Tutorial for creating housings in FreeCAD

Introduction

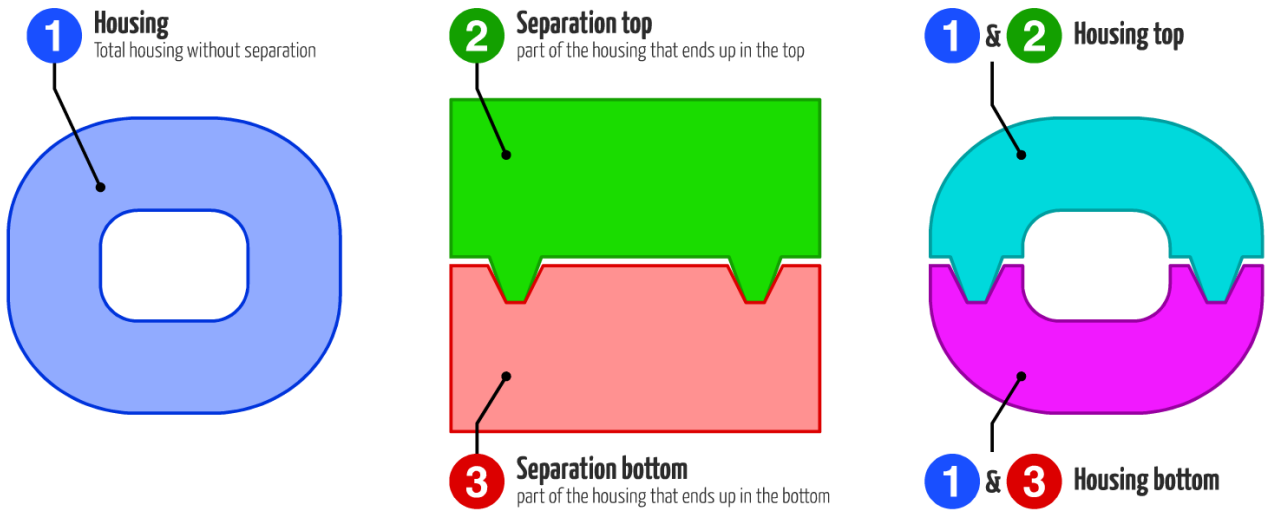
This tutorial demonstrates how to design 3D printable housings in FreeCAD. The advantage of this approach is that it is quite structured, and you can manage changes in the design quite well, even if the design gets complex.

To follow this tutorial, you need to be familiar with the part design workbench and the sketcher. I'll try to just focus on a high conceptual level.

Topics in this document

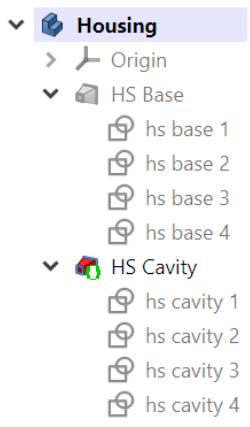
Concept of making a housing using boolean operation of bodies

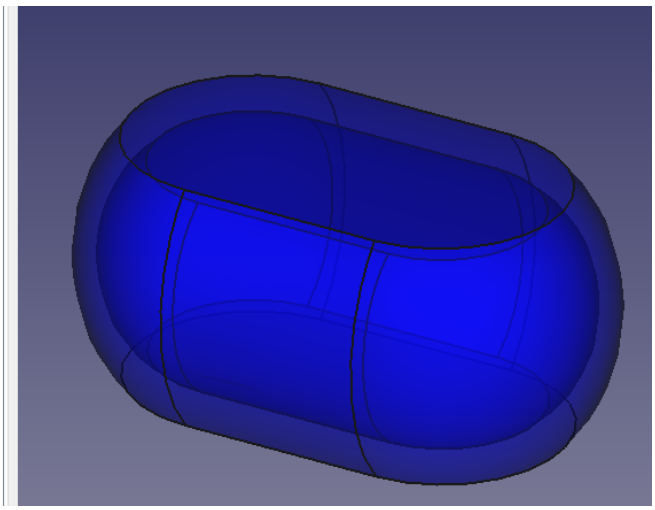
In this example, we will demonstrate the process of constructing a housing through the application of boolean operations on various bodies. The housing will consist of two shells that can be seamlessly assembled together. Follow these steps:



1. Housing Creation:

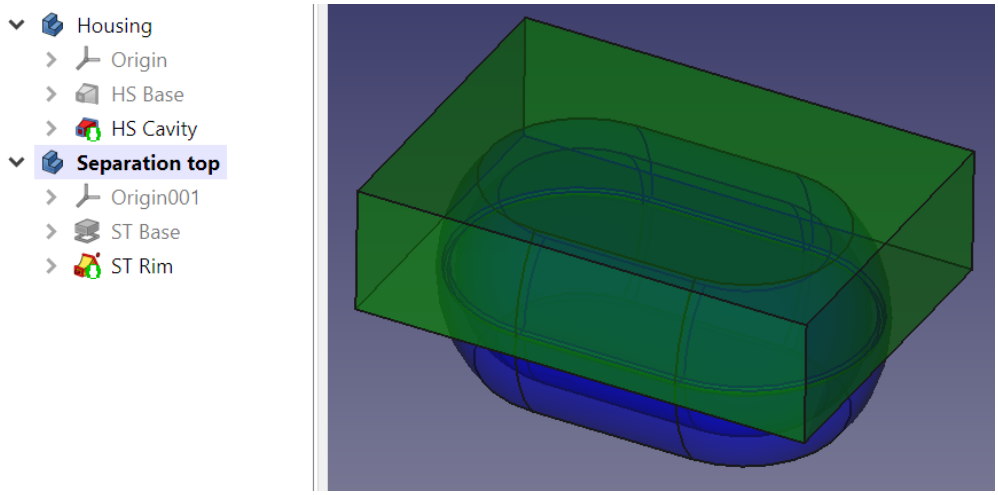
Begin by forming the **housing body** (1) as a singular body, disregarding any separation concerns for now.





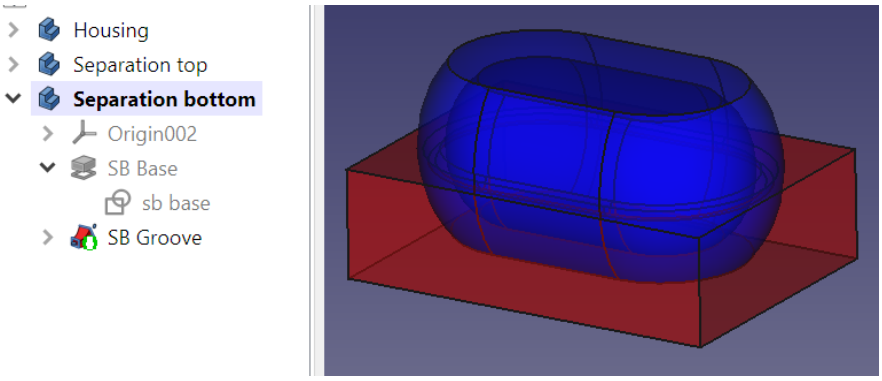
2. Top Separation:

Generate a second body (2) called **Separation top**. This component envelops the volume of the upper section of the housing.



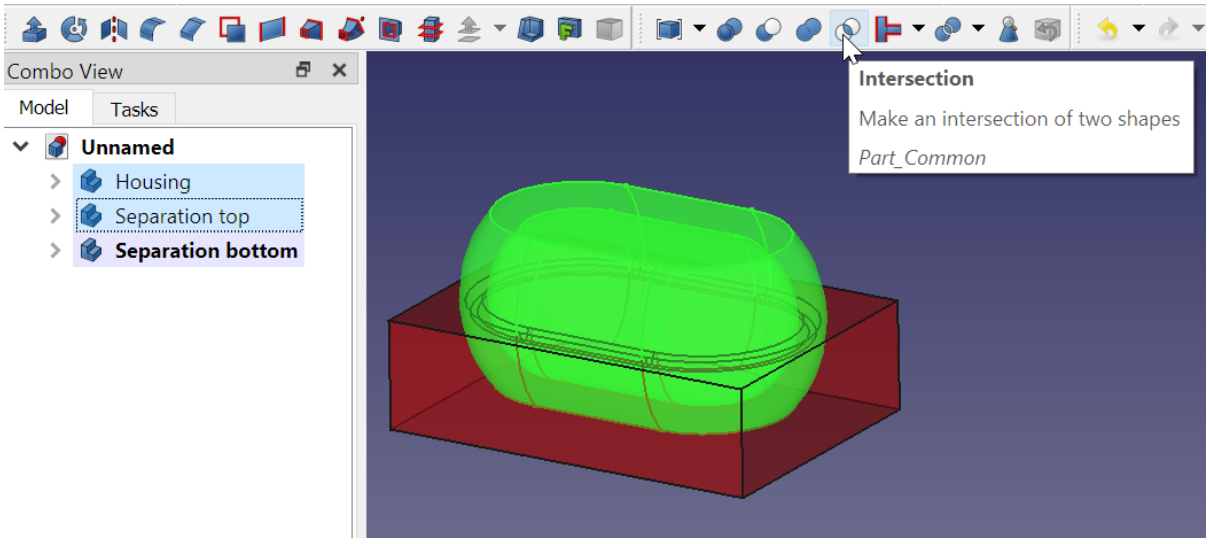
3. Bottom Separation:

Similarly, create another body (3) named **Separation bottom**. This part defines the lower separation boundary.



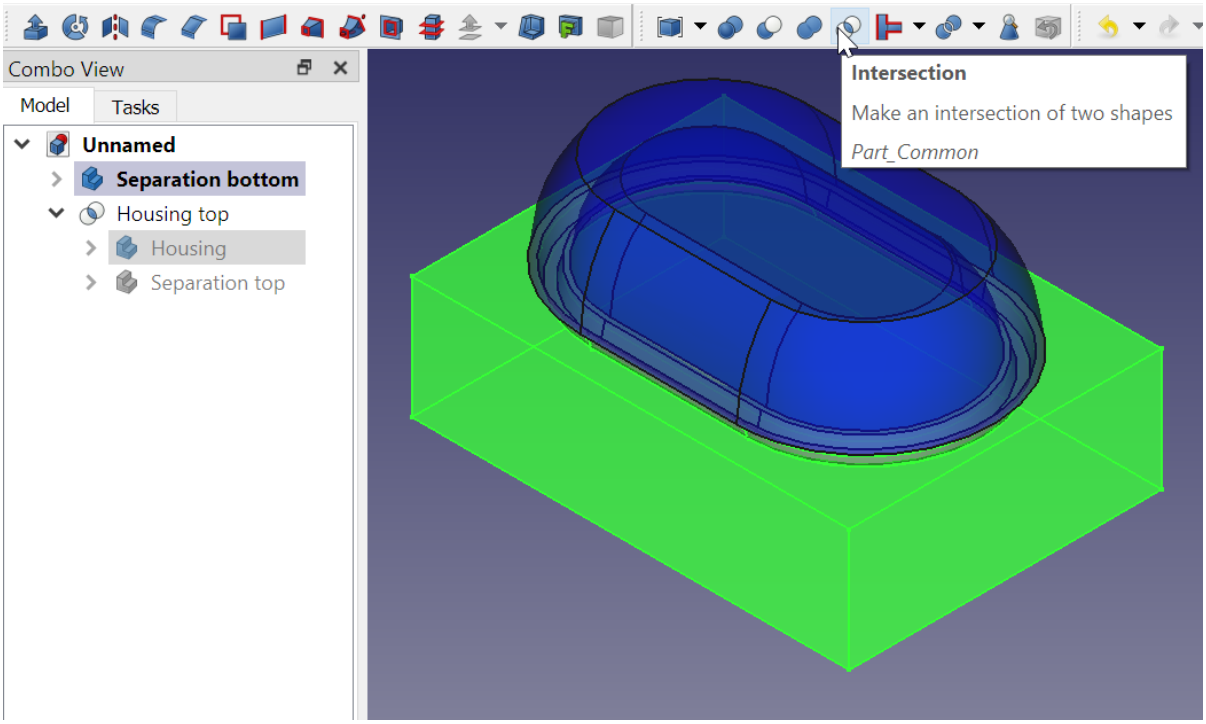
4. Boolean Operations for Top Housing:

Switch to the Part workbench. Select the **Housing** body and the **Separation top** body. Use the `Intersection` command from the toolbar to generate a new body, initially named **Common**, which embodies the boolean intersection of the selected bodies. Rename this new body as **Housing top** (1 & 2).



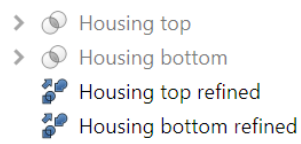
5. Boolean Operations for Bottom Housing:

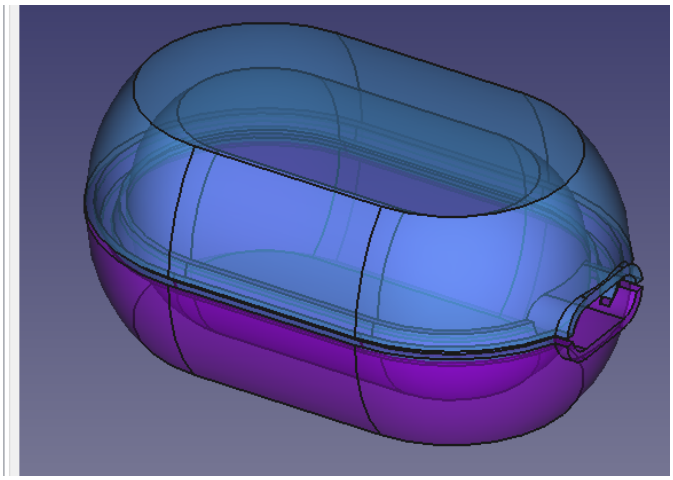
Although the **Housing** body may appear to have vanished from the model tree, it still exists within the **Housing top** body, as it contributed to its formation. By expanding the **Housing top** body, you'll find the **Housing** body within. To establish the **Bottom housing** body, select the **Housing** body along with the **Separation bottom** body. Reapply the `Intersection` command to these chosen bodies, resulting in a new body named **Common 001**. Rename this resultant body as **Housing bottom** (1 & 3).



6. Finalizing Appearance:

Conclude the process by adjusting the color and transparency attributes of both housing components.



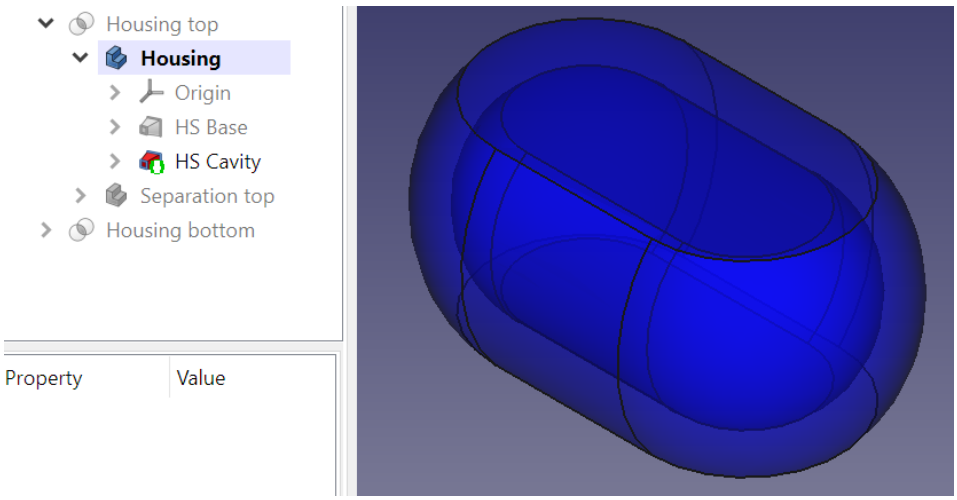


Modifying the Housing Design

An aspect to consider within this approach is that the ability to directly modify both the **Housing top** and **Housing bottom** components using the part design workbench becomes limited. Nonetheless, in practical scenarios, this limitation is inconsequential as modifications can still be efficiently executed on the three original bodies. However, the decision-making process gains significance concerning which specific body to target for the intended alteration. A prime example underscores this point: when the housing serves to accommodate electronic components and requires integration with a power cable, it proves simpler to incorporate these features into the original **Housing** body.

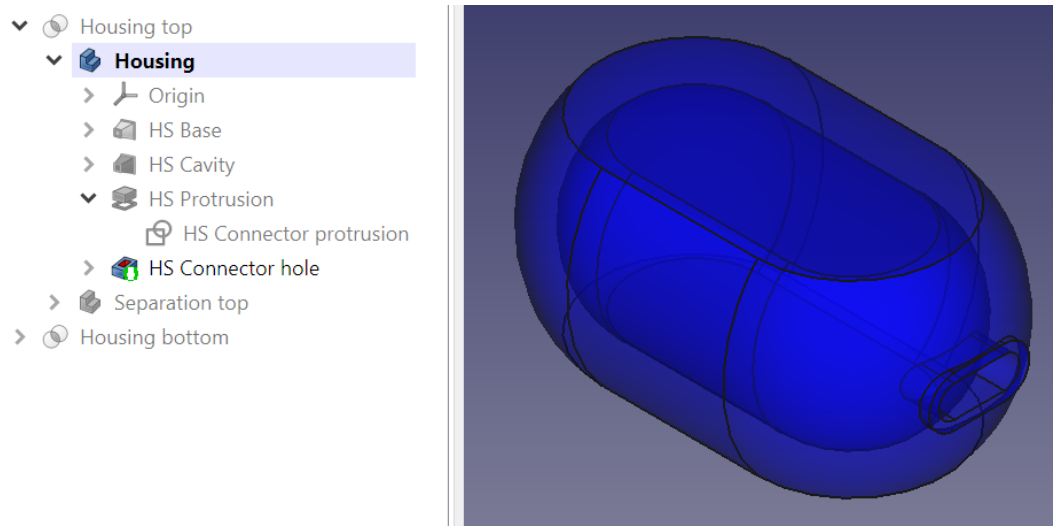
1. Making Housing the active component

Begin by ensuring that the **Housing** body is exclusively visible and designated as the active component. This entails switching to the Part design workbench and a double-click action on the **Housing** body.



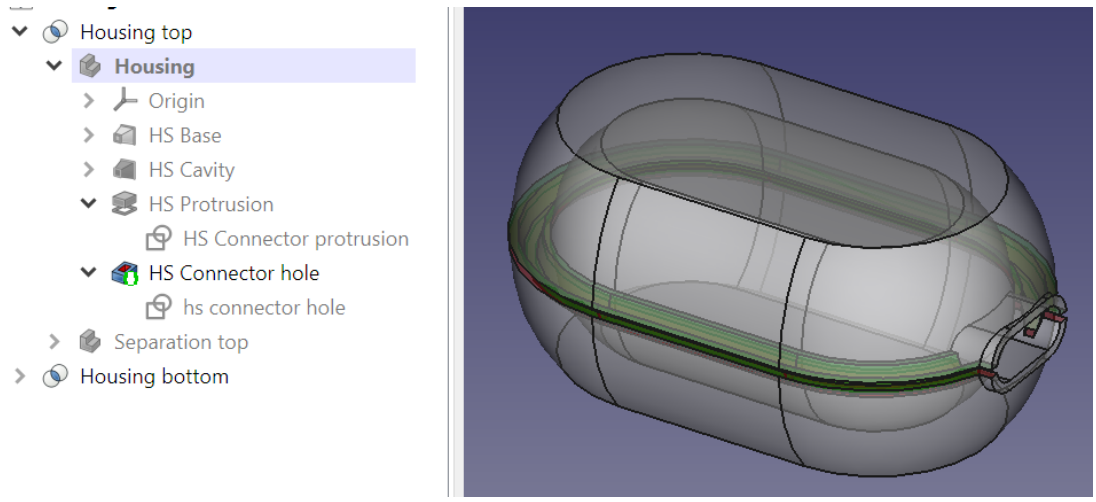
2. Modifying the housing body

With the "Housing" body selected, proceed to introduce necessary changes, such as adding a protrusion and a hole to facilitate the integration of a power connector.



3. Reviewing the result

Following the modification process, by concealing the **Housing** body and making the **Housing bottom** and **Housing top** bodies visible once again, it becomes evident that both these components have undergone alterations as well.



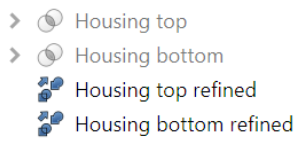
This operation overwrites the original colors assigned to the **Housing bottom** and **Housing top** bodies. While it is indeed possible to rectify these colors once more, the repetitive nature of this task can become cumbersome. To circumvent this, a simple workaround is applied to avoid recurrent color adjustments.

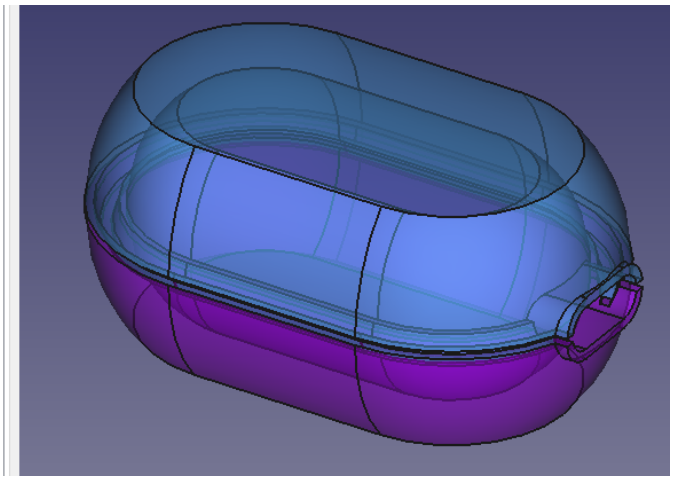
Maintaining the colors of both housing bodies

One of the most straightforward solutions to maintain the colors of the bodies involves the creation of duplicates of the bodies in question, followed by the application of the desired color scheme to these replicated entities.

1. Transition to the Part workbench
2. Select the **Housing top** body.
3. Choose `Part > Create a copy > Refine shape` from the menu
4. Rename the duplicated body as "Housing top refined"
5. Adjust both color and transparency of the copied object

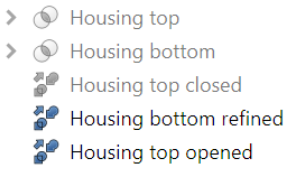
Repeat the process for the **Housing bottom** body.

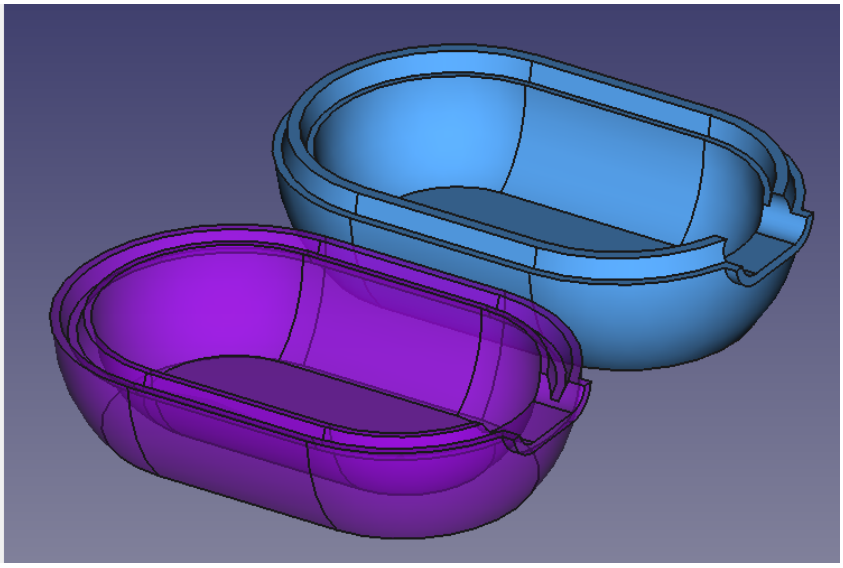




It's worth noting that the color alterations applied to the refined duplicates will remain unaffected even when modifications are made to the original bodies.

An additional benefit stemming from this workaround is the ability to generate multiple copies, each positioned differently. This grants the convenience of inspecting the components from various angles by selectively rendering specific combinations visible.





Applying a naming convention

As the quantity of bodies and features within the model tree expands, the process of identifying the precise feature for necessary modifications becomes progressively complex. To mitigate this challenge, adopting distinct and meaningful names for features proves invaluable. In this context, I have devised a systematic naming approach, outlined as follows:

Type of feature	Case	Code
Volumetric features	Sentence case	`Bb Nnnnn`
Non-volumetric features	lowercase	`bb ttt nnnnn eee`

Non-volumetric features such as sketches and planes employ lowercase for their names. For these instances, a combination of body abbreviation, feature type code, and name extension is employed (bd typ name ext).

Volumes use Sentence case names (e.g., a sketch may be named **hs base**, and the pad that is created using that sketch is named **HS Base**)

Typically, the last sketch to define a volumetric feature has the same name as the volumetric feature, but they become distinct by the use of different case.

Code	Meaning
`bb`	A unique abbreviation of the body (**HS** for housing, **ST** for separation top, **SB** for separation bottom, etc.)
`ttt`	A 3 letter code for the type of feature after that abreviation: - **pln** for a plane - **axs** for an axis - **ref** for a shape binder
`nnnnn`	The name of the feature
`eee`	For additive and subtractive pipes: - the trajectory of the pipe is followed by **trj** - the cross section is followed by **crs**

Examples:

Name of the feature	Purpose
`sk top`	Top view of the part in the **Skeleton** body
`hs ref top`	Shape binder in the housing body, referencing the top view in the **Skeleton** body
`hs pln bottom`	Datum plane in the housing body, representing the bottom of the **Housing** body
`sb groove crs`	Cross section of the groove in the **Separation bottom** body
`sb groove trj`	Trajectory of the groove in the **Separation bottom** body
`SB Groove`	The 3D groove in the **Separation bottom** body, made up of **sb groove crs** and **sb groove trj**

It is helpful to choose a pragmatic approach: for simple projects, the overhead of renaming every feature may not be worth the effort.

Using a skeleton to drive dimensions of the bodies

First steps

To make the design truly parametric, it is helpful to create links between the different bodies. For instance, the rim is defined in the **Separation top** and **Separation bottom** bodies, but they need to follow the contour that is defined in the **Housing** body. It's crucial to note that the referencing between bodies is one-way; once Body B's features reference Body A, a reciprocal reference from A to B is no longer possible to prevent circular references.

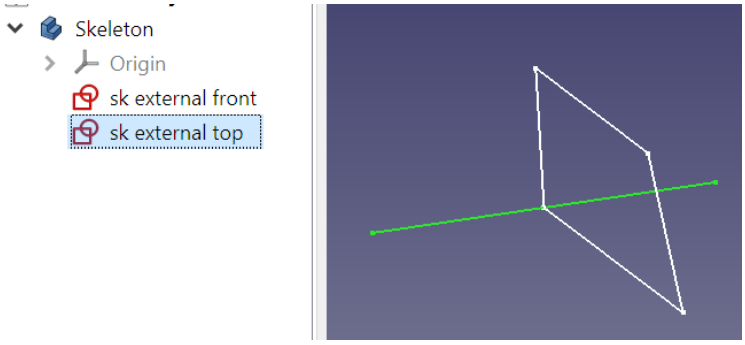
To maintain a structured approach, an effective strategy is to initiate with a **Skeleton** body. This specialized body encapsulates fundamental shapes and dimensions, devoid of volumetric representations. Consequently, this **Skeleton** body serves as a reference for other bodies.

Additionally, a **Skeleton** body enhances model robustness. In cases where sketches reference 3D geometry, such as body edges or faces, making minor alterations can trigger instability due to the notorious Topological Naming Problem. To circumvent this, sketches should refer to sketches, rendering them less prone to naming changes. Simple, smaller sketches are preferable over complex ones.

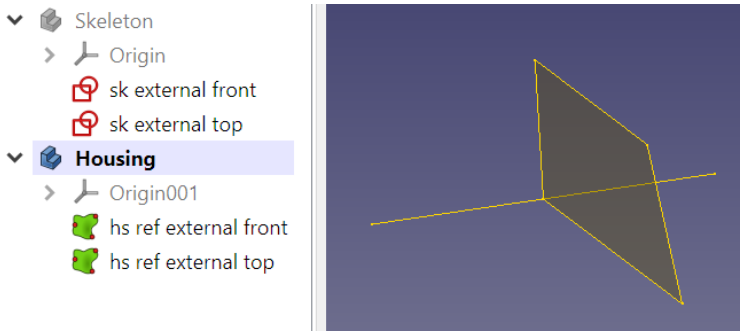
When we use tools such as a pad or a pocket to extrude geometry, the geometry is partially driven by a sketch, but also by the number that defines how long the extrusion is. In order to make the design fully driven by sketches from the **Skeleton** part, we choose another approach.

Here's how it works:

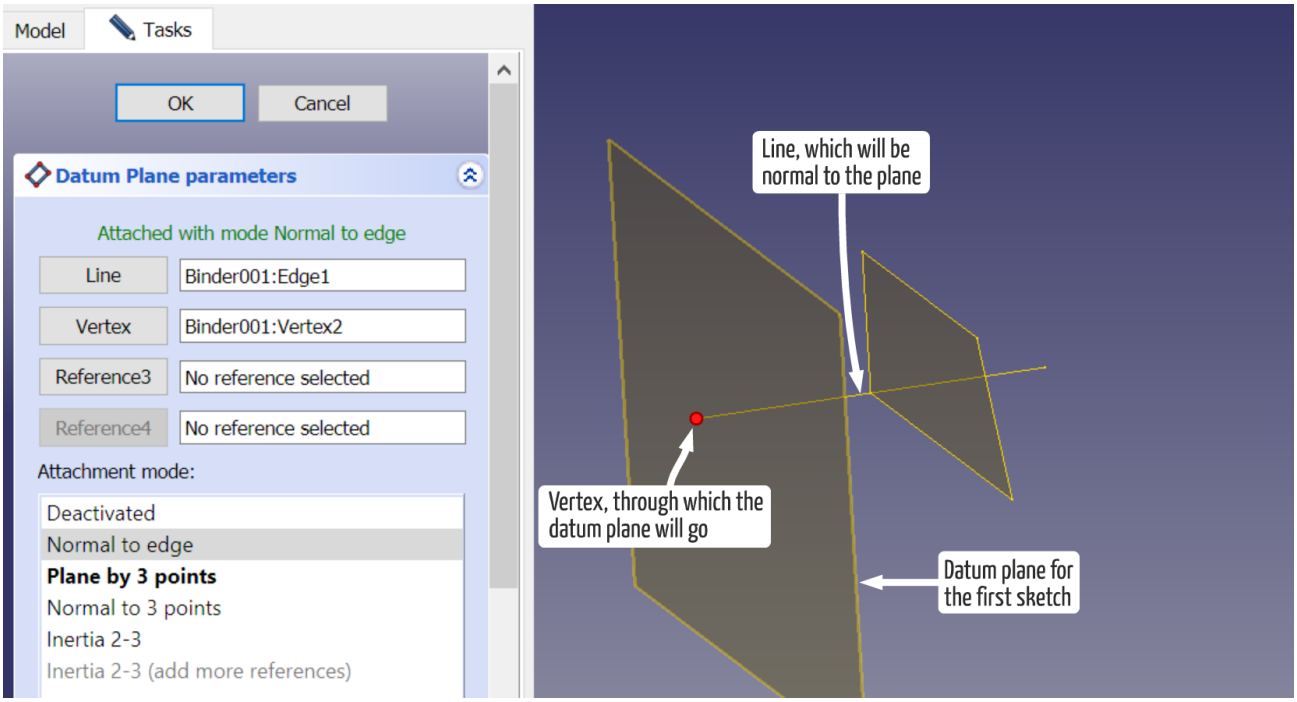
1. Create a sketch in the **Skeleton** part, which contains an edge with an endpoint in the plane where the extrude begins and another where the extrude stops. Also create a sketch which can be referred to for the shape.



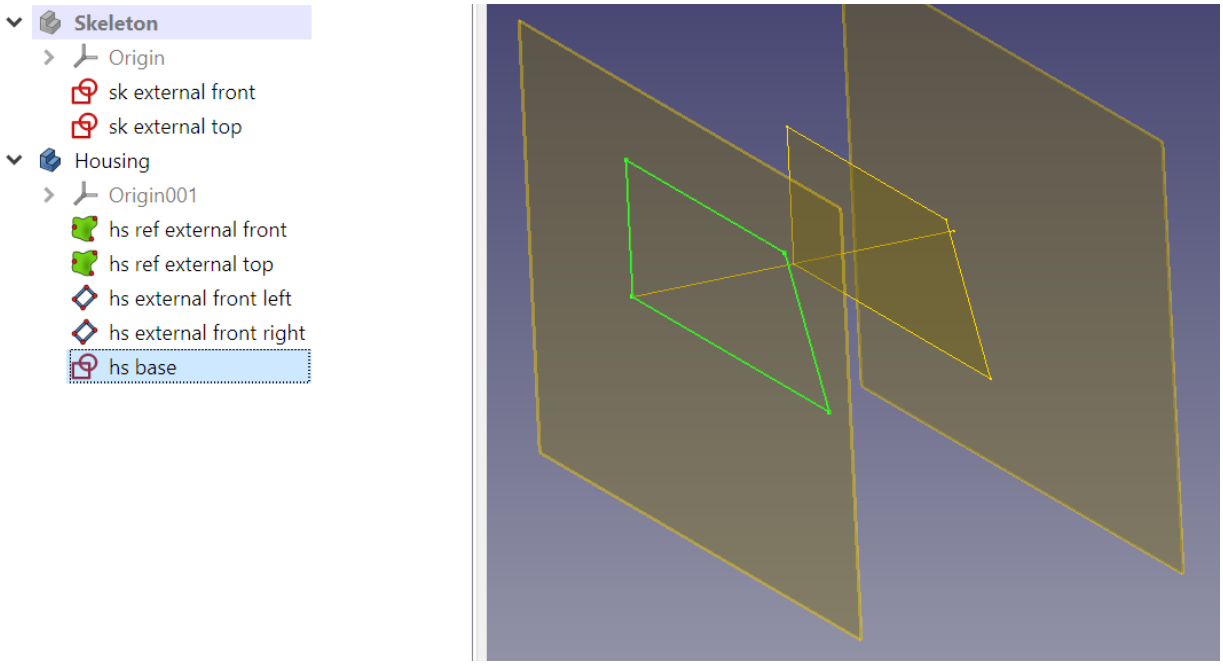
2. Create a body named **Housing**
3. Create a copy of the sketches we need using a Shape Binder



4. Create two datum planes: one at the beginning of the extrude and one at the end. To define the planes, we use an edge and a point, and we define the datum plane 'normal to edge'.

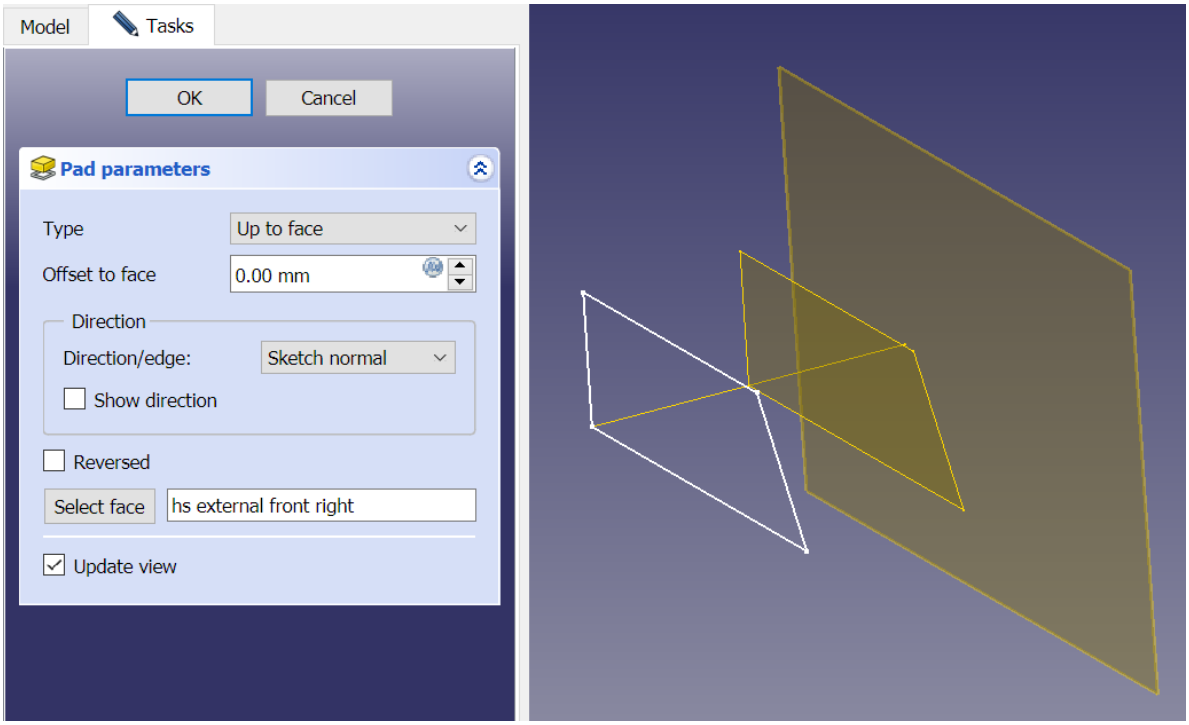


5. Ensure that the **Skeleton** body is made invisible, to avoid referring to sketches in this body.
6. Create a sketch on the first plane. The geometry in the sketch can refer to another sketch that is derived from the **Skeleton** part using a Shape binder



Note: The 'front side' of the datum plane, on which a sketch is created, is sometimes counter intuitive, so it seems as if you need to draw a mirrorred sketch. To solve this, set the 'Map reversed' property of the datum plane to 'True'. It is best to do this early on in the process, since it often corrupts the sketch.

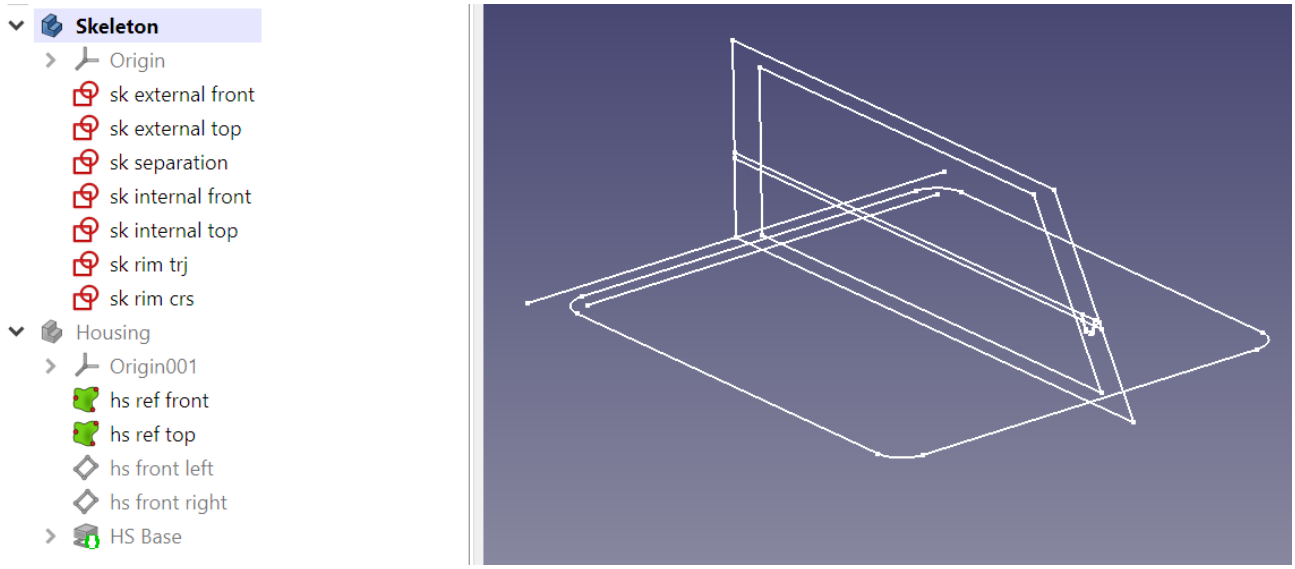
7. Extrude the up until the other datum plane. When using the 'select face' button, the plane can also be selected in the model tree.



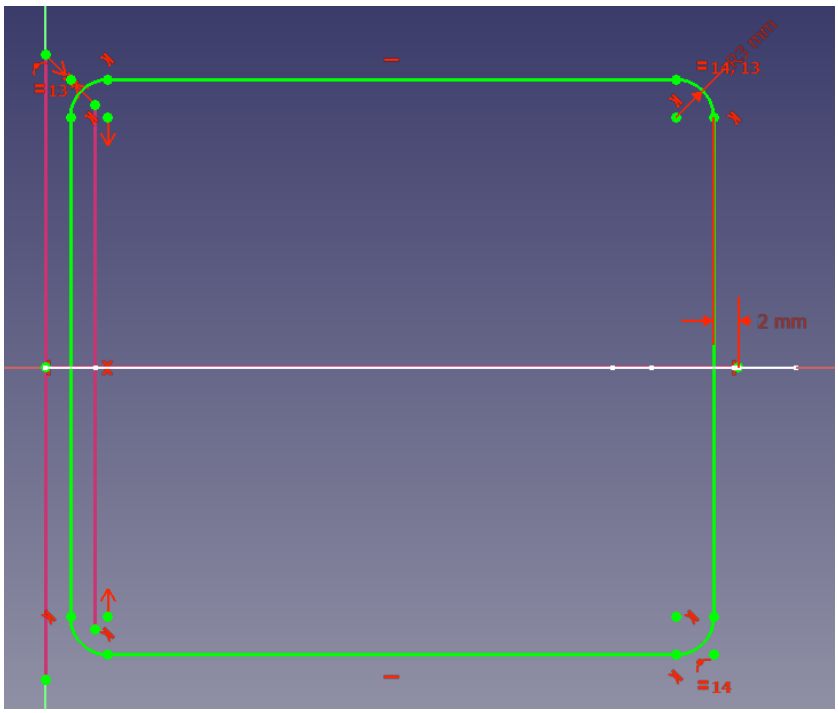
Finalization

Next, we take some bigger steps to complete the housing. We add a number of sketches to the **Skeleton** body:

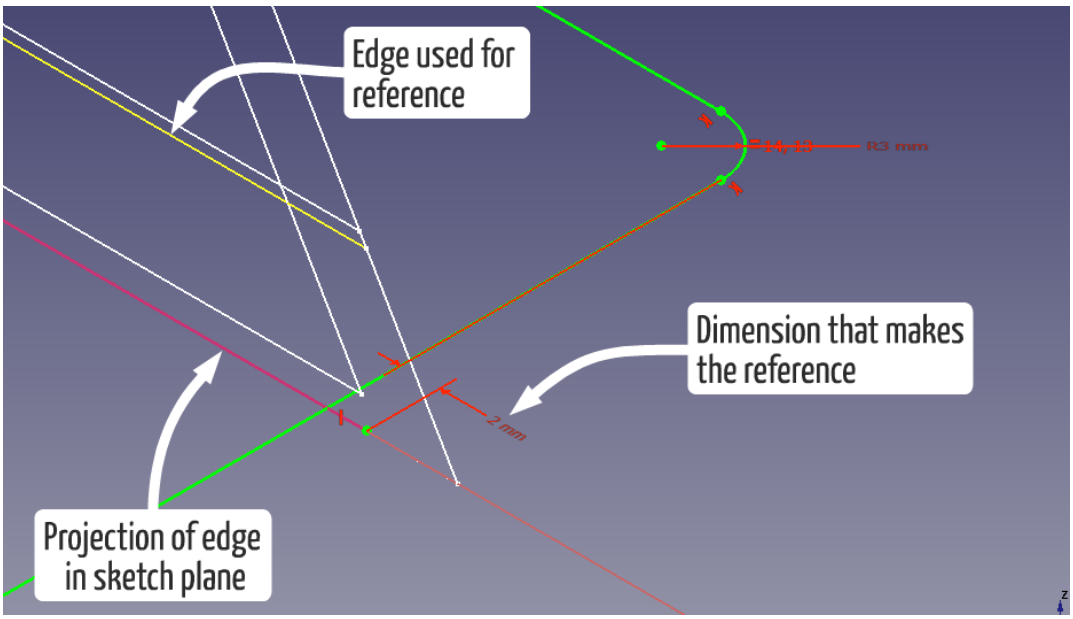
- **sk separation**: the separation lines for both the top and the bottom separations
- **sk internal front**: the front view of the cavity inside housing
- **sk internal top**: the top view of the cavity inside housing
- **sk rim trj**: the trajectory that the rim and the groove must follow
- **sk rim crs**: the cross sections of both the rim and the groove



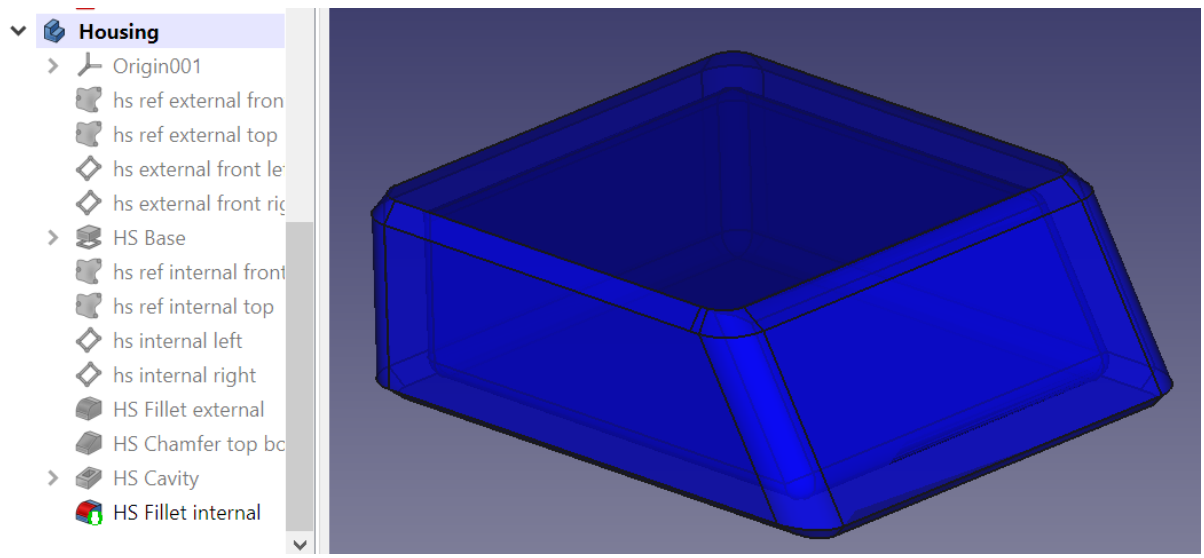
Note that a sketch can make references to sketches in other planes. For instance, the right edge of **sk rim trj** references a line in the **sk separation** sketch:



To make such references, switch to ISO view:

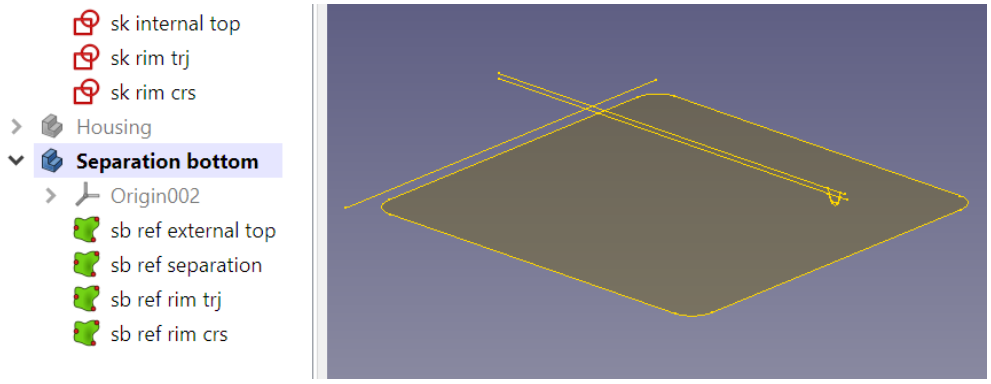


The finish the **Housing** body:

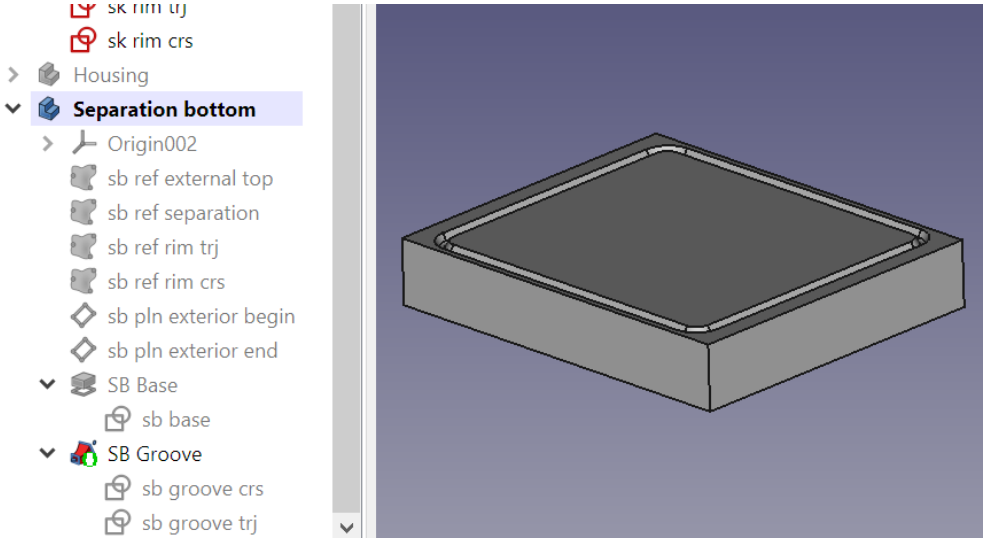


Note that chamfers and fillets were added, but they are just defined in the **Housing** body, without references to the **Skeleton** body.

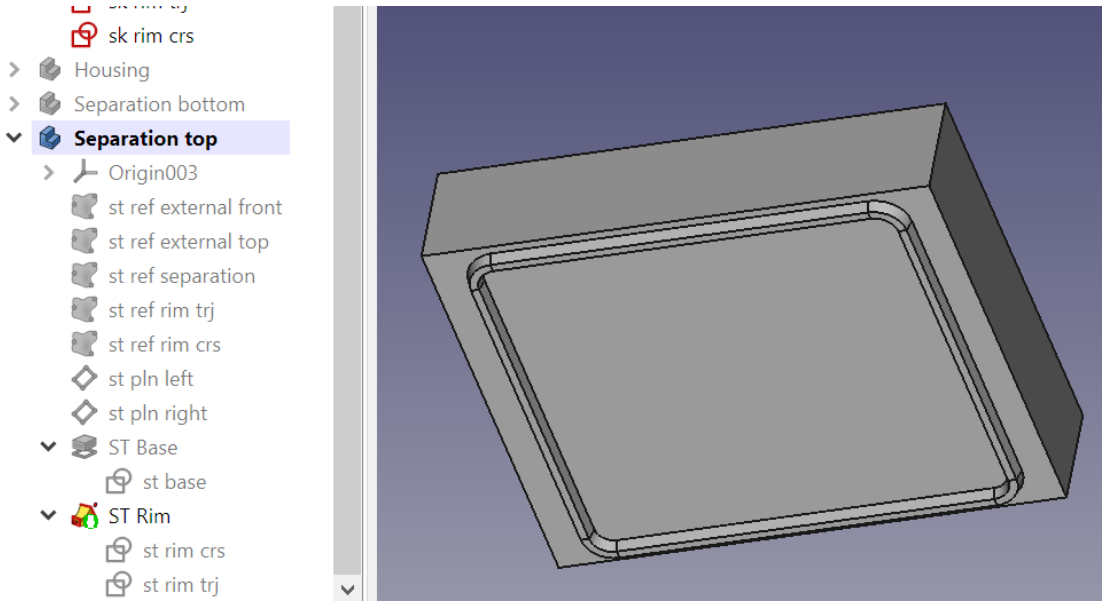
The **Separation bottom** body also has a few necessary shape binders referencing the **Skeleton** body:



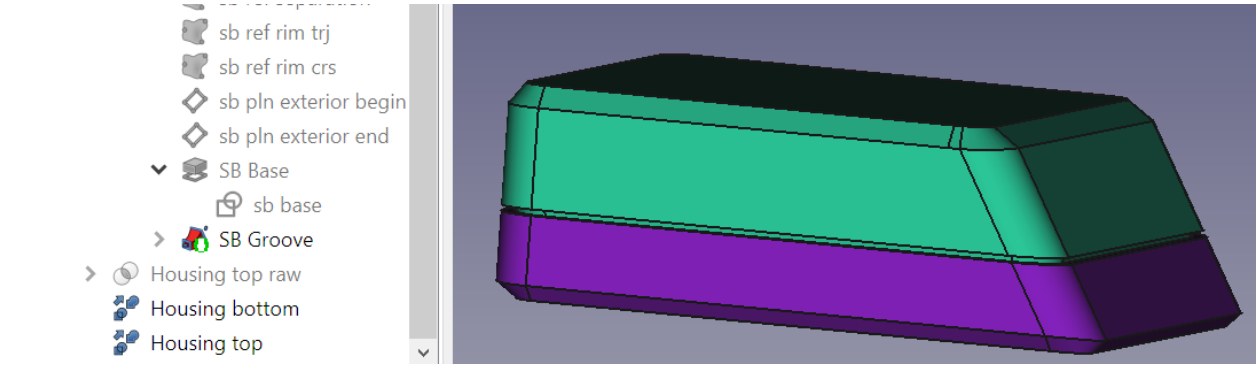
This is wat **Separation bottom** looks like when it is completed:



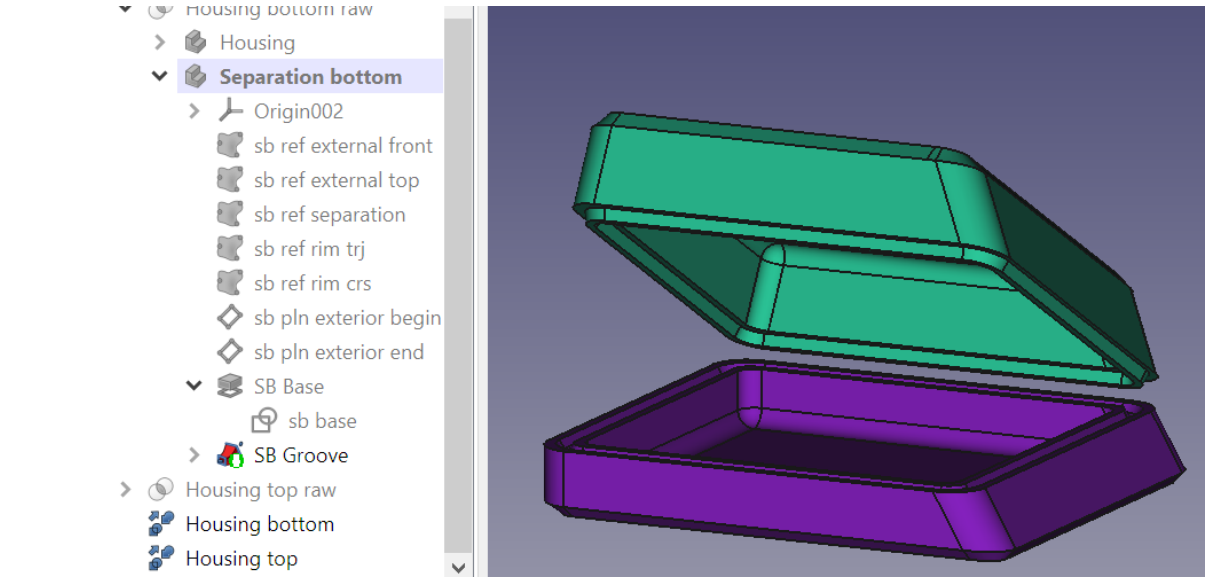
Separation top is very similar:



The final result looks like this:



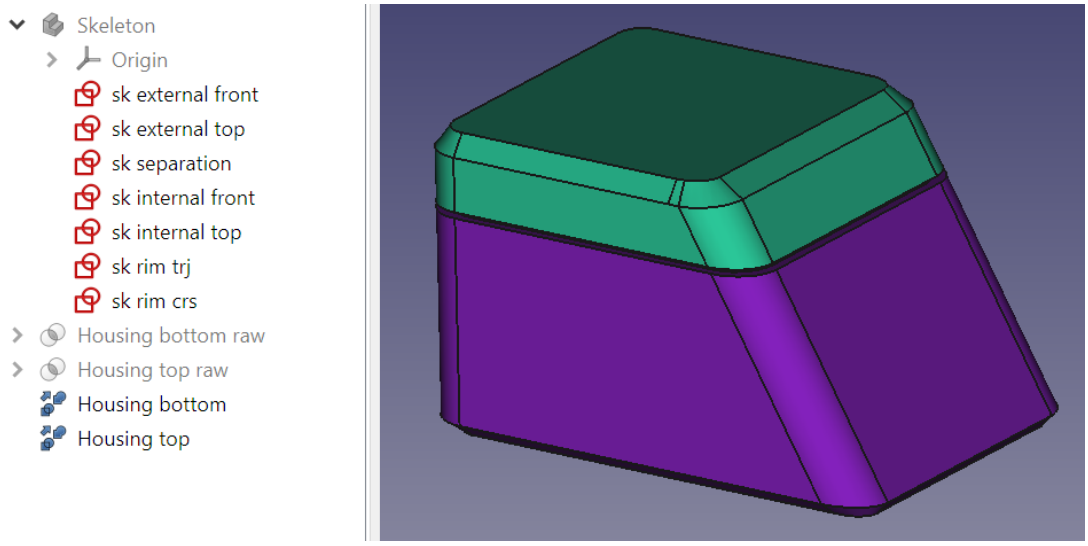
And with the top off:

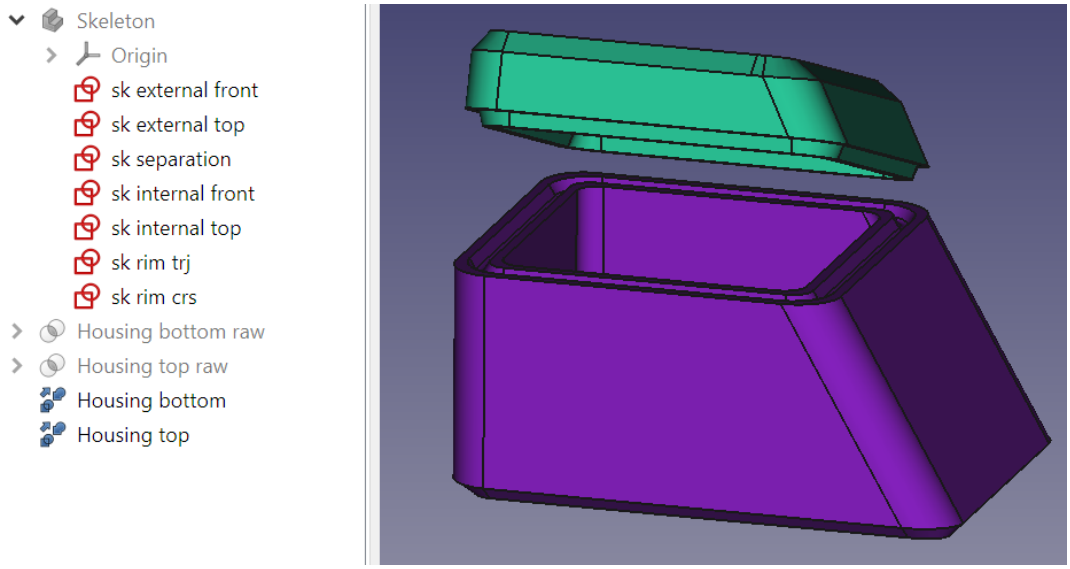


Note that:

- the main outer dimensions of the housing can be changed by only changing dimensions in the **Skeleton** body
- not all sketches from the **Skeleton** body have been imported, e.g. the rim is not needed in the **Housing** body
- details which are independent from other bodies (such as the chamfer), were only defined in the **Housing** body

The proof of the pudding is in the eating. We change a few dimensions in the **Skeleton** body to see if the model is indeed parametric. The result is as expected:





Checking the model

Using the Check geometry tool

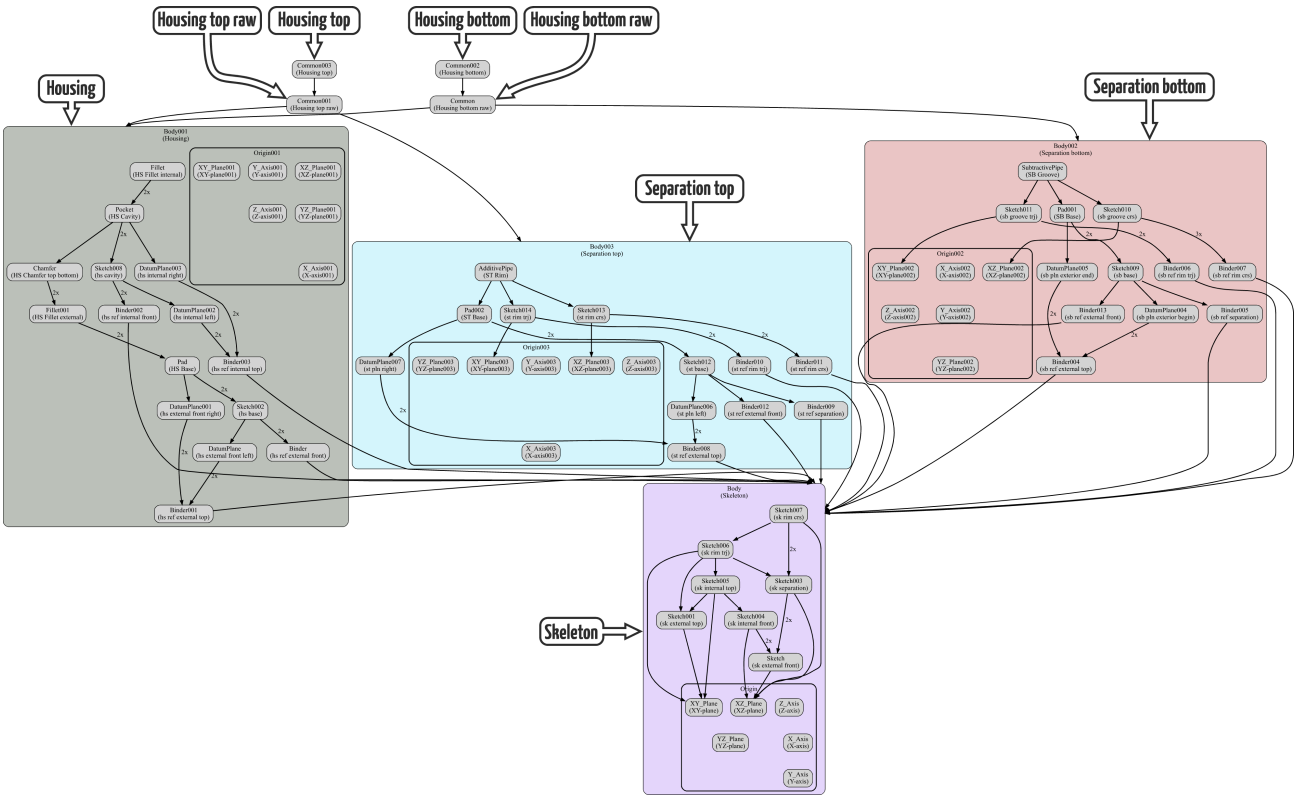
The Check geometry tool from the part workbench can be used to check if the 3D model is valid (Part workbench > Part > Check geometry). It is beyond the scope of this tutorial to explain how to solve common problems. MangoJelly has an excellent video on this tool. If causes are hard to find, the FreeCAD community is also willing to help out.

Dependency graph

It can sometimes (although rarely) occur that links between bodies cause errors that are very hard to find. Sometimes the problem is that there are crosslinks between bodies, i.e. body A refers to body B and body B refers back to body A. This circular reference causes FreeCAD to stop automatic recalculation of the part.

The dependency graph (menu Tools > Dependency Graph) can be very helpful to spot those errors. To use this tool, the third party software Graphviz must be installed (see https://wiki.freecad.org/Std_DependencyGraph).

The dependency graph of the housing looks like this (text balloons were added manually to improve readability):

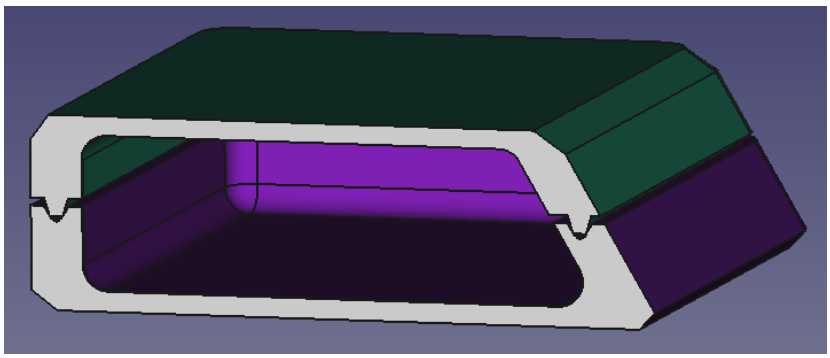


The graph shows that:

- All bodies directly or indirectly refer to the **Skeleton** body
- Body **Housing top** refers to **Separation top** and **Housing**
- References made by the **Part workbench** act on bodies, while references made by the **Part design workbench** act on features
- None of the arrows are red, indicating there are no errors in this graph

Persistent section cut

Using the persistent section cut (View > Persistent section cut) interfaces can be visually inspected in detail:



This tool was significantly improved in FreeCAD 0.21.

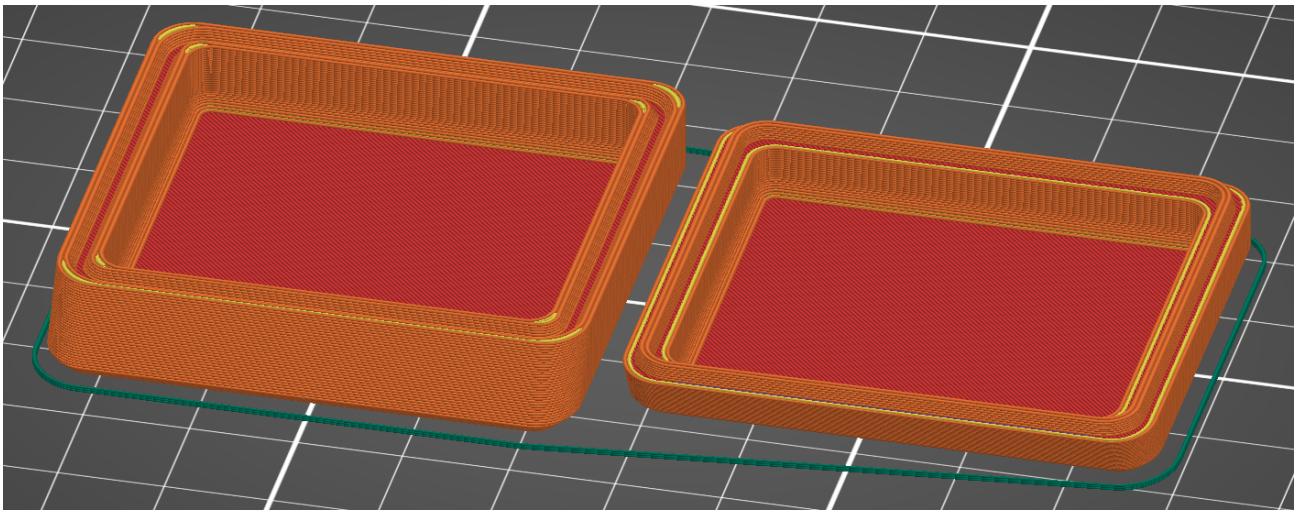
Checking the result in the slicer

I'm using this technique often for 3D printing projects. One of the lessons I learned the hard way is that it is important to regularly check if the parts are printable.

Things to specifically pay attention to:

- are all details still large enough to print?
- would a different orientation of the separation plane make printing easier?
- is it possible to avoid support structures easily?

- is it possible to reduce print time by making other design choices?



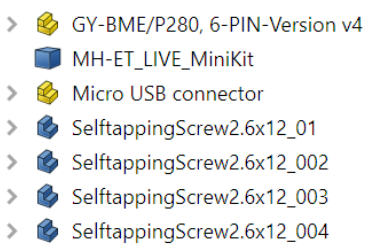
As can be seen in this screenshot, both the top of the rim and the sides of the the groove are printale with multiple adjacent tracks.

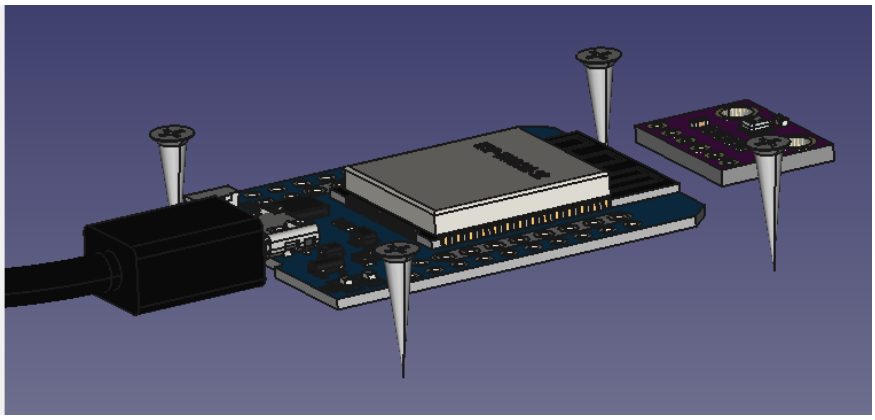
Creating references to the internal components of the housing

In the next example, we will build a housing for an internet of things application. The device will contain a thermometer/barometer/hygrometer connected to a microcontroller. The microcontroller can record the environmental conditions and report logged data over a wireless link.

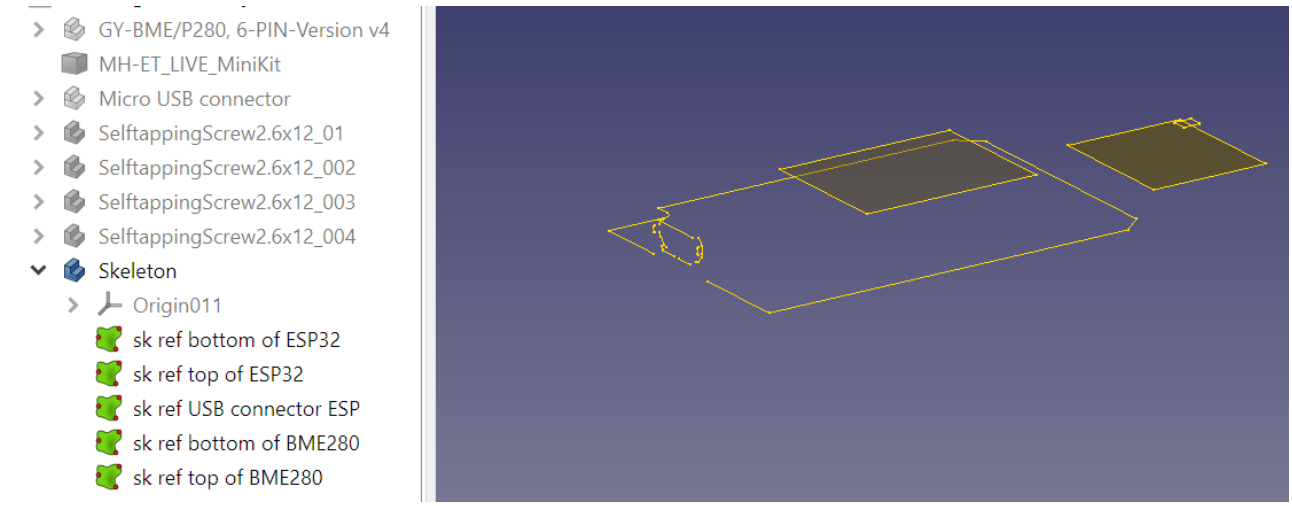
For projects like this, it is important to obtain accurate 3D models. Usually they are available as STEP file or in another format which can be imported in FreeCAD.

Import the electronic components in the FreeCAD file and orient them well. In this project there is a risk that the heat of the wifi module of the microcontroller affects the temperature measurement of the sensor, so it is important to minimize thermal crosstalk when designing the housing.

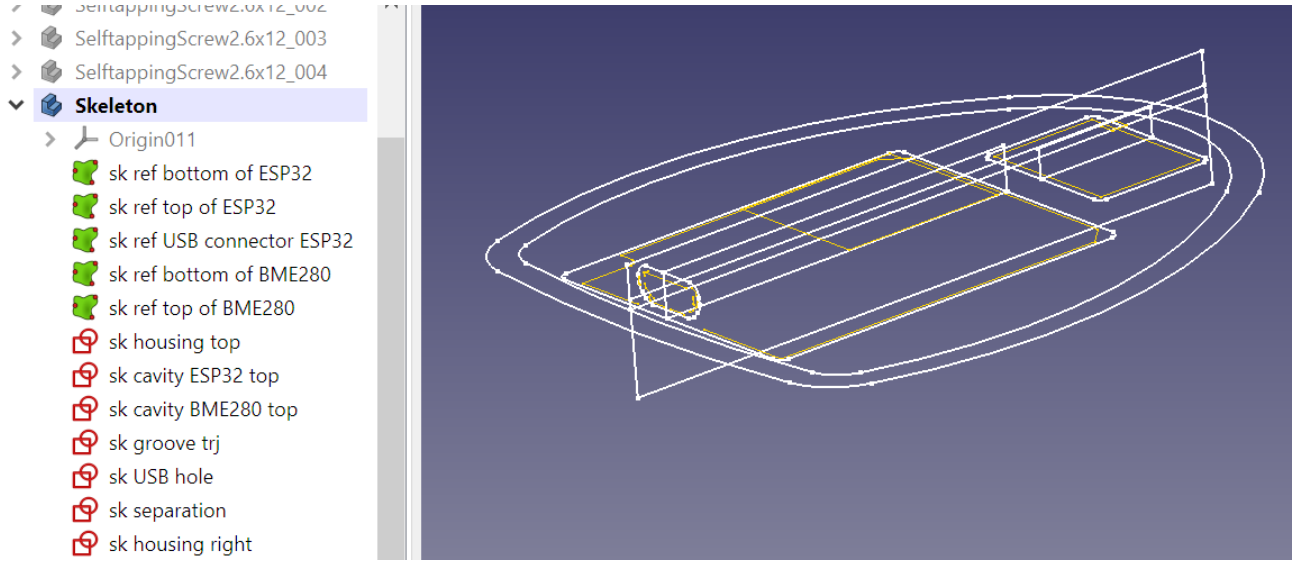




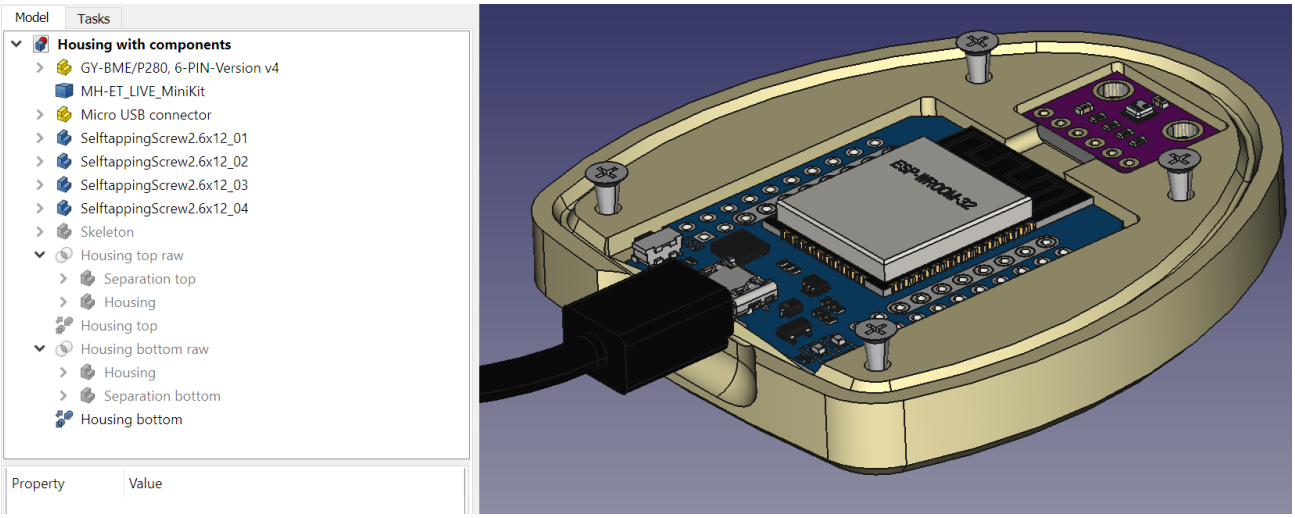
Create the **Skeleton** body. In the skeleton, import important geometry of the components using shape binders. This way, the sketches in the skeleton will dynamically follow the components when the components are moved.

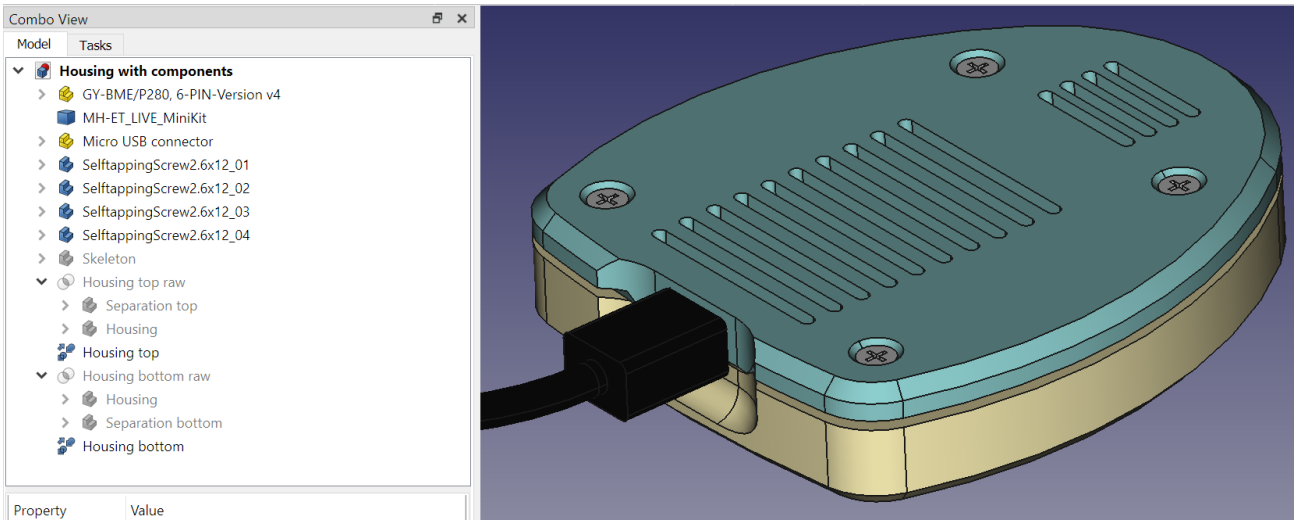


Create the sketches in the **Skeleton** body.



Create the **Housing**, **Separation top**, **Separation bottom**, **Housing top** and **Housing bottom** like in the previous example.



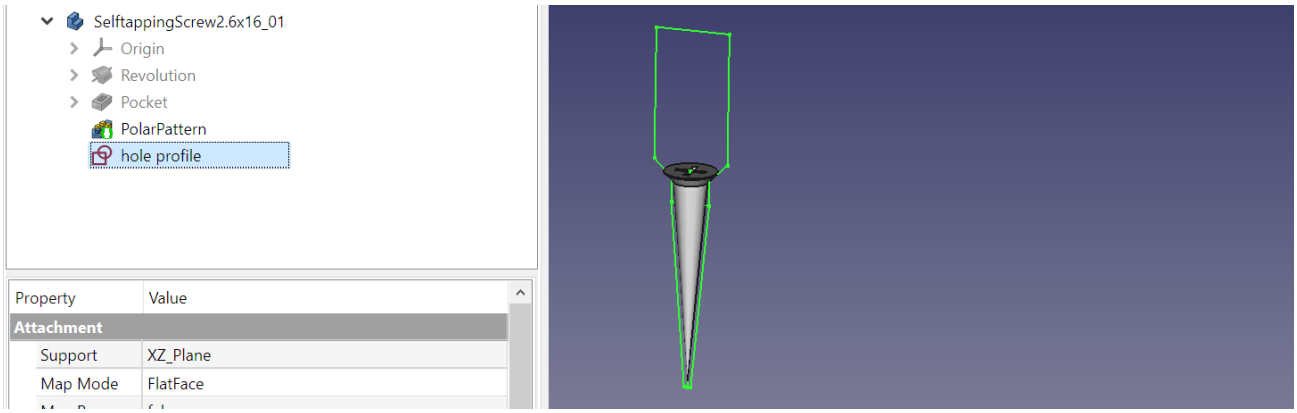


We can now move the boards around, and (within certain limits), the cavities in the housing will follow the components.

Using self tapping screws to close the housing

I often use self tapping screws for such housings. With the right tolerances, these screws work really well and require no post processing (tapping, inserts) in the parts, which makes it quite fast. These screws are available from many different suppliers at AliExpress.

In order to make the screw holes parametric, I created a model of the screw which contains an additional sketch representing the hole in the housing.



⚠ In reality, these screws are not conical. Some day I will make a model that more closely resembles the shape of these screws. Nonetheless, this shape works in my printed models.

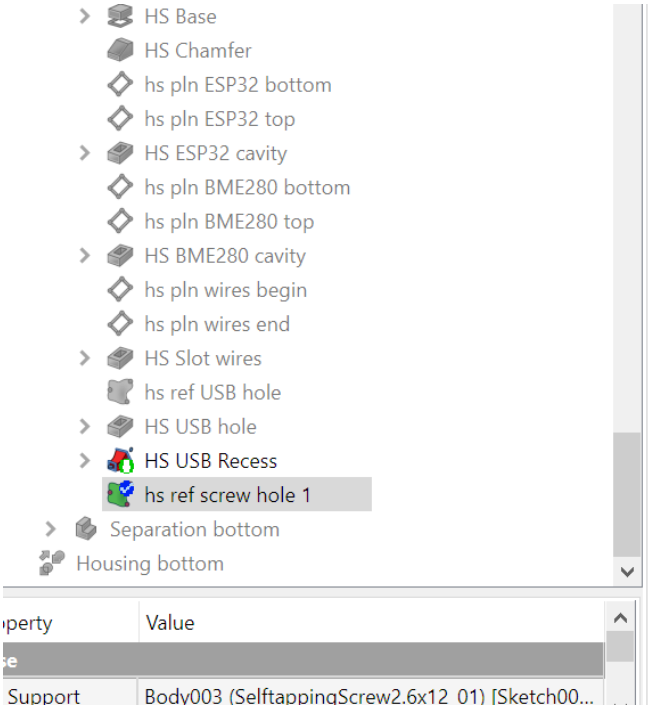
⚠ By tweaking the shape of the hole, I achieved a good fit for these screws for PET printed on my Prusa Mk3s. With a different printer or material the tweaking may have a different outcome.

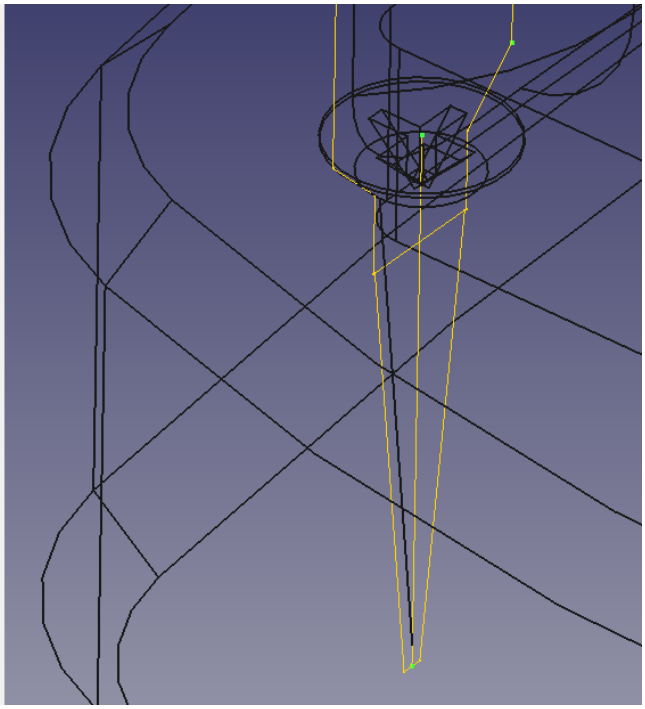
Creating a screw hole

This is how it works:

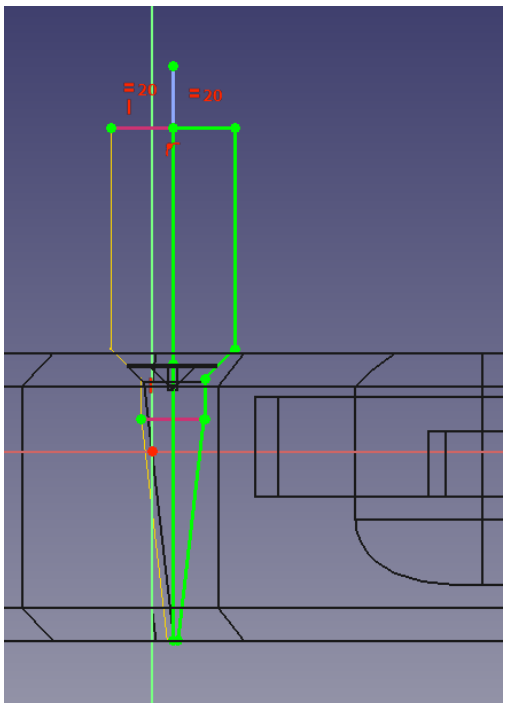
Insert screw in the model using File > Merge project

1. Create a shape binder **hs ref screw hole 1** in the **Housing** body, referencing the screw hole sketch from the model of the screw (no need to make an intermediate reference in the **Skeleton** body)
2. Make the model of the original screw invisible (so we can only select elements fro the shape binder)
3. Select three points on the shape binder of the hole, and create a datum plane **hs pln screw hole 1** through these points

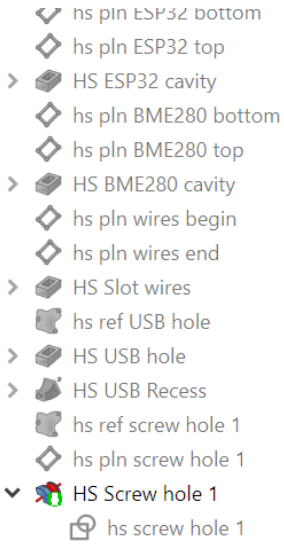


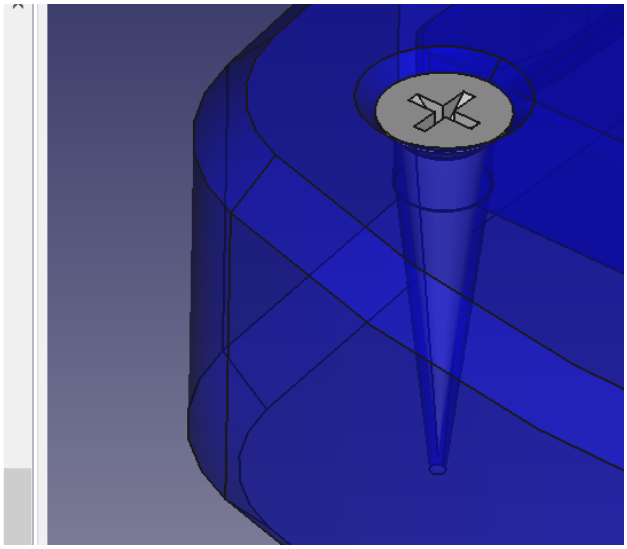


4. Create a sketch **hs screw hole 1** on this datum plane, tracing one half of the screw hole
5. Add a construction geometry line to this sketch, representing the centerline of the screw. I usually make the length equal to an arbitrary other line of the sketch to make the sketch fully defined.



6. Create a Groove **HS Screw hole 1** based on this sketch, choosing the construction line as a centerline.





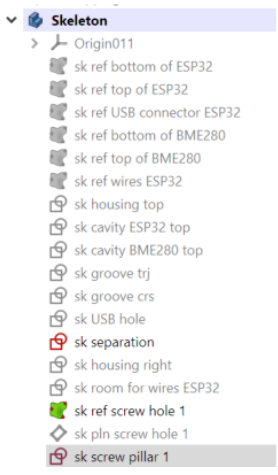
Now if we move the screw to another location in x, y or z-direction, the shape binder **hs ref screw hole 1** and the datum plane **hs pln screw hole 1** will move with it, and thus the hole in the part will be fully parametric.

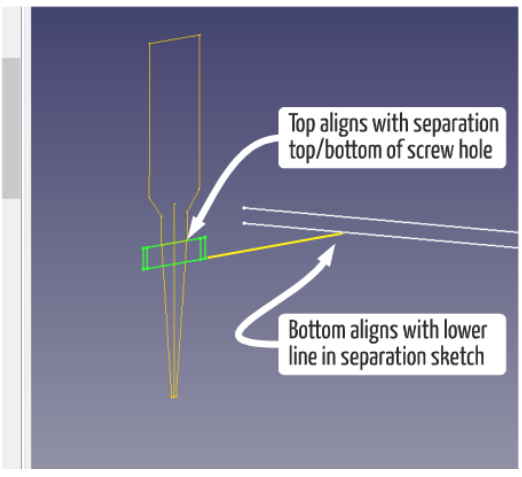
Creating a pillar for the screw

Sometimes the seperation of the housing does not line up with the separation in the screw hole model. For instance, In the housing model I lined up the separation in the middle of the USB port, so the housing can be closed easily.

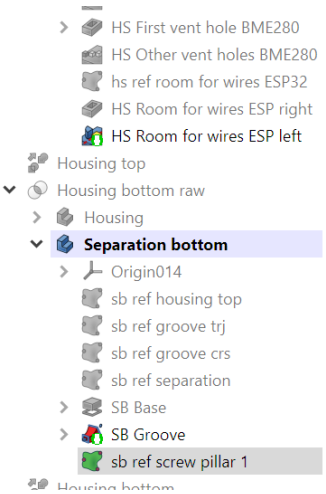
To solve this, we can make a local pillar in the bottom housing and a hole in the top housing.

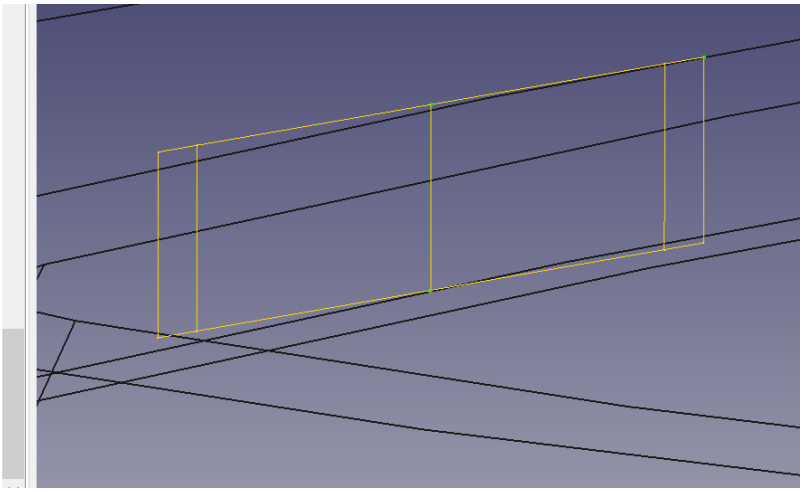
1. Create a shape binder **sk ref screw pillar 1** in the **Skeleton** body.
2. Make the model of the original screw invisible
3. Create a datum plane **sk pln screw pillar 1** like we did in the housing
4. Create a sketch **sk screw pillar 1** that represents both the pillar in the **Bottom housing** and the hole in the **Top housing**. The top of the pillar must align with the separation plane in the screw hole, the bottom of the pillar is aligned with the lower line of the separation. Ensure the centerline of the pillar/hole is also a geometry line. We want to refer to it lateron.



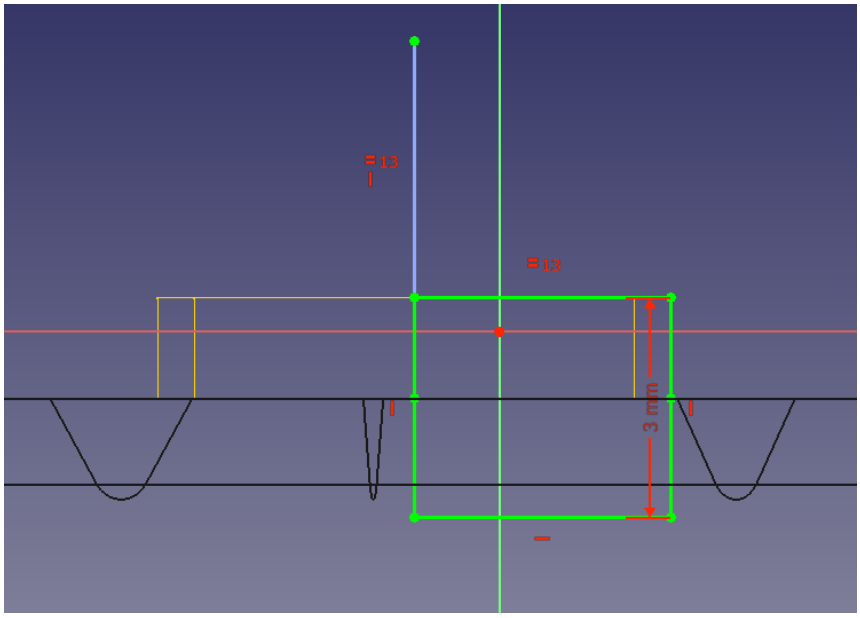


5. Make **Separation bottom** the active body
6. Create a shape binder of **sk screw pillar 1** from the **Skeleton** body and rename it **sb ref screw pillar 1**
7. Create a datum plane **sb pln screw pillar 1** like we did in the housing





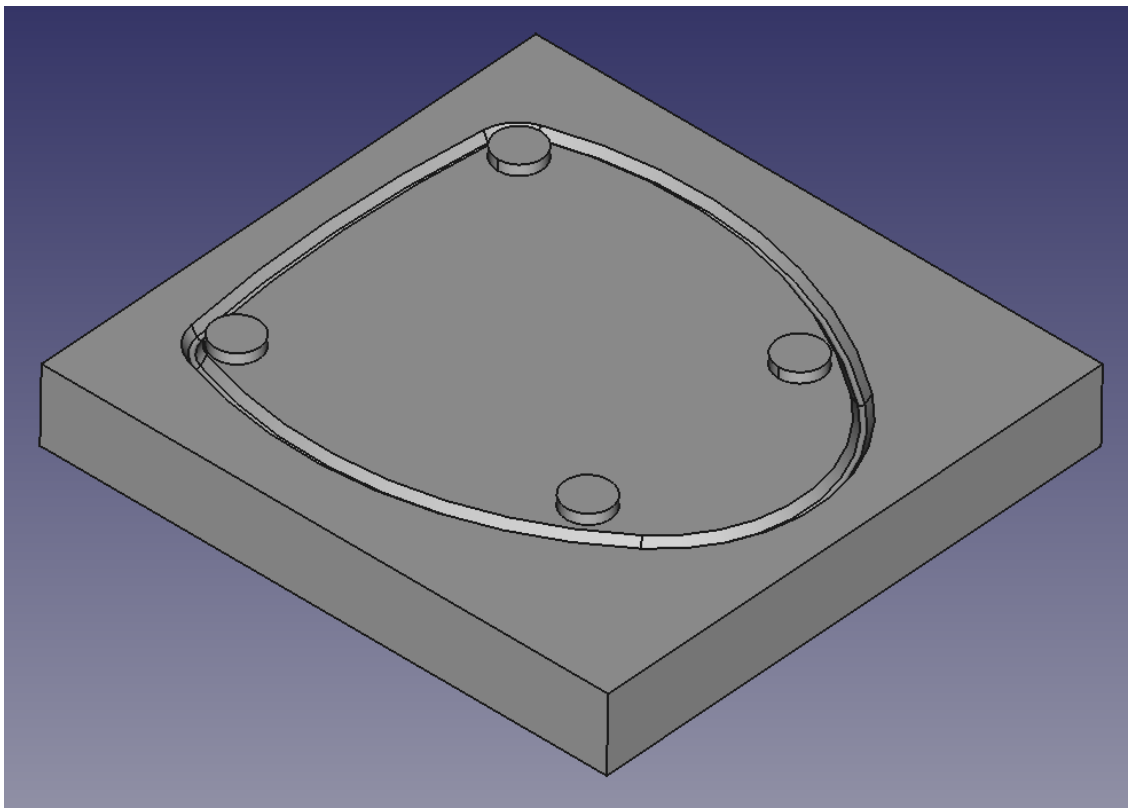
8. Create a sketch **sb screw pillar 1** to create the pillar, and add a geometry line that will be the center line of the pillar



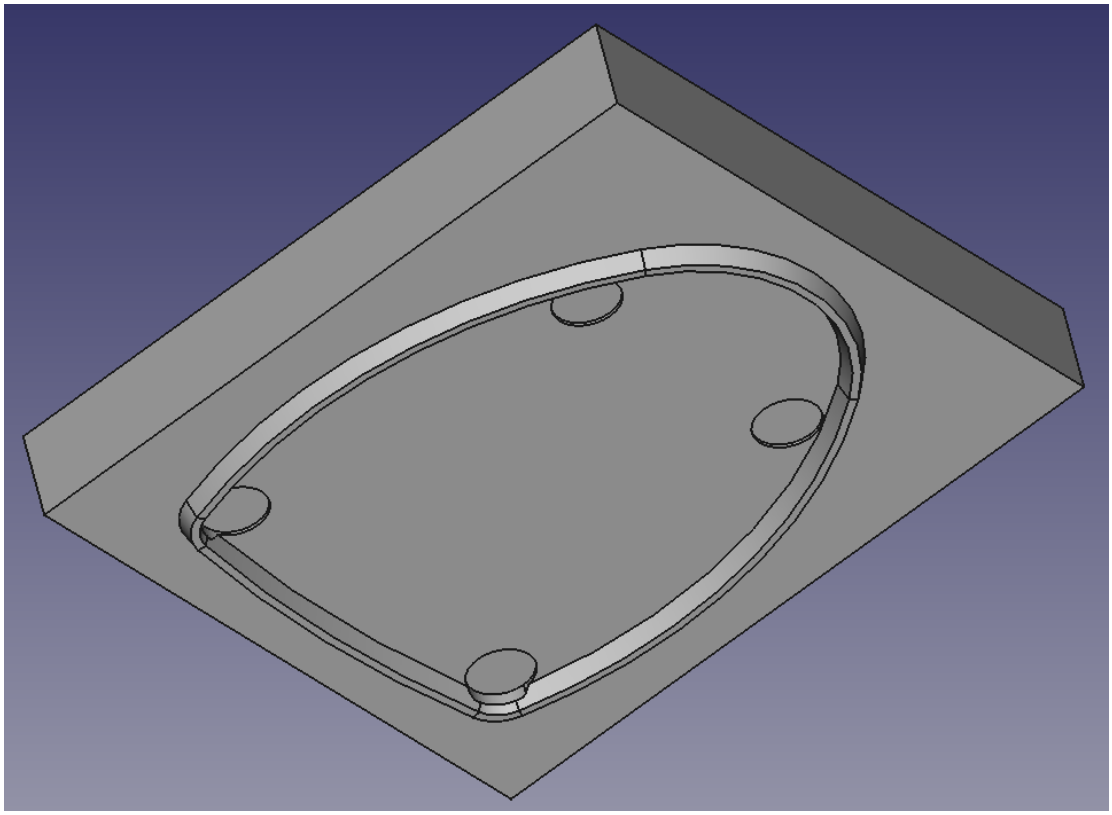
In this case, I extended the bottom of the pillar so it would also fit the V-groove. Also, do not forget the centerline.

9. Create a revolution and name it **SB Screw pillar 1**
10. Repeat the same procedure for the other pillars. In this case, I did pillar 3 in the same way and created pillars 2 and 4 by mirroring.

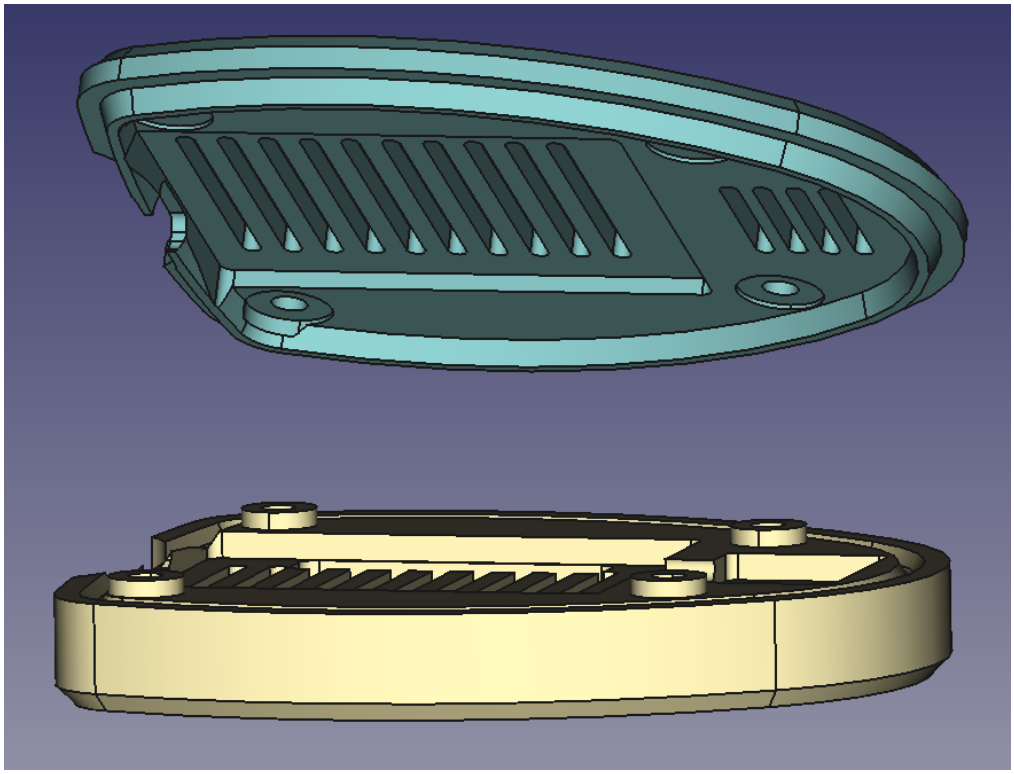
README.md



Also repeat the procedure to create the holes in **Separation top**.



The changes will now automatically come through in both housing parts:

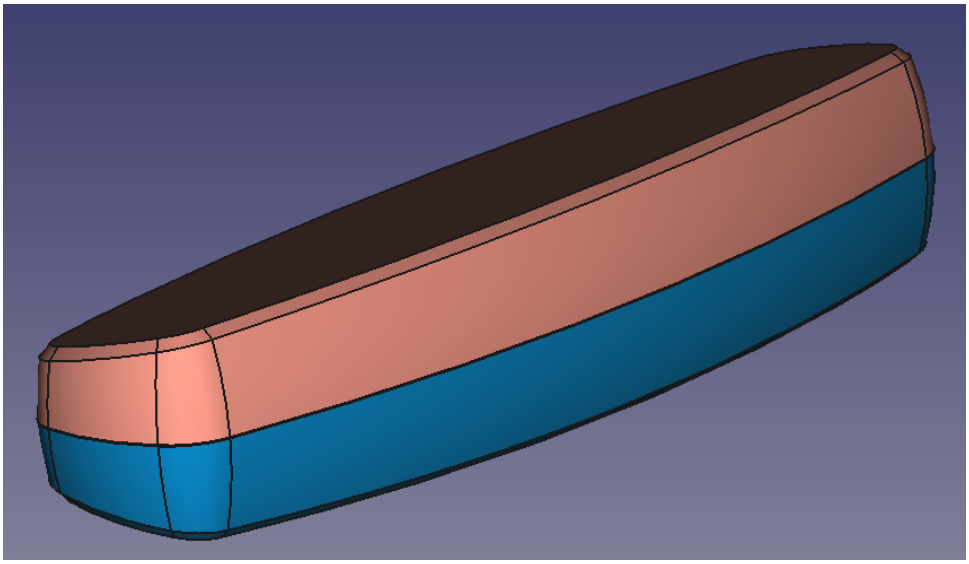


This is a good example to demonstrate why some changes need modifications in the housing part, while others require changes in the separation parts.

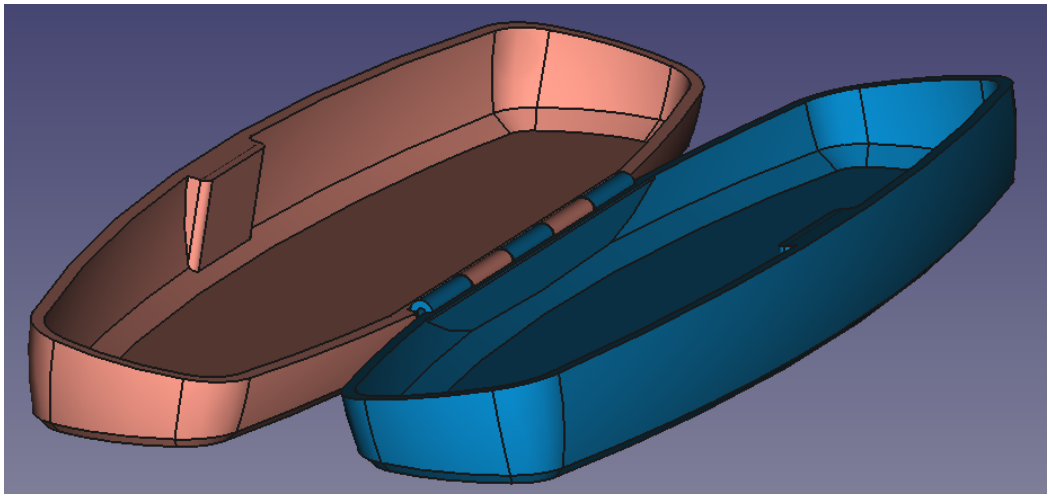
Creating a complex hinge

The next project is a housing with a hinge. It can for instance be used for pencils or glasses. There is a magnet in each shell to keep the housing closed. An advantage of 3D printing is that we can pause printing at a designated layer to insert the magnets manually. When completed, the magnets are fully enveloped by the printed part.

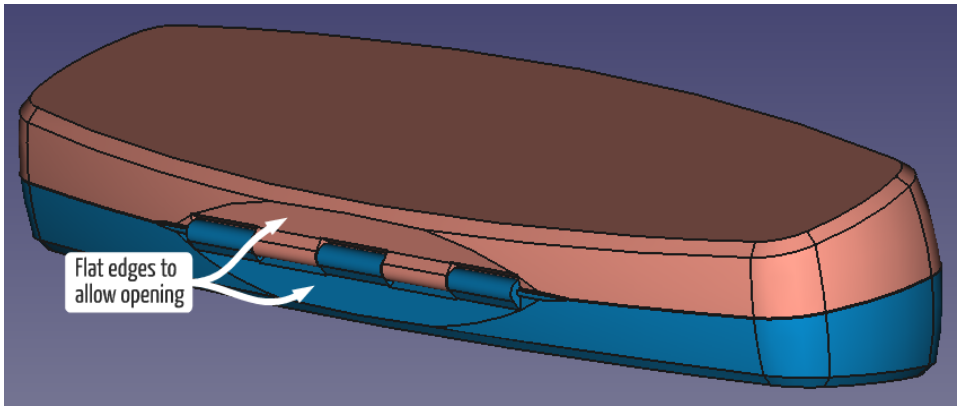
This is the front view of the case when it is closed:



This is the front view of the case when it is open:

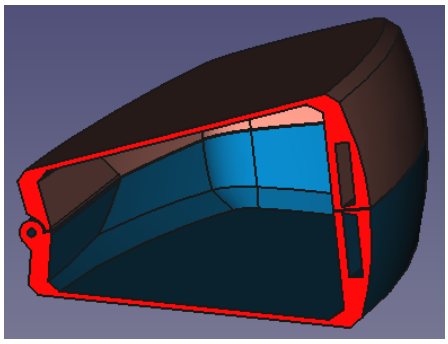


The details of the hinge are quite complex:

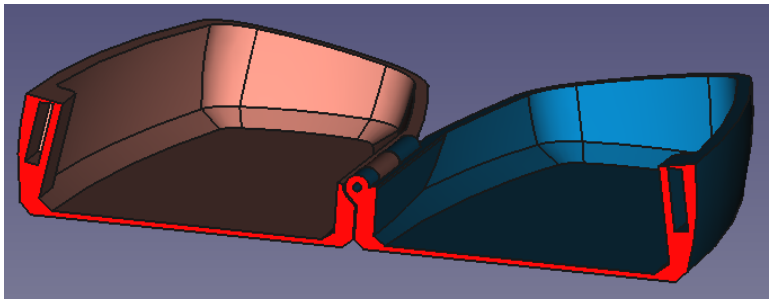


The flat edges in the rear view are needed to avoid mechanical interference when the case is fully open, and they act as an end stop.

This is a cross section through the middle of the casing when it is closed:

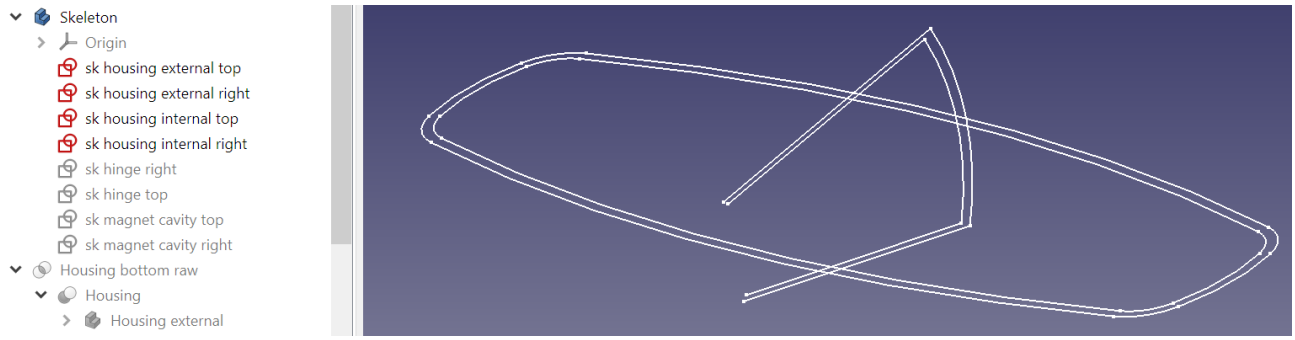


This is a cross section through the middle of the casing when it is open:

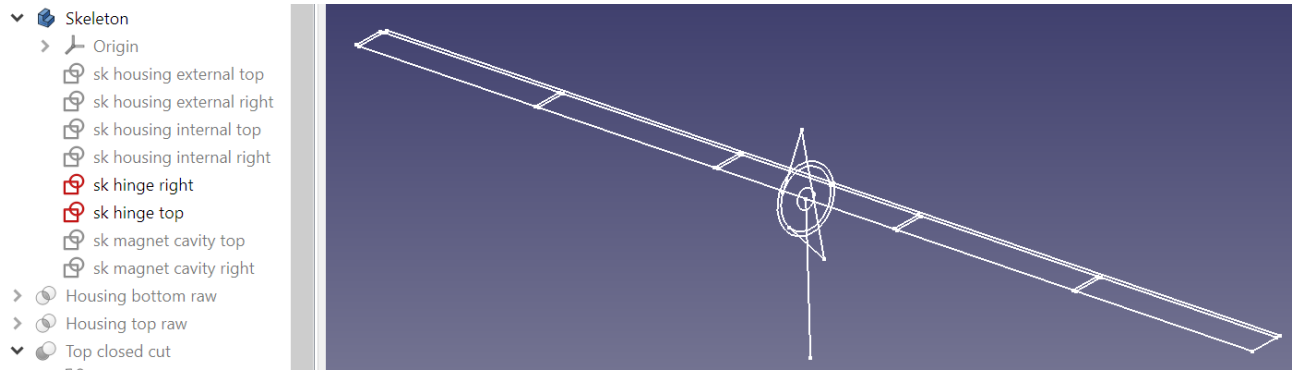


The orientation of the parts during printing is the same as when the case is open. To bridge the openings of the magnets well, the top of the magnet opening needs to be horizontal during printing. This is why the magnet opening is not rectangular.

Four sketches define the general shape of the housing, for the top view and the right view, and for the internal and the external shape:

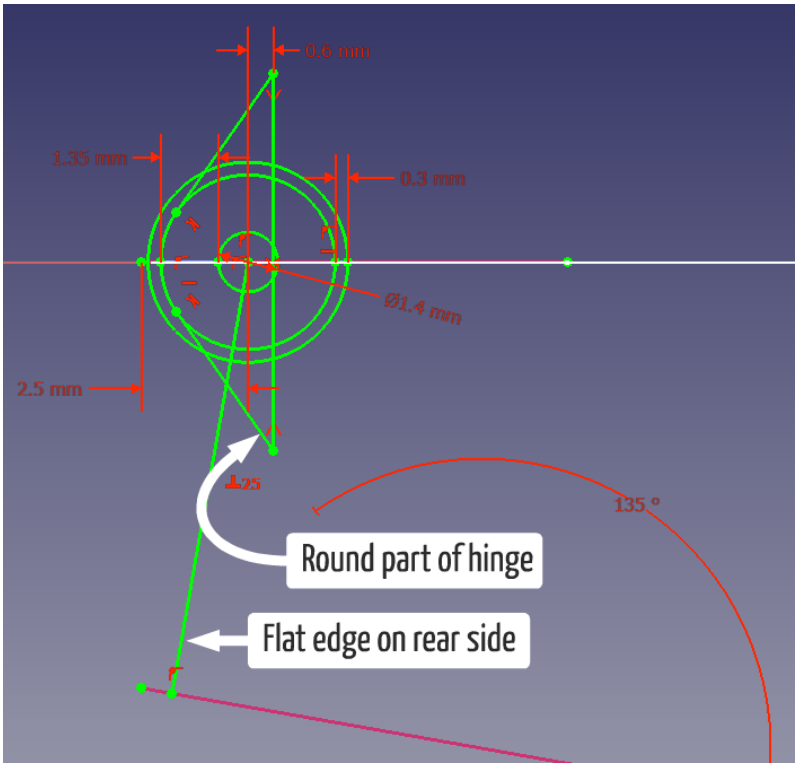


Two sketches define the hinge:



For the design of a 3D printed hinge it is important to take into account the accuracy of printing. There needs to be a slit of about 0.3 mm between the parts in all directions.

sk hinge right defines the right view of the hinge. The smallest circle represents the hole for the hinge pin. The circle around that represents the cilindrical shape of the hinge. The outermost circle is a reference for the play between both parts of the housing.



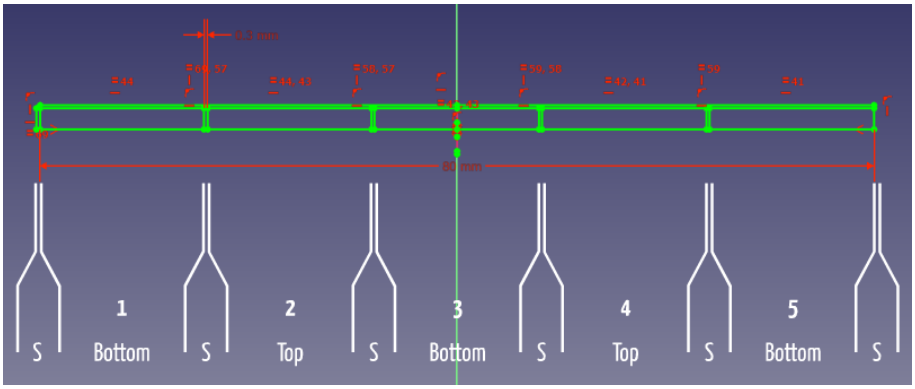
The line going down is perpendicular to the bottom flat side of the housing. This line is a reference for the flat face mentioned above.

There is also geometry representing the round parts of the hinge.

sk hinge top basically divides the length of the hinge in three parts:

- elements that are connected to the bottom part of the housing
- elements that are connected to the top part of the housing
- space between the parts (S)

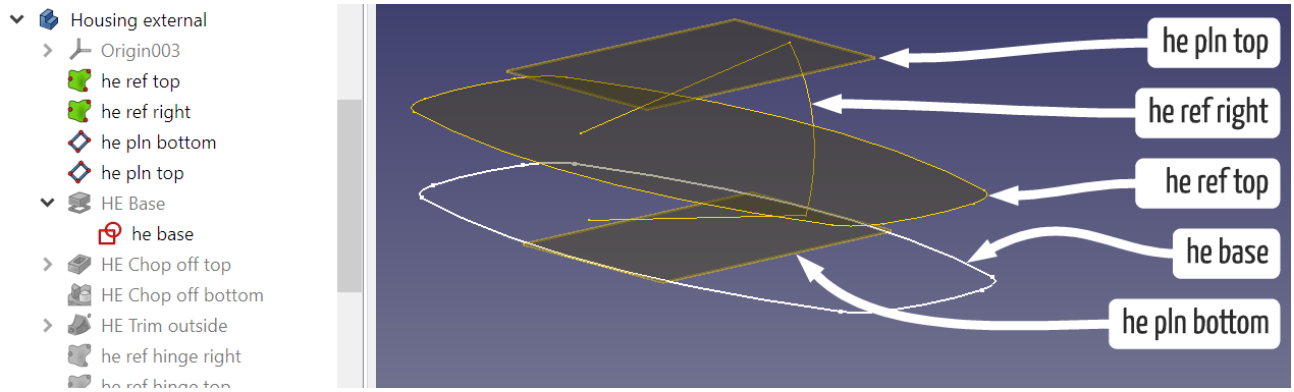
The sketch contains only two dimensions: the total length of the hinge and the space between the parts. The radius of the cilinders is also modelled, but this has been derived from **sk hinge right**



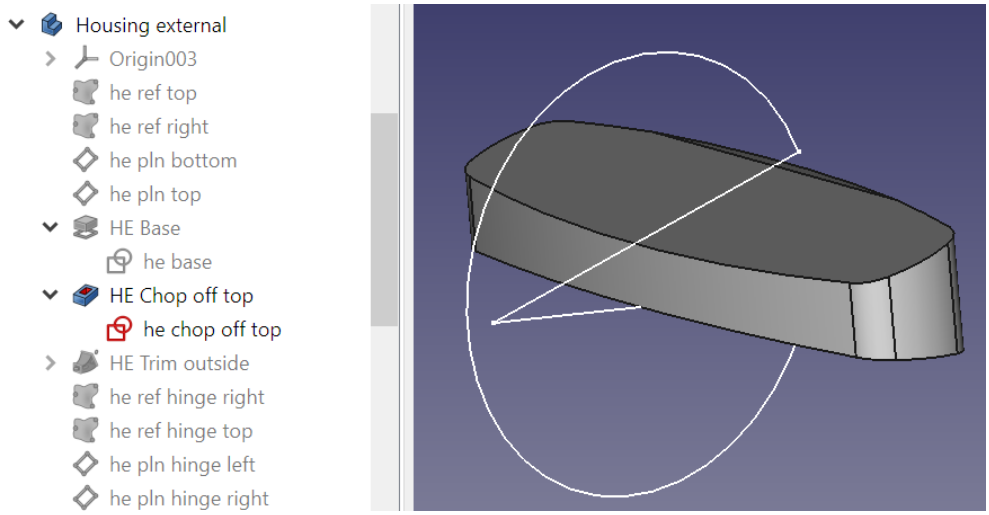
The housing is modelled in two different bodies: **Housing external** represents the outside of the housing, **Housing internal** represents the cavity inside. The final housing is obtained by boolean subtraction in the part workbench.

Housing external

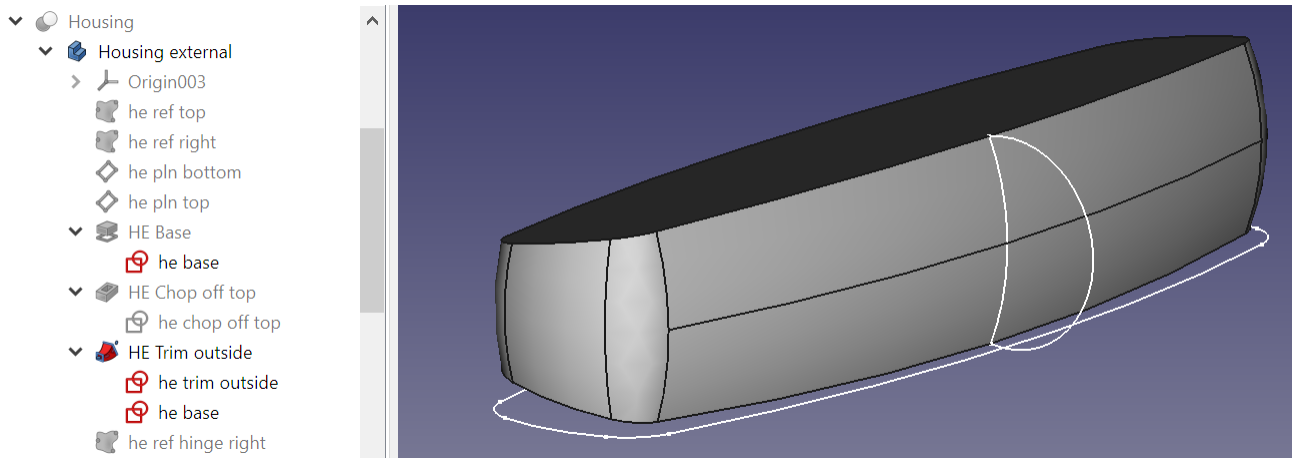
The relevant sketches from the skeleton are imported as shape binders. The bottom and top datum plane are defined as 'normal to edge', referencing the Z-axis and the bottom and top most points. The contour **he base** is modelled on **he pln bottom** and extruded until **he pln top**.



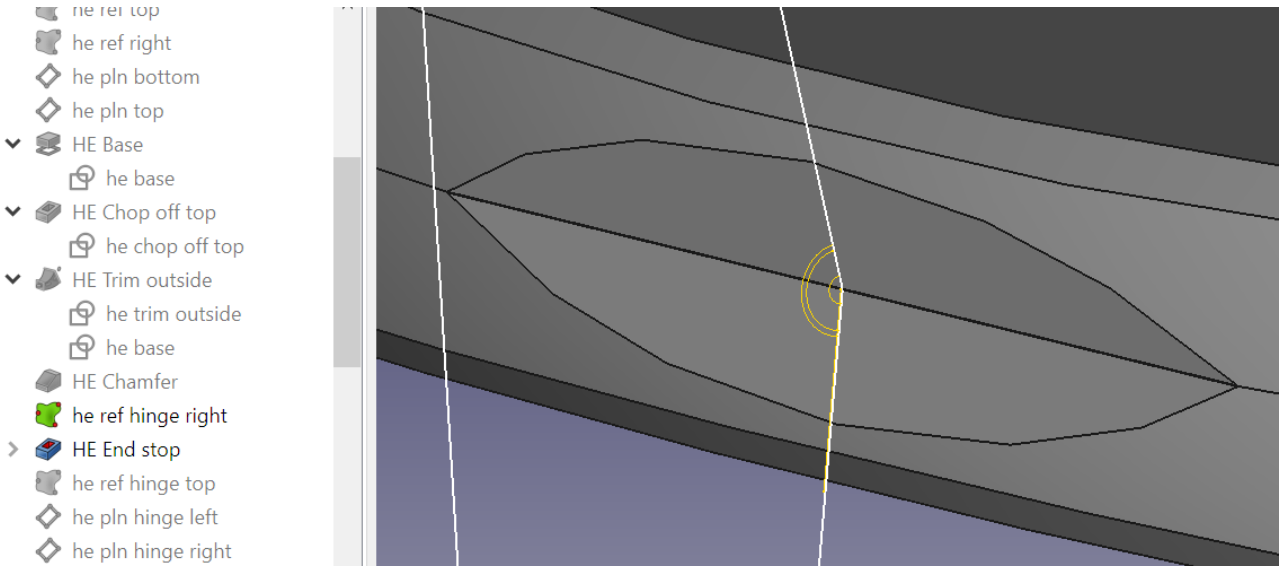
he chop off top chops off the oblique surfaces of **HE Base**.



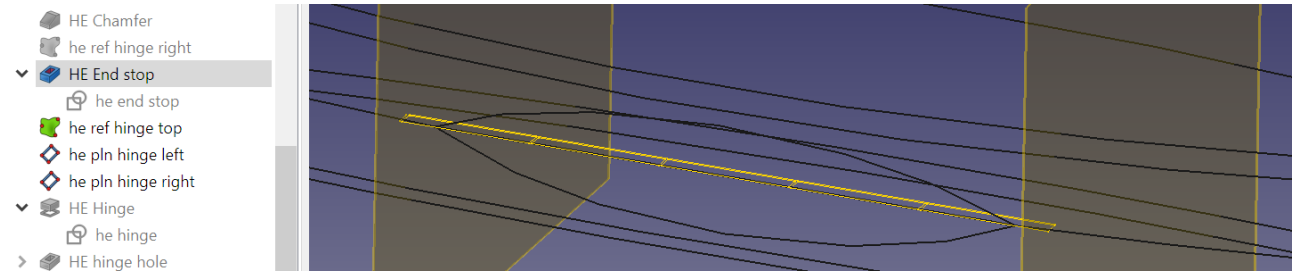
A curve along the outside is made with a subtractive pipe using **he trim outside** along **he base**:



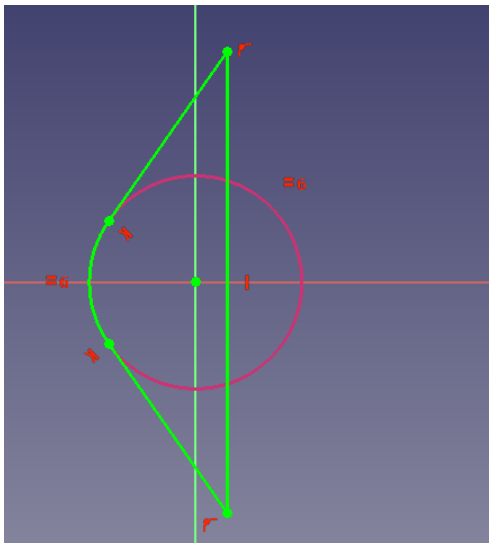
Chamfers are added, **he ref hinge right** is imported and the flat edges for the end stop when opening the case are created:



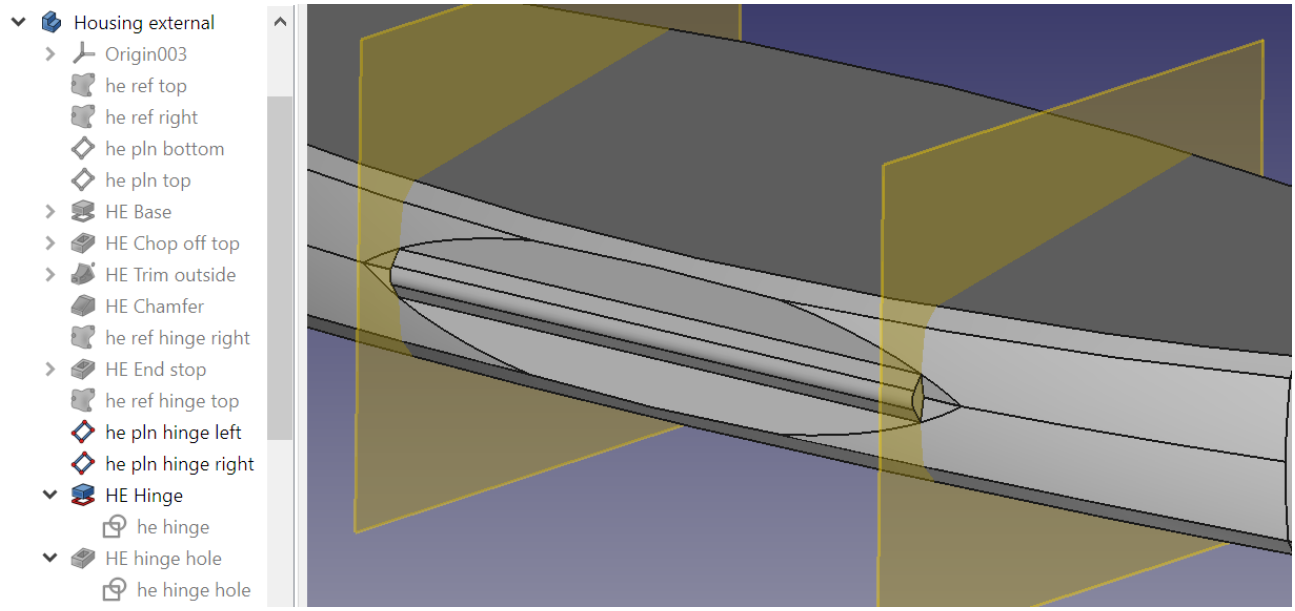
he ref hinge top is imported as shape binder, and the beginning- and end datum planes for the hinge are created. They exclude the space next to the hinge:



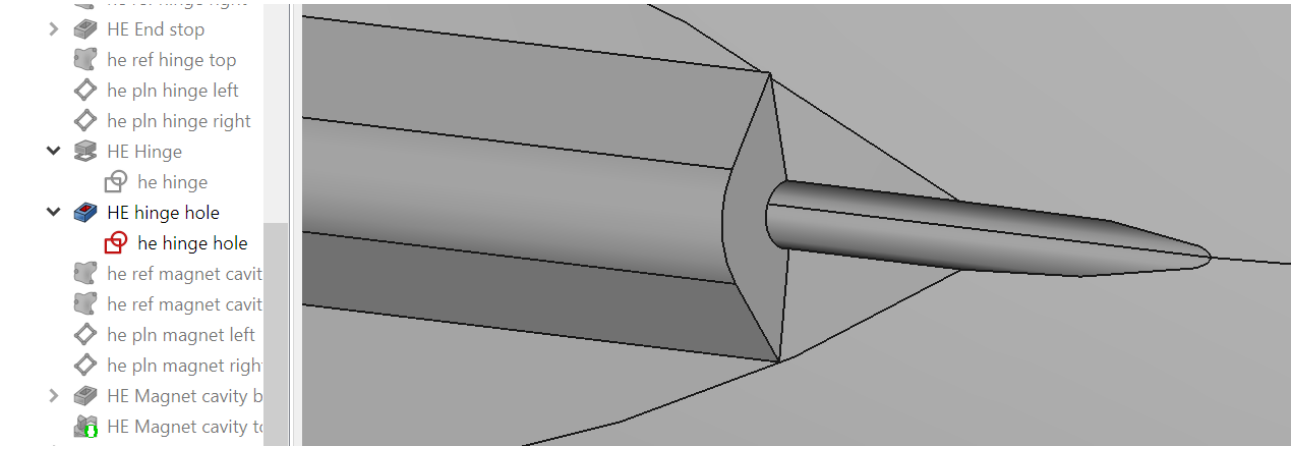
The cross section of the hinge **he hinge** is created on **he pln hinge left**, referring to **he ref hinge right** for the shape:



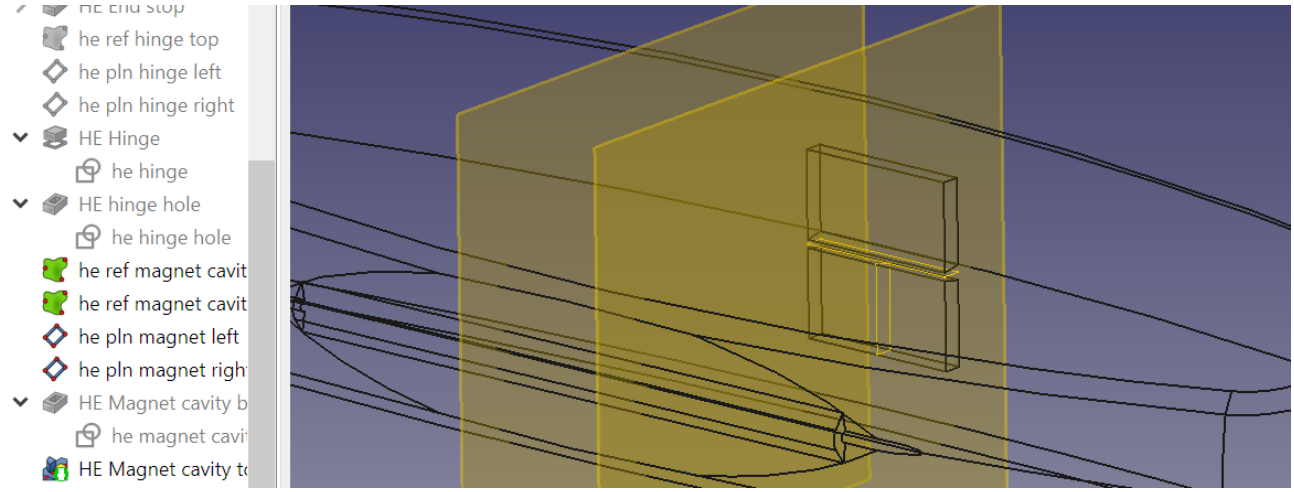
he hinge is extruded from **he pln hinge left** to **he pln hinge right**, forming HE Hinge:



As a final step for the hinge, the hole through the hinge is created:

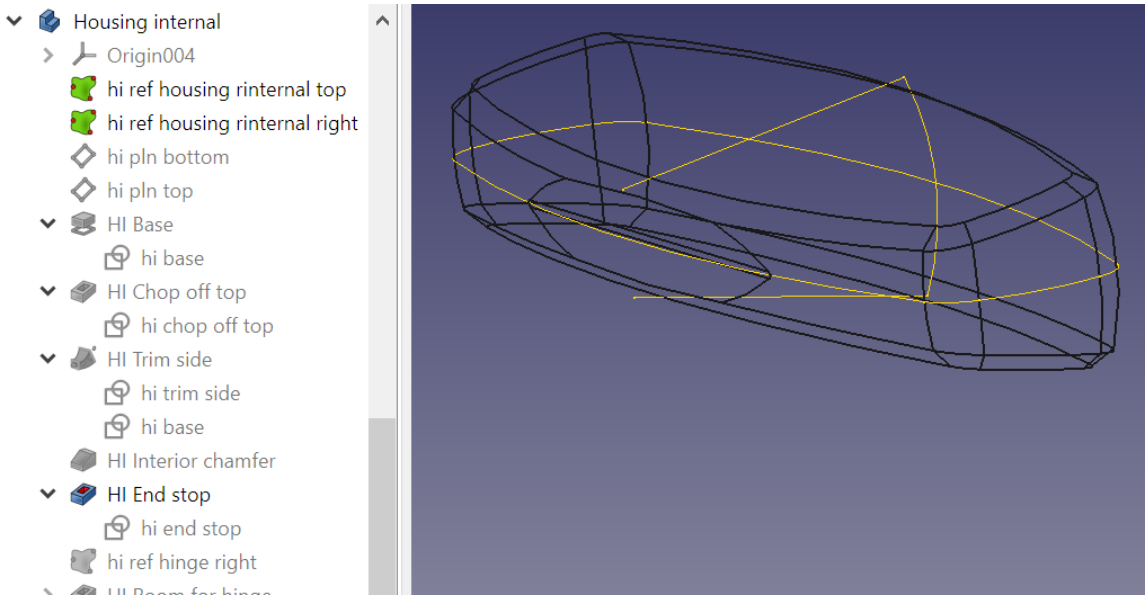


Both sketches determining the magnet pockets are imported, two planes defining the beginning and end of the magnet pockets are created, the lower magnet pocket is created, and the upper magnet pocket is mirrored from the bottom one:

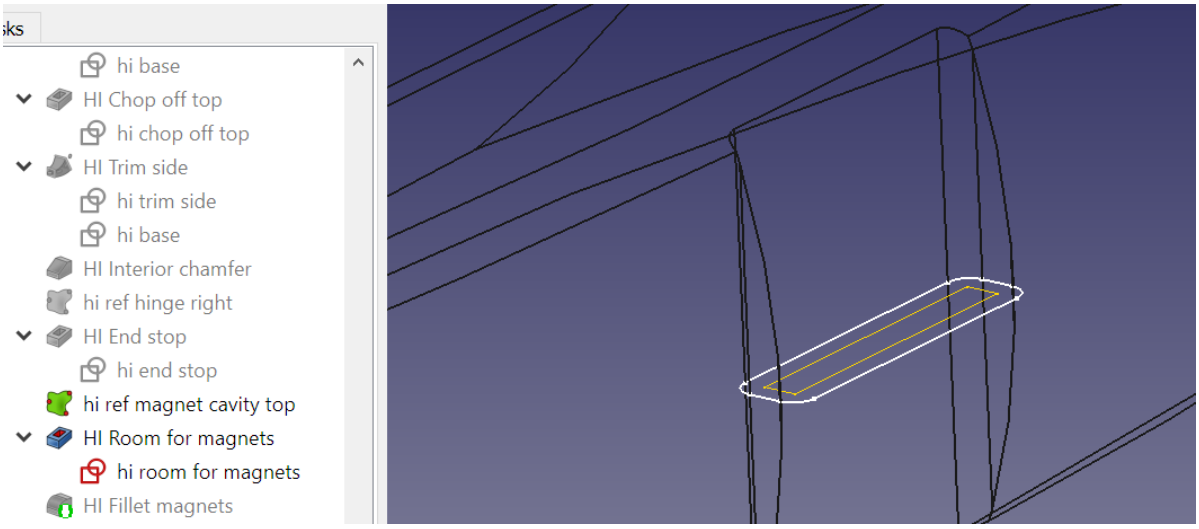


Housing internal

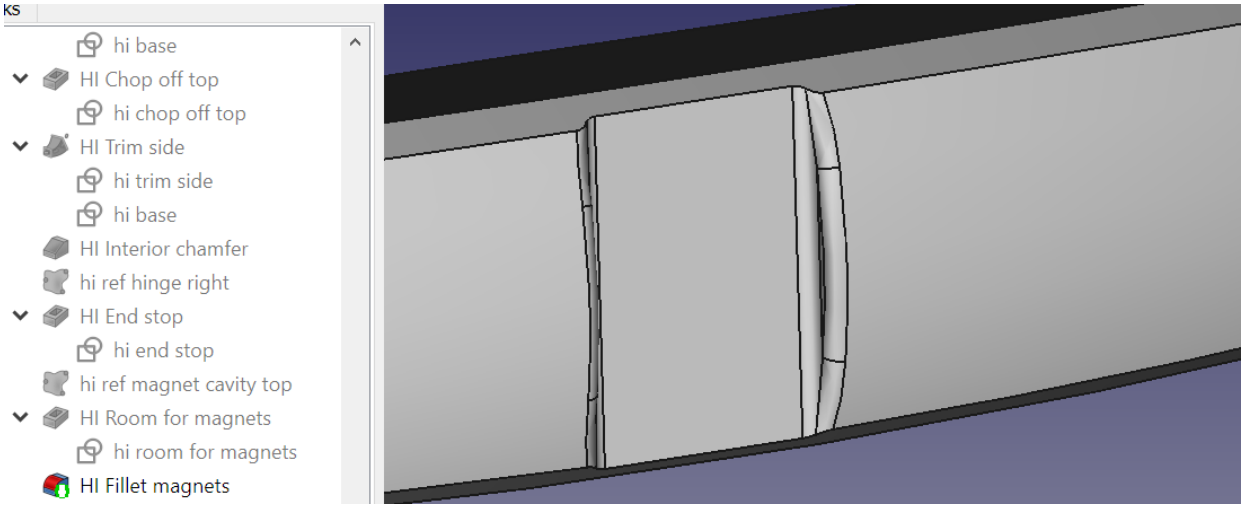
The first steps of the Housing internal body are basically the same, but now referring to **sk housing internal top** and **sk housing internal right** instead of **sk housing external top** and **sk housing external right**:



hi ref magnet cavity top is imported as shape binder. A rounded rectangle is sketched with a wall thickness around this shape in **hi room for magnets**. This is extruded from the shape in both directions as **HI Room for magnets**:

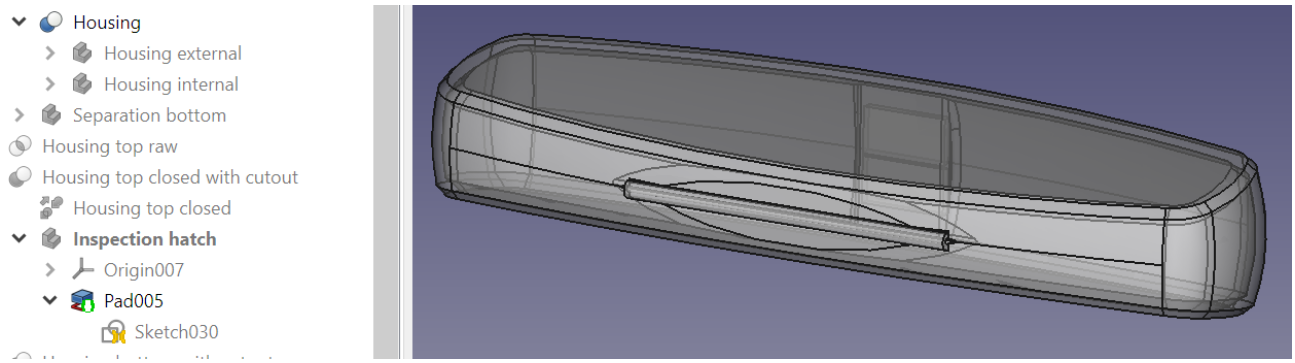


Finally, a fillet is added to the magnet bump:



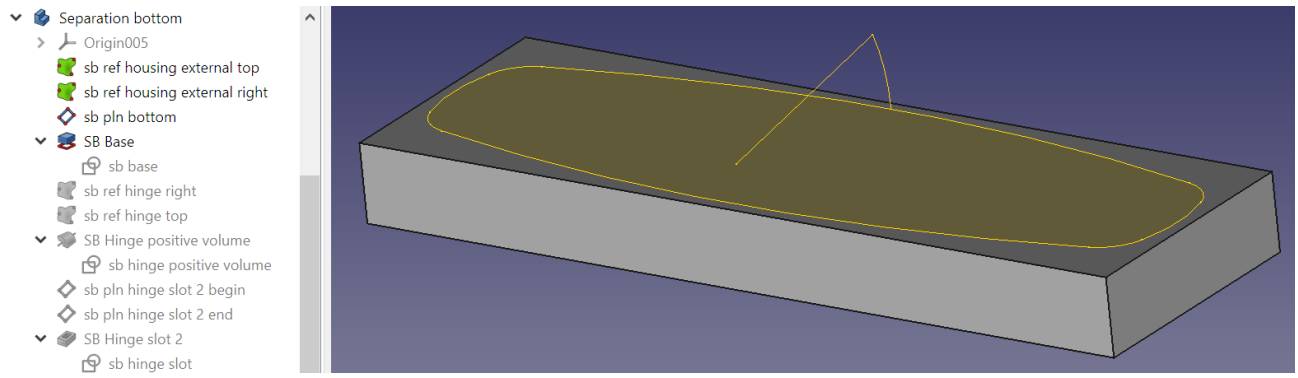
Housing

The housing is created by boolean subtraction of **Housing external** and **Housing internal** in the Part workbench:

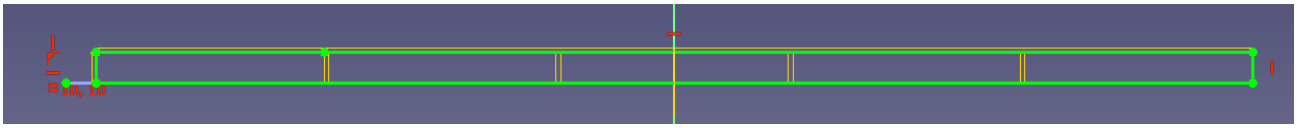


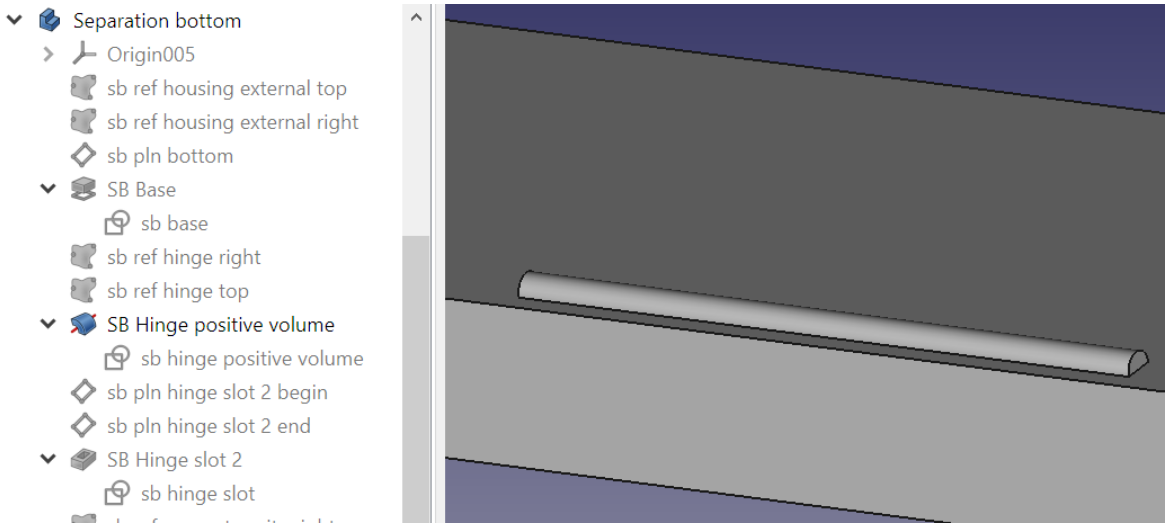
Separation bottom

The **Separation bottom** body starts with importing **sb ref housing external top** and **sb ref housing external right**, and then construction only **sb pln bottom**. A rectangle is drawn in a plane 0.1 mm below the XY plane, to ensure there is 0.2 mm space between both shells when the housing is closed. The rectangle is 3 mm larger than the outer shape.

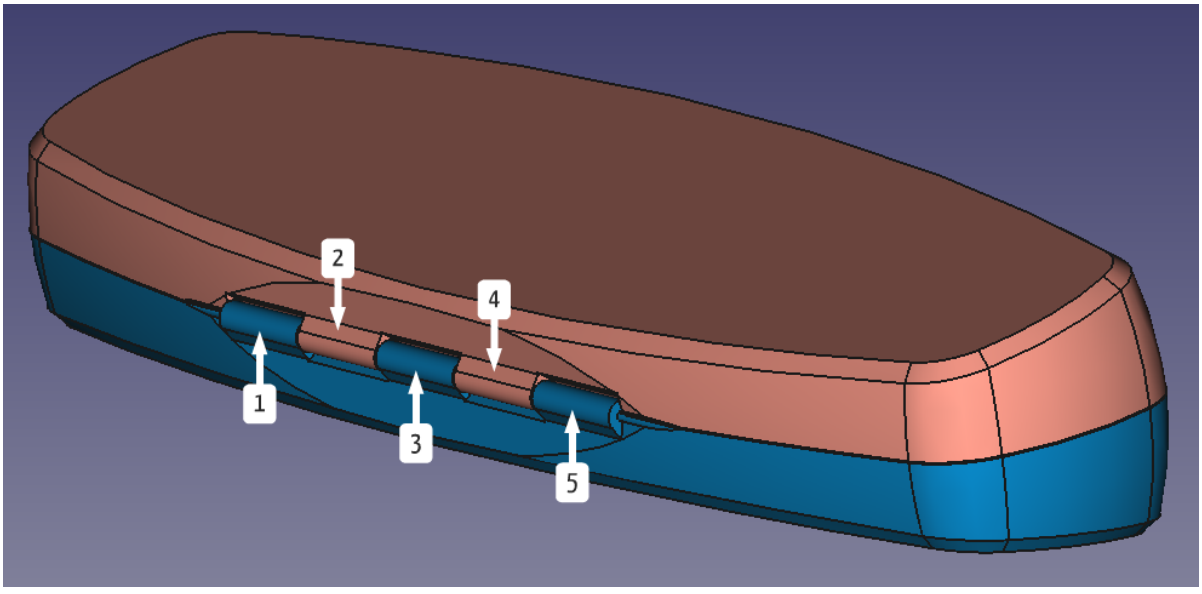


sb ref hinge right and **sb ref hinge top** are imported as shape binders. A positive revolve **SB Hinge positive volume** is added to the shape. The sketch **sb hinge positive volume** is a direct trace from **sb ref hinge top**.

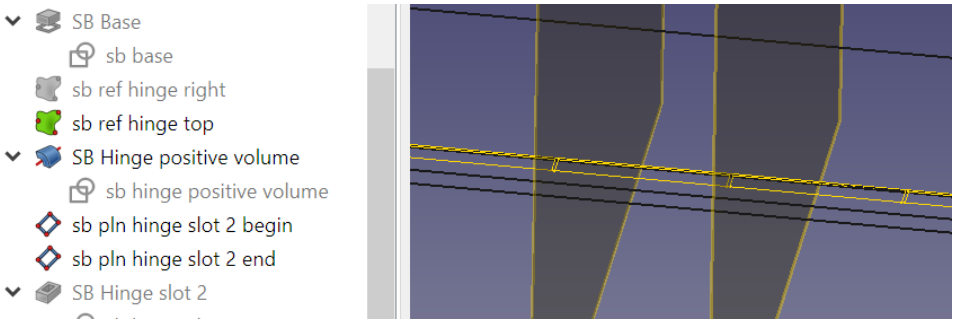




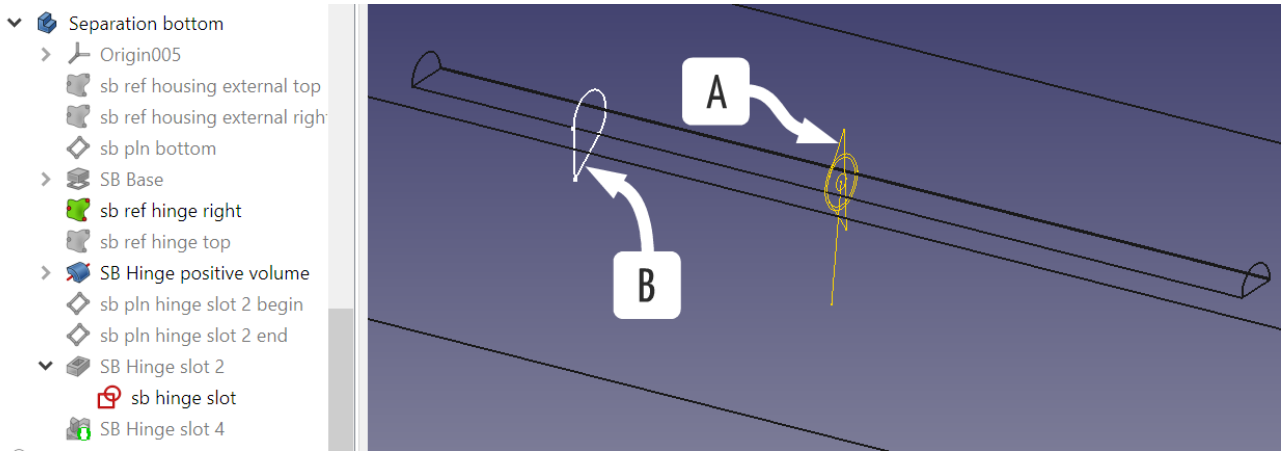
Next we will create the slot for protrustion 2 from the top housing.



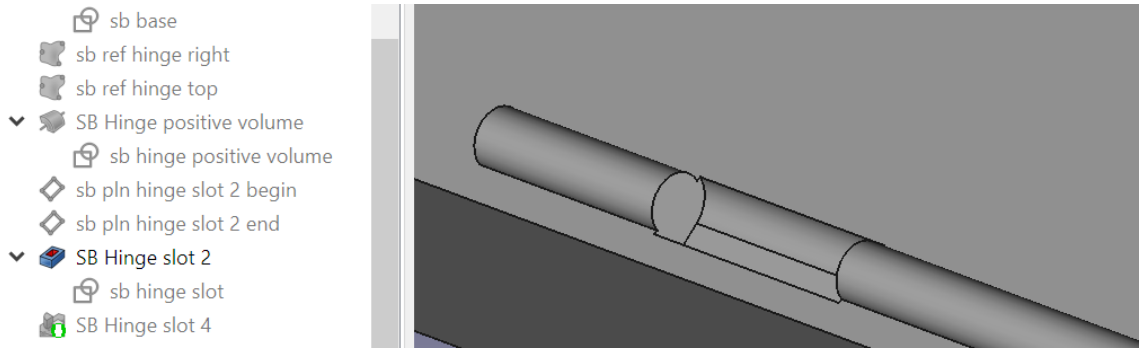
Two datum planes are created, **sb pln hinge slot 2 begin** and **sb pln hinge slot 2 end**, which will be used for the second slot. They will include the space next to the slot.



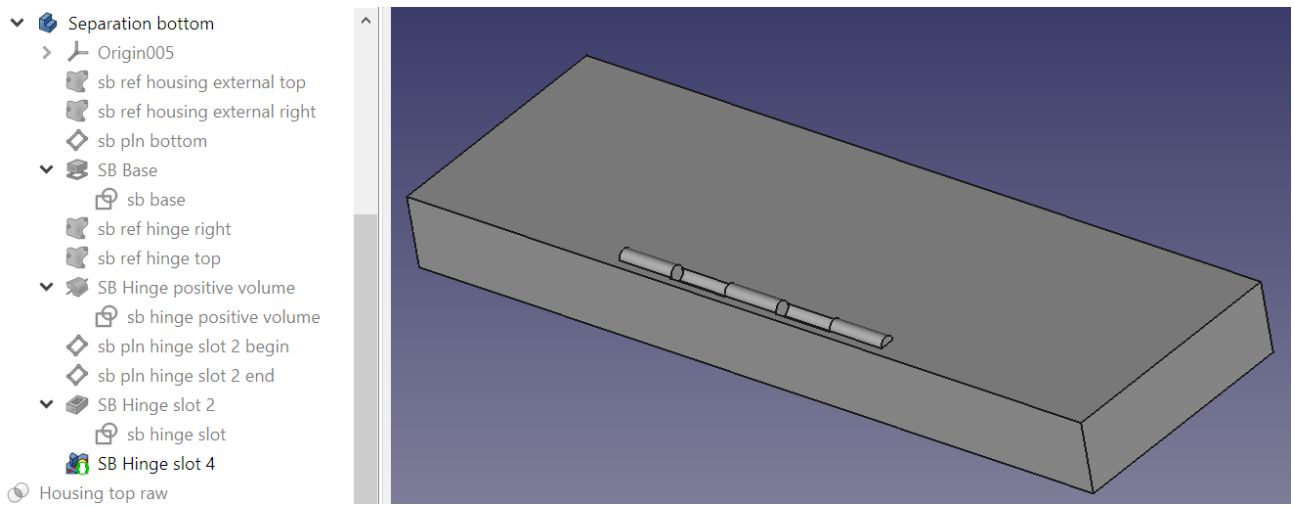
Line B of sketch **sb hinge slot** on datum plane **sb pln hinge slot 2 begin** is under an angle of 160° relative to line A of **sb ref hinge right**. This is because the top housing can be opened 160°, ensuring sufficient space between both parts. The larger circle in **sb ref hinge right** is used, also to create space in radial direction.



SB Hinge slot 2 is extruded until **sb pln hinge slot 2 end**:

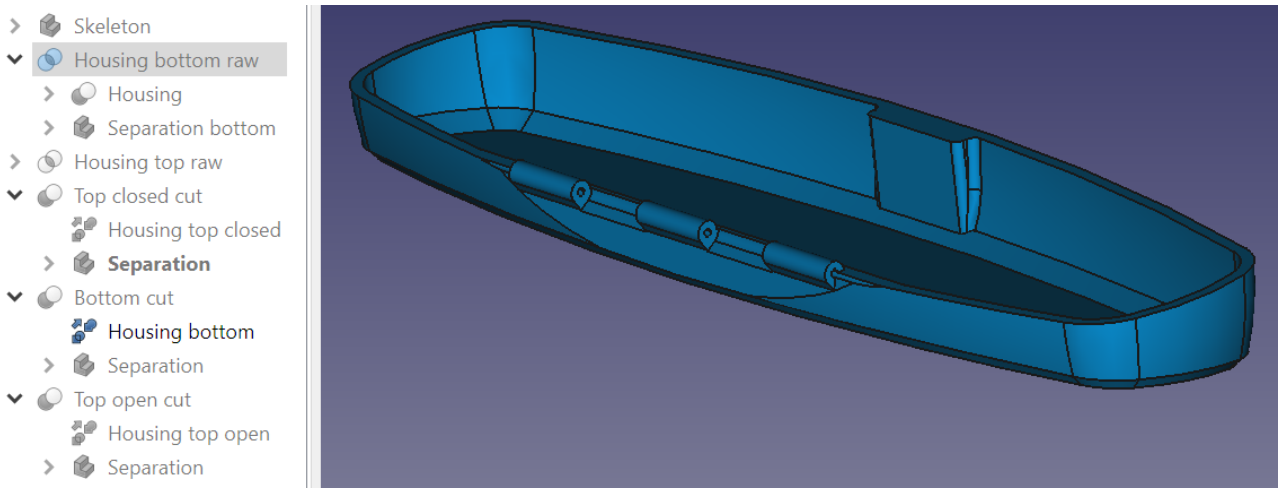


SB Hinge slot 4 is created by mirroring **SB Hinge slot 2** over the YZ plane.



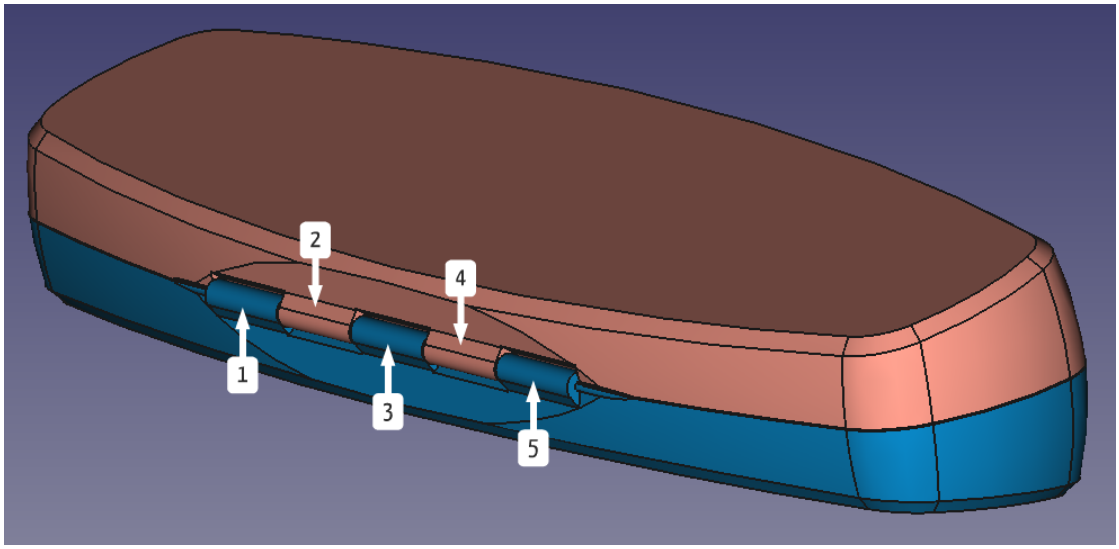
Housing bottom

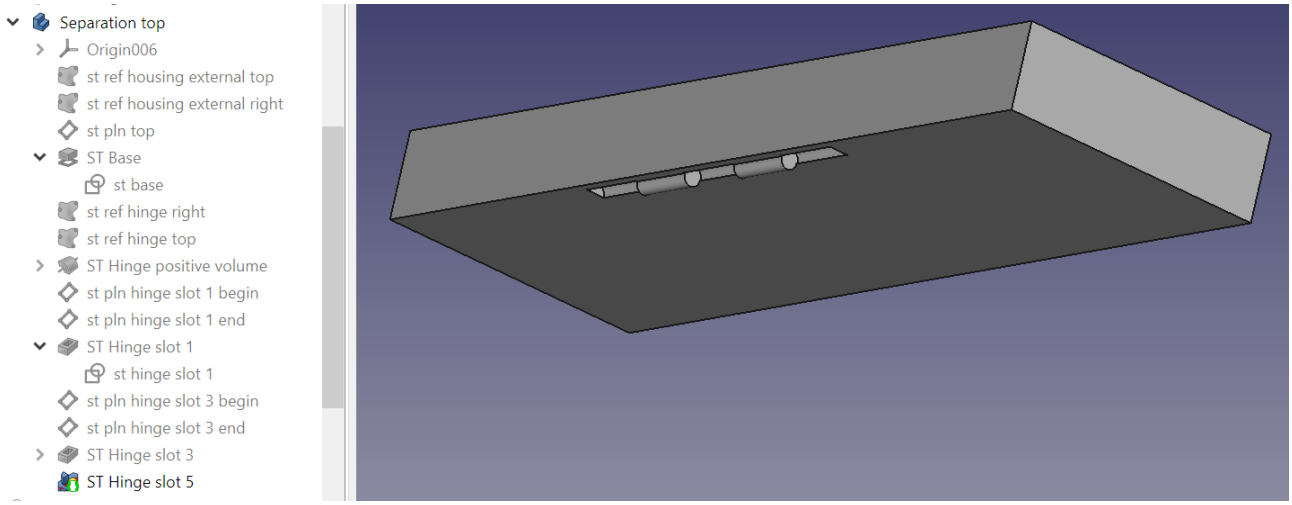
Housing bottom is a boolean intersection of **Housing** and **Separation bottom**



Separation top

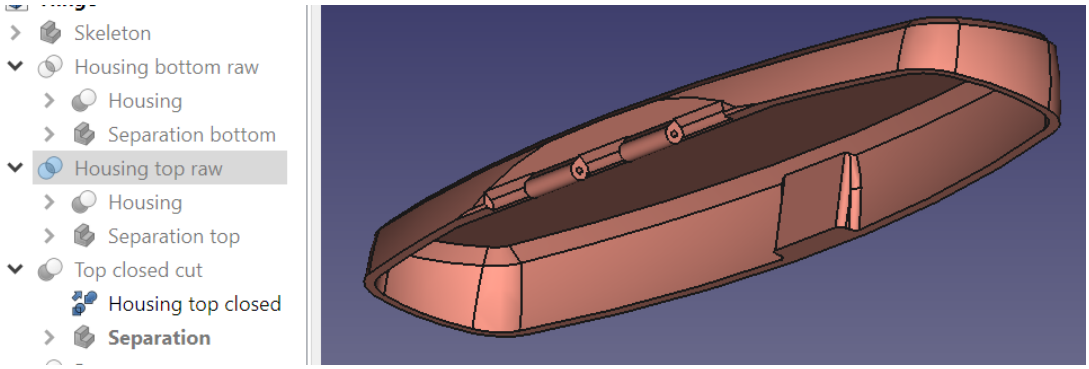
Separation top is very similar to **Separation bottom**, only now the slots are in locations 1, 3 and 5. Slots 1 and 3 were created individually, slot 5 is a mirror of slot 1.





Housing top

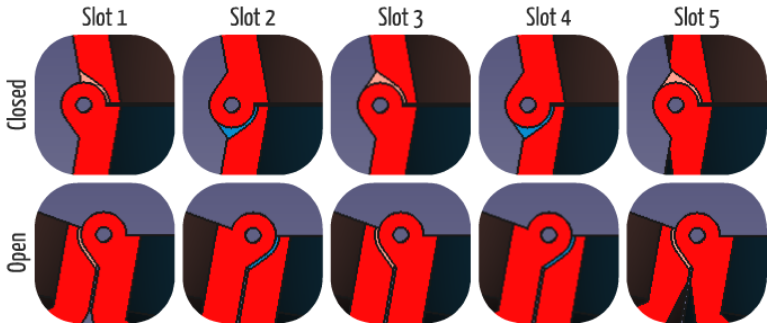
Housing top is a boolean intersection of **Housing** and **Separation top**



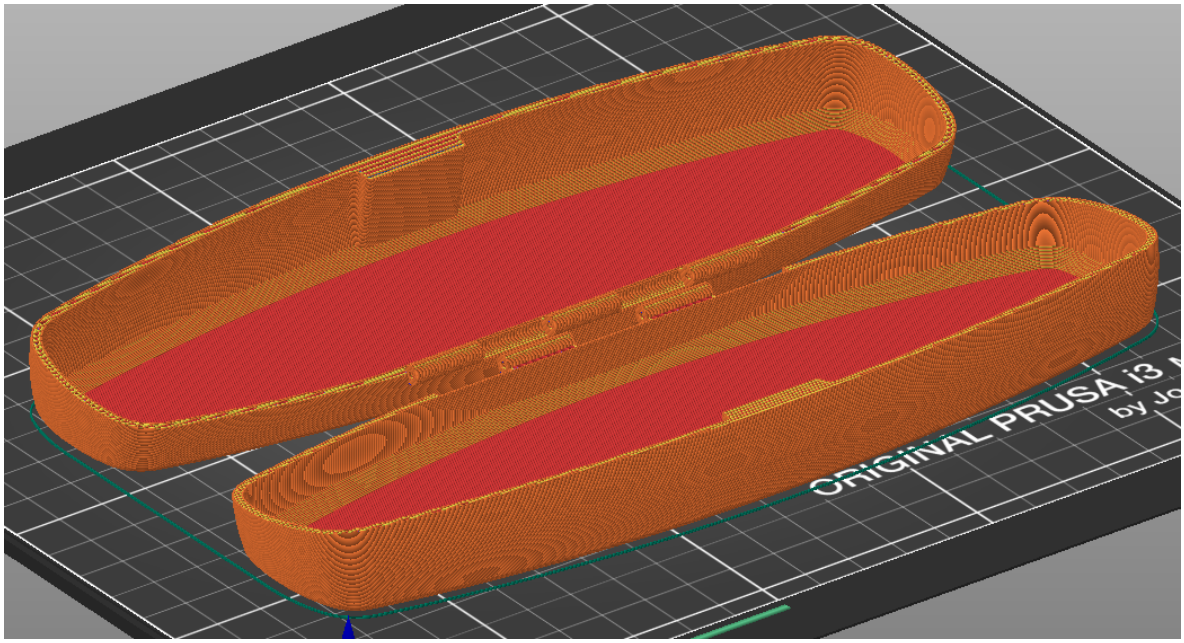
Final checks

The dependency graph and check geometry tool that as described earlier reported no errors.

The persistent section cut also did not reveal problems:



The printability inspection also looks good:



Referencing external parts

Releases

No releases published

Create a new release

Packages

No packages published
Publish your first package

Languages

- **Python** 100.0%

Suggested Workflows

Based on your tech stack

Actions Importer

Automatically convert CI/CD files to YAML for GitHub Actions.

Set up

SLSA Generic generator

Generate SLSA3 provenance for your existing release workflows

Configure

Python application

Create and test a Python application.

Configure

More workflows

Dismiss suggestions