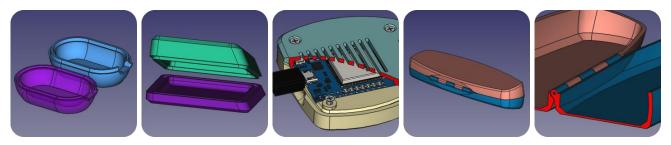# Introduction

This tutorial demonstrates how to design 3D printable housings in FreeCAD. The advantage of the proposed approach is that it is quite structured, and changes to the design can be managed 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.

In Chapter 1, the general concept is explained on how to setup the design of a housing using boolean operation of different bodies. Chapter 2 demonstrates that it is still quite manageable to make modifications to each body when setting up a design like this. Chapter 3 proposes a workaround for the fact that the colors of the parts are overwritten when making modifications to the design. Chapter 4 proposes a naming convention, so it remains easy to locate a feature when modifications are needed, even when designs become complex. Chapter 5 explains the concept of using a skeleton body to manage links between different bodies of the model without the risk of circular references. Skeletons can also make the model more robust. Chapter 6 explains different ways to check if the model and the design are still valid. Chapter 7 extends the idea of the skeleton, but then using stock components as a resource to drive the model. Chapter 8 proposes an elegant way to create holes for fasteners, and also a way to create pillars for screws if the hole does not line up with the separation line of the housing. Chapter 9 takes it a little bit further in using the same concept to create a quite complex hinge. Finally, chapter 10 describes a method to split the design in multiple interlinked files, avoiding a very long model tree in a single file.
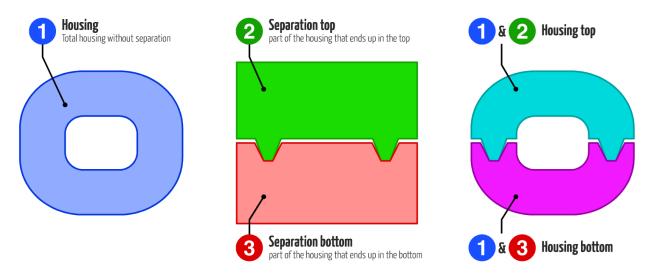
My first thought was to create a Youtube tutorial, but I decided would become quite long. A document is perhaps easier to speed up for the reader. A disadvantage of a document is that I will not learn how many views I have received. Please respond in the Issues section of Github if you appreciate this tutorial or if you have ideas for improvement.

HenkJan van der Pol

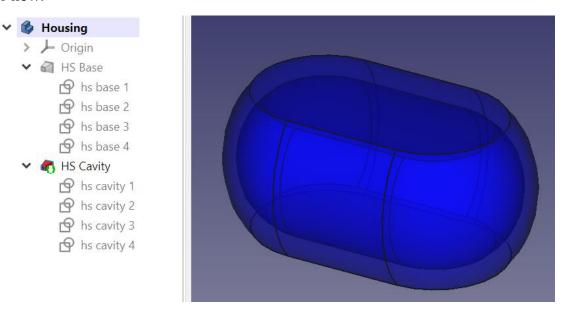# 1. Concept of making a housing using boolean operation of bodies

This example demonstrates the concept 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.
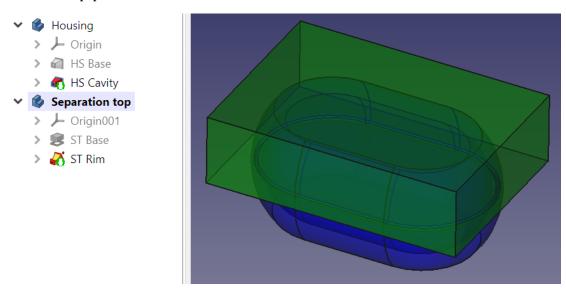

Follow these steps:

1. Housing Creation

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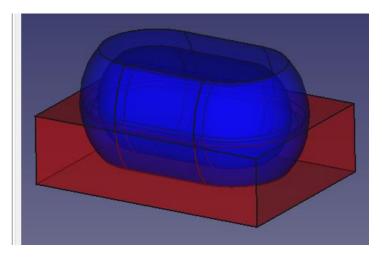
2. Top Separation

Generate a second body (2) called **Separation top**. This component envelops the volume of the upper section of the housing. In this example, the **Separation top** simply consists of a block and an extruded pipe:
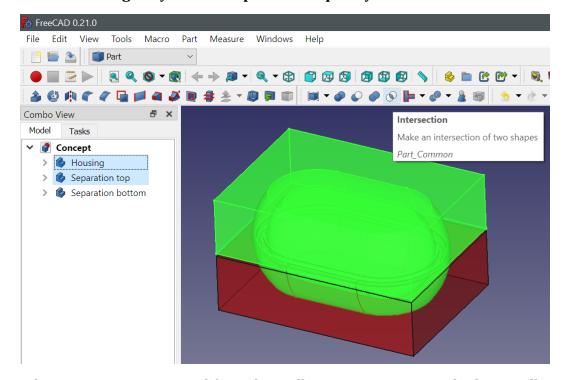

3. Bottom Separation

Similarly, create another body (3) named **Separation bottom**. This part defines the lower separation boundary.
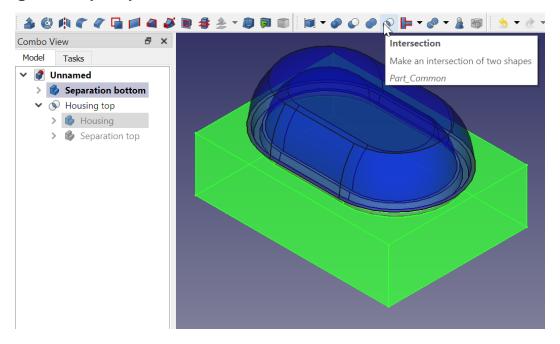
4. Boolean Operations for Top Housing

- 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.
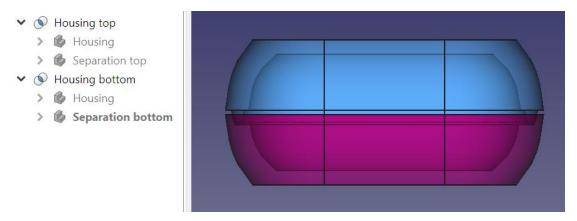- Rename this new body as **Housing top** (1 & 2).

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. Modifying colors

Conclude the process by adjusting the color and transparency attributes of both housing components.
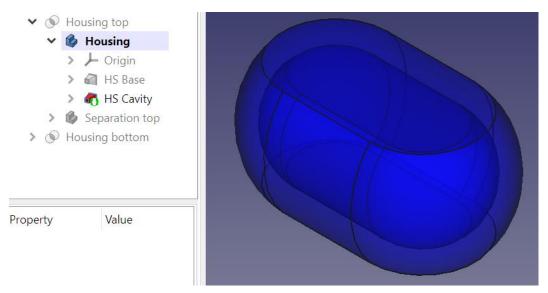
2. Modifying the Housing Design

An aspect to consider within this approach is that it is no longer possible to make changes to **Housing top** and **Housing bottom** using the part design workbench. Nonetheless, modifications can still be efficiently executed on the three original bodies. However, it is vital to make the right decision concerning which specific body to target for the modification. This will be demonstrated in the next example, where we will add a feature to accommodate a power cable to the **Housing** body.
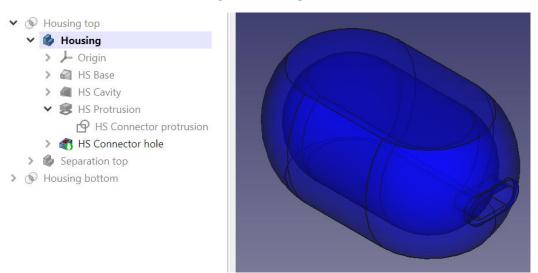
1. Making Housing the active component

Begin by ensuring that **Housing** is the only visible body and ensure it is the active component by double-clicking it in the Part design workbench.
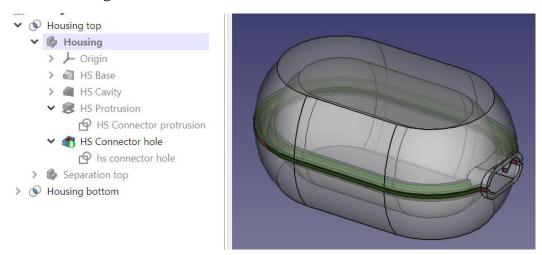

2. Modifying the housing body

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. Review 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 of course possible to rectify these colors once more, the repetitive nature of this task will soon become annoying. To circumvent this, a simple workaround is proposed.

In chapter 8, pillars are created for mounting screws. This is a good example where modifications are not made to the housing body, but to the bottom and top separation bodies instead.

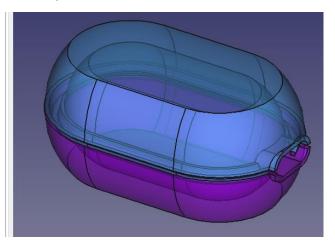# 3. Maintaining the colors of both housing bodies

When a body is created by a boolean operation of bodies, colors are overwritten upon each change of these bodies. To circumvent this, a simple workaround is proposed to prevent having to reset the colors after each change.

One of the simplest 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 to the replicated bodies.

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

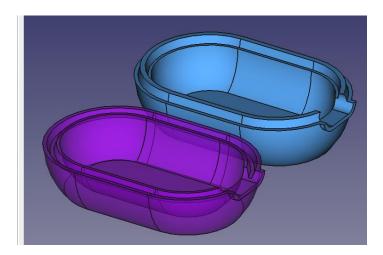Repeat the process for the **Housing bottom** body.

The colors of 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.

This workaround comes at a cost. Every boolean operation and copy makes the model more computationally expensive, so when the model tree gets long (see the dependency graph in Chapter 6), recalculation takes progressively longer when changes are made.

4. Applying a naming convention

As the number of bodies and features expands, identifying features becomes progressively complex. To mitigate this, it is helpful to adopt distinct and meaningful names for features. I have devised a systematic naming approach, outlined as follows:

Non-volumetric features

I use lowercase for non-volumetric features such as sketches and datum planes in the following format:

```
bb ttt nnnnn eee
```

where:

`bb`	A unique abbreviation of the body (e.g., `hs` for housing, `st` for separation top, `sb` for separation bottom, etc.)
`ttt`	An optional 3 letter code for the type of feature: - `pln` for a plane - `axs` for an axis - `ref` for a shape binder
`nnnnn`	The name of the feature
`eee`	Optional extension for additive and subtractive pipes: - the trajectory of the pipe is followed by `trj` - the cross section is followed by `crs`

Volumetric features

For volumetric features, such as pads, cuts and revolves, I use Sentence case:

```
Bb Nnnnn
```

where:

`BB`	A unique abbreviation of the body (`HS` for housing, `ST` for separation top, `SB` for separation bottom, etc.)
`Nnnnn`	The name of the feature

Often, 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 (e.g., a sketch may be named **hs base**, and the pad that is created using that sketch is named **HS Base**)

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

Examples

`sk top`	Top view sketch 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 this 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 groove created as a pipe from `sb groove crs` and `sb groove trj`

5. Using a skeleton to drive dimensions of the bodies

As the number of bodies grows, it becomes increasingly important that the dimensions of each body are driven by a common model. The concept of a skeleton offers a central entity to manage mechanical interfaces and drive major dimensions.

5.1 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, without volumetric features. Consequently, this **Skeleton** body serves as a reference for other bodies.
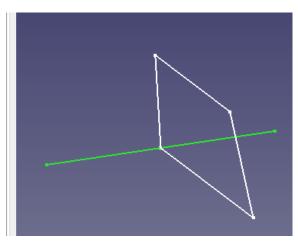
A **Skeleton** body also improves robustness of the model. In cases where sketches reference 3D geometry, such as body edges or faces, making minor alterations can trigger instability due to the notorious Topological Naming Problem. To circumvent this, sketches should refer to other sketches, rendering them less prone to naming changes. A separate **Skeleton** body allows to make multiple simple sketches, which are preferable over complex ones.

When we use tools such as a pad or a pocket to extrude geometry, the depth of extrusion is usually a fixed number. To make the design fully driven by sketches from the **Skeleton** part, we choose a different approach.
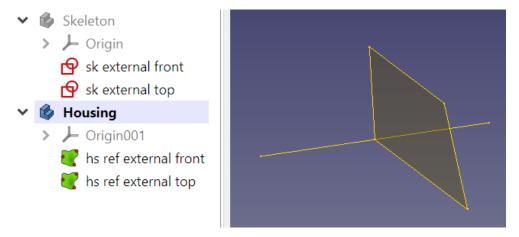
Here's how it works:

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.
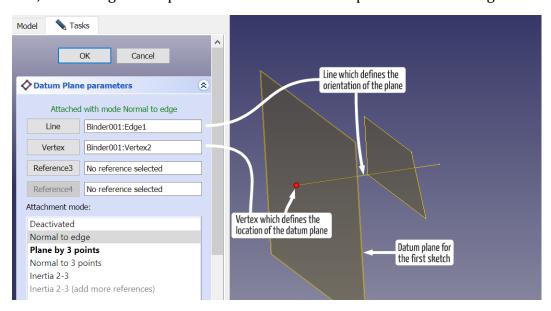
2. Also create a body named **Housing**

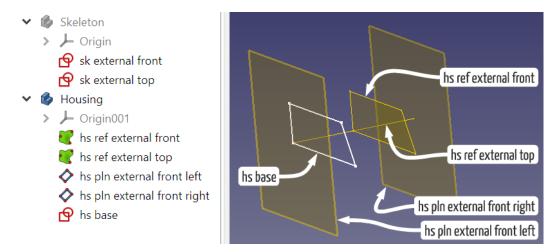3. Create a copy of the sketches we need using a Shape Binder

4. Create two datum planes: one at the beginning of the extrude, named **hs pln external front left** and another at the end, named **hs pln external front right**. To define these planes, use an edge and a point and define the datum plane Normal to edge:
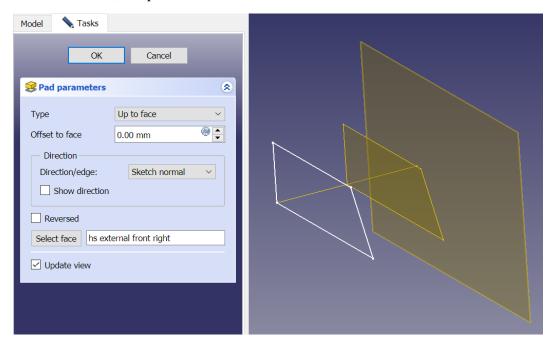

5. Ensure that the **Skeleton** body is invisible, to avoid referring to sketches in this body.

6. Create a sketch **hs base** on the first plane. The geometry in the sketch can refer to a Shape binder derived from the **Skeleton** part using the Create external geometry ![icon] button in the sketcher

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 mirrored sketch. To solve this, set the Map reversed property of the datum plane to true. It is best to do this early in the process, since it often corrupts the sketch.
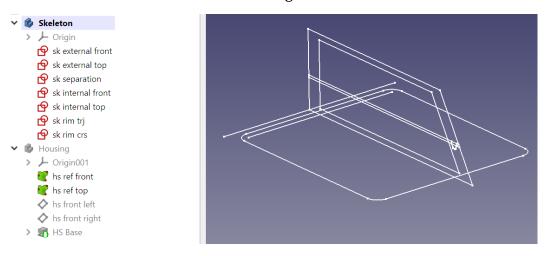
7. Extrude the up until the other datum plane using the Up to face option. When using the Select face button, the plane can also be selected in the model tree.
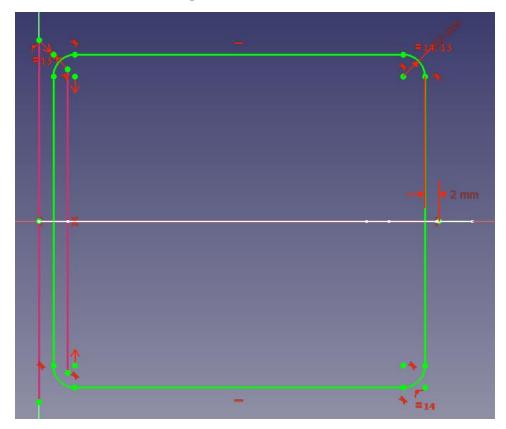
The method proposed here requires more effort. It wil prove its benefits when modifications are necessary later.

5.2 Finalization
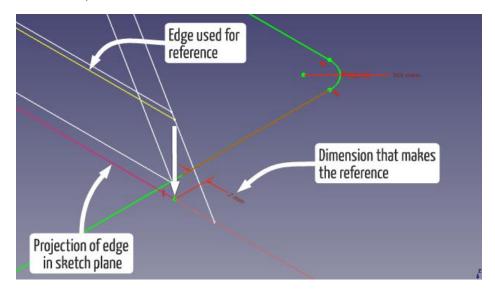
Next, we take some bigger steps to complete the housing. We add a number of sketches to the **Skeleton** body: * **sk separation**: the separation lines for both the top and the bottom separations * **sk internal front**: the front view of the cavity inside housing * **sk internal top**: the top view 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
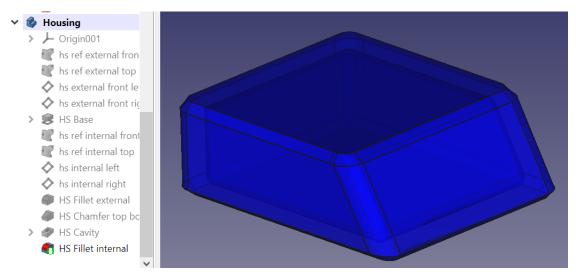
Note that a sketch can make references to sketches in other planes. For instance, the right edge of **sk rim trj** references a line in the **sk separation** sketch:

To make such references, switch to ISO view:
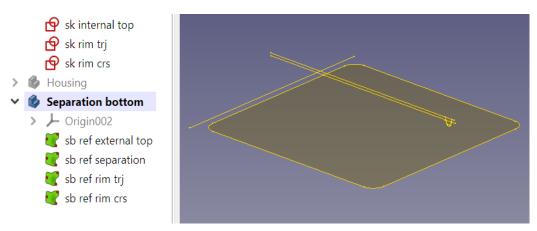
The finish the **Housing** body:
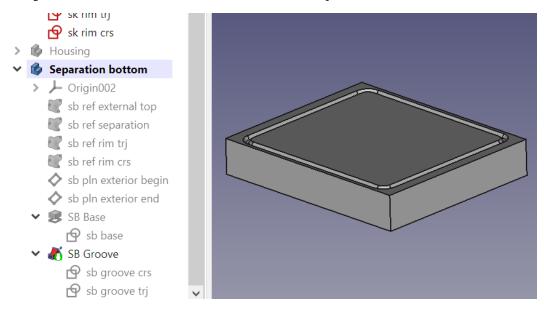
Note that chamfers and fillets were added, but they are just defined in the **Housing** body, without references to the **Skeleton** body.
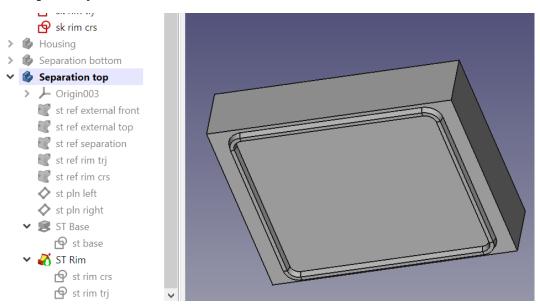
The **Separation bottom** body also has a few necessary shape binders referencing the **Skeleton** body:
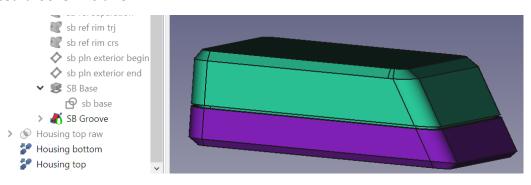
This is wat **Separation bottom** looks like when it is completed:
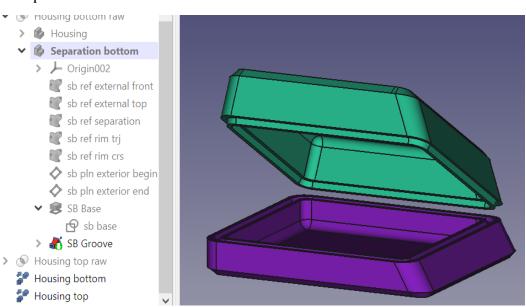
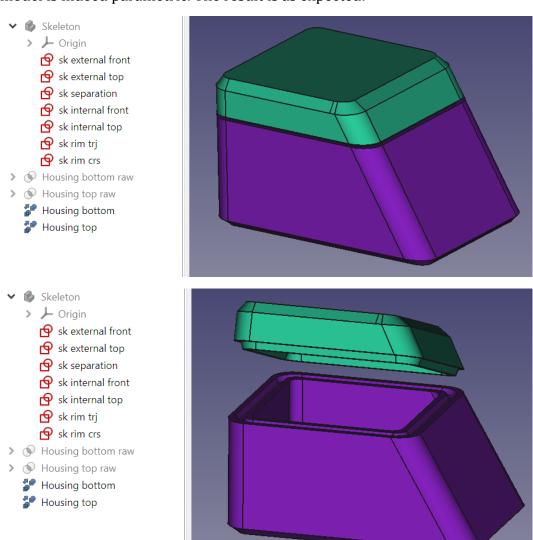**Separation top** is very similar:

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 (i.e. not derived from the **Skeleton**)

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:

6. Checking the model

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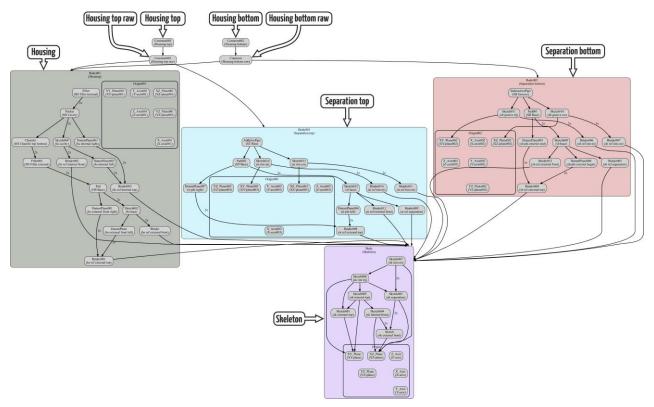
The Check geometry tool from the part workbench can be used to check if the 3D model is valid (Part workbench > Part > Check geometry 👤). It is beyond the scope of this tutorial to explain how to solve common problems. MangoJelly has an excellent video on this tool. If causes are hard to find, the FreeCAD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

6.2 Dependency graph

Sometimes 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
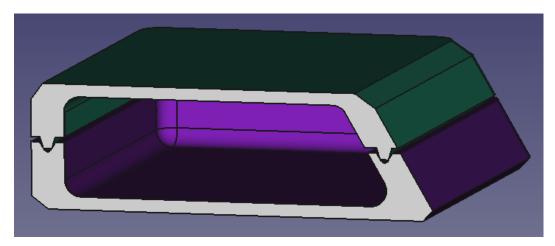
- Body **Housing top raw** refers to **Separation top** and **Housing**

- 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

6.3 Interference check

Although FreeCAD lacks a mechanical interference check function, it is easy to perform an interference check on two bodies. Simply select both bodies and execute the Intersection command from the part workbench. If the result is a body without volume, apparently there is no interference.

6.4 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.
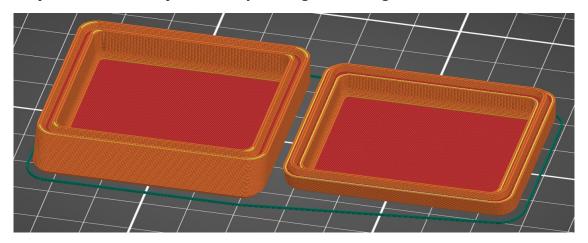
6.5 Checking the result in the slicer

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 groove are printable with multiple adjacent tracks.

7. 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. If not, you need to create them yourself. If this is the case, ensure you take enough headroom for tolerances.
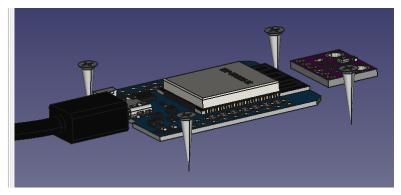
1. Import the components in the FreeCAD file

I usually use File > Merge project.

Orient the components well relative to each other.
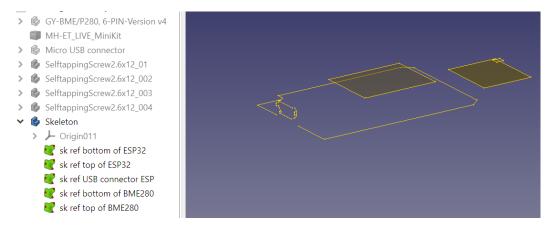
If you can use symmetry in your design, it also makes sense to orient the components relative to the coordinate system, but for more flexibility you can also create planes for that in the **Skeleton** body in the next step.
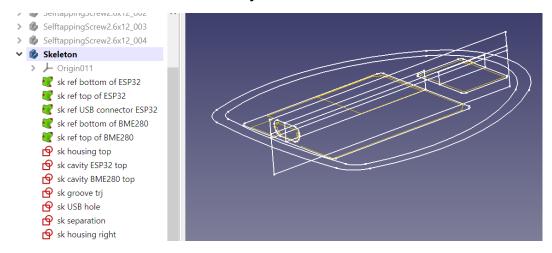
2. Create the Skeleton body

In the skeleton, import important geometry of the components using shape binders. Do this by selecting the important edges and clicking the green Shape binder button. This way, the sketches in the skeleton will dynamically follow the components when components are moved.
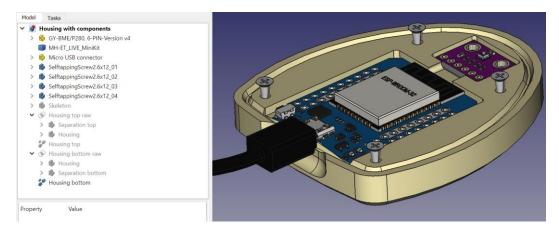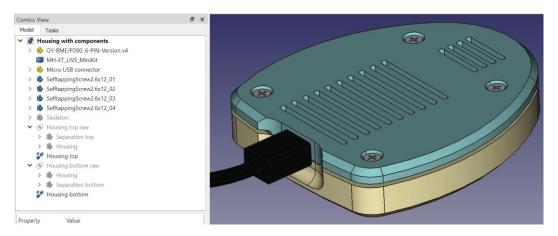
3. Create the sketches in the Skeleton body

4. Create the other bodies

Create the **Housing**, **Separation top**, **Separation bottom**, **Housing top** and **Housing bottom** like in the previous example.
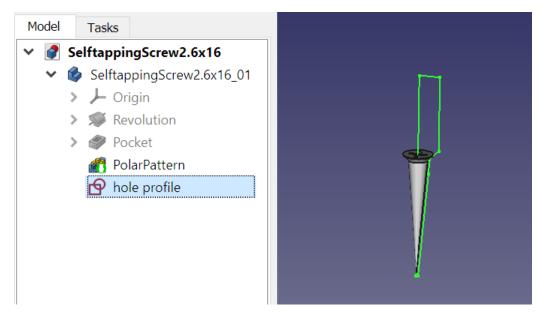
We can now move the boards around, and (within certain constraints), the cavities in the housing will follow the components.

8. Using self tapping screws

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.

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

Below the head of the screw, the sketch has a cylindrical section which is intended to end up in the part that needs to be fixed, and a conical part in which the thread will be formed.

⚠️ In reality, these screws are not conical. Some day I will make a model that more closely resembles the shape of the screws. Nonetheless, this shape works for my printed models
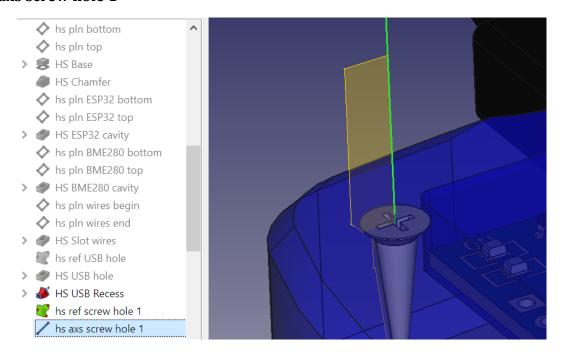
⚠️ By tweaking the shape of the hole, I achieved a good fit for these screws for PET printed on my Prusa Mk3s. With a different printer or material, the tweaking may have a different outcome.
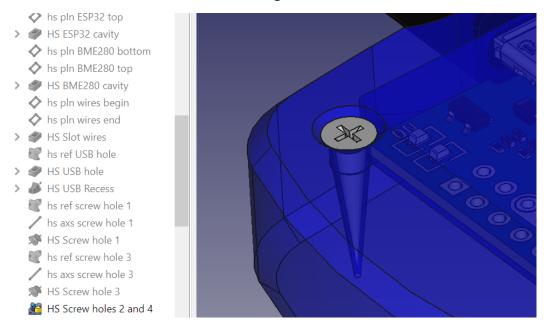
8.1 Creating a screw hole

This is how it works:

1. Insert screw in the model using File > Merge project and orient it using Transform under the right mouse button

2. Create a shape binder **hs ref screw hole 1** in the **Housing** body, referencing the screw hole sketch from the model of the screw

3. Make the sketch in the original model invisible (so we can only select elements from the shape binder)

4. Select the center line of the shape binder, and create a datum axis through it, named **hs axs screw hole 1**

7. Use the groove command on **hs ref screw hole 1** to create thre screw hole, using **hs axs screw hole 1** as a centerline. Rename the groove **HS Screw hole 1**

Now if we move or rotate the screw, the hole will move with it.
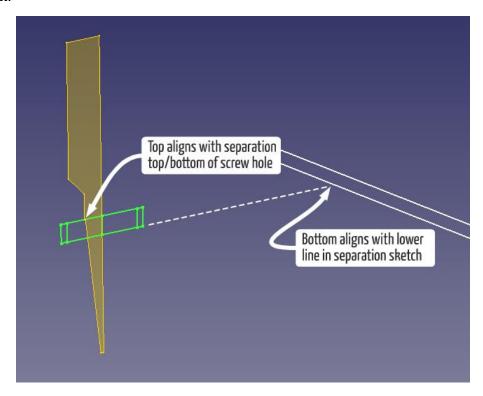
8.2 Creating a pillar for the screw

Sometimes the separation 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.
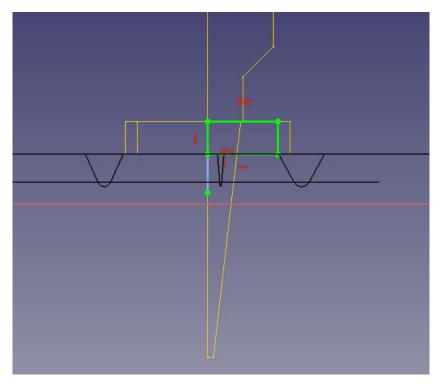
1. Create a shape binder **sk ref screw hole 1** in the **Skeleton** body, referencing the screw hole of the first screw.
2. Make the model of the original screw invisible
3. Select **sk ref screw hole 1** as a plane to draw the sketch of the screw pillar on

4. Create 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.

Top aligns with separation top/bottom of screw hole

Bottom aligns with lower line in separation sketch
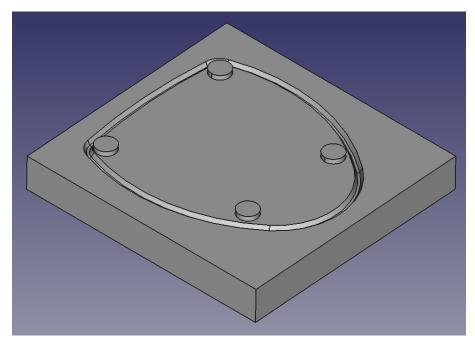
5. Make **Separation bottom** the active body
6. Create a shape binder for the sketch **hole profile** of screw 1 and rename it **sb ref screw hole 1**
7. Create a shape binder for **sk screw pillar 1** from the **Skeleton** body and rename it **sb ref screw pillar 1**
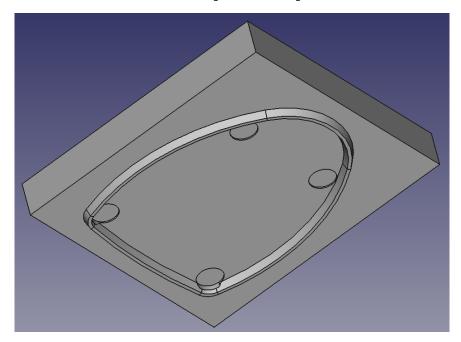
8. Use **sb ref screw hole 1** as a plane for a sketch **sb screw pillar 1** to create the pillar. Add a geometry line that will be the center line of the pillar
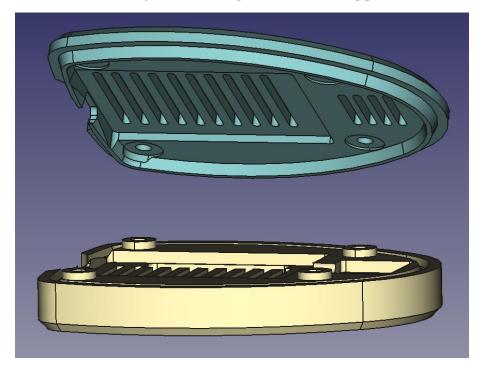
5. Create a revolution and name it **SB Screw pillar 1**
6. Repeat the same procedure for the other pillars. In this case, I did pillar 3 in the same way and created pillars 2 and 4 by mirroring.

Repeat the procedure to create the holes in **Separation top**.

The changes will now automatically come through in both housing parts:
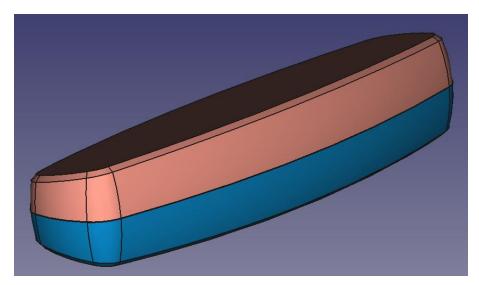
This is a good example to demonstrate why some changes need modifications in the housing part, while others require changes in the separation parts.
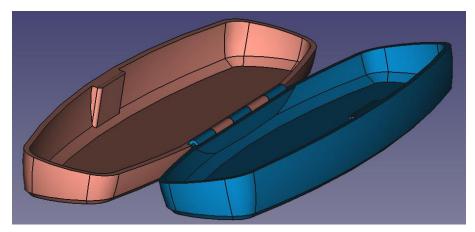
9. Creating a complex hinge

The next project regards a housing with a hinge. It can for instance be used for pencils or glasses. There is a magnet in each shell to lock the housing. 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.
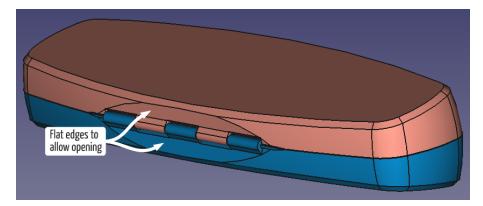
This is the front view of the case when it is closed:

This is the front view of the case when it is open:
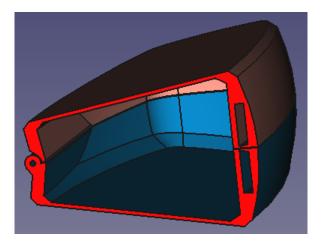
The details of the hinge are quite complex:
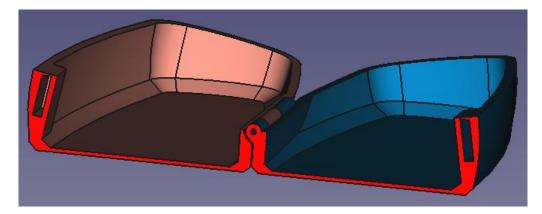
The flat edges in the rear view are needed to avoid mechanical interference when the case is fully open, and they act as an end stop.

This is a cross section through the middle of the casing when it is closed:
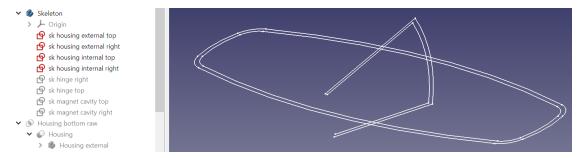
This is a cross section through the middle of the casing when it is open:
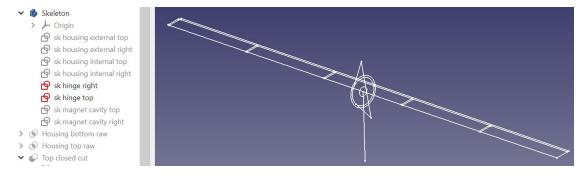
The orientation of the parts during printing is the same as when the case is open. To bridge the openings of the magnets well, the top of the magnet opening needs to be horizontal during printing. This is why the magnet opening is not rectangular.

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:
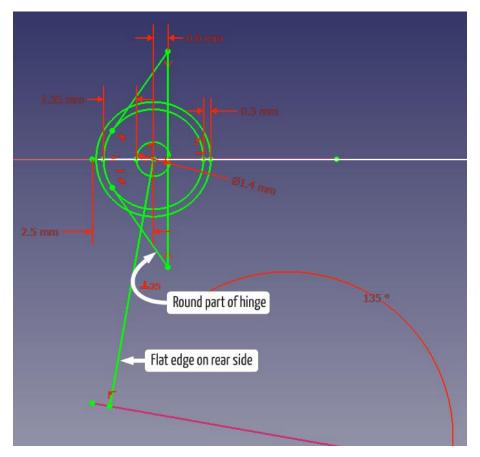
Two sketches define the hinge:

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 (ø1.4 mm) circle represents the hole for the hinge pin. The circle around that represents the cylindrical shape of the hinge (which has a wall thickness of 1.35 mm). The outermost circle is a reference for the play (0.3 mm) between both parts of the housing.
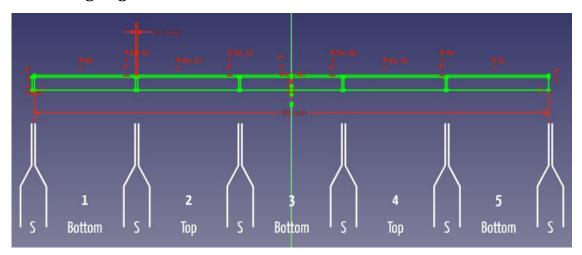
The 135° angle is chosen to ensure both housing shells are printable without support structures.

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, ensuring the housing can be opened fully flat without mechanical interference.

There is also geometry representing the round parts of the hinge, **sk hinge top** basically divides the length of the hinge in three types of sections:

- 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)
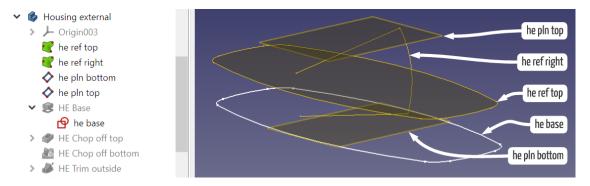
The sketch contains only two dimensions: the total length of the hinge and the space between the parts in axial direction. The radius of the cilinders is also modelled, but this has been derived from **sk hinge right**
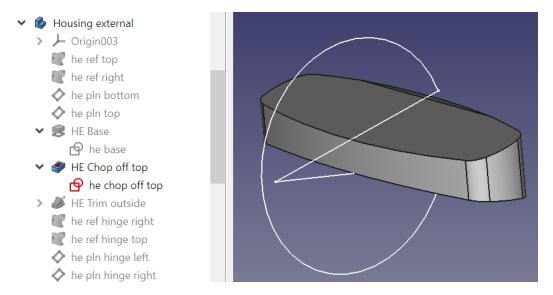
The housing is modelled in two different bodies: **Housing external** represents the outside of the housing, **Housing internal** represents the cavity inside. The final housing is obtained by boolean subtraction in the part workbench.

9.1 Housing external
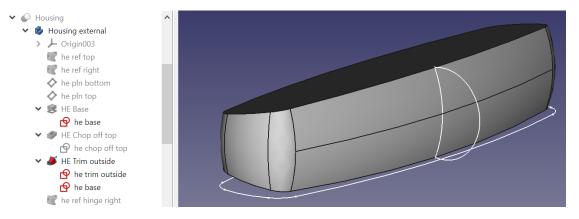
The relevant sketches from the skeleton are imported as shape binders. The bottom and top datum plane are defined as 'normal to edge', referencing the Z-axis and the bottom and top most points. The contour **he base** is modelled on **he pln bottom** and extruded until **he pln top**.
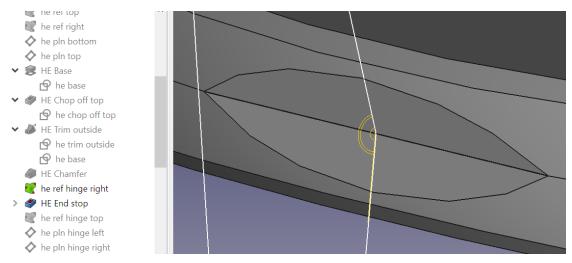
he chop off top chops off the oblique surfaces of **HE Base**.
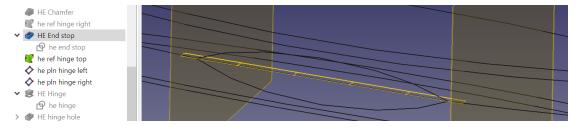
A curve along the outside is made with a subtractive pipe using **he trim outside** along **he base**:
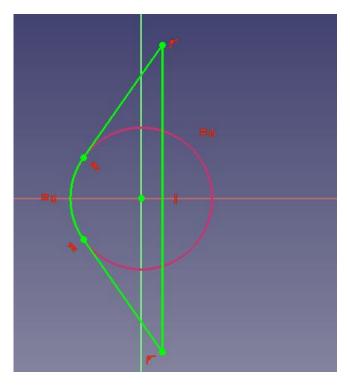
Chamfers are added, **he ref hinge right** is imported and the flat edges for the end stop when opening the case are created:
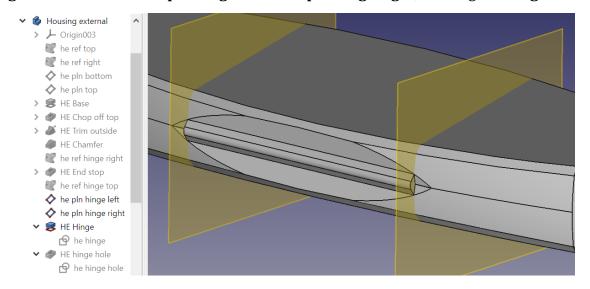
he ref hinge top is imported as shape binder, and the beginning- and end datum planes for the hinge are created. They exclude the space next to the hinge:
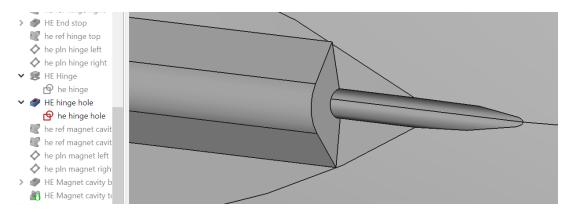

The cross section of the hinge **he hinge** is created on **he pln hinge left**, referring to **he ref hinge right** for the shape:
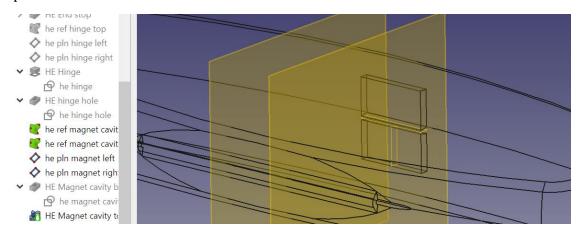
he hinge is extruded from **he pln hinge left** to **he pln hinge right**, forming **HE Hinge**:

As a final step for the hinge, the hole through the hinge is created:
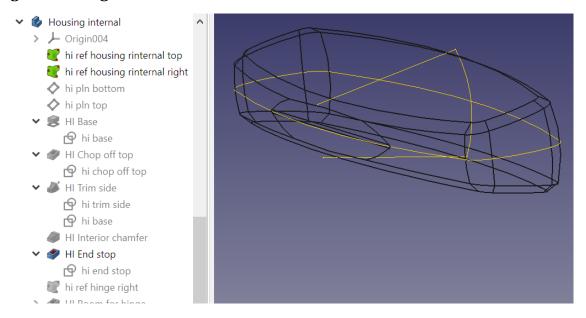
Both sketches determining the magnet pockets are imported, two planes defining the beginning and end of the magnet pockets are created, the lower magnet pocket is created, and the upper magnet pocket is mirrored from the bottom one:
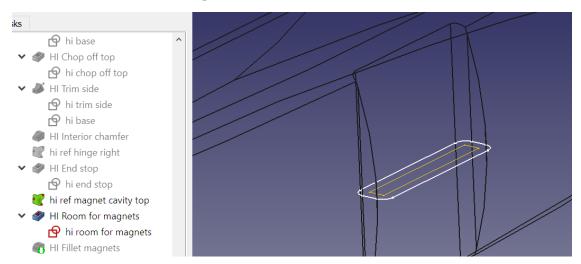
9.2 Housing internal
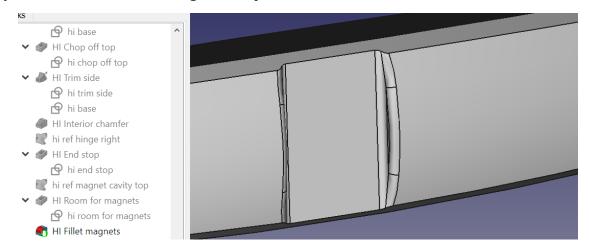
The first steps of the Housing internal body are basically the same, but now referring to **sk housing internal top** and **sk housing internal right** instead of **sk housing external top** and **sk housing external right**:

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**:
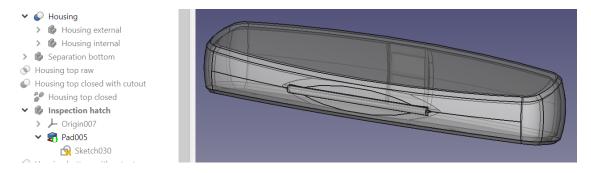
Finally, a fillet is added to the magnet bump:
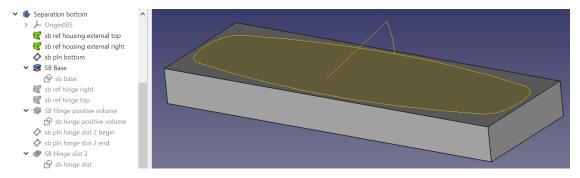
9.3 Housing

The housing is created by boolean subtraction of **Housing external** and **Housing internal** in the Part workbench:
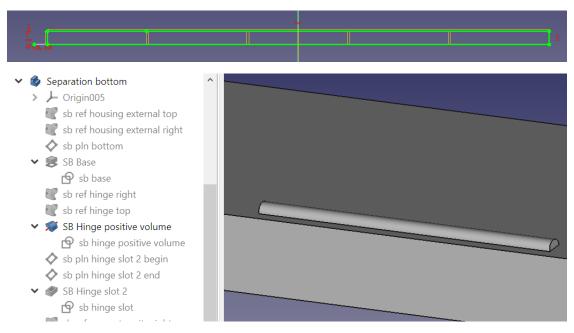
9.4 Separation bottom
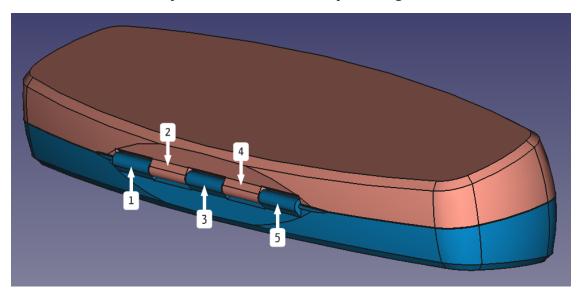
The **Separation bottom** body starts with importing **sb ref housing external top** and **sb ref housing external right**, and then construction only **sb pln bottom**. A rectangle is drawn in a plane 0.1 mm below the XY plane, to ensure there is 0.2 mm space between both shells when the housing is closed, allowing for tolerances of 3D printing. The rectangle is 3 mm larger than the outer shape.
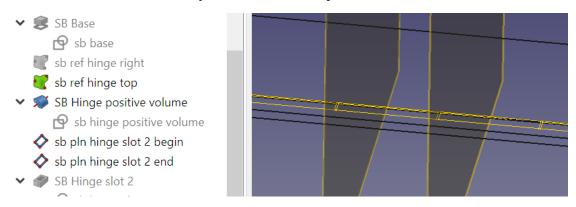
sb ref hinge right and **sb ref hinge top** are imported as shape binders. A positive revolve **SB Hinge positive volume** is added to the shape. The sketch **sb hinge positive volume** is a direct trace from **sb ref hinge top**.
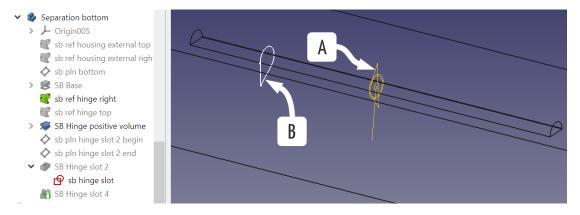

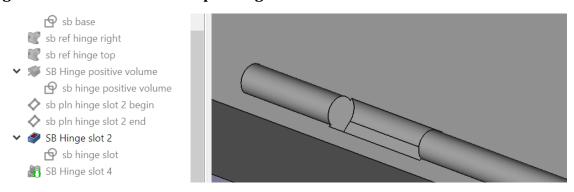Next we will create the slot for protrusion 2 from the top housing.

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.
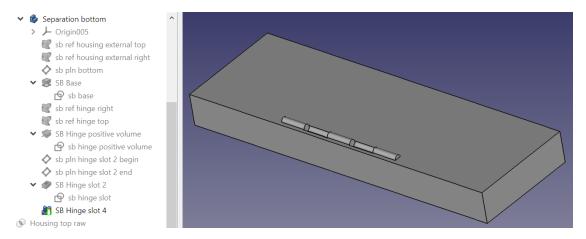

Line B of sketch **sb hinge slot** on datum plane **sb pln hinge slot 2 begin** is under an angle of 160° relative to line A of **sb ref hinge right**. This is because the top housing can be opened 160°, ensuring sufficient space between both parts. The larger circle in **sb ref hinge right** is used,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

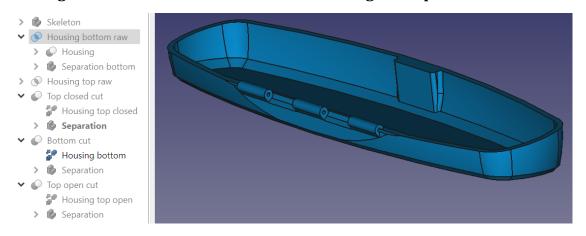**SB Hinge slot 2** is extruded until **sb pln hinge slot 2 end**:

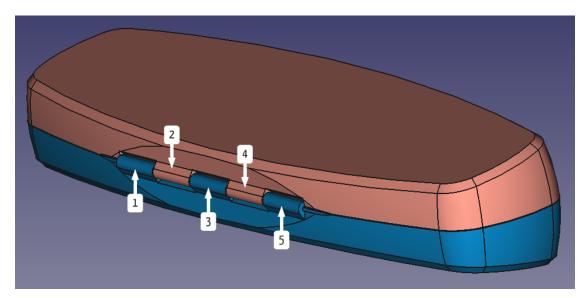**SB Hinge slot 4** is created by mirroring **SB Hinge slot 2** over the YZ plane.

9.5 Housing bottom

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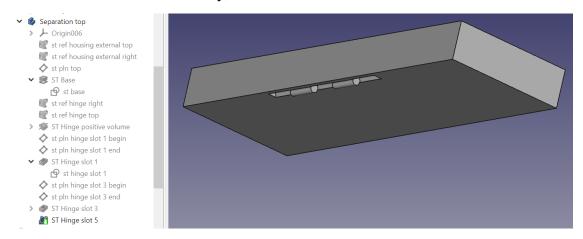
9.6 Separation top

Separation top is very similar to **Separation bottom**, only now the slots are in locations 1, 3 and 5.
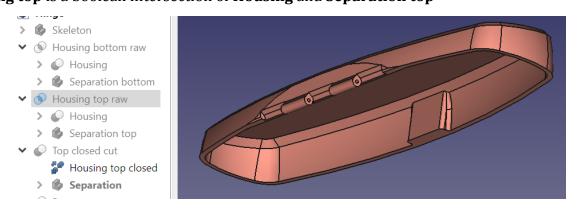

Slots 1 and 3 were created individually, slot 5 is a mirror of slot 1.
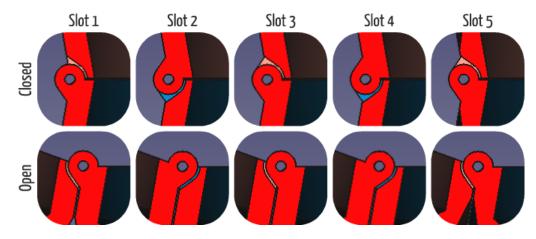

9.7 Housing top

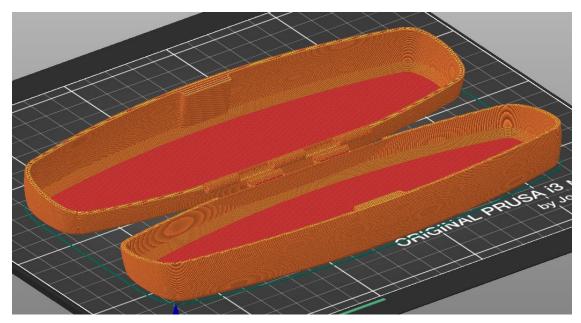**Housing top** is a boolean intersection of **Housing** and **Separation top**

9.8 Final checks

The **Dependency graph** and **Check geometry tool** that as described earlier reported no errors.

The **Persistent section cut** also reveals no problems:

The **Printability inspection** also looks good:

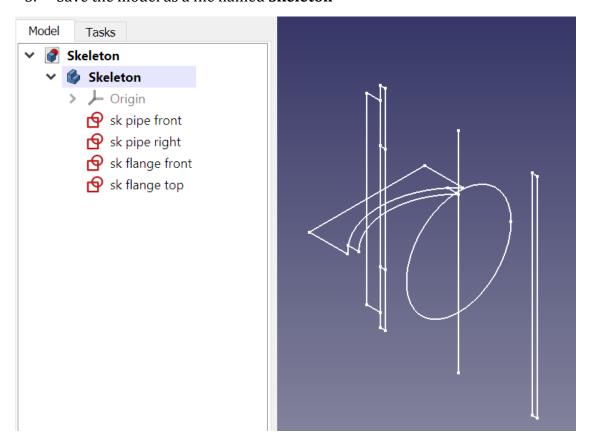# 10. Referencing external parts

When projects become more complex, including many different parts, it may be useful to reduce the number of bodies per FreeCAD file.

Assembly workbenches tend to be prone to the Topological Naming problem: assemblies tend to fall apart if minor changes are made to individual parts.
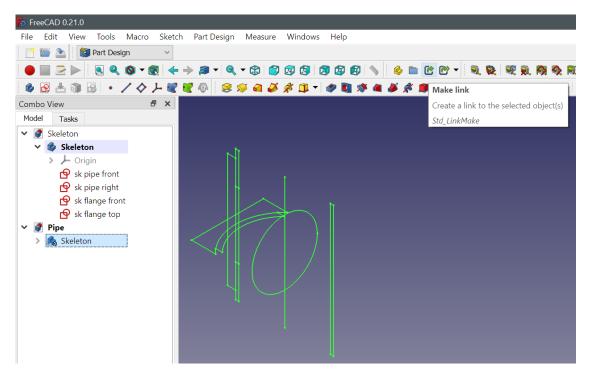
If assemblies are static (no moving parts), it is possible to use the same global coordinate system for all parts. This means that upon importing parts in the general assembly, no orientation is needed, except for parts that occur more than once.
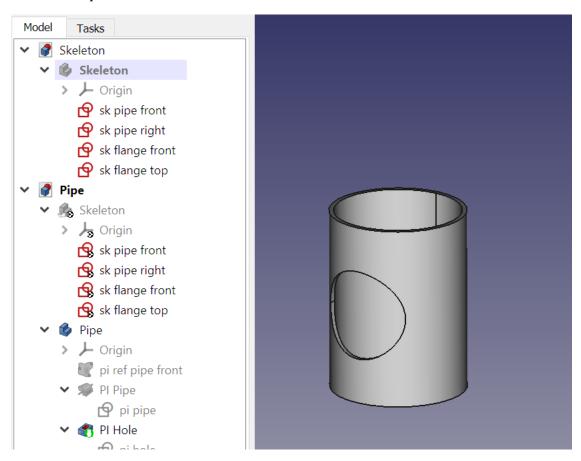
This is how it works:

1. Create the **Skeleton** body as usual,
2. Add sketches that form important envelopes or mechanical interfaces between the parts.
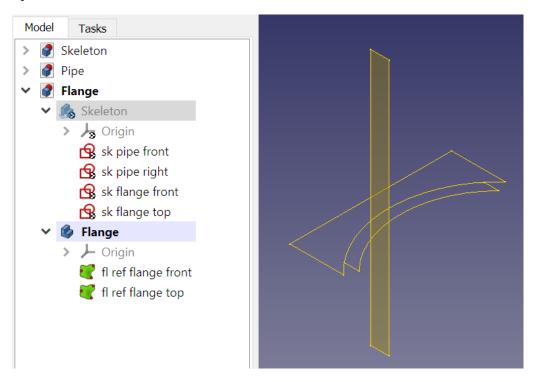3. Save the model as a file named **Skeleton**

5. Create a second file, named **Pipe**, and save that too. It is important that the part is saved to become a recipient for a link.

6. Shape binders cannot directly link to elements in other files. Therefore, we need to first import **Skeleton** via a dynamic link. While the **Pipe** part is open, select the **Skeleton** body in the model tree and choose the Make link button in the toolbar.
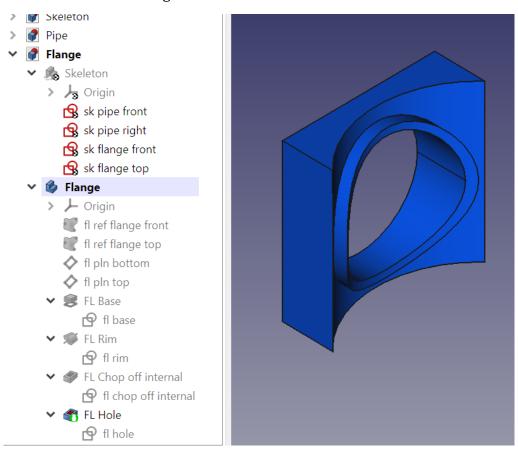
6. Create a new body named **Pipe**.

7. Create the required Shape binders and create the sketches and volumes like in previous examples.

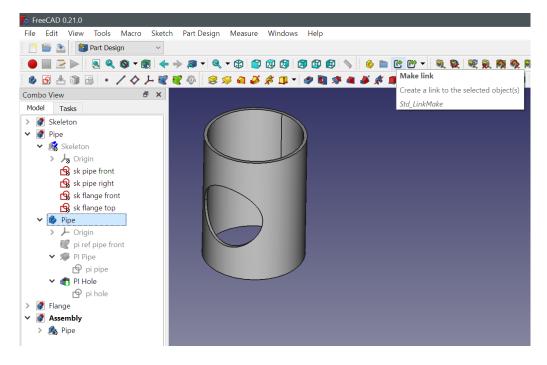8. Create a third file, named **Flange**

9. Select the **Skeleton** body in the model tree and choose the Make link button.

10. Create a new body in the **Flange** part, named **Flange**

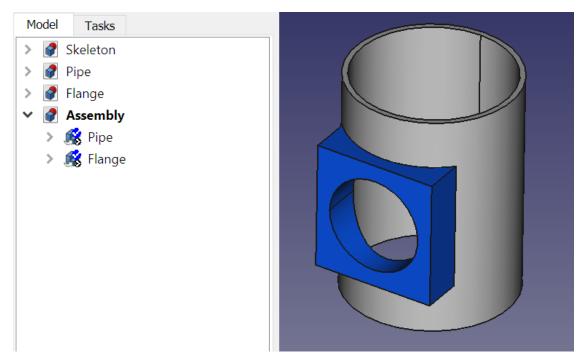11. Create the required Shape binders and create the sketches and volumes like in previous examples.

12. Create the model of the flange

13. Create a fourth file named **Assembly**

14. Select the **Pipe** body in the model tree and choose Make link ↪

15. Select the **Flange** body in the model tree and choose Make link

Since both parts have been created in the same coordinate system, they are already in the right location and orientation.

The skeleton still drives the major dimensions and interfaces in all parts of the assembly.