts
  - Font
  - Base
  - Top
  - Tweaks (fix thing up)
- How to remove floating objects
- How to use symbols in texts
  - On Windows
  - On macOS
  - Using the available cheat sheets
- Version history
- Thanks

A guide through the tabs of parameters

Here's an detailed explanation of every possible setting. Feel free to explore the endless possibilities of combining these values.

Texts

In this tab you can set the basic parameters about displayed texts.

Parameter	Description	unit	min	max	default
text_left	This is the text which is visible: - from left if *text_mode* is *Left-Right*; - from the top in other modes.	text			"SUSANNA"

Parameter	Description	unit	min	max	default
text_right	This is the text which is visible: - from right if *text_mode* is *Left-Right*; - from front in other modes.	text			"CHRISTIAN"
text_mode	This defines the mode of rendering of texts, this value can be: - **Left-Right** - one text is visible from an angled left point of view, the other one is visible from an angled right point of view; - **Front-Top** - one text is visible from the top of the object, the other one from the front; - **45° front-Top** - one text is visible from the top of the object, the other one from a 45 degrees angled position between the top and the front.				Left-Right
render_engine	This enables user to choose which rendering engine to use with this script, providing some sort or retro-compatibility. This value can be: - **v1.0 - legacy** - this uses the legacy rendering engine of version 1.0, which is know to have some limitations; - **v2.0 - NEW! produces a cleaner object** - this value enables the use of new version 2.0 engine which renders better, creating cleaner objects, solving a lot of problems and being more precise with calculations, giving user new opportunities and features;				v2.0
text_angle	This value, which is valid only in the *Left-Right* mode, allows you to set the angle from which the texts will be visible.	defrees	0	180	45

Parameter	Description	unit	min	max	default
text_shape	This value can change the shape of the object, possible values are: - **Rectangle** - the same as in original released version, seen from above the object is a rectangle or a rectangle with arrows: - **Circle** - seen from above the object is a circle; - **Heart** - seen from above the object is an heart; - **Rounded rectangle** - a rectangle with rounded corners (only available when using new rendering engine).				Rounded rectangle
text_roundness	If *text_shape* is set as *Rounded rectangle*, this is the percentage of roundness to give to corners. A value of zero will create a pure rectangle.	%	0	100	50
text_heart_direction	Specifies where the pointing end of the heart is directed, this value can be: - **Left** - the heart points in the same direction of the left text; - **Right** - the heart points in the same direction of the right text; - **Middle** - the heart points in the middle between the two texts.				Left

Font

In this tab you can set all parameters about the font to be used for the texts.

Parameter	Description	unit	min	max	default
font_name	Choose one of the supported fonts from the list to apply it to the text on this side. **Be warned!** Complex fonts and fonts with serifs tend to generate floating objects which will result in waste of material and time. Consider using sans serif fonts and/or check my guide on how to easily get rid of floating objects using Fusion 360 down below.				Merriweather Sans

Parameter	Description	unit	min	max	default
font_custom	Here you can override the above setting by putting the name of a font installed in the system where this script is executed. Please keep in mind that when you use Parametric Model Maker, the system the script is running on is MakerWorld's server, not your computer.				*empty*
font_style	Choose the style of the text. This can also be set to *None*, which will use font's default style. **Note:** not all fonts support all styles.				Bold
font_size	The size of the text, which will result in being also the depth of the object.	mm	1	72	20
font_spacing	Spacing between letters, the greater the number, the far the letters are. Better results are achieved when the letters touch one another, if they don't, a solid base will be required to keep them together.		0.1	5	0.75

Base

To keep the letters merge, you need to generate a base plate, these parameters allows you to personalize it's creation.

Parameter	Description	unit	min	max	default
base_type	The type of base to be created, this value can be: - **None** - no base is created at all; - **Solid** - a solid base is created at bottom of object; - **Text** - a top-text shaped base is being created at bottom of object, please note that this is mode not supported in *Left-Right* model mode and *Solid* will be used instead.				Text
base_height	The height of the base.	mm	0.2	25	3
base_x_correction	Since OpenSCAD does not provide functionalities to compute texts' widths and heights, calculus about the base are approximated, with this parameter you can offset the base position on the X-axis to correct those calculus.	mm	-100	100	0
base_y_correction	Same as *base_x_correction*, but for the Y-axis.	mm	-100	100	0

Parameter	Description	unit	min	max	default
base_z_correction	Same as *base_x_correction*, but for the Z-axis.	mm	-100	100	0.4
base_size_correction	Same reason as for *base_x_correction*, this parameter allows you to shrink on enlarge the base size.	mm	-100	100	0
base_heart_size	The correction of the size of the heart when it's being created.	mm	-100	100	0
base_heart_x	Heart X-axis correction.	mm	-100	100	0
base_heart_y	Heart Y-axis correction.	mm	-100	100	0

Top

Parameters that will create a top plate over the texts.

Parameter	Description	unit	min	max	default
top_type	The type of plate to be created, this value can be: - **None** - no top plate is created at all: - **Solid** - a solid plate is created at top of object: - **Text** - a top-text shaped plate is being created at top of object, please note that this is mode not supported in *Left-Right* model mode and *Solid* will be used instead.				None
top_height	The height of the top plate.	mm	0.2	25	3
top_x_correction	Since OpenSCAD does not provide functionalities to compute texts' widths and heights, calculus about the top plate are approximated, with this parameter you can offset the plate position on the X-axis to correct those calculus.	mm	-100	100	0
top_y_correction	Same as *top_x_correction*, but for the Y-axis.	mm	-100	100	0
top_z_correction	Same as *top_x_correction*, but for the Z-axis.	mm	-100	100	0.4
top_size_correction	Same reason as for *top_x_correction*, this parameter allows you to shrink on enlarge the top plate size.	mm	-100	100	0
top_heart_size	The correction of the size of the heart when it's being created.	mm	-100	100	0
top_heart_x	Heart X-axis correction.	mm	-100	100	0
top_heart_y	Heart Y-axis correction.	mm	-100	100	0

Tweaks (fix things up)

Some parameters that can help you fix things that come out weird.

Parameter	Desccription	min	max	default
curves_quality	Selects the quality of curves to be used when generating the model, the lowest the quality, the faster generation will take. You can work in Draft all the time for fast refresh, but you need to remember to switch back to Normal or something higher before exporting the model to be printed. Possible values: - **Draft** - low quality, not suitable for printing; - **Printable draft** - not for production, yet printable; - **Normal**: normal quality, ready to print; - **Good quality**: good quality for printing; - **High quality**: very good quality for printing; - **Very high quality**: good quality for high resolution printers. **Be warned:** Parametric Model Maker has a short timeout for script execution to avoid DDoS attacks, for this reason setting a quality too high could create a timeout error			Normal
font_ratio	Since OpenSCAD does not provide functionalities to compute texts' widths and heights, calculus are approximated, with this parameter, which rappresents the ration between the height and the width of a single letter of the font used, you can correct those calculations.	0.1	3	1.33
small_characters	This is the list of the characters that will be considered small and will have a smaller depth (see *small_characters_ratio*).			"! ()1l[]fijlt{"
small_characters_ratio	This is the ratio between the depth of a normal character and the depth of a character considered small.	0.01	3	0.3
space_ratio	This is the ratio between the depth of a normal character and the depth of a rectangle created in place of a white space.	0.01	3	0.1

How to remove floating objects

As said before, complex texts and fonts with serifs tend to generate floating objects which will result in waste of material and time. This step by step guide will teach anyone, also with no experience of using AutoDesk Fusion 360 at all, to remove them from the resulting object.

First of all generate the desired design, you will notice some floating objects that are not connected to other object and not supported by material in any way.
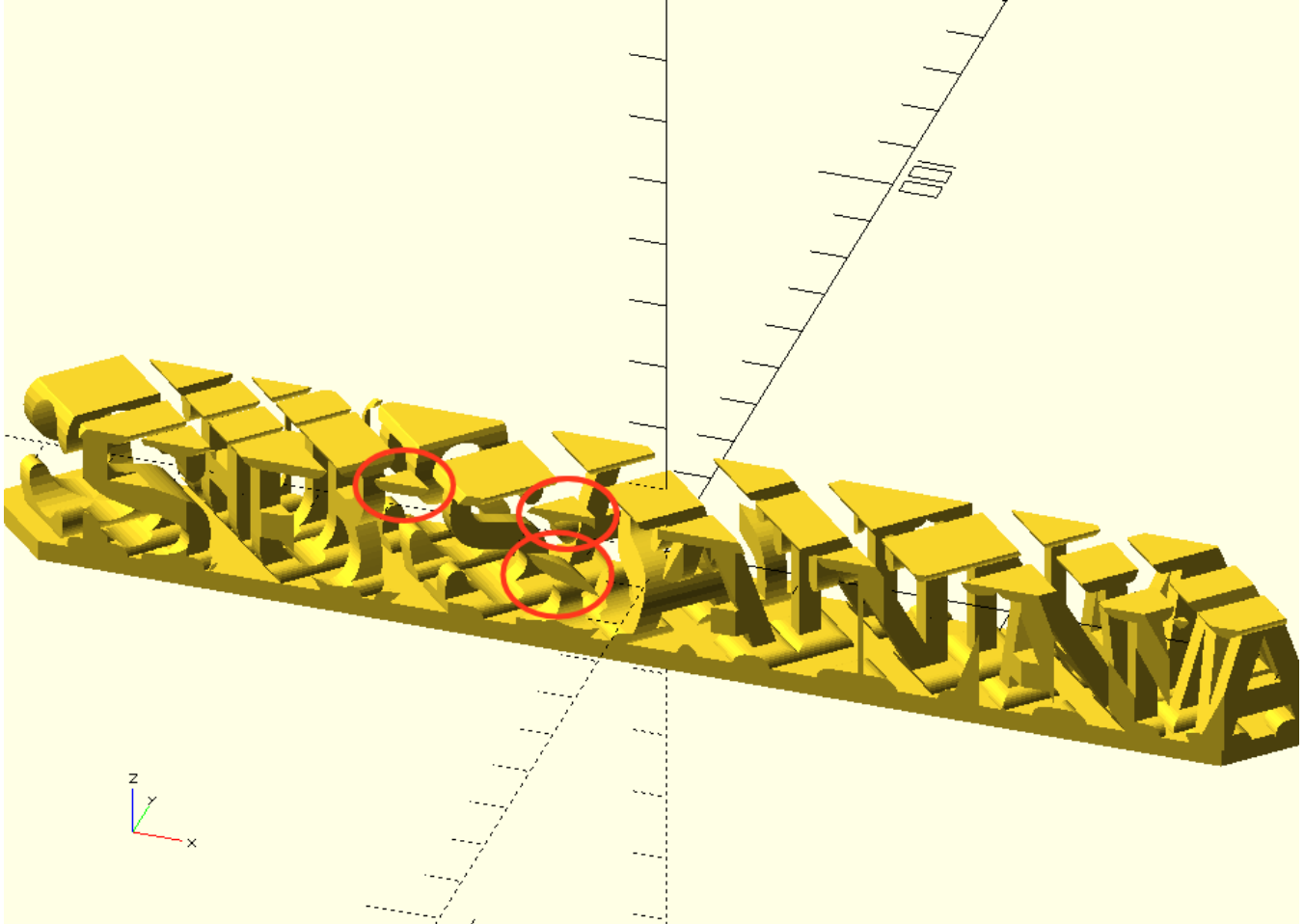
Save the design as STL

If using MakerWorld's Parametric Model Maker, click on the button to have the last modifications to the parameters be reflected on your model, then click on the button.

When this popup shows up
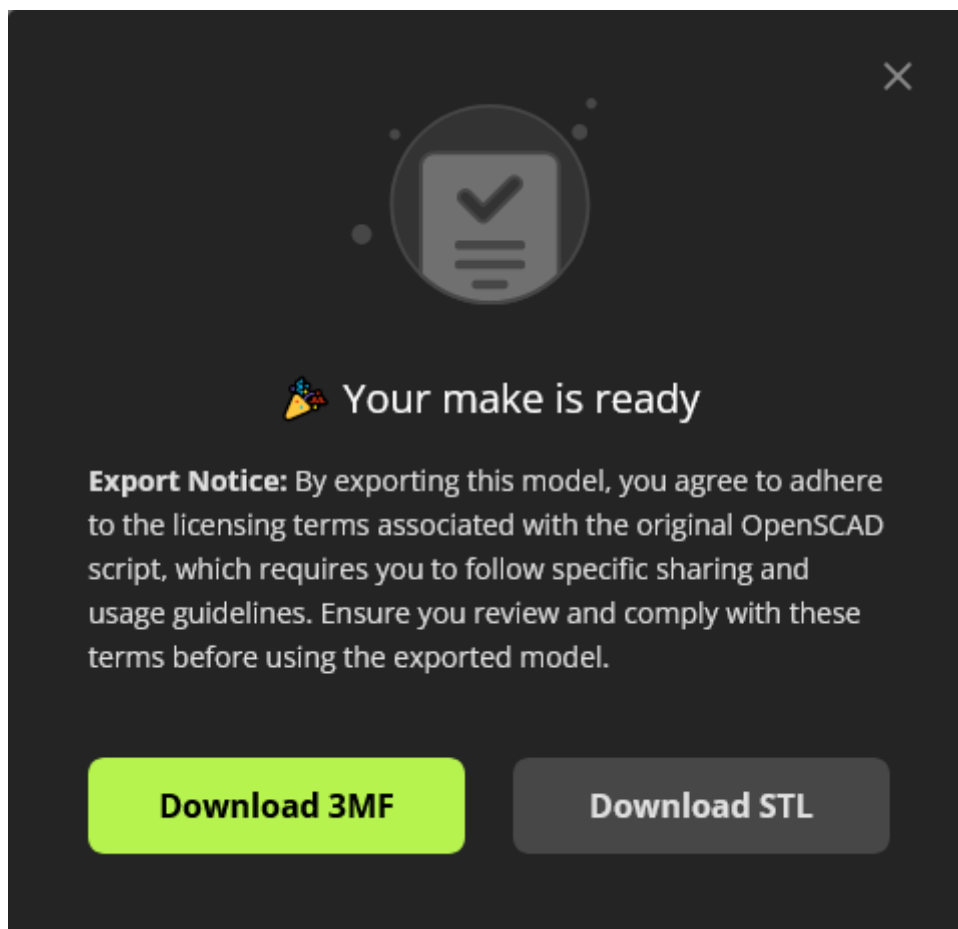
Click on the button to download the STL version.
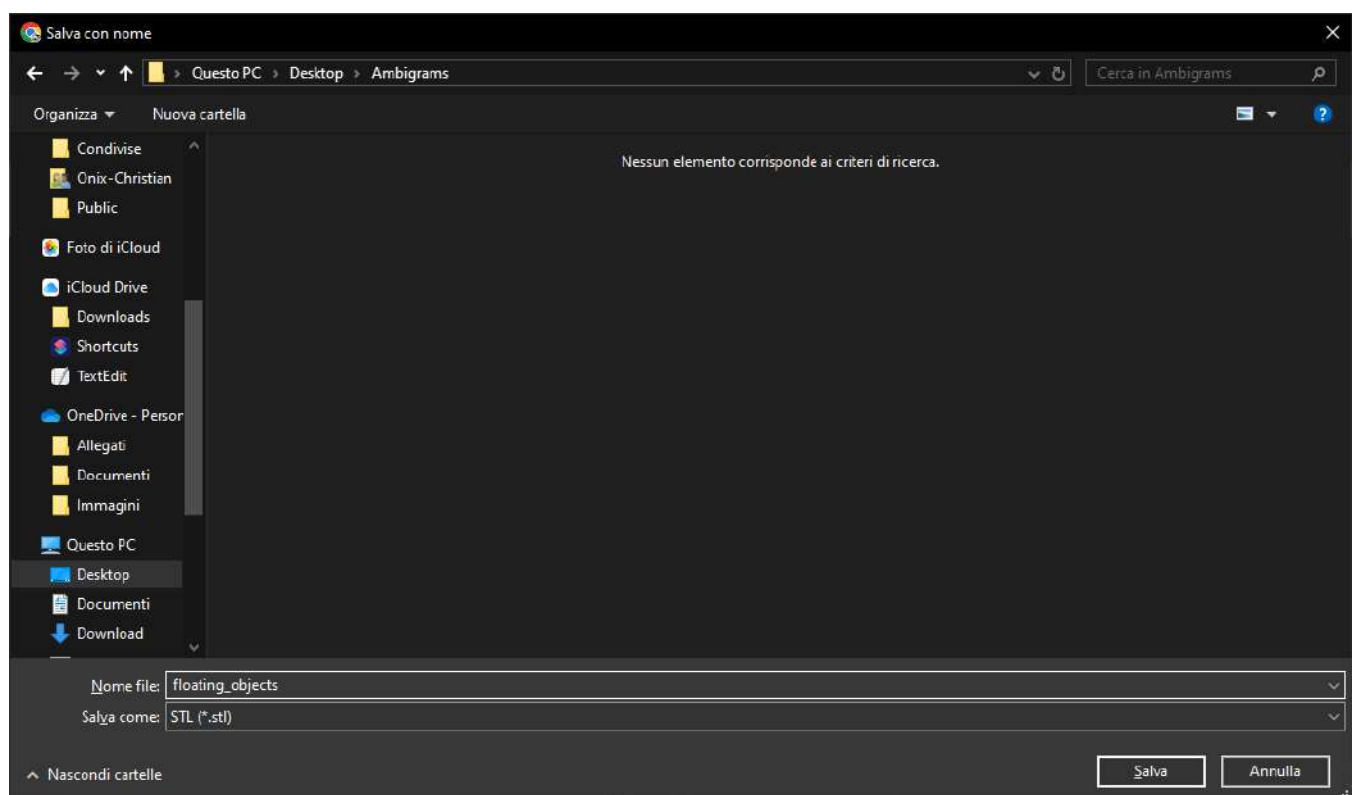
If your computer offers the possibility to save the file to a desired location, select a folder where you want to save it and choose a proper name, e. g. "floating_objects", which fill create "floating_objects.stl".

If your browser is configured to automatically save files, you will find the file named "Ambigram generator.stl" in the location you usually receive downloaded file, usually called the "Downloads" folder.

Now open a new Fusion 360 project, and from the top toolbar, in SOLID tab, choose INSERT => Insert Mesh
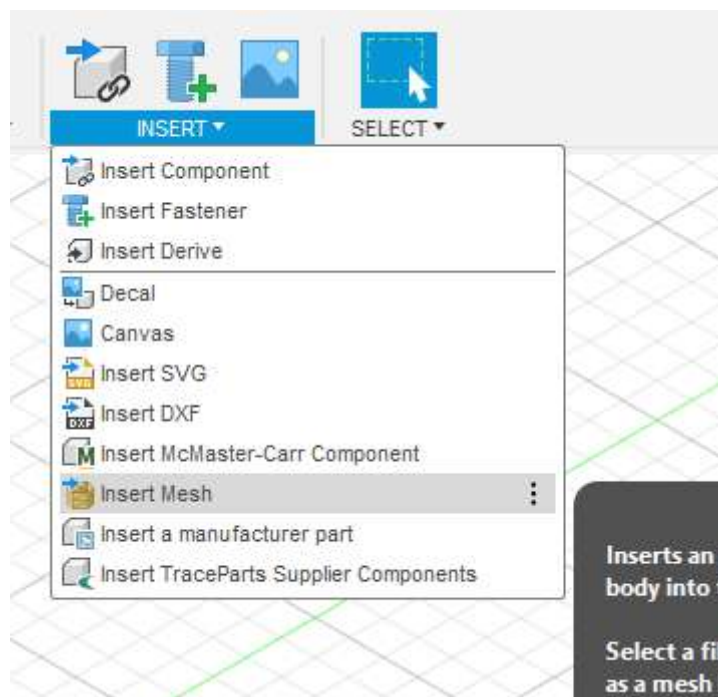
When this popup window will appear, click on **Select from my computer...** button.
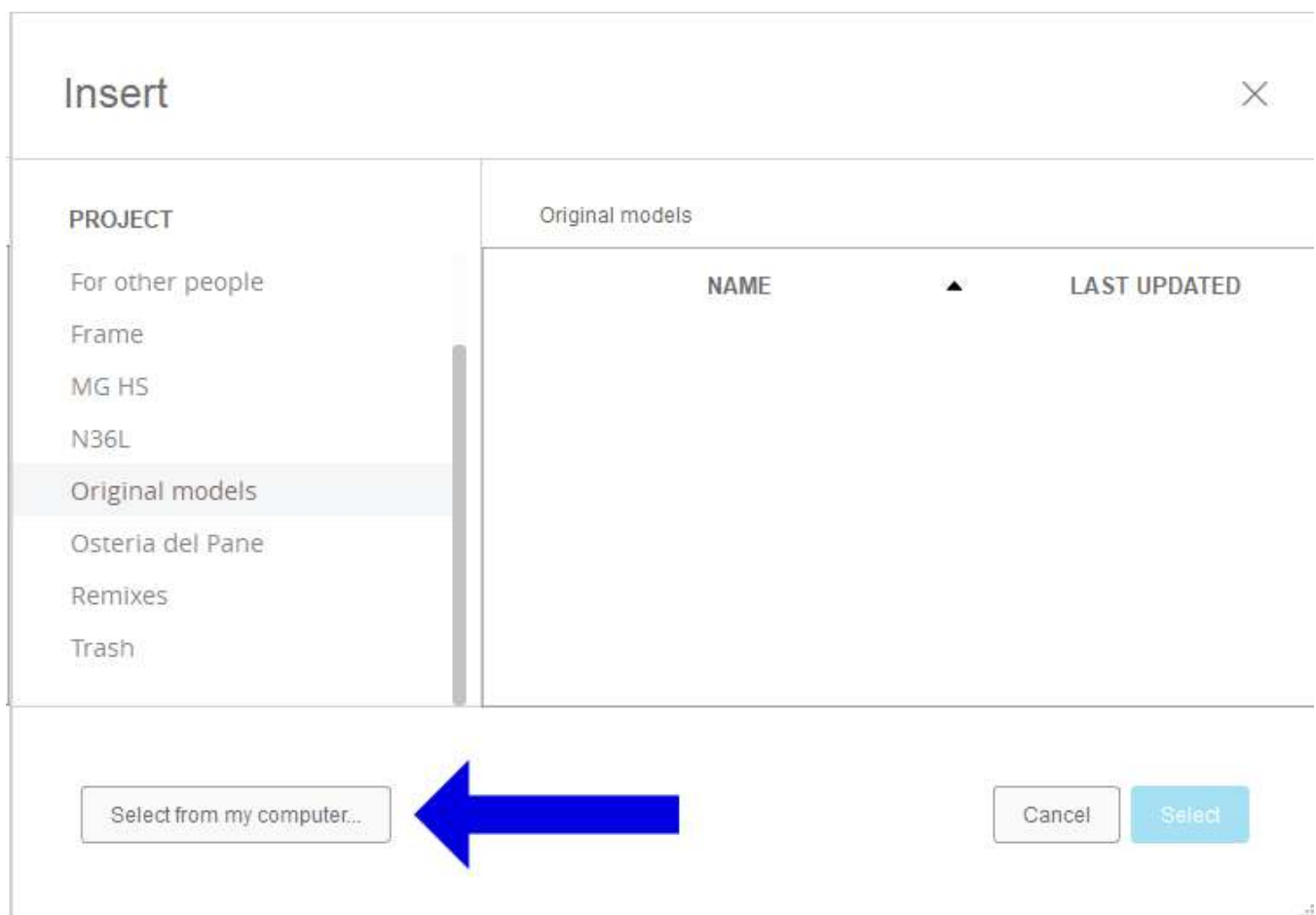
A file selection dialog will be shown, locate the freshly downloaded file and click on **Open**.

The object will be placed in the center of the scene, just click on **OK**.
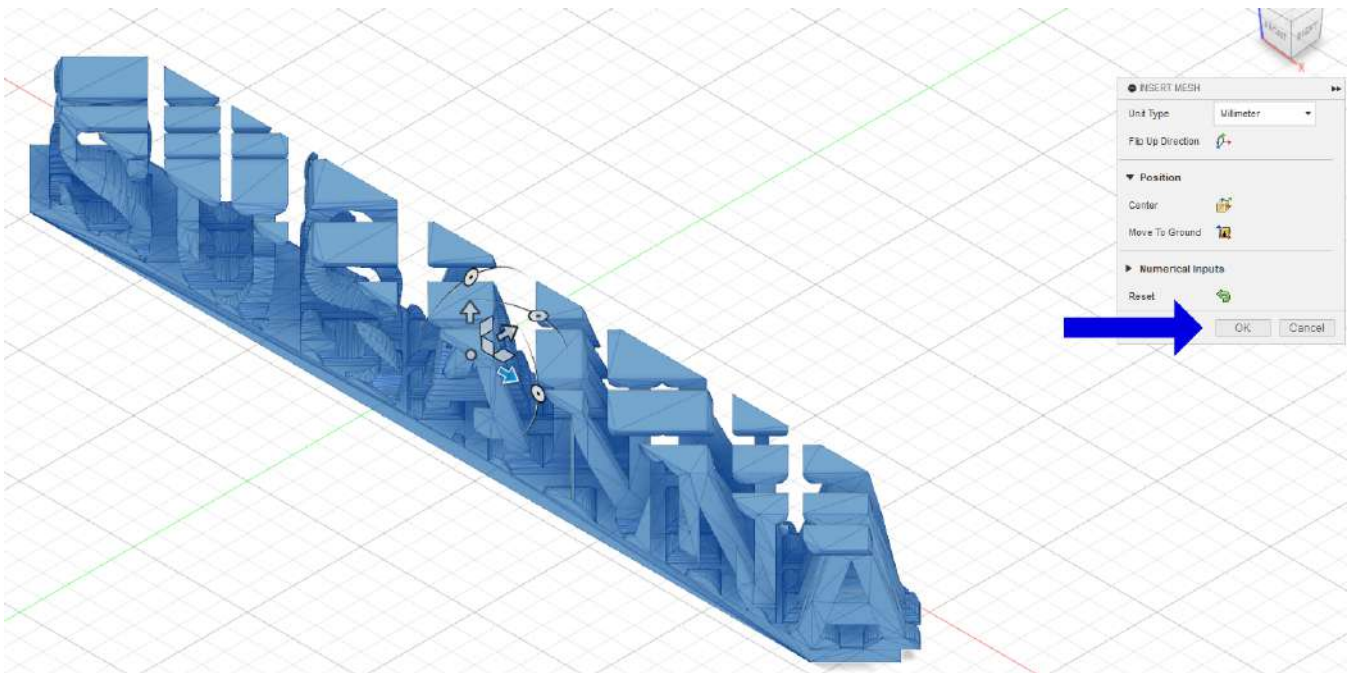
Now we need to convert this to an object body.
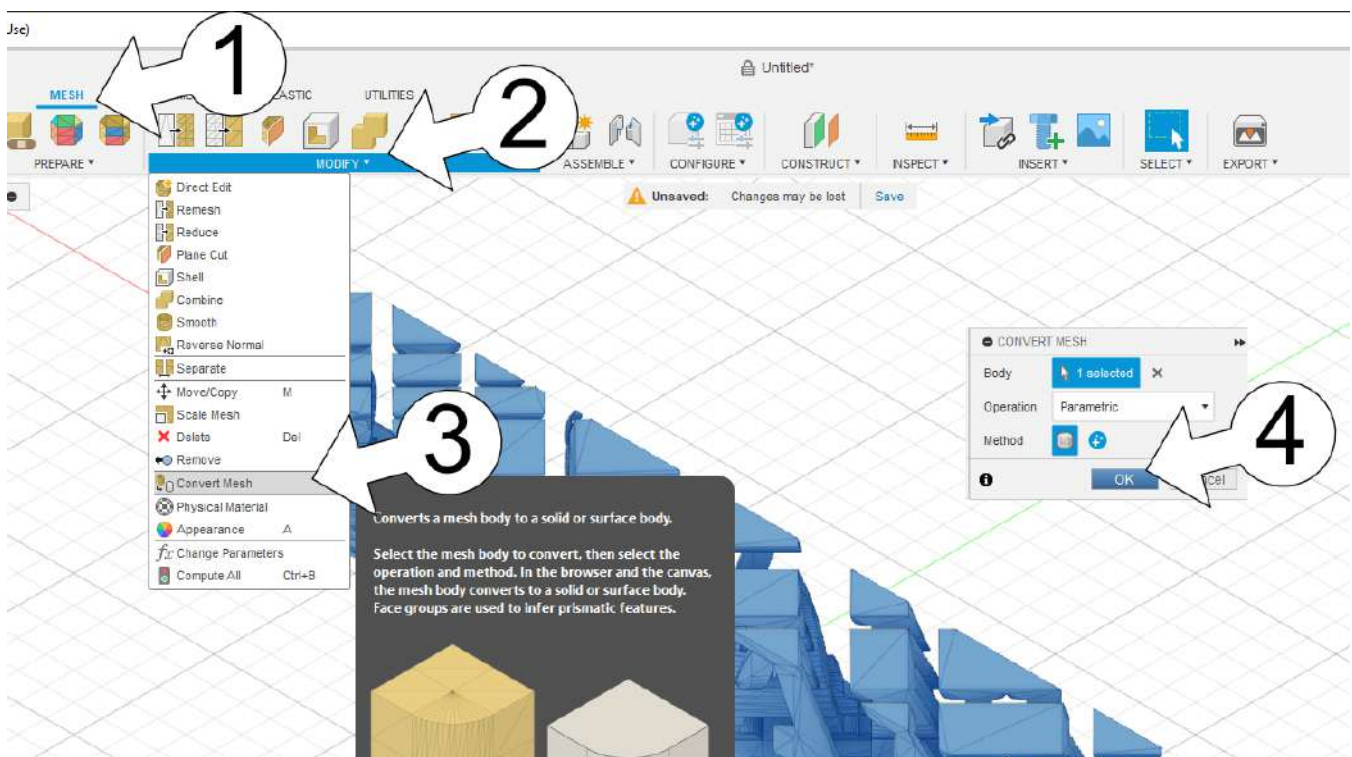
With the object selected, follow these steps:

1. Click on the **MESH** tab in the top toolbar;
2. Click on the **MODIFY** menu in the toolbar;
3. Select the **Convert Mesh** menu from the popup menu;
4. Click on **OK** to start the conversion.

Depending on the object complexity and your computer computational power, this could take some time.

Once finish, we need to disable the design history to be able to work on this freely.

Expand the **Bodies** group in the object browser by clicking on the left-pointing triangle on group name's left side.
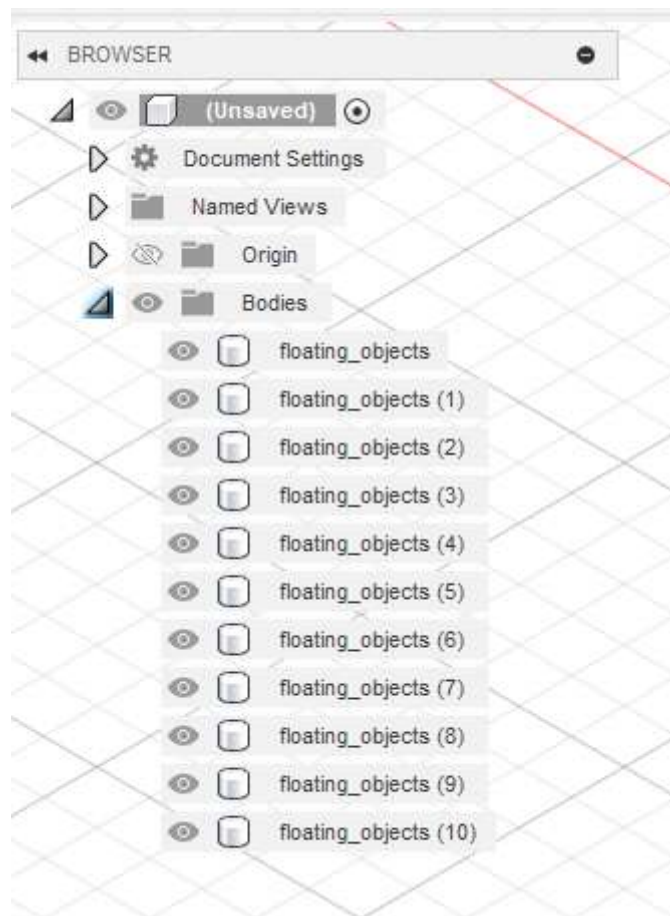
In the object browser, right click on the top nome e. g. *(Unsaved)*, and from the popup menu that appears choose **Do not capture Design History**.
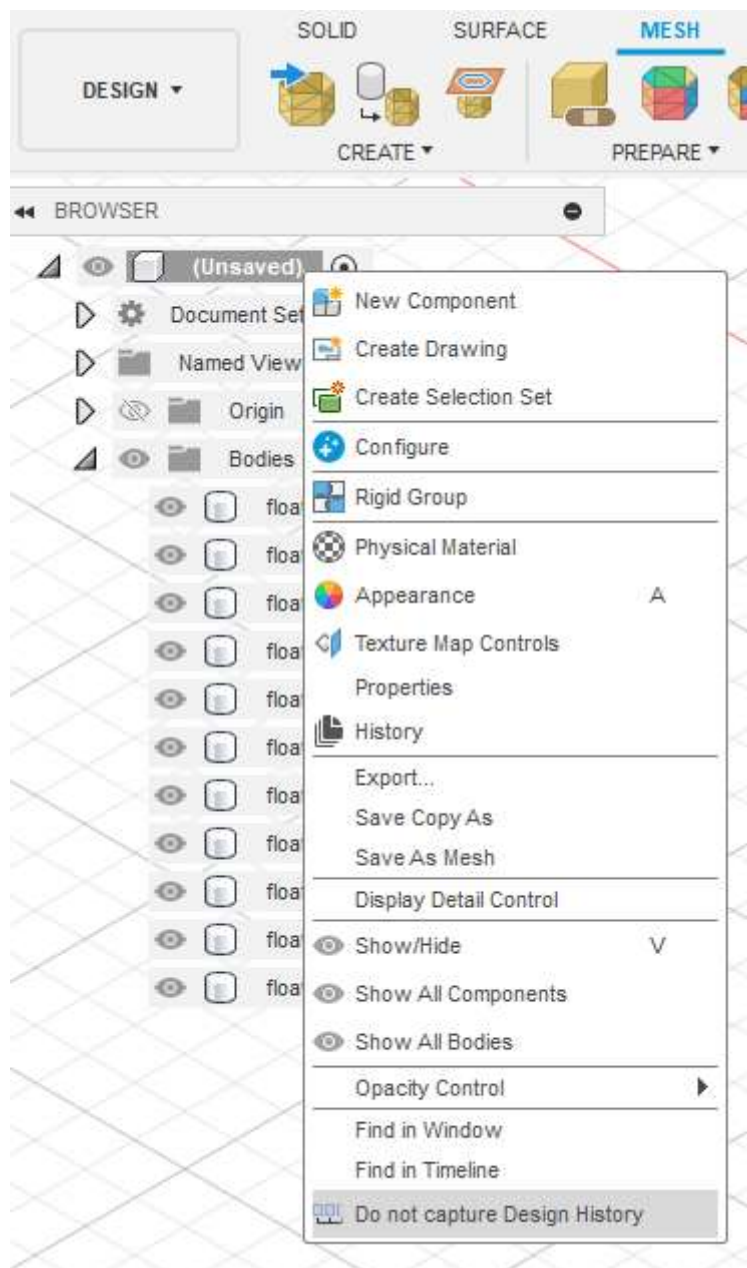
A confirmation dialog will appear, select **Continue** to stop capturing design history.
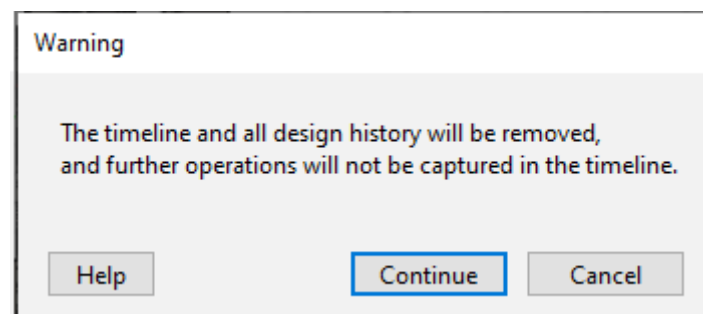

Now start selecting one by one the bodies, in the object browser's Bodies trunk.
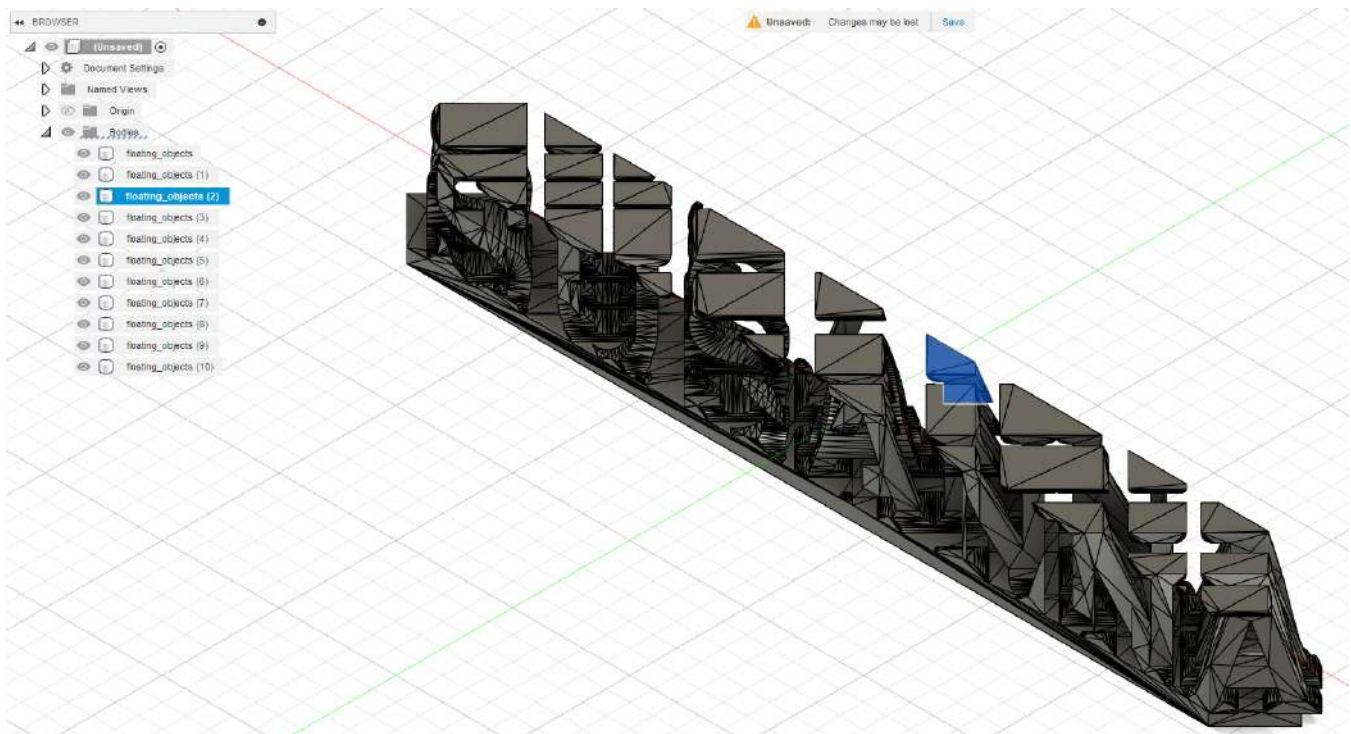
The corresponding object will be blue highlighted on the screen, it this is a floating object, right click on his name in the object browser and select **Delete** from the popup menu that appeared.
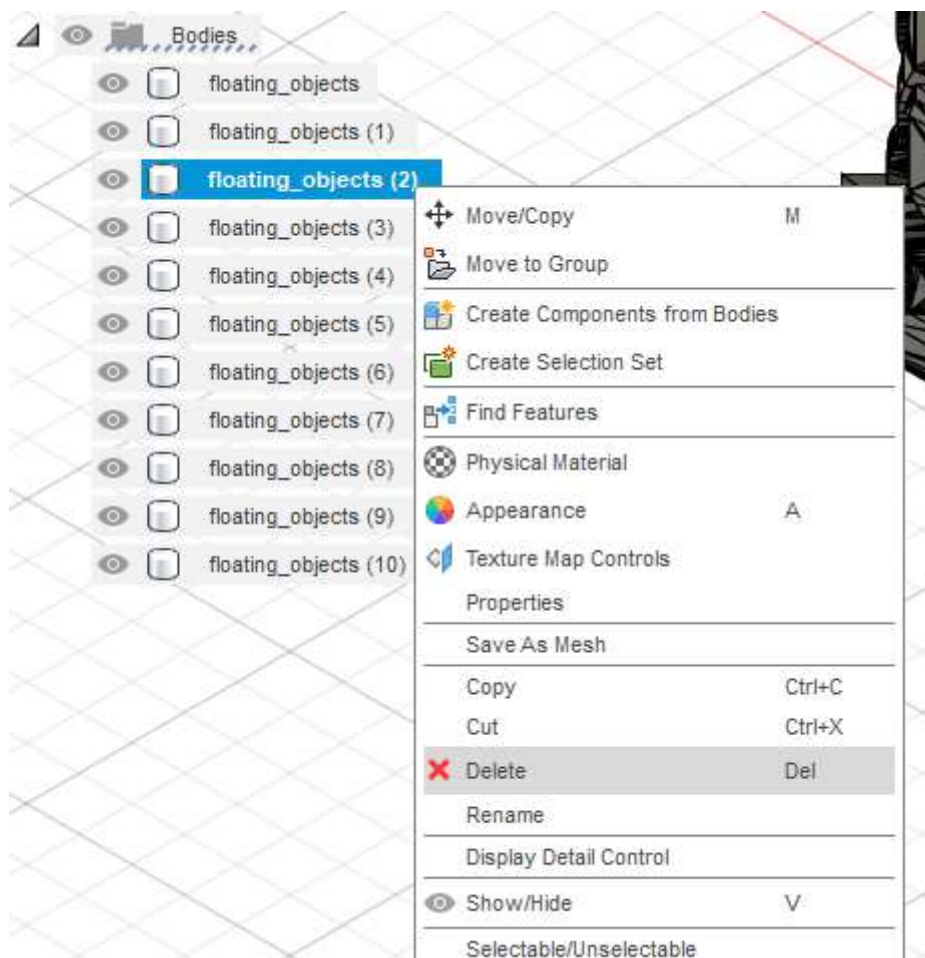
Important: once you deleted all unwanted bodies (you should only have one object left), left click on the **Bodies** trunk in the object browser.
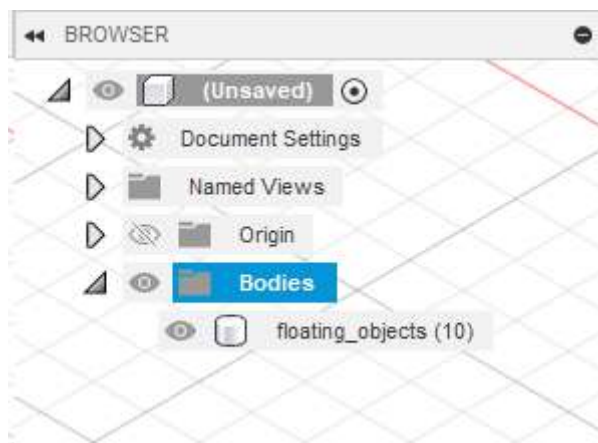
Now that we cleaned the object, we need to export the file back to STL.
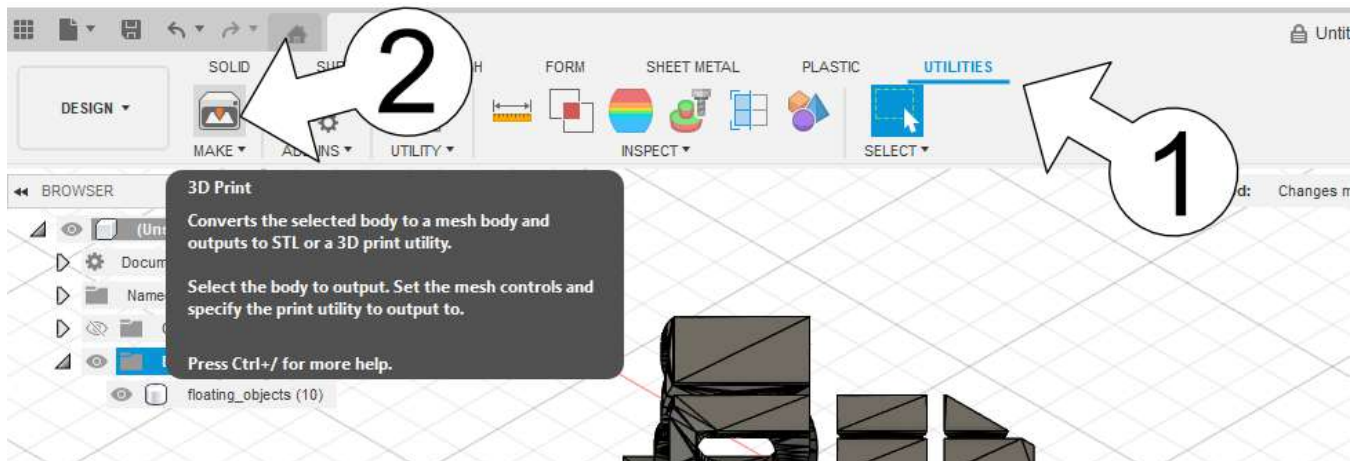
Follow these steps:

1. Select the **UTILITIES** tab in the top toolbar;
2. Select the **MAKE** icon.
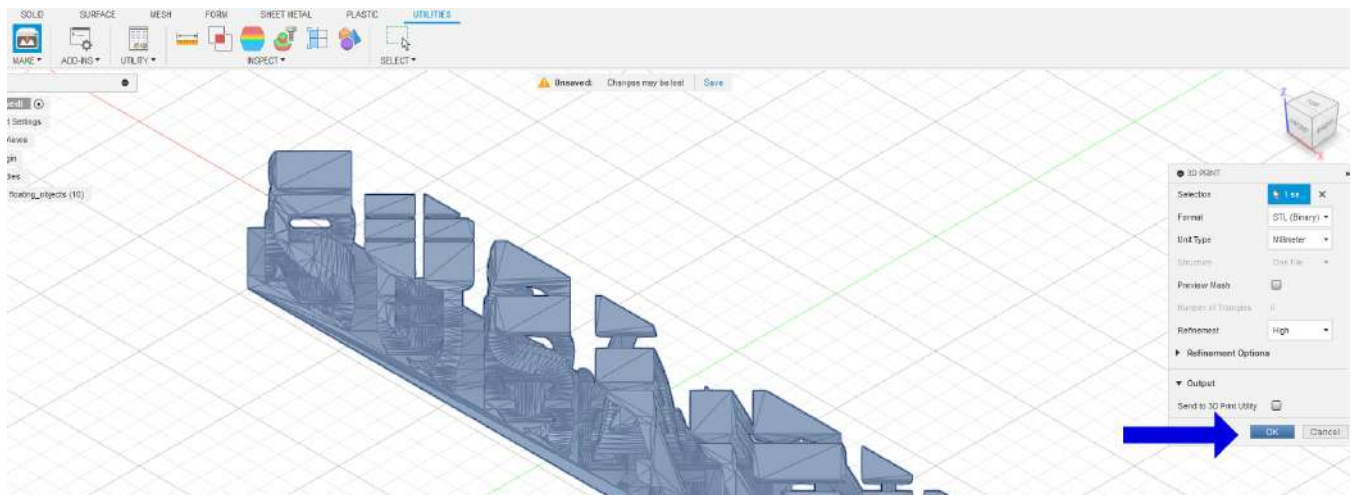
When in Make mode, the object will be light-blue highlighted, please ensure to select the **STL format** (binary is preferred, but not mandatory), then click on **OK**.
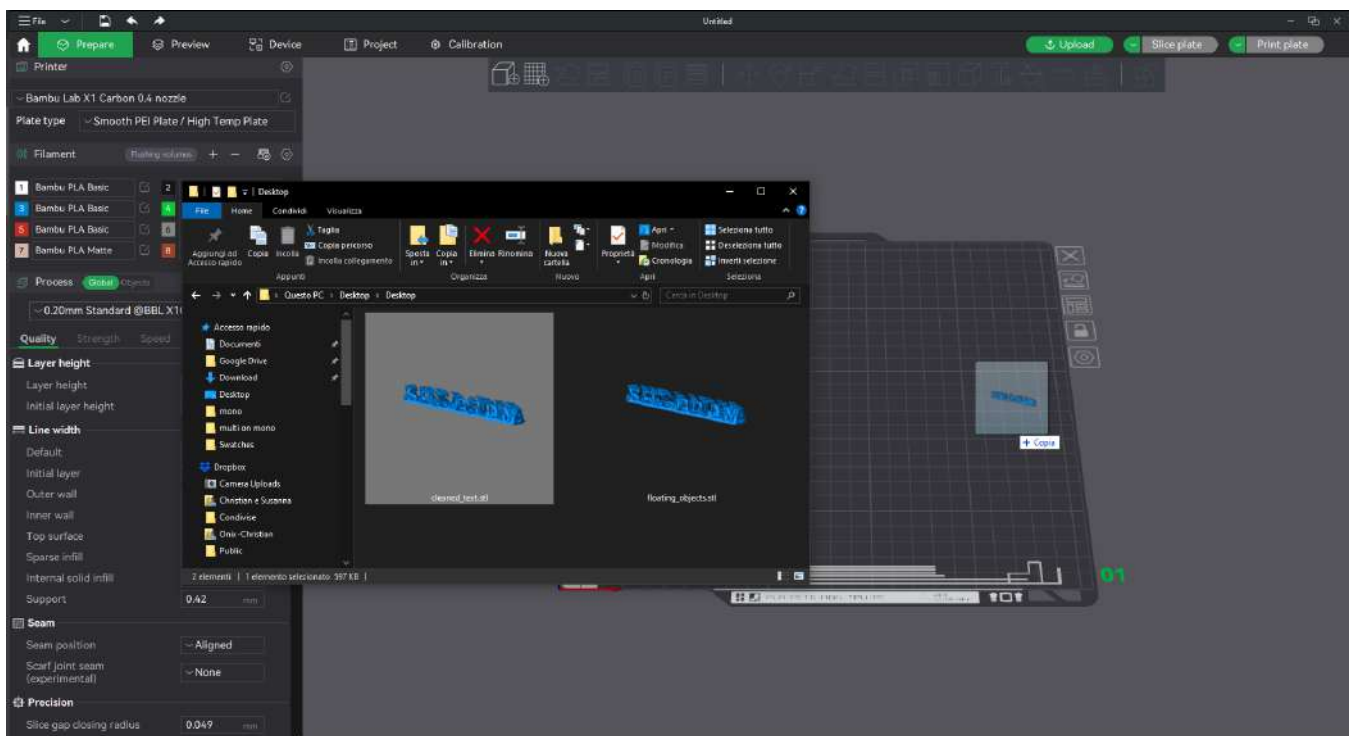
A dialog will popup, please be sure to check the **Save to my computer** option and select a valid folder, enter a name like cleaned_text, which will create cleaned_text.stl file, and finally click on **Save**.

Open Bambu Studio or your preferred slicer, start a new project and drag the file onto it.

You will almost surely need to enable supports for printing this, so please be sure to be in the **Global** section, then go to the **Support** tab, activate **Enable support** and select *tree(auto)* as **Type** - I've got the most efficient and beautiful results using this.
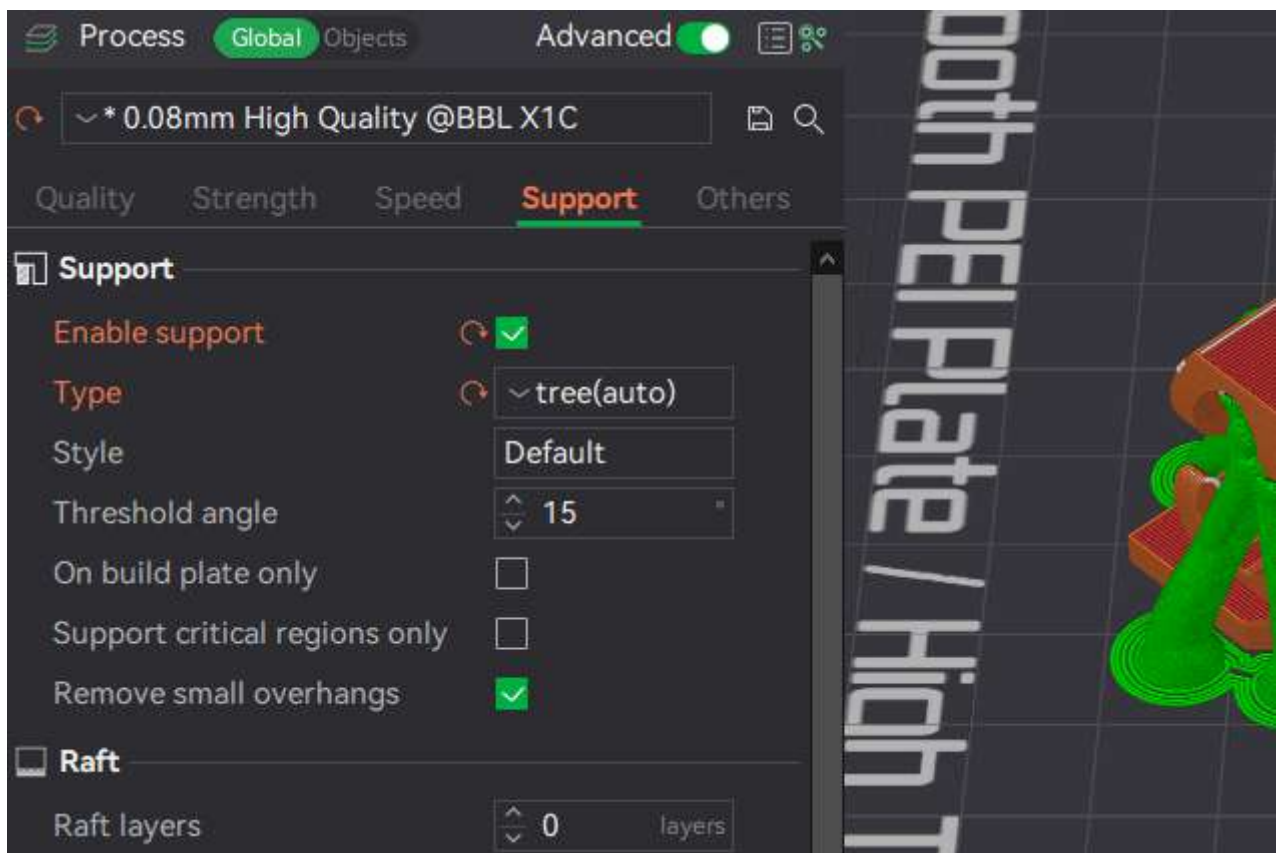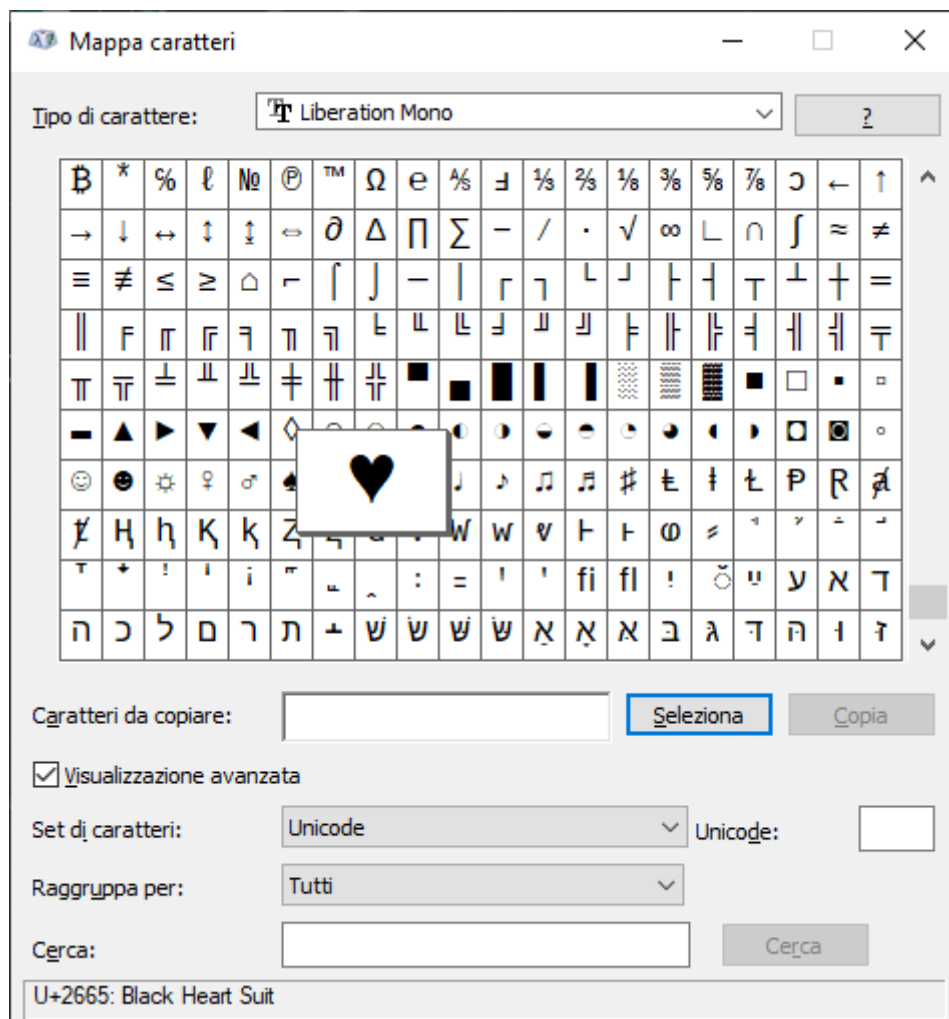
How to use symbols in texts

If you want to use symbols in texts, you can with certain restrictions.

On Windows

From the start menu, open **Characters Map**.

Select from the combobox the font you're using in your design, then scroll through characters to find the glyph you want to use, click on it, and press the **Select** button, then use the **Copy** button to copy it to the clipboard, go to the Parametric Model Maker, focus the input field and paste from the clipboard.

Optionally you can copy the character code in the lower left corner and insert its hexadecimal value in the text field pre-pending the "\u" escape character sequence, e. g. : U+2665 becomes \u2665

Alternatively you can use **Character Map UWP**
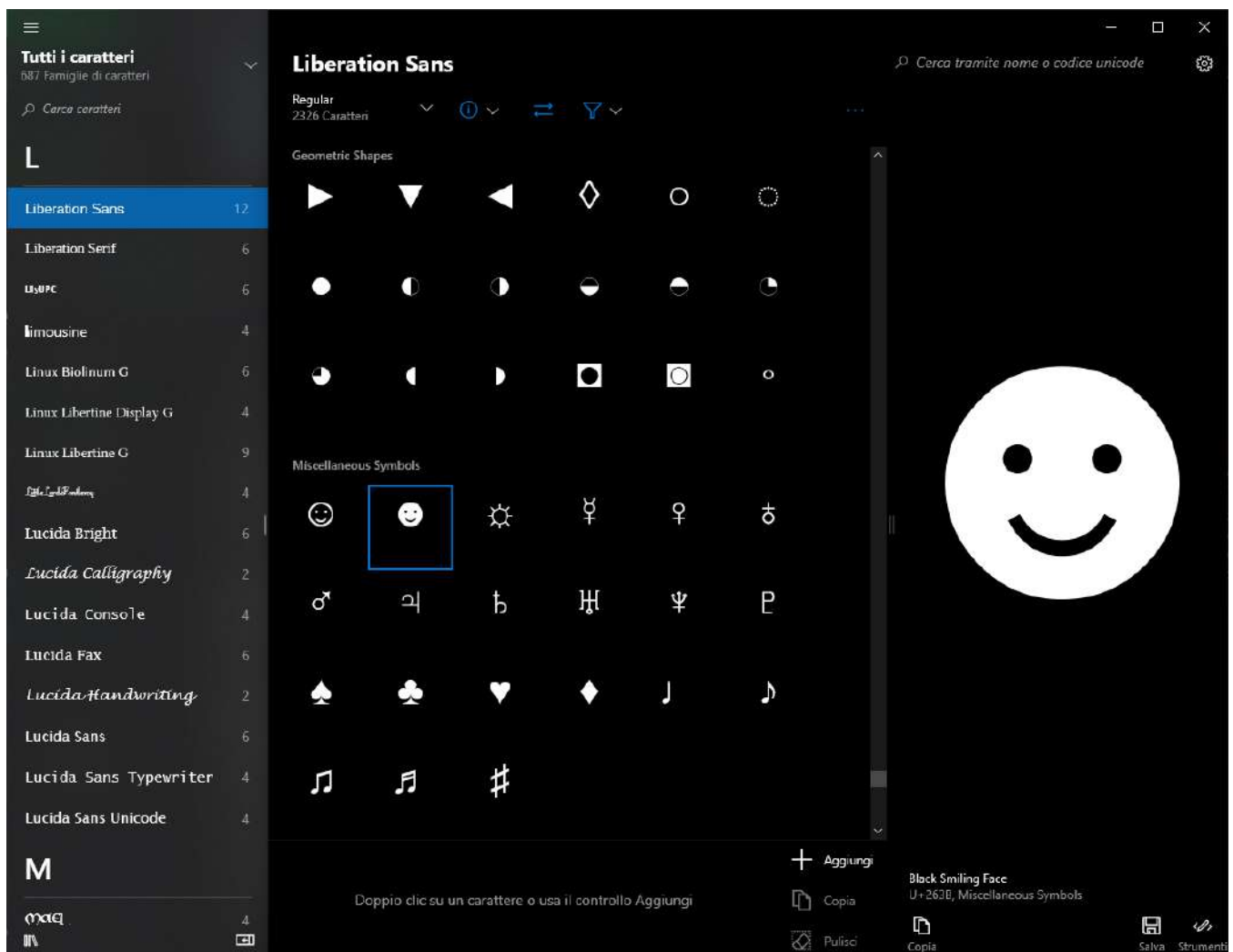
Select the font you're using from the left list, scroll through the glyphs to search the one you like most, then click it to focus it. Use the **Copy** button in the lower-right corner to copy it to the clipboard.
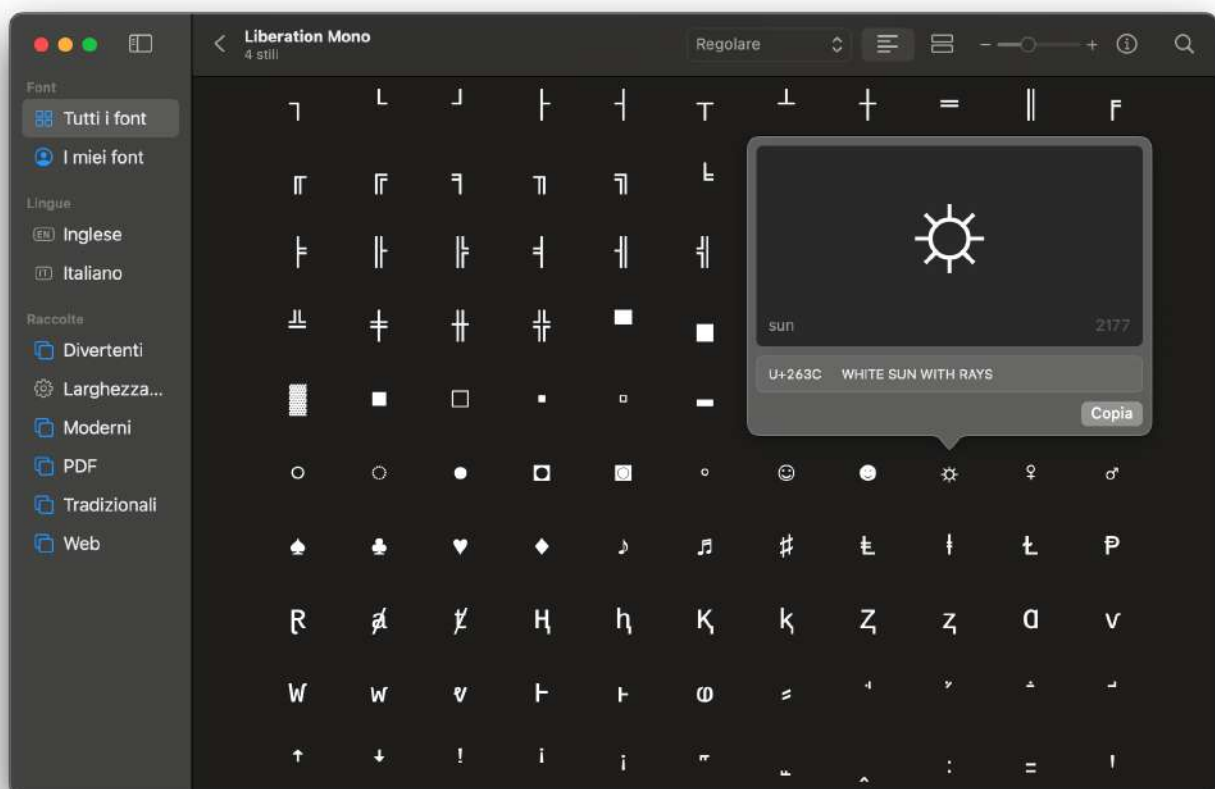
Optionally you can copy the character code in the lower right corner and insert its hexadecimal value in the text field pre-pending the "\u" escape character sequence, e. g. : U+263B becomes \u263B

On macOS

On the Apple system you have a similar tool which is **Font Book**, open it from the Applications folder.
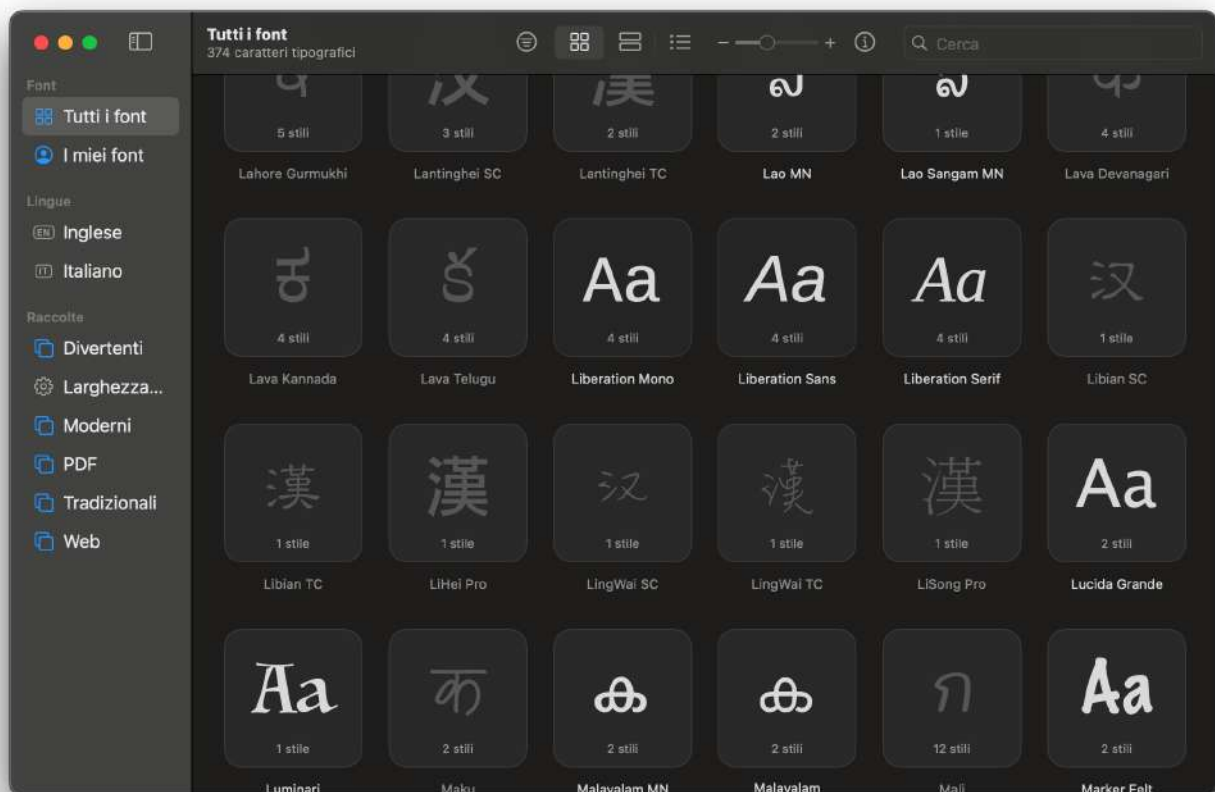
On startup you will be presented the full list of fonts installed on your system, search for the font you're using in the script and double click on it's icon to open up the font details.
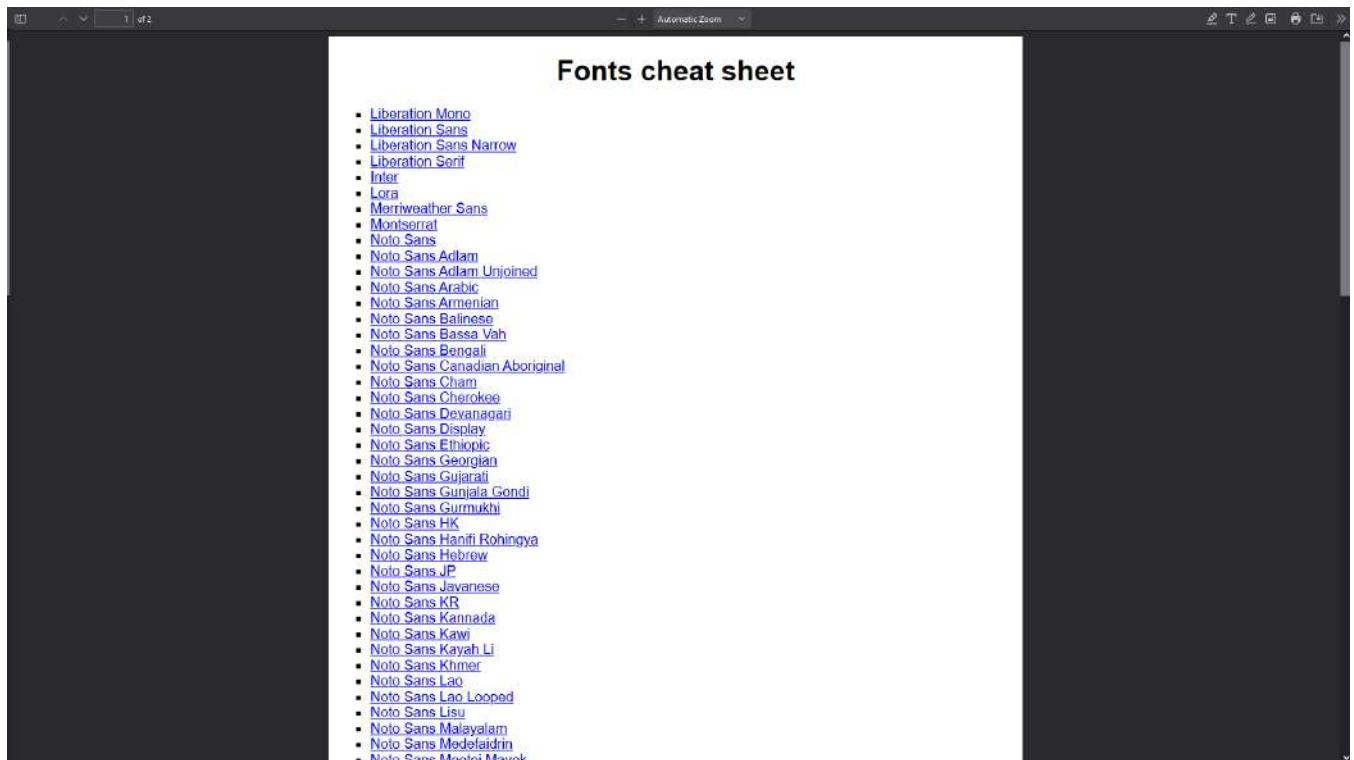

Scroll through the glyphs to search the one you like most, then click it to open it up and use the **Copy** button in the lower-right corner to copy it to the clipboard. Paste it into the script's input field.

Optionally you can copy the character code in the lower left corner and insert its hexadecimal value in the text field pre-pending the "\u" escape character sequence, e. g. : U+263C becomes \u263C.

Using the available cheat sheets

I made some cheat sheets, due to limitations of upload size and count to MakerWorld I only uploaded the index there, you will find all the font-specific files in the Fonts cheat sheet folder here on GitHub.

Open up the Fonts cheat sheet.pdf file and click the font you're using, you will be brought to the PDF file containing the cheat sheet for that font.
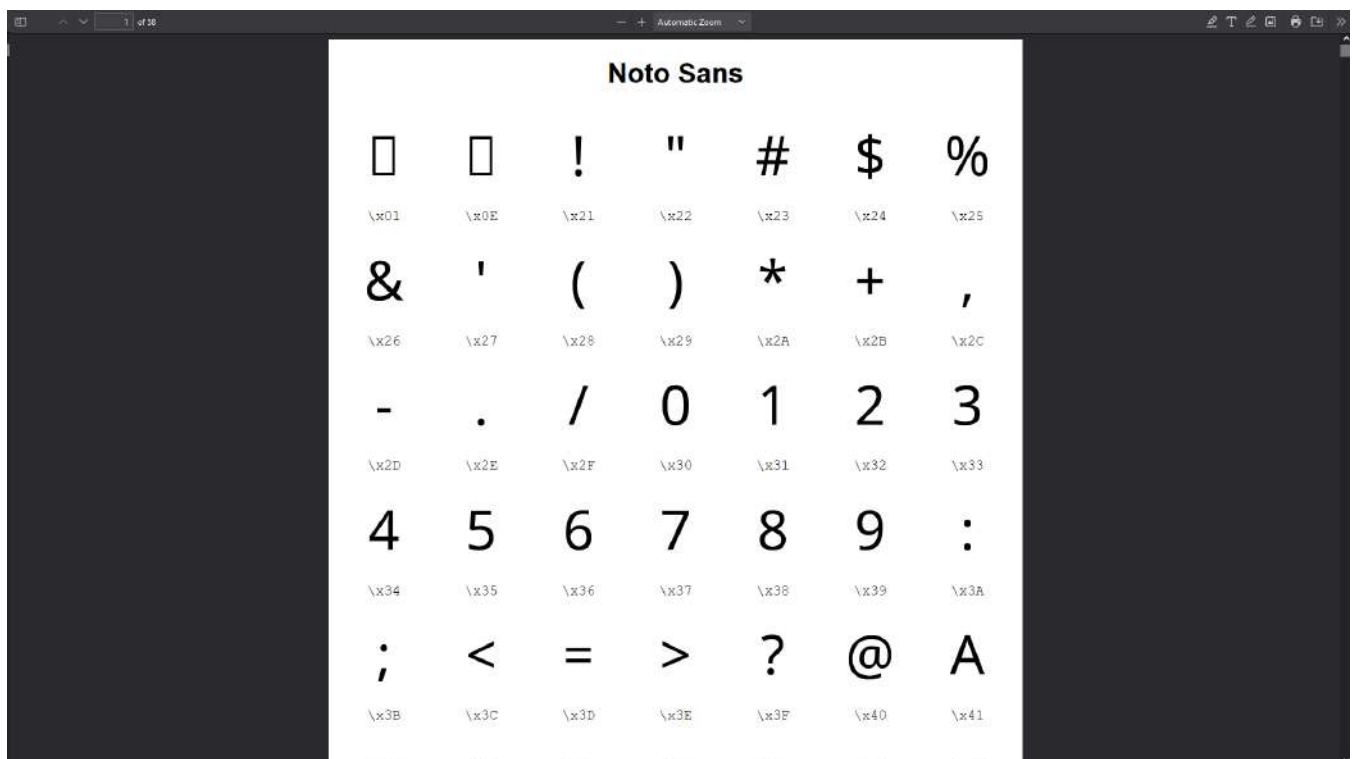
Scroll through the pages and search the glyph you're looking for, copy the code below it and paste it into the input fields.

Version history

Date	Description
2024-06-16	- a completely new rendering engine has been introduced, this engine creates cleaner objects, solves a lot of problems, is more precise with calculations and gives new opportunities for new features; - fix: when text_left was longer than text_right, the font was cut in half vertically; - fix: top was positioned a lot above the end of the object; - other minor fixes that prevented the script running as intended; - feature: new rendering engine allows to create rounded rectangle base and top plate; - feature: escaping of texts allows to specify ASCII and UNICODE characters; - improvement: heart shape is not considered experimental anymore, since with new rendering engine it's almost perfect; - improvement: font_style has a new possible value of "None"; - the new rendering engine has been set as default, but you can switch back to the legacy one by setting the render_engine parameter; - improvement: since rendering engine is default and more precise with calculations, correction offsets and parameters have been zeroed, still available because you can still switch to the legacy engine; - improvement: small characters can be defined and they will be rendered with a specific ratio in depth (only with new rendering engine); - improvement: spaces will be rendered as a very thin shape so corresponding letter can be seen (only with new rendering engine); - improvement: for texts that have different lengths, trailing and leading spaces will be equally added to match the length (only with new rendering engine); - a new PDF called "Fonts cheat sheet.pdf" including all standard fonts with all their glyphs has been added.
2024-06-11	- fix: uses baseline for text creation, this prevents letters like "J" (uppercase j) to break the design; - fix: cuts out all that goes below the baseline, preventing floating letters; - added selection of the shape mode between rectangle (legacy) and circle; - added selection of the shape heart (experimental); - added top plate creation (request from a user).
2024-06-08	First public release.

Thanks

Thanks to @Stephanos for the complete list of fonts and styles supported by MakerWorld he gave me access with his script **Parametric Model Maker ALL TEXT FONTS TEMPLATE** you can find here.